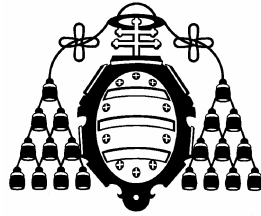


# UNIVERSIDAD DE OVIEDO



## ESCUELA DE INGENIERÍA INFORMÁTICA

### TRABAJO FIN DE GRADO

#### “PLATAFORMA EDUCATIVA BASADA EN REALIDAD AUMENTADA Y DISPOSITIVOS MÓVILES”

**DIRECTOR:** Juan Manuel Cueva Lovelle

**CODIRECTOR:** Oscar Sanjuán Martínez

**AUTOR:** Iván Lorenzo Rubio

VB del Director del Proyecto



## Resumen

Con este trabajo se presenta una nueva plataforma para la representación 3D de todo tipo de objetos. Para ello, se usan técnicas de *Realidad Aumentada* que permiten que, mediante un dispositivo móvil con Android, el usuario perciba el modelo integrado en el mundo real e incluso que pueda interactuar con él.

Los objetos se visualizan utilizando dos técnicas de realidad aumentada: mediante markers o patrones y utilizando la posición GPS. La primera consiste en representar información cuando la cámara reconozca una determinada imagen y la segunda cuando nos encontremos en una determinada posición geográfica.

Se pretende ofrecer la máxima información posible de cada modelo. No sólo información visual, sino también información auditiva. El usuario podrá trasladar y escalar los modelos a través de la pantalla táctil de los dispositivos móviles.

Todo ello se ha desarrollado de forma que haya una independencia entre la plataforma y los datos cargados. Así se podrá aplicar a diferentes temáticas simplemente con un cambio de información, ya que la base será la misma para todas.

## Palabras Clave

Realidad aumentada; markers; GPS; dispositivos móviles; Android; representación 3D;



## Abstract

With this project new platform is presented for the 3D picture of all kinds of objects. For this to be possible, *Augmented Reality* techniques are used and these allow the user to perceive the integrated model in the real world with just one simple Android mobile device and even being able to interact with it.

Objects are displayed using two augmented reality techniques: using markers or patterns and using the GPS position. The first is to represent information when the camera recognizes a certain image and the second when we are in a particular geographic location.

The maximum possible information of each model is offered. Not just visual information but also auditory information. The user may move and increase the size of the models through the touch screen mobile devices.

This has been developed so that there is independence between the platform and the loaded data. This may apply to different subjects simply with a change of information, because the base is the same for all.

## Keywords

Augmented Reality, markers, GPS, mobile devices, Android, 3D picture;



## **Agradecimientos:**

A mis padres y hermana  
por su apoyo en los buenos y en los malos  
momentos y por sus consejos cuando  
las cosas no iban como yo desearía.

A Susi  
por apoyarme y animarme en todo momento,  
por escucharme y preocuparse por todo lo que hacía  
y por admirar siempre mi trabajo.

A toda mi familia  
por el interés mostrado en saber  
como se desarrollaba mi proyecto.

A mis amigos  
por los buenos momentos que hemos compartido  
y que seguiremos compartiendo.

A mis directores de proyecto  
por su ayuda y colaboración durante  
todo el proceso de desarrollo de este proyecto.

En general, a todas las personas con las que  
me he encontrado en estos últimos años  
y que me han aportado cosas importantes.

A tod@s ell@s  
**Gracias.**



# Índice general

|                                                                               |           |
|-------------------------------------------------------------------------------|-----------|
| <b>1. Introducción</b>                                                        | <b>1</b>  |
| 1.1. Motivación del proyecto . . . . .                                        | 1         |
| 1.1.1. En el contexto educativo . . . . .                                     | 1         |
| 1.1.2. En otras áreas . . . . .                                               | 2         |
| 1.1.3. ¿Qué se pretende con este proyecto? . . . . .                          | 3         |
| 1.2. Metodología empleada . . . . .                                           | 4         |
| 1.2.1. Introducción . . . . .                                                 | 4         |
| 1.2.2. SCRUM . . . . .                                                        | 5         |
| 1.2.3. Adecuación de la metodología al proyecto . . . . .                     | 7         |
| 1.2.4. Seguimiento del proyecto . . . . .                                     | 9         |
| 1.3. Estructura del documento . . . . .                                       | 9         |
| 1.3.1. Capítulo 1. Introducción . . . . .                                     | 9         |
| 1.3.2. Capítulo 2. Estado del arte . . . . .                                  | 10        |
| 1.3.3. Capítulo 3. Análisis . . . . .                                         | 10        |
| 1.3.4. Capítulo 4. Diseño . . . . .                                           | 10        |
| 1.3.5. Capítulo 5. Modelo de datos . . . . .                                  | 10        |
| 1.3.6. Capítulo 6. Pruebas software . . . . .                                 | 10        |
| 1.3.7. Capítulo 7. Pruebas de usabilidad . . . . .                            | 11        |
| 1.3.8. Capítulo 8. Presupuesto y Plan Temporal . . . . .                      | 11        |
| 1.3.9. Capítulo 9. Manual del programador . . . . .                           | 11        |
| 1.3.10. Capítulo 10. Manual de internacionalización . . . . .                 | 11        |
| 1.3.11. Capítulo 11. Manual del instalador . . . . .                          | 11        |
| 1.3.12. Capítulo 12. Manual de usuario . . . . .                              | 11        |
| 1.3.13. Capítulo 13. User's manual . . . . .                                  | 11        |
| 1.3.14. Capítulo 14. Conclusiones y líneas de investigación futuras . . . . . | 11        |
| 1.3.15. Apéndice A. Tecnologías . . . . .                                     | 12        |
| 1.3.16. Apéndice B. Documentación . . . . .                                   | 12        |
| 1.3.17. Apéndice C. Código Fuente . . . . .                                   | 12        |
| <b>2. Estado del arte</b>                                                     | <b>13</b> |
| 2.1. Introducción . . . . .                                                   | 13        |
| 2.2. Realidad Aumentada . . . . .                                             | 14        |
| 2.2.1. Introducción . . . . .                                                 | 14        |
| 2.2.2. Usando markers . . . . .                                               | 15        |
| 2.2.3. Usando la posición GPS . . . . .                                       | 16        |
| 2.2.4. Diferencias entre realidad virtual y realidad aumentada . . . . .      | 17        |

|         |                                                                 |    |
|---------|-----------------------------------------------------------------|----|
| 2.2.5.  | ARToolKit y sus variantes . . . . .                             | 18 |
|         | ¿Qué es ArToolKit? . . . . .                                    | 18 |
|         | ¿Cómo funciona? . . . . .                                       | 18 |
|         | Variantes . . . . .                                             | 19 |
| 2.3.    | Tecnologías empleadas . . . . .                                 | 20 |
| 2.3.1.  | Comunicación . . . . .                                          | 20 |
|         | Wi-Fi (estándar 802.11b) . . . . .                              | 20 |
|         | Telefonía móvil 3G y 4G . . . . .                               | 21 |
|         | WiMAX . . . . .                                                 | 22 |
|         | Bluetooth . . . . .                                             | 23 |
| 2.3.2.  | Posicionamiento . . . . .                                       | 23 |
|         | Sistema de posicionamiento global (GPS) . . . . .               | 24 |
|         | Global Orbiting Navigation Satellite System o GLONASS . . . . . | 27 |
|         | Galileo . . . . .                                               | 28 |
| 2.4.    | Magic Book . . . . .                                            | 31 |
| 2.4.1.  | Introducción . . . . .                                          | 31 |
| 2.4.2.  | Ventajas . . . . .                                              | 31 |
| 2.4.3.  | Inconvenientes . . . . .                                        | 31 |
| 2.5.    | Proyecto ARiSE . . . . .                                        | 32 |
| 2.5.1.  | Introducción . . . . .                                          | 32 |
| 2.5.2.  | Ventajas . . . . .                                              | 33 |
| 2.5.3.  | Inconvenientes . . . . .                                        | 33 |
| 2.6.    | LearnAR . . . . .                                               | 33 |
| 2.6.1.  | Introducción . . . . .                                          | 33 |
| 2.6.2.  | Ventajas . . . . .                                              | 34 |
| 2.6.3.  | Inconvenientes . . . . .                                        | 34 |
| 2.7.    | SciMorph . . . . .                                              | 35 |
| 2.7.1.  | Introducción . . . . .                                          | 35 |
| 2.7.2.  | Ventajas . . . . .                                              | 35 |
| 2.7.3.  | Inconvenientes . . . . .                                        | 35 |
| 2.8.    | ZooBurst . . . . .                                              | 36 |
| 2.8.1.  | Introducción . . . . .                                          | 36 |
| 2.8.2.  | Ventajas . . . . .                                              | 36 |
| 2.8.3.  | Inconvenientes . . . . .                                        | 36 |
| 2.9.    | ARSights . . . . .                                              | 37 |
| 2.9.1.  | Introducción . . . . .                                          | 37 |
| 2.9.2.  | Ventajas . . . . .                                              | 38 |
| 2.9.3.  | Inconvenientes . . . . .                                        | 38 |
| 2.10.   | Wikitude . . . . .                                              | 39 |
| 2.10.1. | Introducción . . . . .                                          | 39 |
| 2.10.2. | Ventajas . . . . .                                              | 40 |
| 2.10.3. | Inconvenientes . . . . .                                        | 40 |
| 2.11.   | Layar . . . . .                                                 | 41 |
| 2.11.1. | Introducción . . . . .                                          | 41 |
| 2.11.2. | Ventajas . . . . .                                              | 41 |

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| 2.11.3. Inconvenientes . . . . .                                            | 42        |
| 2.12. Museografía Virtual para el museo de sitio de Calakmul . . . . .      | 42        |
| 2.12.1. Introducción . . . . .                                              | 42        |
| 2.12.2. Ventajas . . . . .                                                  | 45        |
| 2.12.3. Inconvenientes . . . . .                                            | 45        |
| 2.13. Proyecto ARCO . . . . .                                               | 45        |
| 2.13.1. Introducción . . . . .                                              | 45        |
| 2.13.2. Ventajas . . . . .                                                  | 47        |
| 2.13.3. Inconvenientes . . . . .                                            | 47        |
| 2.14. ARCHEOGUIDE . . . . .                                                 | 47        |
| 2.14.1. Introducción . . . . .                                              | 47        |
| 2.14.2. Ventajas . . . . .                                                  | 48        |
| 2.14.3. Inconvenientes . . . . .                                            | 48        |
| 2.15. Ename 974 . . . . .                                                   | 49        |
| 2.15.1. Introducción . . . . .                                              | 49        |
| 2.15.2. Ventajas . . . . .                                                  | 50        |
| 2.15.3. Inconvenientes . . . . .                                            | 50        |
| 2.16. Dinosaurios en el Museo del Jurásico de Asturias . . . . .            | 51        |
| 2.16.1. Introducción . . . . .                                              | 51        |
| 2.16.2. Ventajas . . . . .                                                  | 51        |
| 2.16.3. Inconvenientes . . . . .                                            | 52        |
| 2.17. Conclusiones . . . . .                                                | 53        |
| <b>3. Análisis</b>                                                          | <b>55</b> |
| 3.1. Descripción general del sistema . . . . .                              | 55        |
| 3.1.1. Perspectiva del producto . . . . .                                   | 55        |
| Interfaces del sistema . . . . .                                            | 55        |
| Interfaces de usuario . . . . .                                             | 56        |
| Interfaces hardware . . . . .                                               | 56        |
| Interfaces software . . . . .                                               | 56        |
| Interfaces de comunicaciones . . . . .                                      | 56        |
| Restricciones de memoria . . . . .                                          | 56        |
| Funcionamientos . . . . .                                                   | 56        |
| Requisitos de adaptación . . . . .                                          | 57        |
| 3.1.2. Funcionalidad del producto . . . . .                                 | 57        |
| 3.1.3. Características de los usuarios . . . . .                            | 57        |
| 3.2. Requisitos específicos . . . . .                                       | 58        |
| 3.2.1. Requisitos comunes de los interfaces . . . . .                       | 58        |
| Interfaces de usuario . . . . .                                             | 58        |
| Interfaces de hardware . . . . .                                            | 59        |
| Interfaces de software . . . . .                                            | 59        |
| Interfaces de comunicación . . . . .                                        | 60        |
| 3.2.2. Requisitos funcionales . . . . .                                     | 60        |
| Requisitos funcionales generales . . . . .                                  | 60        |
| Requisitos funcionales de la aplicación para dispositivos móviles . . . . . | 61        |
| Requisitos funcionales del servidor web . . . . .                           | 62        |

|                                                                                 |           |
|---------------------------------------------------------------------------------|-----------|
| Requisitos funcionales de los ficheros de configuración . . . . .               | 62        |
| 3.2.3. Requisitos de rendimiento . . . . .                                      | 63        |
| 3.2.4. Restricciones de diseño . . . . .                                        | 63        |
| 3.2.5. Atributos del software del sistema . . . . .                             | 64        |
| Seguridad . . . . .                                                             | 64        |
| Fiabilidad . . . . .                                                            | 64        |
| Disponibilidad . . . . .                                                        | 64        |
| Mantenibilidad . . . . .                                                        | 65        |
| Portabilidad . . . . .                                                          | 65        |
| 3.2.6. Otros requisitos . . . . .                                               | 66        |
| 3.3. Descripción de los actores . . . . .                                       | 66        |
| 3.3.1. Actor usuario del sistema . . . . .                                      | 66        |
| Actor Usuario de la aplicación . . . . .                                        | 66        |
| Actor Administrador del sistema . . . . .                                       | 66        |
| 3.4. Modelo de Casos de Uso . . . . .                                           | 66        |
| 3.4.1. Límites del sistema . . . . .                                            | 66        |
| 3.4.2. Casos de uso específicos . . . . .                                       | 67        |
| Obtención de información adaptada al contexto . . . . .                         | 67        |
| Manipular objetos 3D . . . . .                                                  | 67        |
| Configurar opciones de la aplicación . . . . .                                  | 68        |
| Publicar datos . . . . .                                                        | 69        |
| 3.5. Especificación de Casos de Uso . . . . .                                   | 70        |
| 3.6. Análisis de Interfaces de Usuario . . . . .                                | 74        |
| 3.6.1. Descripción de la Interfaz . . . . .                                     | 74        |
| Interfaz principal de la aplicación . . . . .                                   | 75        |
| Interfaz de la pantalla de configuración . . . . .                              | 76        |
| <b>4. Diseño</b>                                                                | <b>77</b> |
| 4.1. Descripción de los módulos del sistema . . . . .                           | 78        |
| 4.1.1. Módulo de Carga de Datos . . . . .                                       | 78        |
| 4.1.2. Módulo de Localización . . . . .                                         | 79        |
| 4.1.3. Módulo de Renderización o Parser de objetos 3D . . . . .                 | 79        |
| 4.1.4. Módulo de Sonido . . . . .                                               | 80        |
| 4.1.5. Módulo de Visualización de información . . . . .                         | 80        |
| 4.1.6. Internacionalización de la aplicación . . . . .                          | 81        |
| 4.2. Descripción del diseño de clases . . . . .                                 | 83        |
| 4.2.1. Diseño del Módulo de Carga de Datos . . . . .                            | 83        |
| 4.2.2. Descripción del Módulo de Localización . . . . .                         | 86        |
| 4.2.3. Descripción del Módulo de Renderización o Parser de objetos 3D . . . . . | 87        |
| 4.2.4. Descripción del Módulo de Sonido . . . . .                               | 89        |
| 4.2.5. Descripción del Módulo de Visualización de información . . . . .         | 90        |
| 4.3. Descripción del modelo dinámico . . . . .                                  | 94        |
| 4.3.1. Diagrama de actividad general de la aplicación . . . . .                 | 94        |
| 4.3.2. Diagramas de secuencia . . . . .                                         | 96        |
| Caso de uso 1.1: Arrancar Aplicación Cliente . . . . .                          | 96        |
| Caso de uso 1.2: Mostrar patrón y obtener información . . . . .                 | 98        |

|                                                                                                                      |            |
|----------------------------------------------------------------------------------------------------------------------|------------|
| Caso de uso 1.3: Obtener geoposición e información asociada . . . . .                                                | 98         |
| Caso de uso 1.4: Desplazar Objeto . . . . .                                                                          | 99         |
| Caso de uso 1.5: Escalar Objeto . . . . .                                                                            | 100        |
| Casos de uso 1.6 y 1.7: Configuración de patrones y posiciones y<br>Configuración de objetos e información . . . . . | 101        |
| Caso de uso 1.8: Salir Aplicación Cliente . . . . .                                                                  | 103        |
| 4.4. Validación de los requisitos . . . . .                                                                          | 104        |
| 4.4.1. Requisitos comunes de los interfaces . . . . .                                                                | 104        |
| 4.4.2. Requisitos funcionales generales . . . . .                                                                    | 104        |
| 4.4.3. Requisitos funcionales de la aplicación para dispositivos móviles . .                                         | 105        |
| 4.4.4. Requisitos funcionales del servidor web . . . . .                                                             | 105        |
| 4.4.5. Requisitos funcionales de los ficheros de configuración . . . . .                                             | 105        |
| 4.4.6. Requisitos no funcionales . . . . .                                                                           | 106        |
| <b>5. Modelo de datos</b>                                                                                            | <b>107</b> |
| 5.1. Diseño conceptual: Modelo Entidad-Relación . . . . .                                                            | 107        |
| 5.1.1. Esquema Entidad-Relación . . . . .                                                                            | 108        |
| 5.1.2. Entidades . . . . .                                                                                           | 108        |
| 5.1.3. Relaciones . . . . .                                                                                          | 109        |
| <b>6. Pruebas software</b>                                                                                           | <b>111</b> |
| 6.1. Pruebas unitarias . . . . .                                                                                     | 111        |
| 6.1.1. Definición . . . . .                                                                                          | 111        |
| 6.1.2. Objetivos . . . . .                                                                                           | 112        |
| 6.1.3. Planificación . . . . .                                                                                       | 112        |
| 6.2. Pruebas de integración . . . . .                                                                                | 112        |
| 6.2.1. Definición . . . . .                                                                                          | 112        |
| 6.2.2. Objetivos . . . . .                                                                                           | 113        |
| 6.2.3. Planificación . . . . .                                                                                       | 113        |
| Módulo Carga de Datos. . . . .                                                                                       | 113        |
| Módulo de Localización . . . . .                                                                                     | 114        |
| Módulo de Renderización o Parser de objetos 3D . . . . .                                                             | 115        |
| Módulo de Sonido . . . . .                                                                                           | 116        |
| Módulo de Visualización de información . . . . .                                                                     | 116        |
| 6.3. Pruebas del sistema . . . . .                                                                                   | 118        |
| 6.3.1. Definición . . . . .                                                                                          | 118        |
| 6.3.2. Objetivos . . . . .                                                                                           | 118        |
| 6.3.3. Planificación . . . . .                                                                                       | 118        |
| 6.4. Pruebas de rendimiento . . . . .                                                                                | 119        |
| <b>7. Pruebas de usabilidad</b>                                                                                      | <b>121</b> |
| 7.1. Versiones Alpha . . . . .                                                                                       | 122        |
| 7.1.1. Alpha 0.1 (Prototipo no funcional) . . . . .                                                                  | 122        |
| Descripción y alcance . . . . .                                                                                      | 122        |
| Errores descubiertos y soluciones . . . . .                                                                          | 123        |
| 7.1.2. Alpha 0.2 (Versión servidor con visualización de archivos <i>3DS</i> ) . .                                    | 123        |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| Descripción y alcance . . . . .                                                | 123        |
| Errores descubiertos y soluciones . . . . .                                    | 123        |
| 7.2. Versiones Beta . . . . .                                                  | 124        |
| 7.2.1. Beta 1.0 (Versión con carga de datos y sonido) . . . . .                | 124        |
| Descripción y alcance . . . . .                                                | 124        |
| Errores descubiertos y soluciones . . . . .                                    | 124        |
| 7.2.2. Beta 1.1 (Versión con localización e interacción del usuario) . . . . . | 125        |
| Descripción y alcance . . . . .                                                | 125        |
| Errores descubiertos y soluciones . . . . .                                    | 126        |
| 7.2.3. Beta 1.2 (Versión con preferencias del usuario) . . . . .               | 127        |
| Descripción y alcance . . . . .                                                | 127        |
| Errores descubiertos y soluciones . . . . .                                    | 127        |
| 7.3. Probadores : Los usuarios finales . . . . .                               | 130        |
| <b>8. Presupuesto y Plan Temporal</b>                                          | <b>133</b> |
| 8.1. Presupuesto . . . . .                                                     | 133        |
| 8.1.1. Desarrollo del proyecto . . . . .                                       | 133        |
| 8.1.2. Tareas globales del proyecto . . . . .                                  | 134        |
| 8.1.3. Dedicación por perfiles . . . . .                                       | 136        |
| 8.1.4. Horas y tarifas por perfiles . . . . .                                  | 137        |
| 8.1.5. Costes finales y presupuesto total del proyecto . . . . .               | 137        |
| 8.2. Plan Temporal . . . . .                                                   | 138        |
| <b>9. Manual del programador</b>                                               | <b>141</b> |
| 9.1. Paquete com.ivanlorenzo.connection . . . . .                              | 141        |
| 9.2. Paquete com.ivanlorenzo.data . . . . .                                    | 141        |
| 9.2.1. Paquete com.ivanlorenzo.data.exceptions . . . . .                       | 142        |
| 9.2.2. Paquete com.ivanlorenzo.data.objects . . . . .                          | 142        |
| 9.2.3. Paquete com.ivanlorenzo.data.objects.audio . . . . .                    | 143        |
| 9.3. Paquete com.ivanlorenzo.gui . . . . .                                     | 143        |
| 9.3.1. Paquete com.ivanlorenzo.gui.dialogs . . . . .                           | 143        |
| 9.3.2. Paquete com.ivanlorenzo.gui.preferences . . . . .                       | 144        |
| 9.4. Paquete com.ivanlorenzo.location . . . . .                                | 144        |
| 9.4.1. Paquete com.ivanlorenzo.location.observer . . . . .                     | 144        |
| 9.5. Paquete com.ivanlorenzo.parser . . . . .                                  | 144        |
| <b>10. Manual de internacionalización</b>                                      | <b>147</b> |
| 10.1. Internacionalización de la interfaz . . . . .                            | 147        |
| 10.2. Internacionalización de los contenidos . . . . .                         | 149        |
| <b>11. Manual del instalador</b>                                               | <b>151</b> |
| 11.1. Requisitos . . . . .                                                     | 151        |
| 11.2. Instalación . . . . .                                                    | 151        |

|                                                           |            |
|-----------------------------------------------------------|------------|
| <b>12. Manual de usuario</b>                              | <b>155</b> |
| 12.1. Introducción . . . . .                              | 155        |
| 12.2. Guía de uso . . . . .                               | 155        |
| 12.2.1. Acciones del usuario sobre los objetos . . . . .  | 156        |
| Mover un objeto . . . . .                                 | 156        |
| Escalar un objeto . . . . .                               | 157        |
| 12.2.2. Opciones del menú de la aplicación . . . . .      | 157        |
| Establecer las preferencias de configuración . . . . .    | 158        |
| Salir de la aplicación . . . . .                          | 159        |
| <b>13. User's manual</b>                                  | <b>161</b> |
| 13.1. Introduction . . . . .                              | 161        |
| 13.2. Use guide . . . . .                                 | 161        |
| 13.2.1. User actions on objects . . . . .                 | 162        |
| Move an object . . . . .                                  | 162        |
| Scale an object . . . . .                                 | 162        |
| 13.2.2. Menu options of the application . . . . .         | 163        |
| Set configuration preferences . . . . .                   | 164        |
| Exit of the application . . . . .                         | 165        |
| <b>14. Conclusiones y Líneas de Investigación Futuras</b> | <b>167</b> |
| <b>Bibliografía</b>                                       | <b>171</b> |
| <b>A. Tecnologías</b>                                     | <b>173</b> |
| A.1. Android . . . . .                                    | 173        |
| A.1.1. Historia . . . . .                                 | 173        |
| A.2. JAVA . . . . .                                       | 175        |
| A.3. NyArtoolkit . . . . .                                | 175        |
| A.4. min3D . . . . .                                      | 175        |
| A.5. Tecnologías XML . . . . .                            | 176        |
| A.5.1. XML . . . . .                                      | 176        |
| A.5.2. XML Schema . . . . .                               | 176        |
| A.5.3. DOM . . . . .                                      | 176        |
| A.6. L <sup>A</sup> T <sub>E</sub> X . . . . .            | 177        |
| <b>B. Documentación</b>                                   | <b>179</b> |
| B.1. Paquete com.ivanlorenzo.data.objects.audio . . . . . | 181        |
| B.1.1. Clase <b>Audio</b> . . . . .                       | 181        |
| Declaración . . . . .                                     | 181        |
| Field summary . . . . .                                   | 181        |
| Constructor summary . . . . .                             | 181        |
| Method summary . . . . .                                  | 181        |
| Atributos . . . . .                                       | 181        |
| Constructores . . . . .                                   | 181        |
| Métodos . . . . .                                         | 182        |

|        |                                                                                                                    |     |
|--------|--------------------------------------------------------------------------------------------------------------------|-----|
| B.1.2. | <i>Clase</i> <b>LocalAudio</b>                                                                                     | 182 |
|        | Declaración                                                                                                        | 182 |
|        | Constructor summary                                                                                                | 182 |
|        | Method summary                                                                                                     | 182 |
|        | Constructores                                                                                                      | 182 |
|        | Métodos                                                                                                            | 183 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.data.objects.audio.Audio</code><br>(in B.1.1, page 181)       | 183 |
| B.1.3. | <i>Clase</i> <b>MediaPlayerAdapter</b>                                                                             | 183 |
|        | Declaración                                                                                                        | 183 |
|        | Field summary                                                                                                      | 183 |
|        | Constructor summary                                                                                                | 183 |
|        | Method summary                                                                                                     | 184 |
|        | Atributos                                                                                                          | 184 |
|        | Constructores                                                                                                      | 184 |
|        | Métodos                                                                                                            | 185 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.location.observer.ARObservable</code><br>(in B.3.3, page 213) | 185 |
| B.1.4. | <i>Clase</i> <b>RemoteAudio</b>                                                                                    | 185 |
|        | Declaración                                                                                                        | 186 |
|        | Field summary                                                                                                      | 186 |
|        | Constructor summary                                                                                                | 186 |
|        | Method summary                                                                                                     | 186 |
|        | Atributos                                                                                                          | 186 |
|        | Constructores                                                                                                      | 186 |
|        | Métodos                                                                                                            | 186 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.data.objects.audio.Audio</code><br>(in B.1.1, page 181)       | 187 |
| B.2.   | <b>Paquete</b> <code>com.ivanlorenzo.parser</code>                                                                 | 187 |
| B.2.1. | <i>Interfaz</i> <b>IParser</b>                                                                                     | 187 |
|        | Declaración                                                                                                        | 187 |
|        | Method summary                                                                                                     | 187 |
|        | Métodos                                                                                                            | 187 |
| B.2.2. | <i>Clase</i> <b>AParser</b>                                                                                        | 187 |
|        | Declaración                                                                                                        | 188 |
|        | Field summary                                                                                                      | 188 |
|        | Constructor summary                                                                                                | 188 |
|        | Method summary                                                                                                     | 188 |
|        | Atributos                                                                                                          | 189 |
|        | Constructores                                                                                                      | 189 |
|        | Métodos                                                                                                            | 190 |
| B.2.3. | <i>Clase</i> <b>AParser.BitmapAsset</b>                                                                            | 191 |
|        | Declaración                                                                                                        | 191 |
|        | Field summary                                                                                                      | 191 |
|        | Constructor summary                                                                                                | 191 |

|        |                                                                                                              |     |
|--------|--------------------------------------------------------------------------------------------------------------|-----|
|        | Atributos . . . . .                                                                                          | 191 |
|        | Constructores . . . . .                                                                                      | 192 |
| B.2.4. | <i>Clase</i> <b>AParser.Material</b> . . . . .                                                               | 192 |
|        | Declaración . . . . .                                                                                        | 192 |
|        | Field summary . . . . .                                                                                      | 192 |
|        | Constructor summary . . . . .                                                                                | 192 |
|        | Atributos . . . . .                                                                                          | 193 |
|        | Constructores . . . . .                                                                                      | 193 |
| B.2.5. | <i>Clase</i> <b>AParser.TextureAtlas</b> . . . . .                                                           | 193 |
|        | Declaración . . . . .                                                                                        | 193 |
|        | Field summary . . . . .                                                                                      | 193 |
|        | Constructor summary . . . . .                                                                                | 193 |
|        | Method summary . . . . .                                                                                     | 193 |
|        | Atributos . . . . .                                                                                          | 194 |
|        | Constructores . . . . .                                                                                      | 194 |
|        | Métodos . . . . .                                                                                            | 194 |
| B.2.6. | <i>Clase</i> <b>AParser.TextureAtlas.BitmapHeightComparer</b> . . . . .                                      | 195 |
|        | Declaración . . . . .                                                                                        | 195 |
|        | Constructor summary . . . . .                                                                                | 195 |
|        | Method summary . . . . .                                                                                     | 195 |
|        | Constructores . . . . .                                                                                      | 195 |
|        | Métodos . . . . .                                                                                            | 196 |
| B.2.7. | <i>Clase</i> <b>Max3DSParser</b> . . . . .                                                                   | 196 |
|        | Declaración . . . . .                                                                                        | 196 |
|        | Field summary . . . . .                                                                                      | 196 |
|        | Constructor summary . . . . .                                                                                | 196 |
|        | Method summary . . . . .                                                                                     | 196 |
|        | Atributos . . . . .                                                                                          | 197 |
|        | Constructores . . . . .                                                                                      | 197 |
|        | Métodos . . . . .                                                                                            | 198 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.parser.AParser</code><br>(in B.2.2, page 187) . . . . . | 199 |
| B.2.8. | <i>Clase</i> <b>MD2Parser</b> . . . . .                                                                      | 200 |
|        | Declaración . . . . .                                                                                        | 200 |
|        | Field summary . . . . .                                                                                      | 200 |
|        | Constructor summary . . . . .                                                                                | 200 |
|        | Method summary . . . . .                                                                                     | 200 |
|        | Atributos . . . . .                                                                                          | 201 |
|        | Constructores . . . . .                                                                                      | 201 |
|        | Métodos . . . . .                                                                                            | 201 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.parser.AParser</code><br>(in B.2.2, page 187) . . . . . | 202 |
| B.2.9. | <i>Clase</i> <b>MD2Parser.MD2Header</b> . . . . .                                                            | 202 |
|        | Declaración . . . . .                                                                                        | 202 |
|        | Field summary . . . . .                                                                                      | 202 |

|                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------|-----|
| Constructor summary . . . . .                                                                                | 203 |
| Method summary . . . . .                                                                                     | 203 |
| Atributos . . . . .                                                                                          | 203 |
| Constructores . . . . .                                                                                      | 203 |
| Métodos . . . . .                                                                                            | 203 |
| B.2.10. <i>Clase</i> <b>ObjParser</b> . . . . .                                                              | 204 |
| Declaración . . . . .                                                                                        | 204 |
| Field summary . . . . .                                                                                      | 204 |
| Constructor summary . . . . .                                                                                | 204 |
| Method summary . . . . .                                                                                     | 204 |
| Atributos . . . . .                                                                                          | 204 |
| Constructores . . . . .                                                                                      | 205 |
| Métodos . . . . .                                                                                            | 205 |
| Miembros heredados de la clase <code>com.ivanlorenzo.parser.AParser</code><br>(in B.2.2, page 187) . . . . . | 206 |
| B.2.11. <i>Clase</i> <b>ObjParser.ObjFace</b> . . . . .                                                      | 206 |
| Declaración . . . . .                                                                                        | 206 |
| Constructor summary . . . . .                                                                                | 206 |
| Constructores . . . . .                                                                                      | 206 |
| Miembros heredados de la clase <code>min3d.parser.ParseObjectFace</code> .                                   | 207 |
| B.2.12. <i>Clase</i> <b>ParseObjectData</b> . . . . .                                                        | 207 |
| Declaración . . . . .                                                                                        | 207 |
| Field summary . . . . .                                                                                      | 207 |
| Constructor summary . . . . .                                                                                | 207 |
| Method summary . . . . .                                                                                     | 207 |
| Atributos . . . . .                                                                                          | 208 |
| Constructores . . . . .                                                                                      | 208 |
| Métodos . . . . .                                                                                            | 208 |
| B.2.13. <i>Clase</i> <b>Parser</b> . . . . .                                                                 | 209 |
| Declaración . . . . .                                                                                        | 209 |
| Constructor summary . . . . .                                                                                | 209 |
| Method summary . . . . .                                                                                     | 210 |
| Constructores . . . . .                                                                                      | 210 |
| Métodos . . . . .                                                                                            | 210 |
| B.2.14. <i>Clase</i> <b>Parser.Type</b> . . . . .                                                            | 210 |
| Declaración . . . . .                                                                                        | 210 |
| Field summary . . . . .                                                                                      | 211 |
| Constructor summary . . . . .                                                                                | 211 |
| Method summary . . . . .                                                                                     | 211 |
| Atributos . . . . .                                                                                          | 211 |
| Constructores . . . . .                                                                                      | 211 |
| Métodos . . . . .                                                                                            | 211 |
| Miembros heredados de la clase <code>java.lang.Enum</code> . . . . .                                         | 211 |
| B.3. <b>Paquete</b> <code>com.ivanlorenzo.location.observer</code> . . . . .                                 | 212 |
| B.3.1. <i>Interfaz</i> <b>IARObservable</b> . . . . .                                                        | 212 |

|        |                                                                      |     |
|--------|----------------------------------------------------------------------|-----|
|        | Declaración . . . . .                                                | 212 |
|        | Method summary . . . . .                                             | 212 |
|        | Métodos . . . . .                                                    | 212 |
| B.3.2. | <i>Interfaz</i> <b>IARObserver</b> . . . . .                         | 212 |
|        | Declaración . . . . .                                                | 213 |
|        | Method summary . . . . .                                             | 213 |
|        | Métodos . . . . .                                                    | 213 |
| B.3.3. | <i>Clase</i> <b>ARObservable</b> . . . . .                           | 213 |
|        | Declaración . . . . .                                                | 213 |
|        | Field summary . . . . .                                              | 213 |
|        | Constructor summary . . . . .                                        | 213 |
|        | Method summary . . . . .                                             | 213 |
|        | Atributos . . . . .                                                  | 214 |
|        | Constructores . . . . .                                              | 214 |
|        | Métodos . . . . .                                                    | 214 |
| B.4.   | <b>Paquete com.ivanlorenzo.gui</b> . . . . .                         | 214 |
| B.4.1. | <i>Clase</i> <b>ARStatus</b> . . . . .                               | 214 |
|        | Declaración . . . . .                                                | 214 |
|        | Field summary . . . . .                                              | 215 |
|        | Constructor summary . . . . .                                        | 215 |
|        | Method summary . . . . .                                             | 215 |
|        | Atributos . . . . .                                                  | 215 |
|        | Constructores . . . . .                                              | 215 |
|        | Métodos . . . . .                                                    | 215 |
|        | Miembros heredados de la clase <code>java.lang.Enum</code> . . . . . | 215 |
| B.4.2. | <i>Clase</i> <b>ARToolkitDrawer</b> . . . . .                        | 216 |
|        | Declaración . . . . .                                                | 216 |
|        | Field summary . . . . .                                              | 216 |
|        | Constructor summary . . . . .                                        | 216 |
|        | Method summary . . . . .                                             | 216 |
|        | Atributos . . . . .                                                  | 217 |
|        | Constructores . . . . .                                              | 217 |
|        | Métodos . . . . .                                                    | 217 |
| B.4.3. | <i>Clase</i> <b>MessageWhat</b> . . . . .                            | 219 |
|        | Declaración . . . . .                                                | 219 |
|        | Field summary . . . . .                                              | 219 |
|        | Constructor summary . . . . .                                        | 219 |
|        | Method summary . . . . .                                             | 219 |
|        | Atributos . . . . .                                                  | 219 |
|        | Constructores . . . . .                                              | 219 |
|        | Métodos . . . . .                                                    | 220 |
|        | Miembros heredados de la clase <code>java.lang.Enum</code> . . . . . | 220 |
| B.4.4. | <i>Clase</i> <b>R</b> . . . . .                                      | 220 |
|        | Declaración . . . . .                                                | 220 |
|        | Constructor summary . . . . .                                        | 220 |

|         |                                          |     |
|---------|------------------------------------------|-----|
|         | Constructores . . . . .                  | 220 |
| B.4.5.  | <i>Clase</i> <b>R.attr</b> . . . . .     | 220 |
|         | Declaración . . . . .                    | 220 |
|         | Constructor summary . . . . .            | 221 |
|         | Constructores . . . . .                  | 221 |
| B.4.6.  | <i>Clase</i> <b>R.drawable</b> . . . . . | 221 |
|         | Declaración . . . . .                    | 221 |
|         | Field summary . . . . .                  | 221 |
|         | Constructor summary . . . . .            | 221 |
|         | Atributos . . . . .                      | 221 |
|         | Constructores . . . . .                  | 221 |
| B.4.7.  | <i>Clase</i> <b>R.id</b> . . . . .       | 221 |
|         | Declaración . . . . .                    | 221 |
|         | Field summary . . . . .                  | 222 |
|         | Constructor summary . . . . .            | 222 |
|         | Atributos . . . . .                      | 222 |
|         | Constructores . . . . .                  | 222 |
| B.4.8.  | <i>Clase</i> <b>R.layout</b> . . . . .   | 222 |
|         | Declaración . . . . .                    | 222 |
|         | Field summary . . . . .                  | 222 |
|         | Constructor summary . . . . .            | 222 |
|         | Atributos . . . . .                      | 222 |
|         | Constructores . . . . .                  | 223 |
| B.4.9.  | <i>Clase</i> <b>R.menu</b> . . . . .     | 223 |
|         | Declaración . . . . .                    | 223 |
|         | Field summary . . . . .                  | 223 |
|         | Constructor summary . . . . .            | 223 |
|         | Atributos . . . . .                      | 223 |
|         | Constructores . . . . .                  | 223 |
| B.4.10. | <i>Clase</i> <b>R.raw</b> . . . . .      | 223 |
|         | Declaración . . . . .                    | 223 |
|         | Field summary . . . . .                  | 224 |
|         | Constructor summary . . . . .            | 224 |
|         | Atributos . . . . .                      | 224 |
|         | Constructores . . . . .                  | 224 |
| B.4.11. | <i>Clase</i> <b>R.string</b> . . . . .   | 224 |
|         | Declaración . . . . .                    | 225 |
|         | Field summary . . . . .                  | 225 |
|         | Constructor summary . . . . .            | 225 |
|         | Atributos . . . . .                      | 225 |
|         | Constructores . . . . .                  | 226 |
| B.4.12. | <i>Clase</i> <b>R.xml</b> . . . . .      | 226 |
|         | Declaración . . . . .                    | 226 |
|         | Field summary . . . . .                  | 226 |
|         | Constructor summary . . . . .            | 227 |

|                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------|-----|
| Atributos                                                                                                  | 227 |
| Constructores                                                                                              | 227 |
| B.4.13. Clase <b>TFGARActivity</b>                                                                         | 227 |
| Declaración                                                                                                | 227 |
| Field summary                                                                                              | 227 |
| Constructor summary                                                                                        | 228 |
| Method summary                                                                                             | 228 |
| Atributos                                                                                                  | 229 |
| Constructores                                                                                              | 230 |
| Métodos                                                                                                    | 231 |
| Miembros heredados de la clase <code>android.app.Activity</code>                                           | 234 |
| Miembros heredados de la clase <code>android.view.ContextThemeWrapper</code>                               | 237 |
| Miembros heredados de la clase <code>android.content.ContextWrapper</code>                                 | 237 |
| Miembros heredados de la clase <code>android.content.Context</code>                                        | 239 |
| B.4.14. Clase <b>TFGARActivity.MainHandler</b>                                                             | 241 |
| Declaración                                                                                                | 241 |
| Constructor summary                                                                                        | 241 |
| Method summary                                                                                             | 241 |
| Constructores                                                                                              | 241 |
| Métodos                                                                                                    | 241 |
| Miembros heredados de la clase <code>android.os.Handler</code>                                             | 241 |
| B.4.15. Clase <b>TFGARActivity.PreviewCallback</b>                                                         | 242 |
| Declaración                                                                                                | 242 |
| Constructor summary                                                                                        | 242 |
| Method summary                                                                                             | 242 |
| Constructores                                                                                              | 242 |
| Métodos                                                                                                    | 242 |
| B.4.16. Clase <b>TFGARActivity.ScaleListener</b>                                                           | 243 |
| Declaración                                                                                                | 243 |
| Constructor summary                                                                                        | 243 |
| Method summary                                                                                             | 243 |
| Constructores                                                                                              | 243 |
| Métodos                                                                                                    | 243 |
| Miembros heredados de la clase <code>android.view.ScaleGestureDetector.SimpleOnScaleGestureListener</code> | 243 |
| B.4.17. Clase <b>Util</b>                                                                                  | 243 |
| Declaración                                                                                                | 243 |
| Constructor summary                                                                                        | 244 |
| Method summary                                                                                             | 244 |
| Constructores                                                                                              | 244 |
| Métodos                                                                                                    | 244 |
| B.5. Paquete <b>com.ivanlorenzo.data.objects</b>                                                           | 245 |
| B.5.1. Clase <b>Max3DSObject</b>                                                                           | 245 |

|        |                                                                                                                     |     |
|--------|---------------------------------------------------------------------------------------------------------------------|-----|
|        | Declaración . . . . .                                                                                               | 245 |
|        | Constructor summary . . . . .                                                                                       | 245 |
|        | Constructores . . . . .                                                                                             | 245 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.data.objects.Object3D</code><br>(in B.5.3, page 247) . . . . . | 246 |
| B.5.2. | <i>Clase</i> <b>MD2Object</b> . . . . .                                                                             | 246 |
|        | Declaración . . . . .                                                                                               | 246 |
|        | Constructor summary . . . . .                                                                                       | 246 |
|        | Constructores . . . . .                                                                                             | 246 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.data.objects.Object3D</code><br>(in B.5.3, page 247) . . . . . | 247 |
| B.5.3. | <i>Clase</i> <b>Object3D</b> . . . . .                                                                              | 247 |
|        | Declaración . . . . .                                                                                               | 247 |
|        | Field summary . . . . .                                                                                             | 247 |
|        | Constructor summary . . . . .                                                                                       | 247 |
|        | Method summary . . . . .                                                                                            | 248 |
|        | Atributos . . . . .                                                                                                 | 248 |
|        | Constructores . . . . .                                                                                             | 248 |
|        | Métodos . . . . .                                                                                                   | 249 |
| B.5.4. | <i>Clase</i> <b>Object3D.File3D</b> . . . . .                                                                       | 250 |
|        | Declaración . . . . .                                                                                               | 250 |
|        | Field summary . . . . .                                                                                             | 250 |
|        | Constructor summary . . . . .                                                                                       | 250 |
|        | Method summary . . . . .                                                                                            | 250 |
|        | Atributos . . . . .                                                                                                 | 250 |
|        | Constructores . . . . .                                                                                             | 251 |
|        | Métodos . . . . .                                                                                                   | 251 |
| B.5.5. | <i>Clase</i> <b>Object3DCreator</b> . . . . .                                                                       | 251 |
|        | Declaración . . . . .                                                                                               | 251 |
|        | Constructor summary . . . . .                                                                                       | 251 |
|        | Method summary . . . . .                                                                                            | 251 |
|        | Constructores . . . . .                                                                                             | 252 |
|        | Métodos . . . . .                                                                                                   | 252 |
| B.5.6. | <i>Clase</i> <b>OBJObject</b> . . . . .                                                                             | 252 |
|        | Declaración . . . . .                                                                                               | 253 |
|        | Constructor summary . . . . .                                                                                       | 253 |
|        | Constructores . . . . .                                                                                             | 253 |
|        | Miembros heredados de la clase <code>com.ivanlorenzo.data.objects.Object3D</code><br>(in B.5.3, page 247) . . . . . | 253 |
| B.6.   | <b>Paquete</b> <code>com.ivanlorenzo.data.exceptions</code> . . . . .                                               | 254 |
| B.7.   | <b>Paquete</b> <code>com.ivanlorenzo.location</code> . . . . .                                                      | 254 |
| B.7.1. | <i>Clase</i> <b>ARLocationListener</b> . . . . .                                                                    | 254 |
|        | Declaración . . . . .                                                                                               | 254 |
|        | Field summary . . . . .                                                                                             | 254 |
|        | Constructor summary . . . . .                                                                                       | 254 |

|                                                                                                                              |     |
|------------------------------------------------------------------------------------------------------------------------------|-----|
| Method summary . . . . .                                                                                                     | 254 |
| Atributos . . . . .                                                                                                          | 254 |
| Constructores . . . . .                                                                                                      | 255 |
| Métodos . . . . .                                                                                                            | 255 |
| Miembros heredados de la clase <code>com.ivanlorenzo.location.observer.ARObservable</code><br>(in B.3.3, page 213) . . . . . | 255 |
| B.7.2. <i>Clase</i> <b>LocationData</b> . . . . .                                                                            | 255 |
| Declaración . . . . .                                                                                                        | 256 |
| Field summary . . . . .                                                                                                      | 256 |
| Constructor summary . . . . .                                                                                                | 256 |
| Method summary . . . . .                                                                                                     | 256 |
| Atributos . . . . .                                                                                                          | 256 |
| Constructores . . . . .                                                                                                      | 256 |
| Métodos . . . . .                                                                                                            | 256 |
| B.8. <b>Paquete</b> <code>com.ivanlorenzo.gui.dialogs</code> . . . . .                                                       | 257 |
| B.8.1. <i>Clase</i> <b>DialogType</b> . . . . .                                                                              | 257 |
| Declaración . . . . .                                                                                                        | 257 |
| Field summary . . . . .                                                                                                      | 257 |
| Constructor summary . . . . .                                                                                                | 257 |
| Method summary . . . . .                                                                                                     | 258 |
| Atributos . . . . .                                                                                                          | 258 |
| Constructores . . . . .                                                                                                      | 258 |
| Métodos . . . . .                                                                                                            | 258 |
| Miembros heredados de la clase <code>java.lang.Enum</code> . . . . .                                                         | 258 |
| B.9. <b>Paquete</b> <code>com.ivanlorenzo.data</code> . . . . .                                                              | 258 |
| B.9.1. <i>Clase</i> <b>AugmentedRealityXMLCompiler</b> . . . . .                                                             | 258 |
| Declaración . . . . .                                                                                                        | 259 |
| Constructor summary . . . . .                                                                                                | 259 |
| Method summary . . . . .                                                                                                     | 259 |
| Constructores . . . . .                                                                                                      | 259 |
| Métodos . . . . .                                                                                                            | 259 |
| Miembros heredados de la clase <code>com.ivanlorenzo.data.XMLCompiler</code><br>(in B.9.7, page 270) . . . . .               | 260 |
| B.9.2. <i>Clase</i> <b>CompilerGenerator</b> . . . . .                                                                       | 260 |
| Declaración . . . . .                                                                                                        | 260 |
| Field summary . . . . .                                                                                                      | 260 |
| Constructor summary . . . . .                                                                                                | 260 |
| Method summary . . . . .                                                                                                     | 261 |
| Atributos . . . . .                                                                                                          | 261 |
| Constructores . . . . .                                                                                                      | 261 |
| Métodos . . . . .                                                                                                            | 261 |
| B.9.3. <i>Clase</i> <b>GeoLocation</b> . . . . .                                                                             | 261 |
| Declaración . . . . .                                                                                                        | 262 |
| Field summary . . . . .                                                                                                      | 262 |
| Constructor summary . . . . .                                                                                                | 262 |

|         |                                                                                                                |     |
|---------|----------------------------------------------------------------------------------------------------------------|-----|
|         | Method summary . . . . .                                                                                       | 262 |
|         | Atributos . . . . .                                                                                            | 262 |
|         | Constructores . . . . .                                                                                        | 262 |
|         | Métodos . . . . .                                                                                              | 263 |
| B.9.4.  | <i>Clase KnowledgeBase</i> . . . . .                                                                           | 263 |
|         | Declaración . . . . .                                                                                          | 263 |
|         | Field summary . . . . .                                                                                        | 263 |
|         | Constructor summary . . . . .                                                                                  | 264 |
|         | Method summary . . . . .                                                                                       | 264 |
|         | Atributos . . . . .                                                                                            | 264 |
|         | Constructores . . . . .                                                                                        | 264 |
|         | Métodos . . . . .                                                                                              | 265 |
| B.9.5.  | <i>Clase ObjectsXMLCompiler</i> . . . . .                                                                      | 266 |
|         | Declaración . . . . .                                                                                          | 266 |
|         | Constructor summary . . . . .                                                                                  | 266 |
|         | Method summary . . . . .                                                                                       | 267 |
|         | Constructores . . . . .                                                                                        | 267 |
|         | Métodos . . . . .                                                                                              | 267 |
|         | Miembros heredados de la clase <code>com.ivanlorenzo.data.XMLCompiler</code><br>(in B.9.7, page 270) . . . . . | 268 |
| B.9.6.  | <i>Clase Pattern</i> . . . . .                                                                                 | 268 |
|         | Declaración . . . . .                                                                                          | 268 |
|         | Field summary . . . . .                                                                                        | 268 |
|         | Constructor summary . . . . .                                                                                  | 268 |
|         | Method summary . . . . .                                                                                       | 268 |
|         | Atributos . . . . .                                                                                            | 269 |
|         | Constructores . . . . .                                                                                        | 269 |
|         | Métodos . . . . .                                                                                              | 269 |
| B.9.7.  | <i>Clase XMLCompiler</i> . . . . .                                                                             | 270 |
|         | Declaración . . . . .                                                                                          | 270 |
|         | Field summary . . . . .                                                                                        | 270 |
|         | Constructor summary . . . . .                                                                                  | 270 |
|         | Method summary . . . . .                                                                                       | 270 |
|         | Atributos . . . . .                                                                                            | 271 |
|         | Constructores . . . . .                                                                                        | 271 |
|         | Métodos . . . . .                                                                                              | 271 |
| B.10.   | <b>Paquete <code>com.ivanlorenzo.connection</code></b> . . . . .                                               | 271 |
| B.10.1. | <i>Clase ConnectionStatus</i> . . . . .                                                                        | 271 |
|         | Declaración . . . . .                                                                                          | 272 |
|         | Constructor summary . . . . .                                                                                  | 272 |
|         | Method summary . . . . .                                                                                       | 272 |
|         | Constructores . . . . .                                                                                        | 272 |
|         | Métodos . . . . .                                                                                              | 272 |
| B.11.   | <b>Paquete <code>com.ivanlorenzo.gui.preferences</code></b> . . . . .                                          | 272 |
| B.11.1. | <i>Clase TFGPreferencesActivity</i> . . . . .                                                                  | 272 |

|                                                                                                |            |
|------------------------------------------------------------------------------------------------|------------|
| Declaración . . . . .                                                                          | 272        |
| Constructor summary . . . . .                                                                  | 272        |
| Method summary . . . . .                                                                       | 273        |
| Constructores . . . . .                                                                        | 273        |
| Métodos . . . . .                                                                              | 273        |
| Miembros heredados de la clase <code>android.preference.PreferenceActivity</code><br>. . . . . | 273        |
| Miembros heredados de la clase <code>android.app.ListActivity</code> . . .                     | 273        |
| Miembros heredados de la clase <code>android.app.Activity</code> . . . . .                     | 274        |
| Miembros heredados de la clase <code>android.view.ContextThemeWrapper</code><br>. . . . .      | 276        |
| Miembros heredados de la clase <code>android.content.ContextWrapper</code><br>. . . . .        | 277        |
| Miembros heredados de la clase <code>android.content.Context</code> . . . .                    | 278        |
| <b>C. Código Fuente</b>                                                                        | <b>281</b> |
| C.1. Ficheros de configuración XML . . . . .                                                   | 281        |
| C.1.1. Fichero <code>patterns.xml</code> . . . . .                                             | 281        |
| <code>patterns.xml</code> . . . . .                                                            | 281        |
| C.1.2. Fichero <code>objects.xml</code> . . . . .                                              | 282        |
| <code>objects.xml</code> . . . . .                                                             | 282        |
| C.1.3. Fichero <code>patterns.xsd</code> . . . . .                                             | 284        |
| <code>patterns.xsd</code> . . . . .                                                            | 284        |
| C.1.4. Fichero <code>objects.xsd</code> . . . . .                                              | 284        |
| <code>objects.xsd</code> . . . . .                                                             | 284        |
| C.2. Ficheros de recursos de la aplicación Android . . . . .                                   | 285        |
| C.2.1. Layout . . . . .                                                                        | 285        |
| <code>main.xml</code> . . . . .                                                                | 285        |
| C.2.2. Menu . . . . .                                                                          | 286        |
| <code>menu.xml</code> . . . . .                                                                | 286        |
| C.2.3. Preferencias . . . . .                                                                  | 286        |
| <code>preferences.xml</code> . . . . .                                                         | 286        |
| C.2.4. Internacionalización . . . . .                                                          | 287        |
| <code>values/strings.xml</code> . . . . .                                                      | 287        |
| <code>values-en/strings.xml</code> . . . . .                                                   | 287        |
| <code>values-es/strings.xml</code> . . . . .                                                   | 288        |
| C.2.5. <code>AndroidManifest</code> . . . . .                                                  | 289        |
| <code>AndroidManifest.xml</code> . . . . .                                                     | 289        |
| C.3. JNI . . . . .                                                                             | 291        |
| C.4. Clases Java . . . . .                                                                     | 295        |
| C.4.1. Paquete <code>com.ivanlorenzo.data</code> . . . . .                                     | 295        |
| C.4.2. Paquete <code>com.ivanlorenzo.data.objects</code> . . . . .                             | 313        |
| C.4.3. Paquete <code>com.ivanlorenzo.data.objects.audio</code> . . . . .                       | 321        |
| C.4.4. Paquete <code>com.ivanlorenzo.data.exceptions</code> . . . . .                          | 327        |
| C.4.5. Paquete <code>com.ivanlorenzo.location</code> . . . . .                                 | 328        |
| C.4.6. Paquete <code>com.ivanlorenzo.location.observer</code> . . . . .                        | 332        |

|                                                           |     |
|-----------------------------------------------------------|-----|
| C.4.7. Paquete com.ivanlorenzo.parser . . . . .           | 334 |
| C.4.8. Paquete com.ivanlorenzo.connection . . . . .       | 369 |
| C.4.9. Paquete com.ivanlorenzo.gui . . . . .              | 370 |
| C.4.10. Paquete com.ivanlorenzo.gui.preferences . . . . . | 405 |
| C.4.11. Paquete com.ivanlorenzo.gui.dialogs . . . . .     | 406 |
| C.5. Pruebas JUnit . . . . .                              | 406 |
| DataTest.java . . . . .                                   | 406 |
| ParserTest.java . . . . .                                 | 408 |

# Índice de figuras

|                                                                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1. Esquema del proceso de <i>Scrum</i> . . . . .                                                                                                                             | 6  |
| 1.2. Página web para seguimiento del proyecto . . . . .                                                                                                                        | 9  |
| 2.1. Logotipo de Realidad Aumentada . . . . .                                                                                                                                  | 14 |
| 2.2. Realidad Aumentada . . . . .                                                                                                                                              | 15 |
| 2.3. Distintos markers . . . . .                                                                                                                                               | 15 |
| 2.4. Juego Invizimals de la PSP . . . . .                                                                                                                                      | 16 |
| 2.5. Diferencia entre VR y AR . . . . .                                                                                                                                        | 17 |
| 2.6. Funcionamiento de ARToolKit . . . . .                                                                                                                                     | 18 |
| 2.7. Estándares para redes de área local inalámbricas. . . . .                                                                                                                 | 21 |
| 2.8. Estándares de Wireless LAN. . . . .                                                                                                                                       | 21 |
| 2.9. El sistema de posicionamiento global es el principal sistema de localización<br>en entornos exteriores desarrollado hasta la fecha . . . . .                              | 24 |
| 2.10. Se emplean cuatro satélites para determinar las coordenadas tridimensio-<br>nales del receptor . . . . .                                                                 | 25 |
| 2.11. El sistema GLONASS . . . . .                                                                                                                                             | 27 |
| 2.12. Satélites GLONASS . . . . .                                                                                                                                              | 28 |
| 2.13. Diagrama del sistema “Galileo” . . . . .                                                                                                                                 | 29 |
| 2.14. <i>MagicBook</i> y su aplicación de enseñanza para volcanes . . . . .                                                                                                    | 31 |
| 2.15. Aplicación del proyecto ARiSE . . . . .                                                                                                                                  | 32 |
| 2.16. Es necesario un equipamiento especial para su uso . . . . .                                                                                                              | 32 |
| 2.17. Aplicación de <i>Brazos y Músculos</i> . . . . .                                                                                                                         | 33 |
| 2.18. Aplicación de <i>Biología</i> en la que se muestran los órganos . . . . .                                                                                                | 34 |
| 2.19. SciMorph . . . . .                                                                                                                                                       | 35 |
| 2.20. ZooBurst . . . . .                                                                                                                                                       | 36 |
| 2.21. ARSights . . . . .                                                                                                                                                       | 37 |
| 2.22. Wikitude . . . . .                                                                                                                                                       | 39 |
| 2.23. Wikitude ofrece información en tiempo real del sitio observado . . . . .                                                                                                 | 40 |
| 2.24. Layar . . . . .                                                                                                                                                          | 41 |
| 2.25. Layar muestra información de una capa . . . . .                                                                                                                          | 42 |
| 2.26. Imagen aérea de Calakmul . . . . .                                                                                                                                       | 43 |
| 2.27. Situación de Calakmul, Campeche . . . . .                                                                                                                                | 43 |
| 2.28. Demo de Realidad Aumentada donde los objetos digitales quedan super-<br>puestos al objeto real, en este caso, un libro en el que el usuario va pasando<br>hojas. . . . . | 45 |
| 2.29. Escena del proyecto ARCO . . . . .                                                                                                                                       | 46 |

|                                                                                                         |     |
|---------------------------------------------------------------------------------------------------------|-----|
| 2.30. Escena virtual del proyecto ARCHEOGUIDE . . . . .                                                 | 48  |
| 2.31. Sistema de ARCHEOGUIDE . . . . .                                                                  | 49  |
| 2.32. Simulación del sistema para la visualización de reconstrucciones en Ename . . . . .               | 50  |
| 2.33. Dinosaurios en el MUJA . . . . .                                                                  | 51  |
| 3.1. Diagrama de casos generales de uso . . . . .                                                       | 67  |
| 3.2. Diagrama de casos de uso específico para “Obtención de información adaptada al contexto” . . . . . | 68  |
| 3.3. Diagrama de casos de uso específico para “Manipular objetos 3D” . . . . .                          | 68  |
| 3.4. Diagrama de casos de uso específico para “Configurar opciones de la aplicación” . . . . .          | 69  |
| 3.5. Diagrama de casos de uso específico para “Publicar datos” . . . . .                                | 69  |
| 3.6. Boceto de la interfaz de la aplicación móvil . . . . .                                             | 75  |
| 3.7. Boceto de la interfaz al mostrar las opciones . . . . .                                            | 76  |
| 3.8. Boceto de la interfaz de configuración . . . . .                                                   | 76  |
| 4.1. Descomposición modular del sistema . . . . .                                                       | 77  |
| 4.2. Ejemplo de programa en otro idioma . . . . .                                                       | 81  |
| 4.3. Diagrama de clases del Módulo Carga de Datos . . . . .                                             | 85  |
| 4.4. Diagrama de clases del Módulo de Localización . . . . .                                            | 87  |
| 4.5. Diagrama de clases del módulo de Renderización o Parser de objetos 3D. . . . .                     | 88  |
| 4.6. Diagrama de clases del módulo de Sonido. . . . .                                                   | 90  |
| 4.7. Diagrama de clases del Módulo Visualización de información. . . . .                                | 92  |
| 4.8. Diagrama de actividad general del sistema. . . . .                                                 | 95  |
| 4.9. Diagrama de secuencia para el caso de uso 1.1 Arrancar Aplicación Cliente. . . . .                 | 96  |
| 4.10. Ciclo de vida de una Activity en Android . . . . .                                                | 97  |
| 4.11. Diagrama de secuencia para el caso de uso 1.2 Mostrar patrón y obtener información. . . . .       | 98  |
| 4.12. Diagrama de secuencia para el caso de uso 1.3 Obtener geoposición e información asociada. . . . . | 99  |
| 4.13. Diagrama de secuencia para el caso de uso 1.4 Desplazar Objeto. . . . .                           | 100 |
| 4.14. Diagrama de secuencia para el caso de uso 1.5 Escalar Objeto. . . . .                             | 100 |
| 4.15. Diagrama de secuencia para los casos de uso 1.6 y 1.7. . . . .                                    | 101 |
| 4.16. Diagrama de secuencia de la carga de datos. . . . .                                               | 102 |
| 4.17. Diagrama de secuencia para el caso de uso 1.8 Salir Aplicación Cliente. . . . .                   | 103 |
| 5.1. Diagrama entidad-relación del modelo de datos. . . . .                                             | 108 |
| 7.1. Una captura de la versión prototipo. . . . .                                                       | 122 |
| 7.2. Interfaz de la versión Alpha 0.2 . . . . .                                                         | 124 |
| 7.3. Interfaz de la versión Beta 1.1 . . . . .                                                          | 125 |
| 7.4. Prototipo para la versión final . . . . .                                                          | 126 |
| 7.5. Diseño de la pantalla de configuración . . . . .                                                   | 127 |
| 7.6. Editando en la pantalla de configuración . . . . .                                                 | 128 |
| 7.7. Pantala principal final . . . . .                                                                  | 128 |
| 7.8. Configuración final 1 . . . . .                                                                    | 129 |
| 7.9. Configuración final 2 . . . . .                                                                    | 129 |

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| 8.1. Estimación de horas del desarrollo del proyecto . . . . .                   | 134 |
| 8.2. Estimación de horas de todas las tareas del proyecto . . . . .              | 135 |
| 8.3. Dedicación de cada uno de los perfiles en las tareas del proyecto . . . . . | 136 |
| 8.4. Horas y tarifas para cada perfil . . . . .                                  | 137 |
| 8.5. Costes finales del presupuesto . . . . .                                    | 138 |
| 8.6. Diagrama de Gantt . . . . .                                                 | 139 |
| 8.7. Tareas y sus recursos asignados . . . . .                                   | 140 |
|                                                                                  |     |
| 11.1. Activar “Orígenes desconocidos” en Ajustes del teléfono . . . . .          | 152 |
| 11.2. Instalador de la aplicación . . . . .                                      | 152 |
| 11.3. Instalación finalizada . . . . .                                           | 153 |
|                                                                                  |     |
| 12.1. Pantalla principal de la aplicación . . . . .                              | 155 |
| 12.2. Desplazamiento de un objeto . . . . .                                      | 156 |
| 12.3. Escalado de un objeto . . . . .                                            | 157 |
| 12.4. Opciones del menú de aplicación . . . . .                                  | 157 |
| 12.5. Opciones de configuración . . . . .                                        | 158 |
| 12.6. Salir de la aplicación . . . . .                                           | 159 |
|                                                                                  |     |
| 13.1. Main application screen . . . . .                                          | 161 |
| 13.2. Moving an object . . . . .                                                 | 162 |
| 13.3. Scaling an object . . . . .                                                | 163 |
| 13.4. Menu options of the application . . . . .                                  | 163 |
| 13.5. Configuration options . . . . .                                            | 164 |
| 13.6. Exit of the application . . . . .                                          | 165 |
|                                                                                  |     |
| A.1. Sistema operativo Android . . . . .                                         | 174 |

# Índice de cuadros

|                                                                                                                                |     |
|--------------------------------------------------------------------------------------------------------------------------------|-----|
| 1.1. Comparativa entre las características básicas ágiles y los rasgos observados en el desarrollo de software móvil . . . . . | 7   |
| 3.35. Especificación de Caso de Uso 1.1 . . . . .                                                                              | 70  |
| 3.36. Especificación de Caso de Uso 1.2 . . . . .                                                                              | 70  |
| 3.37. Especificación de Caso de Uso 1.3 . . . . .                                                                              | 71  |
| 3.38. Especificación de Caso de Uso 1.4 . . . . .                                                                              | 71  |
| 3.39. Especificación de Caso de Uso 1.5 . . . . .                                                                              | 71  |
| 3.40. Especificación de Caso de Uso 1.6 . . . . .                                                                              | 72  |
| 3.41. Especificación de Caso de Uso 1.7 . . . . .                                                                              | 72  |
| 3.42. Especificación de Caso de Uso 1.8 . . . . .                                                                              | 72  |
| 3.43. Especificación de Caso de Uso 1.9 . . . . .                                                                              | 73  |
| 3.44. Especificación de Caso de Uso 1.10 . . . . .                                                                             | 73  |
| 3.45. Especificación de Caso de Uso 1.11 . . . . .                                                                             | 73  |
| 3.46. Especificación de Caso de Uso 1.12 . . . . .                                                                             | 74  |
| 3.47. Especificación de Caso de Uso 1.13 . . . . .                                                                             | 74  |
| 4.1. Validación de los requisitos comunes de los interfaces . . . . .                                                          | 104 |
| 4.2. Validación de los requisitos funcionales generales . . . . .                                                              | 104 |
| 4.3. Validación de los requisitos funcionales de la aplicación para dispositivos móviles . . . . .                             | 105 |
| 4.4. Validación de los requisitos funcionales del servidor web . . . . .                                                       | 105 |
| 4.5. Validación de los requisitos funcionales de los ficheros de configuración . . . . .                                       | 105 |
| 4.6. Validación de los requisitos no funcionales . . . . .                                                                     | 106 |

# Capítulo 1

## Introducción

El ser humano se ha distinguido por la capacidad de elaborar herramientas con el fin de aumentar sus capacidades físicas; desde los primeros cuchillos de piedra hasta los modernos aviones de reacción, podemos decir que hemos recorrido un buen trecho en aumentar las capacidades de nuestros sentidos a través de tejidos sintéticos, televisores, perfumes, lentes y microscopios.

Mediante la tecnología hemos logrado avances que, hasta hace poco, parecían de ficción, tal es el caso de lo que hoy en día conocemos como *Realidad Aumentada*[1]; que tiene como finalidad superponer, al entorno real, la información que nos interesa visualizar.

La realidad aumentada se centra en la integración de elementos virtuales en el entorno físico que nos rodea. A diferencia de la realidad virtual, que utiliza dispositivos especiales para interpretar las acciones del usuario, la realidad aumentada pretende que los dispositivos potencien sus actividades proporcionándoles en cada momento información no intrusiva acerca de los objetos reales con los que interacciona y mediante un seguimiento de las mismas.

### 1.1. Motivación del proyecto

#### 1.1.1. En el contexto educativo

Durante años, los sistemas educativos optaron por una escuela más formal y garantizadora de que un mínimo es aprendido y de que todos aprenden lo mismo, eliminando la riqueza de la diversidad en el aprendizaje en el proceso, pero asegurando (al menos así lo pretenden) un mínimo de competencia académica. Esto, desde el punto de vista de muchos alumnos, suponía una educación “aburrida” y poco motivante.

Estudios de Jerome S. Bruner [2] resaltan el papel de la actividad como parte esencial de todo proceso de aprendizaje. Bruner añade que la condición indispensable para aprender una información de manera significativa, es tener la experiencia personal de descubrirla: **el descubrimiento fomenta el aprendizaje significativo.**

## 1. Introducción

---

Atribuye una gran importancia a la actividad directa de los individuos sobre la realidad. Por otro lado plantea que los profesores deberían variar sus estrategias metodológicas de acuerdo al estado de evolución y desarrollo de los alumnos.

En este tipo de aprendizaje el alumno tiene una gran participación. El docente no expone los contenidos de un modo acabado, su actividad se dirige a darles a conocer una meta que ha de ser alcanzada y además de servir como mediador y guía para que los alumnos sean los que recorran el camino y alcancen los objetivos propuestos.

En otras palabras, el aprendizaje por descubrimiento se produce cuando el docente le presenta todas las herramientas necesarias al alumno para que este descubra por sí mismo lo que se desea aprender.

Constituye un aprendizaje muy efectivo, pues cuando se lleva a cabo de modo idóneo, asegura un conocimiento significativo y fomenta hábitos de investigación y rigor en los individuos.

Afortunadamente, en los últimos años hemos presenciado una revolución en la tecnología que ha afectado irreversiblemente al desarrollo de las técnicas de exposición. Ahora es posible incorporar a nuestro sistema tradicional y conservador técnicas y recursos que hacen posible recuperar los criterios antes mencionados e incorporarlos al ejercicio de la docencia de un modo que divierta y motive a los alumnos sin necesidad de perder ni un ápice de rigor formal.

Mediante este proyecto y utilizando técnicas de realidad aumentada queremos contribuir a ello.

La realidad aumentada puede ayudar a los estudiantes a reforzar y contextualizar mejor su aprendizaje. La creación de contenidos altamente interactivos apoya el proceso de aprendizaje de diversas formas, entre ellas brindando soporte a la adquisición de conocimientos procedimentales que son esenciales para relacionar y entender los conceptos aprendidos, mediante la interacción con los recursos que se encuentran alrededor del individuo, es decir, que hacen parte de la realidad.

### 1.1.2. En otras áreas

Aunque el proyecto se centra en el ámbito educativo, se podría haber aplicado a cualquier otro área como la arquitectura, cirugía, industria, publicidad, entretenimiento, decoración de interiores o exposiciones museísticas.

Centrándonos en las visitas a museos, el visitante obtiene la información procedente de folletos, libros, paneles informativos, guías, etc. Este tipo de visitas no distingue entre los diferentes tipos de usuarios, limitando a aquellos que deseen profundizar en algún aspecto en concreto.

Como se ha presentado, la información que un visitante actualmente puede obtener en una visita a un museo está muy dispersa, lo cual le influye negativamente. Sería ideal disponer de un sistema que sea capaz de reunir todo el conocimiento, que esté disponible durante la visita y además se adapte a las necesidades del visitante, garantizando así la plena satisfacción del mismo.

La mayoría de los museos no tienen el espacio y los recursos requeridos para exhibir sus colecciones enteras. Además, la naturaleza y la fragilidad de algunos objetos obligan al museo a no ponerlos a disposición del público. También, la interacción de los visitantes con los artefactos exhibidos es muy limitada, por ejemplo no pueden mirar los artefactos desde todos los ángulos, comparar los artefactos o estudiarlos en diversos contextos.

Hoy en día, los visitantes de un museo o un centro de divulgación, sea cual sea su disciplina, desean conocer, vivir, comprender, ver y tocar, aunque sea de forma simulada; en definitiva pasarlo bien y, a lo mejor y en el mejor de los casos, aprender algo. Es necesario, si se desea lograr una comunicación efectiva, implicarlos, hacerles sentir partícipes. Es básico no ahorrar esfuerzos en el intento de conseguir que el visitante establezca un diálogo a su nivel con el propio contenido.

La utilización de estas nuevas tecnologías permite técnicas de exhibición diferente, participativa e interactiva, que permiten un contacto mayor con el público. El visitante logra un papel activo en el proceso de difusión y le proporciona una herramienta que suple su falta de conocimientos, invitándole a entablar el diálogo con las piezas o acontecimientos. La participación activa proporciona al visitante una experiencia multisensorial.

Algunos exámenes recientes en Europa demuestran que cerca del 35 % de los museos ya han comenzado a realizar una presentación 3D de los objetos. En muchos casos, éstos son solamente proyectos en una etapa inicial, pero el número está creciendo rápidamente y es evidente, que los museos comienzan a reconocer el potencial ofrecido por estas tecnologías.

### 1.1.3. ¿Qué se pretende con este proyecto?

En definitiva, con este proyecto se intenta crear una plataforma de soporte a realidad aumentada para la representación y obtención de información extra de objetos 3D. Gracias a ello aumentamos la interactividad de los usuarios entre ellos y con el propio entorno.

Se pretende crear un nuevo concepto de visita, que se adapte a las necesidades del usuario, que proporcione mayor precisión en la información y que, gracias a una gran interactividad, determine en todo momento la cantidad y calidad de información que desee obtener, adaptándose, por tanto, a su propio perfil. El visitante verá reconstrucciones virtuales de los cuadros u objetos donde se encuentre, permitiéndole entender mejor lo que está observando.

## 1. Introducción

---

Así se permitirá al usuario liberarse de la pasividad receptora e introducirse en una dinámica activa y participativa. Con ello se estimulará el interés y el conocimiento con una orientación interdisciplinar que permite un aprendizaje lúdico, frente a los modelos tradicionales de educación que son considerados pasivos y aburridos.

Por lo tanto, gracias a esta plataforma se podrán realizar diferentes aplicaciones, simplemente con un cambio de información, ya que la base será la misma para todas.

Así, gracias a este proyecto el usuario de este modelo de navegación podrá:

- Ver los objetos tridimensionales e interactuar con ellos.
- Mostrar objetos bien en función de la posición en la que nos encontremos, o bien al reconocer unos patrones o *markers* a través de la cámara
- Escuchar información de cada objeto.
- Se podrán manipular los objetos: aumentar y reducir el tamaño y trasladarlos en todas las direcciones
- Todo lo anterior, se ejecutará en dispositivos móviles

## 1.2. Metodología empleada

### 1.2.1. Introducción

En muchas ocasiones, las metodologías tradicionales no nos sirven para afrontar un reto que hoy en día resulta fundamental: **incorporar cambios con rapidez y en cualquier fase del proyecto.**

Las metodologías ágiles surgen para ser una alternativa a los procesos de desarrollo tradicionales caracterizados por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas.

Su propósito es el de **evitar que los proyectos se retrasen o fracasen y aumentar la calidad del software** que se desarrolla.

Dado que los cambios nunca van a dejar de existir, lo que necesitamos es ser capaces de gestionar los proyectos de una forma más ágil.

Con ese objetivo, en los años 80 los japoneses Takeuchi y Nonaka estudiaron las prácticas de empresas con buenos resultados de rapidez y flexibilidad en la producción: Xerox, Canon, Honda, NEC, Epson, Brother, 3M o Hewlett-Packard. De ahí extrajeron la base de la metodología **SCRUM** que, aunque nació en el ámbito tecnológico, ha ido creciendo hasta consolidarse en campos de actividad muy diferentes.

### 1.2.2. SCRUM

*Scrum* distingue entre dos elementos principales: actores y acciones.

Las acciones son las distintas fases del ciclo de desarrollo de Scrum y los actores son las personas que ejecutarán las acciones.

En cuanto a los actores se pueden diferenciar básicamente cuatro roles:

- **Product Owner:** es el responsable del proyecto por parte del cliente. Se encarga de definir las prioridades de las tareas a realizar, evaluar el avance del proyecto desde el punto de vista del cliente y, en general, servir de punto de comunicación entre el cliente y el equipo de desarrollo.
- **Scrum Master:** es la persona que se encarga de que se cumplan los principios de la metodología, así como servir de guía para los integrantes del equipo a través de reuniones.
- **Scrum Team:** son el equipo de desarrollo del proyecto. Se encargan de implementar las funciones requeridas por el *Product Owner*.
- **Usuarios o Clientes:** son los usuarios finales de la aplicación. A partir del progreso de la aplicación, pueden aportar nuevas ideas de modo que la aplicación final se adapte a sus necesidades personales.

Por otro lado, las acciones de Scrum forman parte de un ciclo iterativo repetitivo, por lo que el mecanismo y forma de trabajar que a continuación se indica, tiene como objetivo minimizar el esfuerzo y maximizar el rendimiento en el desarrollo.

Las acciones fundamentales de Scrum son:

- **Product Backlog:** es una lista de todas las tareas, funcionalidades o requerimientos del proyecto. El *Product Owner* se encarga de priorizar estas tareas y de actualizar la lista con los objetivos conseguidos.
- **Sprint Backlog:** es un periodo de entre una y cuatro semanas en el cual se deben realizar una serie de tareas sacadas del *Product Backlog*. La duración y las tareas a realizar en el *Sprint Backlog* se deciden antes del mismo, y se mantienen invariables hasta la finalización del mismo.
- **Daily Scrum Meeting:** es una reunión diaria, entre los miembros del equipo, que se realiza durante el tiempo que dura el *Sprint*. Debe tratarse de una reunión informal y ágil, de no más de 30 minutos, y en la cual se le plantean las 3 siguientes preguntas a los miembros del equipo:
  - ¿Qué has hecho desde la última reunión?
  - ¿Qué vas a hacer hoy?
  - ¿Qué obstáculos se te presentan? Tras la respuesta a esta pregunta debe actuar el *Scrum Master*, que se encargará de eliminar dichos obstáculos.

## 1. Introducción

---

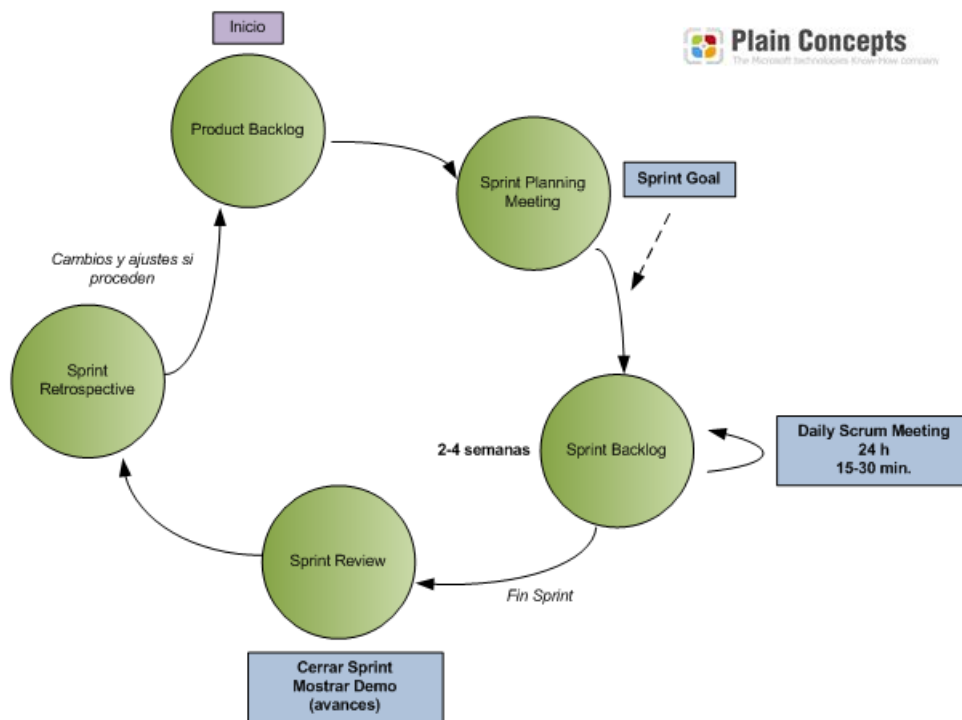


Figura 1.1: Esquema del proceso de *Scrum*

Además de estas acciones, se pueden encontrar otras secundarias que ayudan a mantener un avance ordenado del proyecto, así como permitir que el cliente pueda ir viendo este avance. Estas acciones son:

- **Sprint Planning Meeting:** es una reunión que tiene por objetivo, planificar el *Sprint* a partir del *Product Backlog*. El objetivo de esta reunión es la de mover las tareas del *Product Backlog* al *Sprint Backlog*. En esta reunión, suelen participar el *Product Owner* que es quien prioriza las tareas, el *Scrum Master* y el *Scrum Team*. Del *Sprint Planning Meeting*, sale también el **Sprint Goal**, que es un pequeño documento o una breve descripción que indica lo que el *Sprint* intentará alcanzar.
- **Sprint Review:** se revisa en unas 2 horas como máximo el *Sprint* finalizado. Al llegar a este punto, debemos tener “algo” que el Cliente o el Usuario pueda ver y tocar. En esta reunión, suelen asistir el *Product Owner*, el *Scrum Master*, el *Scrum Team* y personas que podrían estar involucradas en el proyecto. El *Scrum Team* es quién muestra los avances realizados en el *Sprint*.
- **Sprint Retrospective:** Se inicia al finalizar un *Sprint Backlog* y el *Sprint Review*. El *Product Owner* revisará con el equipo los objetivos marcados inicialmente en el *Sprint Backlog* concluido, se aplicarán los cambios y ajustes si son necesarios, y se marcarán los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) del *Sprint*.

En el diagrama 1.1 se puede ver todo el proceso completo de la metodología.

### 1.2.3. Adecuación de la metodología al proyecto

Las metodologías ágiles poseen ciertas propiedades que las hacen totalmente aplicables al dominio del software en los móviles.

En [3] se hace un análisis comparativo para probar la idoneidad de los métodos ágiles sobre el desarrollo de software para móviles. Esas características ideales y su motivación en cada caso se muestran en la siguiente tabla:

| Características ágiles                      | Motivación logística                                                                                                                                                                        | En el caso del desarrollo para plataformas móviles                                                                       |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Alta volatilidad del entorno                | Debido a la alta frecuencia en el cambio que sufren los requisitos, tendremos menos necesidad de diseño y planificación inicial y mayor necesidad de desarrollos incrementales e iterativos | Alta incertidumbre, entornos dinámicos, cientos de nuevos terminales cada año                                            |
| Equipos de desarrollo pequeños              | Capacidad de reacción más rápida, trabajo basado en la compartición de la información, menos documentación                                                                                  | La mayor parte de los proyectos de desarrollo software para plataformas móviles se lleva a cabo en microempresas y PyME. |
| Cliente identificable                       | Desaparecen los malentendidos                                                                                                                                                               | Potencialmente, hay un número ilimitado de usuarios finales, pero los clientes son fáciles de identificar                |
| Entornos de desarrollo orientados a objetos | Mayoría de las herramientas de desarrollo ágil existen bajo plataformas orientadas a objetos                                                                                                | Por ejemplo, Java y C++ se usan, algunos problemas en herramientas como refactorizaciones o primeros tests               |
| Software crítico no asegurado               | Los fallos no causan gran impacto, como la pérdida de vidas. Se puede buscar mayor agilidad en el desarrollo                                                                                | La mayoría del software es para entretenimiento. Los terminales no son fiables                                           |
| Software a nivel de aplicación              | Sistemas embebidos grandes requieren comunicación exhaustiva y mecanismos de verificación                                                                                                   | Mientras los sistemas móviles son complejos y altamente dependientes, las aplicaciones son muy autónomas                 |
| Sistemas pequeños                           | Menos necesidad de diseño inicial                                                                                                                                                           | Las aplicaciones, aunque variables en tamaño, no suelen superar las 10.000 líneas de código                              |
| Ciclos de desarrollo cortos                 | Propósito de realimentación rápida                                                                                                                                                          | Periodos de desarrollo de 1 a 6 meses                                                                                    |

Cuadro 1.1: Comparativa entre las características básicas ágiles y los rasgos observados en el desarrollo de software móvil

## 1. Introducción

---

Las metodologías de desarrollo tienen como objetivo convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad, que satisfaga las necesidades y expectativas del cliente, y a su vez, se convierta en un proceso estándar en la organización y en los equipos de desarrollo.

Pero las metodologías están pensadas para aunar esfuerzos de un amplio grupo de personas involucradas en el desarrollo.

Partiendo de ese punto se buscará la forma de aplicar la metodología de forma que verdaderamente repercuta beneficios teniendo presente que éstas nos ayudarán a gestionar las necesidades del proyecto de forma estructurada, a predecir los posibles cambios, a disminuir costes y retrasos, y en definitiva, a mejorar la calidad del software y evitar rechazos de los clientes.

Para adaptarnos a Scrum en primer lugar tendremos que adoptar varios 4 roles. En un trabajo como este en el que el grupo consta de una única persona, resulta difícil. Podemos establecer algo como lo siguiente:

- **Product Owner:** podría ser el director del proyecto, ya que se encarga de controlar que el trabajo va en la dirección adecuada.
- **Scrum Máster:** este rol podría dividirse entre el director del proyecto y el propio proyectando, aunque la mayor parte del trabajo tendrá que ser realizado por este último.
- **Scrum Team:** el equipo de desarrollo se compondrá del proyectando que implementará las opciones requeridas.
- **Usuarios o Clientes:** el destino final de un trabajo fin de grado es el Tribunal. Podría considerarse que éstos son los usuarios de nuestro “sistema”. También pueden englobarse en este apartado los usuarios que hagan las pruebas de usabilidad del producto.

En cuanto a las acciones podemos adoptar lo siguiente:

- **Product Backlog:** no es más que una declaración de intenciones. Es una lista de tareas que se priorizará por parte del *Product Owner*.
- **Sprint Backlog:** se pueden realizar varios sprints backlogs a lo largo del proyecto.
- **Daily Scrum Meeting:** es muy difícil que un director de proyecto esté dispuesto a realizar la reunión diaria. Los miembros del tribunal serán desconocidos hasta que el proyecto esté terminado. Entonces la reunión diaria se quedará en una introspección por parte del proyectando en el que examinará lo realizado el último día y lo que se va a hacer en el día actual.

### 1.2.4. Seguimiento del proyecto

Para realizar el seguimiento del Trabajo Fin de Grado se ha creado una página web en la que se suben las distintas versiones de la memoria, así como videos mostrando la evolución a lo largo del desarrollo. También se añaden ejemplos de ficheros de configuración.

Esta página web es <http://tfg.ivanlorenzo.es/>



Figura 1.2: Página web para seguimiento del proyecto

## 1.3. Estructura del documento

Esta memoria se encuentra dividida en catorce capítulos y tres apéndices, los cuáles describiré brevemente a continuación.

### 1.3.1. Capítulo 1. Introducción

Este es el capítulo en el que nos encontramos. En él se realiza una descripción de cuál ha sido la motivación del proyecto, para centrar las bases de éste. Por otro lado, se explica

## 1. Introducción

---

la metodología elegida para el desarrollo de éste. Además, también se incorpora una breve descripción del contenido de los diferentes capítulos de la memoria.

### 1.3.2. Capítulo 2. Estado del arte

Es el punto de partida de cualquier proyecto. Antes de ponerse a desarrollar nada, es necesario investigar el mercado y analizar las soluciones similares que ya están creadas, es decir, asegurarse de no “inventar la rueda” sino mejorarla.

En este capítulo lo que se hace es un análisis tanto cualitativo, como cuantitativo y comparativo de algunas de las aplicaciones de Realidad Aumentada existentes en el mercado.

### 1.3.3. Capítulo 3. Análisis

En el análisis lo que se pretende es definir el “qué” queremos desarrollar. En esta sección se hace un estudio de cuáles son los requisitos que ha de satisfacer el proyecto.

Esta sección concluye con la definición de cuáles serán los casos de uso a los que la aplicación ha de dar soporte.

### 1.3.4. Capítulo 4. Diseño

En esta sección, a lo que se pretende responder es al “cómo” será implementada la aplicación. Es aquí donde se definirá de forma detallada la estructura del nuevo sistema y en la que se definirá el comportamiento del sistema ante las condiciones de entrada preestablecidas.

Este capítulo concluye con un checklist que muestra en que puntos del diseño se han satisfecho todos los requisitos del análisis.

### 1.3.5. Capítulo 5. Modelo de datos

En este capítulo, se muestra primero de una manera gráfica y posteriormente de una manera textual, el modelo de datos que utilizará nuestro sistema.

### 1.3.6. Capítulo 6. Pruebas software

En esta sección, se encuentra la documentación de una de las etapas más importantes del proceso de desarrollo de software. En ella se encuentra la documentación de las pruebas de módulos y de las pruebas de conjunto.

### 1.3.7. Capítulo 7. Pruebas de usabilidad

En los tiempos en los que vivimos, la usabilidad es un aspecto fundamental con el que debe de cumplir cualquier aplicación que se preste. Para ello, se realizaron unas exhaustivas pruebas de usabilidad, cuyo resultado se encuentra documentado en esta sección.

### 1.3.8. Capítulo 8. Presupuesto y Plan Temporal

Incluye un listado exhaustivo de las tareas requeridas para el desarrollo del proyecto, su duración y su planificación temporal. Se muestran las tareas que se ejecutan en paralelo en el diagrama correspondiente. Del mismo modo, se presupuesta el coste de ejecución del material fungible y del inventariable, así como posibles dietas por viajes.

### 1.3.9. Capítulo 9. Manual del programador

Éste manual contiene una referencia rápida para que el programador que desee modificar la aplicación en un futuro, sirva de ayuda para saber cómo se encuentran ordenadas las clases y para que sirve cada una.

### 1.3.10. Capítulo 10. Manual de internacionalización

Éste manual contiene una referencia rápida para que el programador que desee introducir un nuevo lenguaje a la aplicación en un futuro, sirva como guía para poder realizar dicho cambio.

### 1.3.11. Capítulo 11. Manual del instalador

En éste capítulo, lo que se pretende es dar una visión descriptiva del proceso de instalación de las diferentes herramientas necesarias para la explotación posterior del proyecto.

### 1.3.12. Capítulo 12. Manual de usuario

En el manual de usuario, se muestra de una forma detallada el proceso para realizar cualquier operación con nuestra herramienta.

### 1.3.13. Capítulo 13. User's manual

In the user's manual, all the actions that can be done by the users with the tools will be described one by one.

### 1.3.14. Capítulo 14. Conclusiones y líneas de investigación futuras

Éste último capítulo recoge las conclusiones sacadas del proceso de elaboración del proyecto. Además, se dejan abiertas una serie de vías de evolución del mismo.

### 1.3.15. Apéndice A. Tecnologías

En este apéndice, se incluye una breve descripción de todas y cada una de las tecnologías que han sido utilizadas durante el proceso de desarrollo de este proyecto.

### 1.3.16. Apéndice B. Documentación

En éste apéndice se incluirá la documentación del código fuente.

### 1.3.17. Apéndice C. Código Fuente

En éste apéndice se mostrarán los listados de código que forman la aplicación.

# Capítulo 2

## Estado del arte

Hoy en día no existe ninguna plataforma de realidad aumentada para dispositivos móviles totalmente configurable según las necesidades. Aunque si que, recientemente, están apareciendo algunas aplicaciones que fomenten el aprendizaje mediante este tipo de técnicas. También existen otro tipos de proyectos de realidad aumentada enfocados a otras áreas y que tienen una base en común con lo que aquí se ofrece. A continuación realizo un estudio significativo de los sistemas que se asemejan a lo que se propone con esta plataforma.

### 2.1. Introducción

El fin último de todo proceso de aprendizaje es adquirir nuevos conocimientos, habilidades o actitudes que produzcan algún cambio en nuestro modo de ser o de actuar.

Sin embargo, no todos los alumnos se sienten motivados y logran estos objetivos. No es ninguna novedad que los alumnos no aprenden igual, ni a la misma velocidad. En un grupo en que se empiece a estudiar una materia partiendo del mismo nivel, se producirá una diferencia de conocimientos entre los estudiantes, en no mucho tiempo, pese a recibir las mismas explicaciones y realizar las mismas actividades.

Para combatir esto, tratamos de aprovechar las ventajas de las nuevas tecnologías y fomentar un método de aprendizaje más participativo, fácil, intuitivo y divertido.

A continuación trataré de clasificar y medir del modo más objetivo posible un grupo de sistemas basados en realidad aumentada, para poder obtener una visión global de lo que hay hasta el momento, sus ventajas, sus puntos fuertes, y sus posibles errores y puntos a mejorar. Pero antes de todo esto veremos una pequeña introducción al concepto de realidad aumentada, las diferencias con la realidad virtual y diversas tecnologías que se emplean en la plataforma como por ejemplo Wi-fi o GPS.

## 2.2. Realidad Aumentada

### 2.2.1. Introducción

La Realidad Aumentada (RA) es un paradigma de interacción que trata de reducir las interacciones con el dispositivo utilizando la información del entorno como una entrada implícita. Con este estilo el usuario será capaz de interactuar con el mundo real, aumentado por la información sintética del ordenador.

La situación del usuario será automáticamente reconocida utilizando un amplio conjunto de métodos de reconocimiento como tiempo, posición o inspección de objetos.



Figura 2.1: Logotipo de Realidad Aumentada

Se trata, pues, de una nueva tecnología que aumenta o mejora la visión que el usuario tiene del mundo real con información adicional sintetizada mediante un modelo computerizado. Los usuarios pueden trabajar y examinar objetos 3D reales mientras reciben información adicional sobre estos objetos o sobre la tarea que se está realizando.

Así, la RA permite al usuario permanecer en contacto con su entorno de trabajo, de forma que su foco de atención esté en el mundo real. Explotando las habilidades visuales y espaciales de los usuarios, la RA traslada información adicional a su mundo real, en vez de introducirlos en el mundo virtual del ordenador o dispositivo.

De esta forma, el papel que juega el dispositivo es el de asistir y mejorar las relaciones e interacciones entre las personas y el mundo real. Este paradigma de la interacción y visualización de información constituye el centro de una nueva y muy prometedora tecnología para muchas aplicaciones en multitud de sectores como medicina, arquitectura, diseño interior, construcción, ingeniería civil, diseño de automóviles, mantenimiento mecánico y reparación, etc. De todas formas, las aplicaciones reales para RA imponen una fuerte demanda de tecnología que todavía no se ha alcanzado.

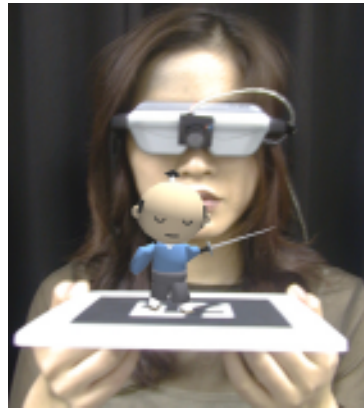


Figura 2.2: Realidad Aumentada

En resumen, usando este paradigma un usuario será capaz de interactuar con el mundo real, el cual está aumentado por la información sintetizada.

Existen diferentes métodos para colocar los objetos 3D. Los describimos a continuación:

### 2.2.2. Usando markers

El marker file o mark es un patrón en forma de imagen que nos sirve para situar un objeto 3D en una posición determinada.



Figura 2.3: Distintos markers

Esta imagen está compuesta por un marco negro y un símbolo en su interior, de tamaño y forma característica. La idea es disponer de un software de reconocimiento de imágenes para determinar qué objeto 3D se debe representar y su posición.

## 2. Estado del arte

---

Este método tiene sus ventajas y sus inconvenientes. Como ventajas podemos señalar que nos permite situar con mucha exactitud un objeto 3D ya que se debe posicionar justo en el lugar la imagen.

Los principales inconvenientes es que la imagen debe estar impresa en algún lugar, y si la imagen no se reconoce, ya sea porque esté oculta una parte o porque esté deteriorada, no se mostrará nada.

Existen algunas librerías que muestran objetos representando mediante markers con técnicas de RA como pueden ser ARToolkit o alguna de sus variantes.



Figura 2.4: Juego Invizimals de la PSP

Otro ejemplo sería videojuego que ha sacado la compañía PlayStation para su consola PSP: *Invizimals*<sup>1</sup> (figura 2.4), el cual nos muestra algún personaje cuando enfocamos con la cámara de la PSP el marcador que viene con el videojuego.

### 2.2.3. Usando la posición GPS

Otra opción es situar los objetos en una determinada posición GPS. Al aproximarse a esta posición se mostrará el objeto en el dispositivo.

Tiene la ventaja, en contraposición a los markers, de que no es necesario realizar ningún reconocimiento de imágenes.

Como inconvenientes hay que destacar la poca precisión de los GPS actuales. Así, si hay varios objetos cercanos y nuestra posición varía un poco, podemos ver saltar los objetos de un punto a otro.

---

<sup>1</sup>**Invizimals** es un videojuego para PSP creado por la compañía Novarama y comercializado por Sony en que se utiliza la innovación de la realidad aumentada, el juego incluye una plataforma en forma de tarjeta sobre la que aparecerán los invizimals y una cámara que conecta con el puerto frontal de la PSP.

Un ejemplo de esta técnica sería la aplicación *Layar*.

### 2.2.4. Diferencias entre realidad virtual y realidad aumentada

La Realidad Virtual (RV) ha sido un tema de gran interés en los últimos tiempos. Menos atención se ha puesto en el campo de la RA a pesar de su similar potencial. En los sistemas de RV el usuario está completamente inmerso en un mundo artificial y no hay manera de interactuar con objetos del mundo real. En contraposición con la RV, en la tecnología de la RA los usuarios pueden interactuar con una mezcla de mundo real y virtual de una forma natural.

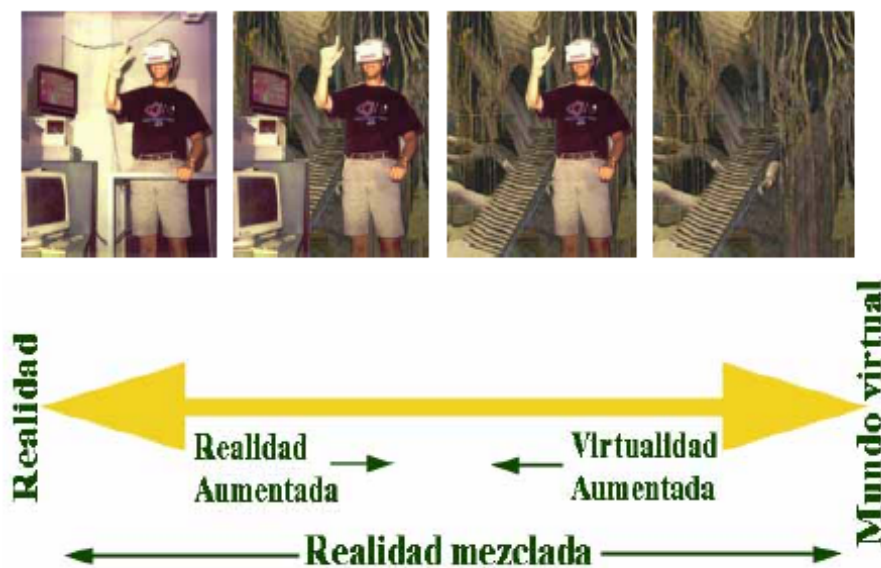


Figura 2.5: Diferencia entre VR y AR

Así, la diferencia entre RV y RA está en el tratamiento que hacen del mundo real. La RV sumerge al usuario dentro de un mundo virtual que reemplaza completamente al mundo real exterior, mientras que la RA deja ver al usuario el mundo real a su alrededor y aumenta la visión que éste tiene de su entorno mediante la superposición o composición de los objetos 3D virtuales. Idealmente, esto daría al usuario la ilusión que los objetos de los mundos real y virtual coexisten.

En definitiva, se podría decir que los sistemas de RA llevan el ordenador al entorno de trabajo real del usuario, mientras que los sistemas de RV intentan llevar el mundo al interior del ordenador. O dicho de otro modo: la RA lleva la información dentro del mundo real del usuario en vez de llevar al usuario dentro del mundo virtual del ordenador.

### 2.2.5. ARToolKit y sus variantes

#### ¿Qué es ARToolKit?

*ArToolKit*<sup>2</sup> es un conjunto de librerías para C/C++ que sirven para la creación de aplicaciones de realidad aumentada. Para ello proporciona una serie de funciones para la captura de vídeo y para la búsqueda de ciertos patrones, en las imágenes capturadas, mediante técnicas de visión por computador. También proporciona una serie de ejemplos y utilidades de gran ayuda al programador que quiera realizar este tipo de aplicaciones.

#### ¿Cómo funciona?

Algo muy importante en las aplicaciones de realidad aumentada, es la necesidad de calcular el punto de vista de la cámara, para así poder realizar las operaciones necesarias sobre los objetos virtuales, para que estos se integren correctamente en el mundo real. Es decir, si queremos mostrar objetos virtuales, de modo que el usuario realmente se crea que existen en el mundo real, tendremos que realizar transformaciones sobre esos objetos de modo que el usuario los vea (a través de la cámara o dispositivo de captura utilizado) en la posición, tamaño, orientación e iluminación, en que esos objetos serían percibidos por el usuario en el mundo real en caso de que realmente estuvieran allí.

Para ello se utilizan unas plantillas de forma cuadrada, que se componen de un cuadrado negro con un cuadrado blanco cuatro veces más pequeño en su centro, y un dibujo sencillo en el interior del cuadrado blanco. La aplicación, utilizando las funciones y utilidades proporcionadas por ARToolKit, será capaz de detectar una de estas plantillas en las imágenes de vídeo capturadas.

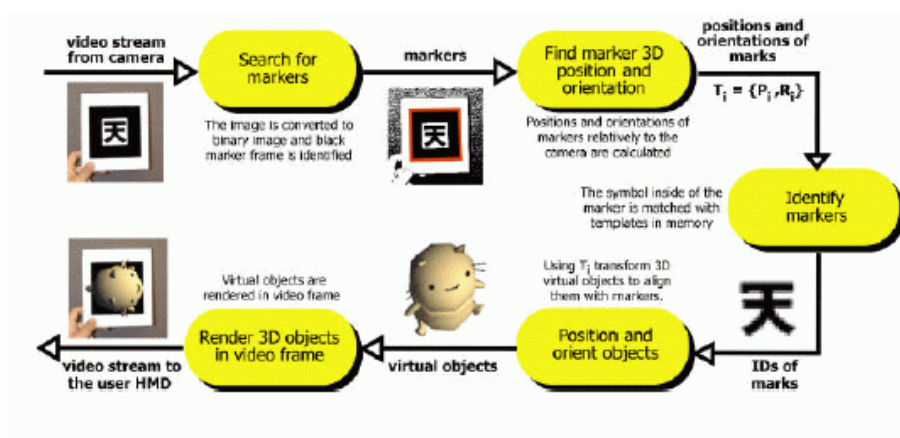


Figura 2.6: Funcionamiento de ARToolKit

Una vez detectada una plantilla en una imagen, estudiando la orientación, posición y tamaño de la plantilla, la aplicación es capaz de calcular la posición y orientación relativa

<sup>2</sup>ARToolKit: <http://www.hitl.washington.edu/artoolkit>

de la cámara respecto a la plantilla, y usando esta información podrá pasar a dibujar el objeto correspondiente sobre la imagen capturada mediante librerías externas a ARToolKit (por ejemplo GLUT <sup>3</sup> y OpenGL <sup>4</sup>), de modo que el objeto aparezca sobre la plantilla en la posición, orientación y tamaño correspondiente al punto de vista de la cámara, siempre que el programador de la aplicación así lo haya decidido, pues las posibilidades son muchas y pudiera ser que una vez obtenida esta información el programador decidiese utilizarla de otra forma, hacer otras operaciones distintas, etc.

### Variantes

Con la proliferación de la realidad aumentada, han surgido distintas variantes de ARToolKit para distintas plataformas y lenguajes de programación. Algunas de ellas son las siguientes:

- **ATOMIC Authoring Tool** es un software Multi-plataforma para la creación de aplicaciones de realidad aumentada, el cual es un Front end para la biblioteca ARToolkit. Fue Desarrollado para no-programadores, y permite crear rápidamente, pequeñas y sencillas aplicaciones de Realidad Aumentada. Está licenciado bajo la Licencia GNU GPL.
- **ATOMIC Web Authoring Tool** es un proyecto hijo de ATOMIC Authoring Tool que permite la creación de aplicaciones de realidad aumentada para exportarlas a cualquier sitio web. Es un Front end para la biblioteca Flartoolkit. Está licenciado bajo la Licencia GNU GPL
- **OSGART** es una combinacion de ARToolKit y OpenSceneGraph
- **ARTag** es alternativa a ARToolKit que usa un procesamiento de imagen y de símbolos digitales, mas complejo, para mayor fiabilidad y resistencia a la luz. Licenciado solo para fines no-comerciales.
- **ARToolKitPlus** es una version extendida de ARToolKit, recomendada solo para usuarios experimentados y desarrolladores de software RA. No continua su desarrollo.
- **Studierstube Tracker** es el sucesor de ARToolKitPlus, muchas características, pero ya no es libre.
- **Mixed Reality Toolkit (MRT)** desarrollado por la University College London

---

<sup>3</sup>**GLUT Mecanismos** (del inglés **OpenGL Utility Toolkit**) es una biblioteca de utilidades para programas OpenGL que principalmente proporciona diversas funciones de entrada/salida con el sistema operativo. Entre las funciones que ofrece se incluyen declaración y manejo de ventanas y la interacción por medio de teclado y ratón. También posee rutinas para el dibujo de diversas primitivas geométricas (tanto sólidas como en modo wireframe) que incluyen cubos, esferas y tetraedros.

<sup>4</sup>**OpenGL (Open Graphics Library)** es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos.

- **FLARToolKit** es un port en ActionScript 3 de ARToolKit para Flash 9+.
- **NyARToolkit** es una biblioteca de clase de ARToolkit desarrollada para maquinas virtuales, particularmente aquellas con host Java, C# y Android.
- **ARDesktop** es una biblioteca de clase de ARToolKit que crea interfaces de escritorio con controles y widgets.

## 2.3. Tecnologías empleadas

### 2.3.1. Comunicación

A la hora de elegir la tecnología destinada a realizar la comunicación, nos decantaremos por aquella que ofrezca un mayor alcance, una mayor tasa de transferencia de datos y una mayor fiabilidad.

En la actualidad existen varias tecnologías que pueden utilizarse para la comunicación en los dispositivos móviles. Las más destacadas son:

- Wi-Fi (estándar 802.11b).
- Telefonía móvil 3G y 4G.
- WiMAX.
- Bluetooth.

#### Wi-Fi (estándar 802.11b)

Hoy en día las tecnologías para red de área local (LAN) inalámbrica han experimentando un boom de implantación.

Numerosos proveedores de redes wireless han instalando sus sistemas en hoteles, cafés, aeropuertos y otros edificios en los que se considera rentable una oferta de acceso a Internet de alta velocidad. Estas nuevas infraestructuras también soportan localización de dispositivos móviles, por lo que las aplicaciones basadas en la posición para entornos de área local resultan viables.

A continuación es necesario explicar de qué tipo de tecnología se está hablando. Las redes inalámbricas cubren áreas de hasta 75 metros en el interior de edificios, y de 300 metros en el exterior, ampliables a varios kilómetros mediante antenas. Están implementadas en un grupo de estándares, conocido como 802.11 que comprende varias modalidades, cada una de ellas con unas características de ancho de banda y alcance determinadas. La modalidad más popular es la que se conoce como 802.11b, introducida en 1997. Opera a una frecuencia de 2,4GHz, y su índice de transferencia de datos es de 11Mbps . Esta tasa se ve ampliamente superada por el estándar 802.11a, que llega a ofrecer 55Mbps en una frecuencia de funcionamiento de 5GHz. El problema es que dicha parte del espectro

no está disponible en países como España, Italia, Portugal o Alemania, pues es de uso restringido militar.

Resumiendo, los estándares actuales 802.11 y algunos relacionados con ellos son los siguientes:

- **IEEE 802.11:** Estándar original de WLANs que soporta velocidades entre 1 y 2 Mbps.
- **IEEE 802.11a:** Estándar de alta velocidad que soporta hasta 54 Mbps en la banda de 5 GHz.
- **IEEE 802.11b:** Estándar dominante de WLAN (conocido también como Wi-Fi), con velocidades de hasta 11 Mbps en la banda de 2.4 GHz.
- **IEEE 802.11g:** Se aprobó el 12 de junio de 2003, e incrementa la tasa de transmisión hasta los 54Mbps a 2.4GHz.
- **HiperLAN2:** Estándar que compite con IEEE 802.11a al soportar velocidades de hasta 54 Mbps en la banda de 5 GHz.
- **HomeRF:** Estándar que compite con el IEEE 802.11b pues soporta velocidades de hasta 10 Mbps en la banda de 2.4 GHz.

Figura 2.7: Estándares para redes de área local inalámbricas.

| Estándar  | Velocidad máxima | Interfaz de aire | Ancho de banda de canal | Frecuencia |
|-----------|------------------|------------------|-------------------------|------------|
| 802.11b   | 11 Mbps          | DSSS             | 25 MHz                  | 2.4 GHz    |
| 802.11a   | 54 Mbps          | OFDM             | 25 MHz                  | 5.0 GHz    |
| 802.11g   | 54 Mbps          | OFDM/DSSS        | 25 MHz                  | 2.4 GHz    |
| HomeRF2   | 10 Mbps          | FHSS             | 5 MHz                   | 2.4 GHz    |
| HiperLAN2 | 54 Mbps          | OFDM             | 25 MHz                  | 5.0 GHz    |

DSSS: Direct Sequence Spread Spectrum  
 OFDM: Orthogonal Frequency Division Multiplexing  
 FHSS: Frequency Hopping Spread Spectrum

Figura 2.8: Estándares de Wireless LAN.

### Telefonía móvil 3G y 4G

Los móviles han tenido diferentes versiones, evoluciones o generaciones en cuanto a conexiones vía radio (Wireless) [4].

La primera generación, **1G**, utilizaba conexiones analógicas y tenía graves problemas de seguridad puesto que el mismo número podía estar en dos móviles distintos.

## 2. Estado del arte

---

Los móviles **2G** supondrían el cambio de tecnología analógica a digital, además codifican y comprimen la señal pudiendo abarcar más llamadas. El 2G o **GPRS** incluye varios protocolos o estándares de transmisión de datos. Entre ellos el usado por los países europeos y americanos: el GSM.

Y por fin la tecnología **3G o UMTS** que utiliza la infraestructura del GSM pero posee un ancho de banda mayor para poder enviar y recibir tanto señales de voz como de datos a una velocidad de hasta 10Mb por segundo.

Nos interesa saber que todo esto se transforma en una señal de cobertura, calidad de sonido y recepción mucho mejor que el convencional GPRS. De ahí que los proveedores de teléfonos móviles ofrezcan nuevos servicios implementados en esta tecnología.

En unos años llegará la siguiente tecnología, denominada como **4G**. Está basada completamente en el protocolo IP, siendo un sistema de sistemas y una red de redes, que se alcanza gracias a la convergencia entre las redes de cables e inalámbricas.

La principal diferencia con las generaciones predecesoras será la capacidad para proveer velocidades de acceso mayores de 100 Mbps en movimiento y 1 Gbps en reposo, manteniendo una calidad de servicio (*QoS*) de punta a punta de alta seguridad que permitirá ofrecer servicios de cualquier clase en cualquier momento, en cualquier lugar, con el mínimo coste posible.

### WiMAX

WiMax es una tecnología inalámbrica para redes de área metropolitana (MAN). Su alcance es de 50 km y ofrece una velocidad de transmisión de 70 Mbps. Las células habituales tienen un alcance menor.

El primer estándar 802.16 operaba en bandas de 10 a 66 GHz y requería que los dispositivos estuvieran en la misma línea de visión. La nueva versión, el estándar 802.16a, funciona en frecuencias que oscilan entre 2 y 11 GHz y, ahora, también permite la conexión entre dispositivos que no se encuentren en la misma línea de visión. La aprobación de esta norma en Europa está sufriendo retrasos debido a cuestiones relacionadas con el uso de los espectros en las bandas de frecuencia de 2.8 y 3.4 GHz.

Esta tecnología es compatible con usuarios móviles que viajen a velocidades de entre 20 y 100 km/h (e incluso a velocidades superiores). El estándar 802.16e ofrecerá a los usuarios movilidad y portabilidad. La norma IEEE 802.16a y el estándar ETSI HiperMAN (Red de área metropolitana de alto rendimiento) comparten las mismas capas físicas (PHY) y MAC. La norma 802.16 se ha diseñado para que sea compatible con el estándar europeo.

Esta tecnología, creada como alternativa a las líneas ADSL y la conexión por cable, es ideal para zonas rurales o áreas en las que el cableado resulta poco viable.

### Bluetooth

Bluetooth es una tecnología diseñada para ofrecer conectividad a redes personales mediante un dispositivo móvil de forma económica. Permite conectar múltiples aparatos Bluetooth: ordenadores portátiles, PDAs, teléfonos móviles, etc., y ofrece conexión a una LAN o WAN a través de un punto de acceso. Bluetooth consigue un canal de comunicación de 721 kbps en un radio de acción de 10 metros, ampliable hasta 100 metros por medio de repetidores. La frecuencia que utiliza está entre 2,4 y 2,48 GHz, cuya gran ventaja es que es un rango de frecuencias abierto. Además, y debido a su concepción de tecnología móvil y económica, tiene un consumo de energía bajo. Para transmitir a una distancia de 10 metros emplea 1mW de potencia, mientras que para llegar a los 100 utiliza 100mW.

La aplicación práctica de esta tecnología es la posibilidad de montar redes inalámbricas en lugares donde haya dificultad para hacerlo de forma convencional, aunque hoy en día para este propósito se oferta otra tecnología, basada en los estándares del IEEE 802.11 que ofrece mayor ancho de banda y radio de conectividad. Por ello, Bluetooth se dirige más a la comunicación de dispositivos.

### 2.3.2. Posicionamiento

Actualmente existen dos grandes sistemas de localización por satélite, aunque en estos momentos se están desarrollando nuevos sistemas:

- Sistema de posicionamiento global
- Global Orbiting Navigation Satellite System

**GPS** El más conocido es el GPS (Global Positioning System - Sistema de posicionamiento global), desarrollado por el Departamento de Defensa de los Estados Unidos.

**GLONASS** Otro sistema es el GLONASS (Global Orbiting Navigation Satellite System) es el sistema de Navegación via satélite desarrollado por los soviéticos.

Ambos sistemas fueron concebidos inicialmente para fines militares, aunque con el paso del tiempo fueron utilizados también para usos civiles.

Estos sistemas proporcionan la posición (altitud, longitud, elevación y tiempo exacto) a millones de usuarios alrededor del mundo a través de las señales que emiten sus satélites y el cálculo de coordenadas desde tierra a través de receptores provistos con relojes muy precisos.

Ambos emplean cada uno 24 satélites ubicados en una órbita media de alrededor de 20,000 Kms. Pero a pesar de ello, ambos sistemas son incompatibles e interoperables entre sí. Aunque GPS y GLONASS ofrecen sus señales a usuarios civiles, su operación sigue estando bajo el control militar.

Por otro lado, estos sistemas de localización presentan los siguientes inconvenientes:

- No hay garantía o cobertura de fiabilidad proveída por sus operadores (e.g. accidentes aéreos)

## 2. Estado del arte

---

- La fiabilidad es incierta en regiones de altas latitudes del norte de Europa
- La precisión es moderada para aplicaciones que requieren una rápida determinación de la posición
- A los usuarios no se les informa inmediatamente de los errores que ocurren en el sistema

Actualmente hablar de localización por satélite es hablar de GPS ya que ha sido el que más auge ha tenido, pero es tan sólo uno de los muchos sistemas de localización existentes.

**Galileo** En 1999 nace un proyecto de origen europeo denominado Galileo. Se compone de unos 30 satélites y estará listo hacia el año 2014. En principio se supone que tendrá compatibilidad con GPS. Tendrá un servicio gratuito de posición con una precisión de unos 5 metros. Existirá otro servicio de pago con mayor precisión.

### Sistema de posicionamiento global (GPS)

El sistema de posicionamiento global o “GPS” de acuerdo a sus siglas en inglés (*Global Positioning System*) constituye en estos momentos actuales el principal sistema de localización en entornos exteriores. Este sistema se basa en el uso de las comunicaciones vía satélite: concretamente, el sistema consta de 21 satélites, los cuales orbitan alrededor de la Tierra a una altura aproximada de 20.000 kilómetros.



Figura 2.9: El sistema de posicionamiento global es el principal sistema de localización en entornos exteriores desarrollado hasta la fecha

Fue desarrollado por el Departamento de Defensa de los Estados Unidos con el objetivo de poder determinar la posición de un punto cualquiera sobre la superficie terrestre en tiempo real y sustituyó al anterior sistema, *Transit*. Este sistema se basaba en el empleo de 6 satélites dispuestos en una órbita polar baja, a una altura de 1074 kilómetros. Esta configuración permitía una cobertura mundial pero no constante, a diferencia del actual sistema “GPS”.

### Cálculo de coordenadas tridimensionales

El sistema de posicionamiento global o “GPS” determina la posición de los agentes a través del cálculo de coordenadas tridimensionales. Lógicamente, para llevar a cabo este objetivo es necesario que los receptores capten adecuadamente las señales emitidas por los satélites. Por otro lado, en el proceso del cálculo de la posición del receptor, basándose en el cálculo de coordenadas tridimensionales sería necesario, en principio, la presencia de tres satélites.

Sin embargo, el empleo de únicamente tres satélites lleva implícito el problema del sincronismo: los tres satélites deben estar perfectamente sincronizados o de lo contrario las coordenadas tridimensionales obtenidas no serán válidas. Para solucionar este problema, la trigonometría nos dice que es posible determinar la posición exacta de un objeto en el espacio empleando cuatro satélites y así eliminar las mediciones erróneas producidas por problemas de sincronismo entre éstos.

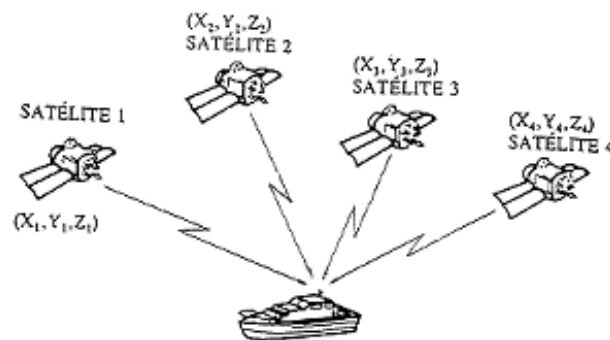


Figura 2.10: Se emplean cuatro satélites para determinar las coordenadas tridimensionales del receptor

### Medición de la distancia

Para determinar la distancia a la que se encuentra el receptor de un satélite y así llevar a cabo el cálculo de las coordenadas tridimensionales de éste es necesario tener en cuenta la fórmula física que permite determinar el espacio recorrido durante un período de tiempo a velocidad constante:

$$e = v * t$$

Puesto que los satélites empleados por el sistema GPS se basan en la emisión de ondas electromagnéticas, sabemos que la velocidad de dichas ondas se corresponde a la velocidad de la luz en el vacío, que denotamos por  $c$  y cuyo valor aproximado es de  $300,000km/s$ .

## 2. Estado del arte

---

De este modo vemos cómo de importante es la correcta sincronización de los dispositivos implicados en el cálculo de las coordenadas, ya que una pequeña diferencia en la medición del tiempo producirá errores muy graves en la posición de receptor debido al elevado valor de la velocidad.

No obstante, surge otro problema a la hora de calcular el tiempo que tardó el receptor en recibir la señal del satélite ya que éste no sabe a priori en momento envió la señal el satélite. Para solventar este problema es preciso que ambos dispositivos se encuentren perfectamente sincronizados mediante señales de reloj. Lógicamente, estas señales no pueden verse alteradas, tanto atrasadas como adelantadas, ni podrán estar condicionadas por factores atmosféricos en el caso de los receptores, por lo que se opta por el empleo de relojes atómicos, con un error de un segundo cada 30.000 años. En esencia, la solución propuesta consiste en que ambos dispositivos (satélite y receptor) generan una misma señal en un mismo instante, el receptor capta la señal recibida del satélite y la diferencia de fase entre las dos señales constiye el tiempo que tardó la señal en ser recibida por el receptor, con lo que el problema queda perfectamente solventado.

### Disponibilidad selectiva

El sistema de posicionamiento global, si bien consta de numerosas aplicaciones civiles, surgió como un proyecto militar y por tanto, dicho sistema es susceptible de ser desactivado o disminuida su precisión para evitar un uso militar del mismo en situaciones de conflicto bélico por parte de naciones no aliadas de Estados Unidos, como fue en el caso de la primera guerra del Golfo.

### Aplicaciones

Las principales aplicaciones del sistema de posicionamiento global se pueden dividir en:

- Militares
- Civiles

#### Militares

Las principales aplicaciones militares del sistema de posicionamiento global únicamente pueden ser utilizadas por Estados Unidos o por naciones aliadas. Dichas aplicaciones se basan en el empleo de armamento “inteligente” como por ejemplo, misiles de crucero.

#### Civiles

Las aplicaciones civiles del sistema de posicionamiento global son muy amplias. Podemos destacar:

- Sistemas de navegación, tanto terrestre como marítima o aérea
- Topografía
- Aplicaciones científicas
- Aplicaciones lúdicas (deportes, acampadas, etcétera)
- Ayuda a enfermos o discapacitados físicos

### Global Orbiting Navigation Satellite System o GLONASS

La contrapartida rusa al “GPS” lleva el nombre *Global Navigation Satellite System* (GLONASS) y es operacional desde el 18 de Enero de 1996, día en el que los 24 satélites estaban operativos y en comunicación al mismo tiempo.

#### Diseño del sistema

Consta de una constelación de 24 satélites (21 en activo y 3 satélites de repuesto) situados en tres planos orbitales con 8 satélites cada uno y siguiendo una órbita inclinada de  $64,8^\circ$  con un radio de 25510 kilómetros. La constelación de “GLONASS” se mueve en órbita alrededor de la tierra con una altitud de 19.100 kilómetros (algo más bajo que el “GPS”) y tarda aproximadamente 11 horas y 15 minutos en completar una órbita.



Figura 2.11: El sistema GLONASS

#### Historia

Los satélites se han lanzado desde Tyuratam, Kazajstán. Los tres primeros fueron colocados en órbita en octubre de 1982. El sistema fue pensado para ser funcional en el año 1991, pero la constelación no fue terminada hasta diciembre de 1995 y comenzó a ser operativo el 18 de enero de 1996. La situación económica de Rusia en los años 90 supuso que en abril de 2002 solo 8 satélites estuvieran completamente operativos. En el 2004, 11 satélites se encuentran en pleno funcionamiento y tras un acuerdo con el gobierno indio se plantea tener de nuevo completamente operativo el sistema para el año 2007.

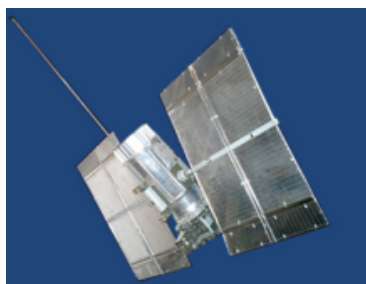


Figura 2.12: Satélites GLONASS

### Compatibilidad con “GPS”

Actualmente, se ha conseguido construir receptores que puedan recibir señales pertenecientes a los dos grupos de satélites “GLONASS” y “GPS”.

Existe por supuesto un gran interés en incorporar los satélites “GLONASS” al sistema debido al incremento potencial del número de satélites visibles simultáneamente. Lógicamente, cuanto mayor es la cantidad de satélites disponibles al mismo tiempo, más rápidas, mejor y más fiables son las técnicas de posicionamiento. Hoy en día las expectativas se centran en un nuevo sistema de gran interés: el “GNSS” (*Global Navigation Satellite System*) integrado por los dos grandes sistemas.

Para aplicaciones temporales ya están en el mercado receptores del sistema dual combinado.

### Galileo

“Galileo” es un sistema global de navegación por satélite propuesto por la Unión Europea con apoyo de la Agencia Espacial Europea y un grupo de inversionistas privados. El proyecto “Galileo” se propuso para fines civiles, y se espera que esté en funcionamiento en el año 2014.

Es un sistema independiente de “GPS”, pero totalmente compatible con él.

Este nuevo sistema es una constelación formada por 30 satélites de nueva generación (27 operativos y 3 en reserva, 6 más que el “GPS”) en tres órbitas circulares en torno a la Tierra, que lanzan durante las 24 horas, con buen o mal tiempo, una señal de radio codificada constante que es recogida y analizada por los receptores en tierra.

Dos centros de control “Galileo” se instalarán en Europa para seguir el funcionamiento de los satélites y encargarse del sistema de navegación.

Los satélites “Galileo” serán capaces de entregar la señal hacia la Tierra con más potencia que la señal del sistema “GPS”, por eso la señal de “Galileo” es menos interferible.

El gran reto del sistema “Galileo” será la sincronización entre los satélites y las terminales en la Tierra. “Galileo” empleará relojes atómicos de cesio en tierra, en contraste con “GPS” que utiliza estos relojes en sus satélites. Los satélites de “Galileo” estarán equipados con relojes de rubidio, que permiten precisiones en el orden de nanosegundos. El cálculo orbital será llevado a cabo en tierra.

### El diseño del sistema

El sistema por el que los receptores calcularán su posición a partir de los satélites es relativamente simple. Cada uno de los satélites está situado de forma equidistante entre sí formando una malla. Cuando un receptor localiza su señal puede medir la distancia que los separa (debido a que una onda de radio viaja a la velocidad de la luz, unos 300 mil kilómetros por segundo, se puede deducir la distancia entre un emisor de radio y un satélite a partir del tiempo que tarda la señal en recorrer esa distancia). De este modo, basta con recibir la señal de tres satélites para saber el lugar exacto en el que se encuentra el receptor.

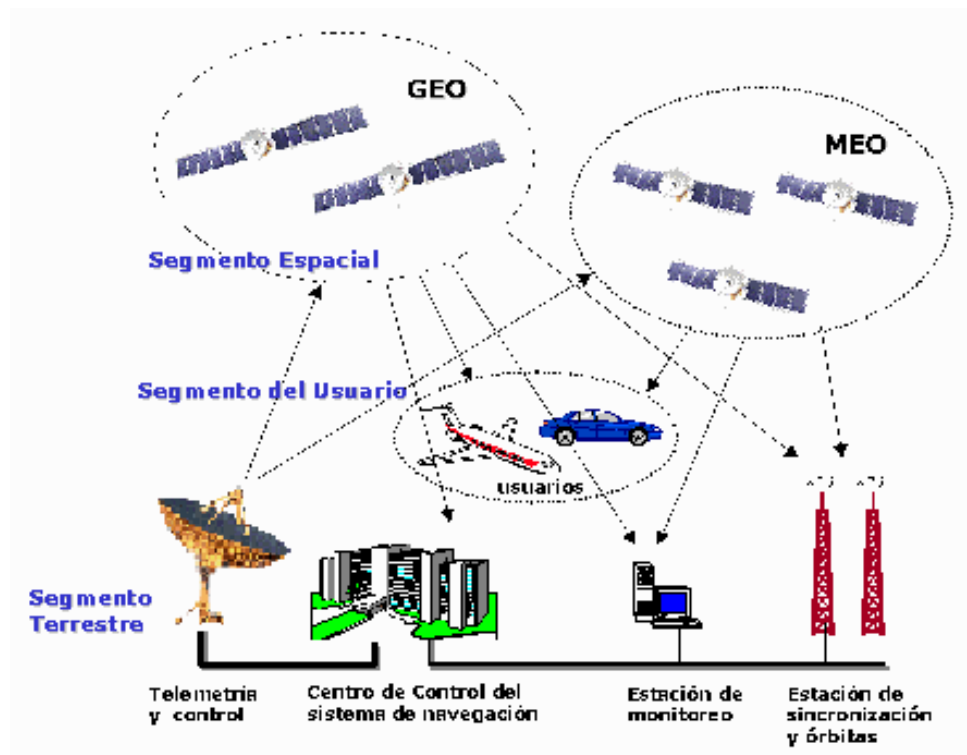


Figura 2.13: Diagrama del sistema “Galileo”

Con una simple trigonometría se puede encontrar una determinada posición terrestre.

### Servicios

## 2. Estado del arte

---

El sistema Galileo tiene tres componentes que representan tres niveles de desempeño:

- **Global** que ofrece cobertura mundial
- **Regional** que da cobertura europea
- **Local**, para aeropuertos o cobertura urbana.

Los tipos de servicios son:

- Servicio con acceso abierto: servicio básico gratuito y abierto para todo el público.
- Servicio con acceso controlado de nivel 1 (SAC 1): servicio con pago de tarifa con acceso controlado para aplicaciones comerciales y aplicaciones profesionales que necesitan niveles de desempeño superior y una garantía de servicio.
- Servicio con acceso controlado de nivel 2 (SAC 2): servicio con pago de tarifa con acceso controlado para aplicaciones de alta seguridad.

### Aplicaciones

“Galileo” revolucionará la administración del tráfico aéreo, mejorará la calidad y seguridad de este medio de transporte en regiones del mundo donde los sistemas existentes son inadecuados, incrementará la precisión y control permitiendo la optimización del uso del espacio aéreo. Esto ayudará en gran medida a los retardos en los vuelos. Además, los conductores de camiones y autos podrán evitar congestionamientos de tráfico al reducir sus tiempos de viaje entre 15 y 25 %, así como también se reducirá el consumo de combustible y emisión de contaminantes.

Los servicios de emergencia llegaran más rápidamente a la escena para proveer asistencia a la gente en peligro. Las compañías de transporte serán capaces de monitorear la posición de sus vehículos o contenedores y la lucha contra el crimen será más efectiva al localizar más rápida y eficientemente los vehículos robados. La lista de aplicaciones potenciales crecerá día a día.

### Conclusión

Aunque el proyecto “Galileo” está aún en sus fases iniciales, se esperan grandes beneficios para los usuarios cuando éste proyecto esté concluido en el 2014. “Galileo” será otra opción para la determinación de la posición que en conjunto con los otros sistemas “GPS” y “GLONASS” brindarán una gama de nuevos servicios y aplicaciones.

## 2.4. Magic Book

### 2.4.1. Introducción

Quizá una de las aplicaciones más conocidas de la Realidad Aumentada en la educación sea el proyecto **Magic Book** [5] del grupo de investigación *Human Interface Technology Laboratory de Nueva Zelanda*.

El alumno lee un libro real a través de un visualizador de mano y ve sobre las páginas reales contenidos virtuales. De esta manera cuando el alumno ve una escena de Realidad Aumentada que le gusta puede introducirse dentro de la escena y experimentarla en un entorno virtual inmersivo.



Figura 2.14: *MagicBook* y su aplicación de enseñanza para volcanes

### 2.4.2. Ventajas

Las principales ventajas son:

- Es una aplicación educativa pionera en el uso de la realidad aumentada.
- Se consiguen mostrar objetos y realizar explicaciones de forma interactiva.

### 2.4.3. Inconvenientes

Los principales inconvenientes son:

- Las aplicaciones son ad-hoc, es decir, están desarrolladas para un contexto específico y, por tanto, no son reutilizables ni adaptables.
- No está pensada para dispositivos móviles.
- No permite la manipulación de los objetos para poderlos ver desde otros ángulos.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

### 2.5. Proyecto ARiSE

#### 2.5.1. Introducción

El propósito del proyecto *ARiSE* (*Augmented Reality in School Environments*) [6] tiene como objetivo la integración de una plataforma de enseñanza de realidad aumentada en el entorno cotidiano de los profesores y estudiantes, mostrando los contenidos audiovisuales y multimedia en línea con las necesidades identificadas en estas áreas.



Figura 2.15: Aplicación del proyecto ARiSE

Los estudiantes tendrán la posibilidad de interactuar en equipo con los objetos virtuales en un espacio virtual compartido proporcionado por un sistema de visualización de realidad aumentada (figura 2.16) y de ese modo se llevará a cabo el aprendizaje práctico en lugar de aprender a leer o escuchar.



Figura 2.16: Es necesario un equipamiento especial para su uso

### 2.5.2. Ventajas

Las principales ventajas son:

- Se permite a los alumnos interactuar con los objetos virtuales.

### 2.5.3. Inconvenientes

Los principales inconvenientes son:

- Es necesario un espacio virtual para poder visualizar y manipular los objetos.
- No está pensada para dispositivos móviles.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.6. LearnAR

### 2.6.1. Introducción

**LearnAR** [7] es una herramienta educativa de realidad aumentada que permite combinar el mundo real con contenido virtual mediante una simple cámara digital de vídeo o web. Dispone de las siguientes aplicaciones:

- **Brazos y músculos.** Los estudiantes pueden ver los huesos y músculos en su propio brazo (figura 2.17).

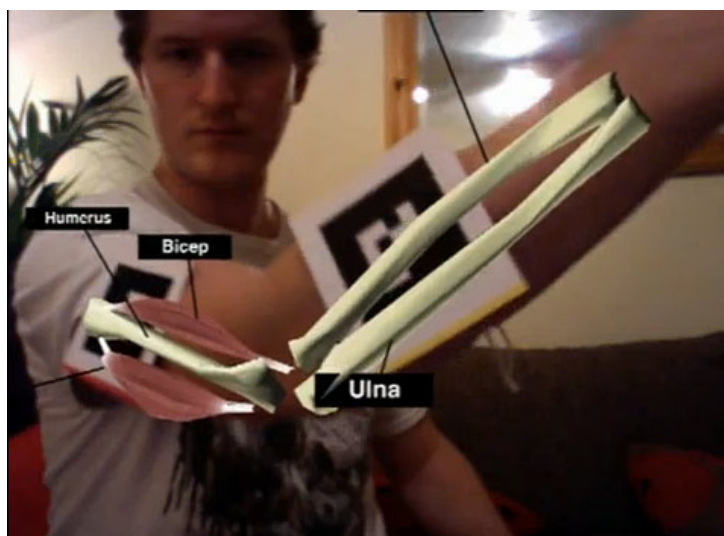


Figura 2.17: Aplicación de *Brazos y Músculos*

## 2. Estado del arte

---

- **Química:** Series de reactividad de los metales. Para experimentar con metales y soluciones y averiguar las reacciones que acontecen cuando un metal se mezcla con una solución.
- **Biología - Órganos:** Herramienta que permite mantener un marcador delante del pecho del alumno y ver sus órganos internos (figura 2.18).

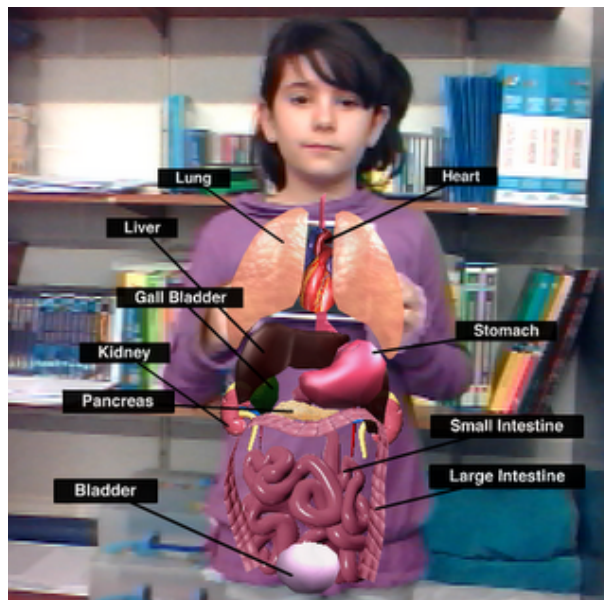


Figura 2.18: Aplicación de *Biología* en la que se muestran los órganos

- **Física:** Radiactividad. Experimentando con los marcadores radiactivos se puede ver cómo afectan a un contador Geiger.
- **Matemáticas:** Objetos en 3D. Se puede calcular el volumen o la superficie del área de objetos en la pantalla del ordenador.
- **Religiones del mundo:** Los 5 K del sijismo. Guru Gobind Singh, el último de los diez gurús, propuso a todos los miembros de la hermandad de Sikhs propagar los cinco K (artículos de fe) como un símbolo para todos sus creyentes.

### 2.6.2. Ventajas

Las principales ventajas son:

- Dispone de varias aplicaciones en su plataforma web.

### 2.6.3. Inconvenientes

Los principales inconvenientes son:

- Tiene desarrolladas varias aplicaciones, pero no nos permite crear las nuestras ni configurar las existentes.

- No está pensada para dispositivos móviles.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.7. SciMorph

### 2.7.1. Introducción

Otra aplicación que utiliza técnicas de realidad aumentada aplicadas al ámbito de la educación es **Scimorph** [8], un juego diseñado para estimular el debate entre los estudiantes y sus profesores acerca de cuestiones científicas tratadas en los escenarios del juego.



Figura 2.19: SciMorph

Scimorph es un extraterrestre al que los alumnos pueden poner en diferentes entornos para observar qué sucede.

### 2.7.2. Ventajas

Las principales ventajas son:

- Aplicación destinada exclusivamente al ámbito educativo.

### 2.7.3. Inconvenientes

Los principales inconvenientes son:

- Tiene desarrolladas varias aplicaciones, pero no nos permite crear las nuestras ni configurar las existentes.
- No está pensada para dispositivos móviles.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

### 2.8. ZooBurst

#### 2.8.1. Introducción

**ZooBurst** [9] es una herramienta que nos permite crear libros en los que los personajes pueden aparecer y dialogar en tres dimensiones.

Enfocado principalmente al público infantil tiene disponible la posibilidad de usar realidad aumentada para, usando la webcam, ofrecer experiencias diferentes a las que estamos acostumbrados en este tipo de sitios. En realidad bastará imprimir un padrón para que, al mostrarlo en la webcam, aparezca el libro abierto en nuestras manos.

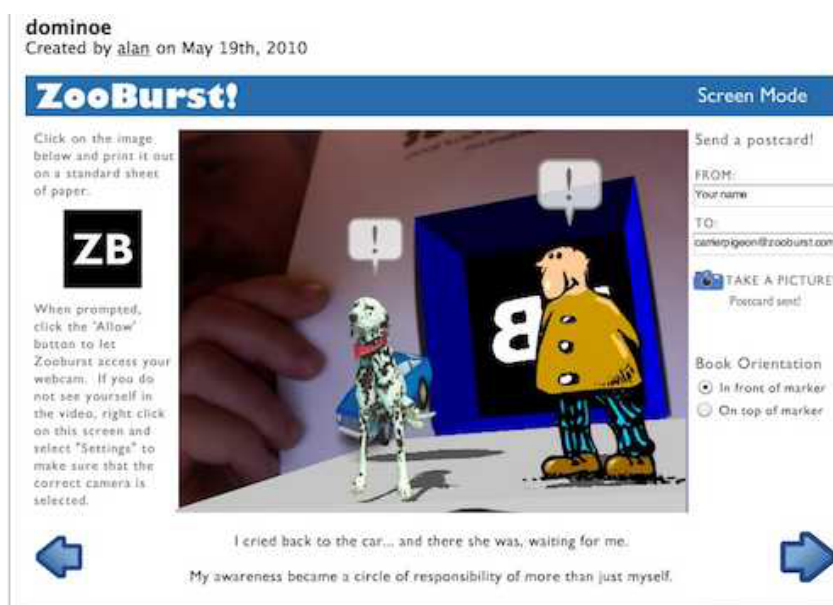


Figura 2.20: ZooBurst

Las historias pueden incluirse en cualquier página web, aumentando la posibilidad de divulgar nuestras creaciones.

#### 2.8.2. Ventajas

Las principales ventajas son:

- Permite crear contenidos personalizados.
- Sólo es necesario usar un navegador web con Flash.

#### 2.8.3. Inconvenientes

Los principales inconvenientes son:

- No está pensado para dispositivos móviles.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.9. ARSights

### 2.9.1. Introducción

**ARSights** [10] es una extensión de *Google Earth* en la que se podrán mostrar los edificios representados en la aplicación en tu propio entorno de realidad aumentada. Sólomente es necesario instalarlo en el navegador, imprimir un patrón y una cámara web.

ARSights ha sido desarrollado por una compañía italiana de nombre *Inglobe*, especializada en el desarrollo de aplicaciones relacionadas tanto con la realidad aumentada como con la virtual. El objetivo principal de ARSights es extender las ya de por sí excelentes capacidades que posee Google Earth, aplicando tecnología de realidad aumentada y haciéndola disponible para el usuario común de una forma bastante sencilla. El primer paso será descargar e instalar la aplicación ARSights.



Figura 2.21: ARSights

Una vez instalado, es necesario imprimir una hoja llamada “marcador”. Esta hoja servirá como base para los objetos (o mejor dicho, los edificios y construcciones destacadas) que aparezcan en la realidad aumentada. La hoja no tiene ningún diseño complicado, sólo

## 2. Estado del arte

---

es necesaria para que la cámara web tenga un punto de referencia. El paso siguiente es ir a la página oficial de ARSights e instalar la extensión para poder visualizar a Google Earth en el navegador. Luego debes buscar el edificio que deseas ver, descargar el modelo, y automáticamente ARSights se iniciará.

El programa tomará control de la cámara web, y cuando enfoques el marcador de forma que pueda verse a través de la cámara, el edificio en cuestión aparecerá.

### 2.9.2. Ventajas

Las principales ventajas son:

- Utiliza los objetos tridimensionales de *Google Earth*, con lo que posee una gran base de datos de éstos

### 2.9.3. Inconvenientes

Los principales inconvenientes son:

- No está pensado para dispositivos móviles.
- No es posible visualizar otro tipo de objetos que no sean los de *Google Earth*.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

Una vez vistas las herramientas existentes en entornos educativos, mostraremos aplicaciones de realidad aumentada implantadas en entornos museísticos.

## 2.10. Wikitude

### 2.10.1. Introducción

Wikitude [11] es una aplicación que combina la base de datos geolocalizados de la Wikipedia con el mundo real. En definitiva, es una guía para viajes y turismo basada en sistemas de geoposicionamiento, brújula digital, sensores de orientación y acelerómetro, mapas, video y contenidos informativos de la Wikipedia.



Figura 2.22: Wikitude

Se trata de una aplicación cuya propuesta de valor es ayudar a un viajero a planificar su viaje o encontrar información relevante sobre un sitio concreto o zona geográfica. Actualmente, Wikitude presenta información de 350.000 *POI* (puntos de interés) de todo el mundo que pueden ser buscados mediante GPS o mediante su dirección. Estos puntos de interés geográficos son mostrados en el móvil en una lista, sobre un mapa o mediante un video geo-referenciado.

Wikitude utiliza 3 capas de información que se superponen en la pantalla del dispositivo móvil. Primero aparece la información de Wikipedia, reseñas locales de Qype y por ultimo información del propio Wikitude. Luego está la posibilidad de añadir notas a los lugares marcados por ti y posteriormente puede ser visto por cualquier persona.

La prestación más interesante de Wikitude es la posibilidad de acceder a información en tiempo real mediante realidad aumentada en la pantalla del móvil sobre los lugares que se van viendo a medida que se recorre el sitio observado con la cámara. La aplicación Wikitude ha sido desarrollada por la empresa *Mobilizy* de Austria.

## 2. Estado del arte

---



Figura 2.23: Wikitude ofrece información en tiempo real del sitio observado

### 2.10.2. Ventajas

Las principales ventajas son:

- Se crean los puntos de interés a través de la web.
- Está diseñada específicamente para dispositivos móviles.
- Es totalmente configurable, al poder visualizar y crear capas totalmente personalizadas.
- Nos muestra información en función de nuestra posición geográfica.

### 2.10.3. Inconvenientes

Los principales inconvenientes son:

- No se muestran objetos en 3 dimensiones, sino que se muestra información en función de la posición que tengamos.
- No utiliza markers para mostrar la información, sino que sólo utiliza la posición GPS.

## 2.11. Layar

### 2.11.1. Introducción

**Layar** [12] es un proyecto de navegador web basado en realidad aumentada para móviles, que emplea el hardware de los teléfonos de última generación, para ubicar espacialmente información de lo más variopinta en nuestra pantalla.



Figura 2.24: Layar

El concepto en sí es sencillo. Layar utiliza el GPS y el acelerómetro de nuestro terminal para saber en qué lugar del globo nos encontramos en cada momento, y una vez situados, busca la información que le pidamos en las distintas capas de mapas existentes: negocios, inmobiliarias, wikipedia, hoteles, oficinas turísticas... Y usando la cámara, nos muestra sobre la imagen real que está captando, en qué lugar o dirección se encuentran dichos lugares.

Lo mejor de todo, es que el proyecto crece día a día, y ya también muestra imágenes en 3D geolocalizadas con lo que posiblemente estemos ante el nacimiento del soporte publicitario del futuro.

### 2.11.2. Ventajas

Las principales ventajas son:

- Se crean los puntos de interés a través de la web
- Está diseñada específicamente para dispositivos móviles.
- Es totalmente configurable, al poder visualizar y crear capas totalmente personalizadas.
- Nos muestra información en función de nuestra posición geográfica.

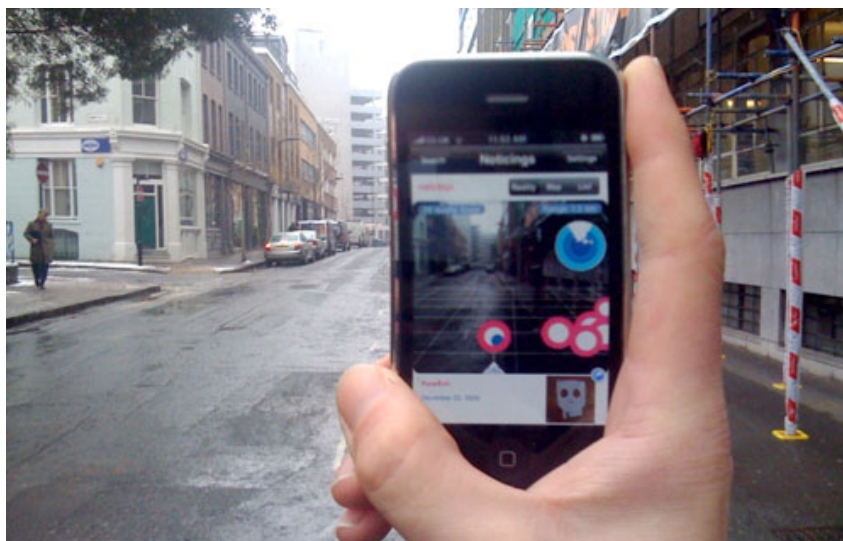


Figura 2.25: Layar muestra información de una capa

### 2.11.3. Inconvenientes

Los principales inconvenientes son:

- El único inconveniente es que no utiliza markers para mostrar la información, sino que sólo utiliza la posición GPS.

## 2.12. Museografía Virtual para el museo de sitio de Calakmul

### 2.12.1. Introducción

Calakmul [13], declarado Patrimonio de la Humanidad por UNESCO en Junio 27 del 2002, es una de las más importantes ciudades de la cultura Maya. Oculta por la selva e inexplorada hasta la década de 1980, Calakmul esta destinada a permanecer parcialmente cubierta por la vegetación de una de las más grandes y saludables reservas nacionales.

Sus inmensas estructuras de 50 metros de altura, se alzan por encima del nivel de los árboles tropicales y sólo subiendo hasta sus cimas pueden verse las cúspides de las demás pirámides de la ciudad. Dado que las plazas alrededor de cuales se encuentran dispuestas las edificaciones, están cubiertas por enormes árboles, resulta muy difícil para los visitantes el poder apreciar el diseño urbano y arquitectónico de Calakmul.

El sitio arqueológico se encuentra 65 Km dentro de la Biosfera de Calakmul y a tan sólo 30 Km de la frontera con Guatemala. El aislamiento de Calakmul es parte de su mayor atractivo y, sin embargo, motivo de que ninguna de las piezas encontradas en las



Figura 2.26: Imagen aérea de Calakmul

17 tumbas descubiertas hasta el momento, puedan ser exhibidas en el sitio por motivos de seguridad.



Figura 2.27: Situación de Calakmul, Campeche

La trascendencia histórica de Calakmul y la importancia de sus hallazgos arqueológicos, merece de una adecuada exhibición en un museo de sitio, sin embargo dadas las circunstancias antes descritas, para poder mostrar sus invaluable máscaras, ofrendas funerarias, su traza urbana y su continuo proceso arquitectónico a lo largo de 1200 años de historia, se requiere de una amplia gama de reproducciones.

## 2. Estado del arte

---

El hecho de que ningún objeto real pudiera mostrarse, originó la hipótesis de que el uso de museografía virtual pudiese resolver la imposibilidad de exhibición material. De esta forma se iniciaron reproducciones digitales con recursos de Realidad Virtual, Realidad Aumentada y VRML interactivo.

Debido a las circunstancias descritas sobre el sitio arqueológico de Calakmul, se consideró la creación de una museografía virtual compuesta de:

1. Entornos de Realidad Virtual para ser recorridos en tiempo real, uno de ellos con asistencia de un sistema inteligente,
2. Una instalación de Realidad Aumentada que permita sobreponer elementos virtuales a un espacio físico. La metáfora de superposición del pasado al presente servirá para presentar a la más importante tumba encontrada en Calakmul, la tumba del Rey “Garra de Jaguar”.
3. Vitrinas virtuales interactivas con reproducciones digitales tridimensionales que cubran las temáticas de:
  - Ubicación de Calakmul
  - Secuencias arquitectónicas
  - Tumbas y ritos funerarios
  - Subestructura IIc
  - Secuencia dinástica
  - Pintura Mural
  - Cerámica
4. Un portal de Realidad Virtual para despliegue en Internet que presente al Museo Virtual de Calakmul a partir de la promoción en línea de la Ciudad de Campeche, capital del Estado y entrada aeroportuaria del turismo nacional e internacional.

A nosotros nos va a interesar el estudio del segundo punto: la visita mediante Realidad Aumentada. Los objetos virtuales que utiliza, están realizados en VRML y su medio de identificación de patrones es a través de webcams convencionales, por lo que hacer aplicaciones híbridas puede ser factible.

Se realiza una novedosa instalación museográfica del entierro del gobernante “Garra de Jaguar” en la cual se tendrá una réplica de parte de la cámara funeraria como fue encontrada por el arqueólogo Carrasco en 1997. Esta parte representará el Presente de la situación en torno a este importante hallazgo. Adicionalmente, los elementos que desaparecieron tras 13 siglos serán añadidos digitalmente ante la mirada del visitante de tal forma que pueda contemplar el aspecto que hipotéticamente tuvo la mortaja del gobernante cuando fue enterrado en el siglo VIII.



Figura 2.28: Demo de Realidad Aumentada donde los objetos digitales quedan superpuestos al objeto real, en este caso, un libro en el que el usuario va pasando hojas.

### 2.12.2. Ventajas

Las principales ventajas son:

- Facilidad de uso de la herramienta, ya que las cámaras colocadas identificarán los patrones y se mostrará el objeto correspondiente.
- Se han añadido elementos que han desaparecido y que por lo tanto los usuarios no podrían ver en su visita al museo.

### 2.12.3. Inconvenientes

Los principales inconvenientes son:

- La aplicación no está destinada a dispositivos móviles.
- Los objetos 3D deberán ser diseñados a medida, no pudiendo importarse del trabajo que anteriormente pudiese haber realizado algún diseñador en la creación de objetos 3DS.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.13. Proyecto ARCO

### 2.13.1. Introducción

ARCO [14] es un proyecto de investigación, financiado por la Unión Europea, dirigido a desarrollar la tecnología para que los museos creen sus exposiciones virtuales 3D en la Web.

## 2. Estado del arte

---

Las exposiciones virtuales son creadas convirtiendo los artefactos del museo a digital. Entonces se transforman en las representaciones virtuales, que pueden ser X3D o modelos o escenas del VRML. Estas representaciones virtuales junto con los objetos asociados de los medios descritos con XML terminan en la exposición virtual. Se enfocó de tal manera que el museo gestione la base de datos y junto con la tecnología X-VRML para así permitir que los museos creen fácilmente estas exposiciones virtuales. X-VRML define la disposición virtual de la exposición y permite actualizaciones dinámicas de la exposición virtual simplemente cambiando el contenido de la base de datos.

El proyecto ARCO se basa en tres componentes principales:

- Producción de contenidos
- Gestión de contenidos
- Visualización

La *producción de contenidos* incluye todas las herramientas y técnicas usadas para crear representaciones digitales de los artefactos del museo. El método de modelar depende de las características de los objetos. Los objetos con geometría simple se pueden modelar con ciertos sistemas como puede ser el caso de *3D Studio MAX*.

Los objetos con geometría compleja se modelan con un sistema digital de cámara fotográfica también desarrollado dentro del proyecto. Este sistema es capaz de la generación automática de 3D.

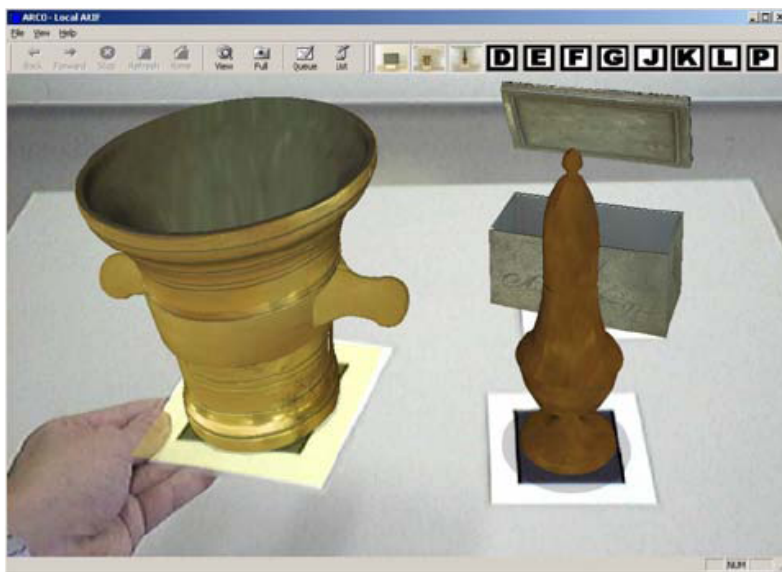


Figura 2.29: Escena del proyecto ARCO

Los *contenidos* se almacenan en una base de datos de ARCO y se manejan con una aplicación de gestión de contenidos. Un diseñador puede crear fácilmente exposiciones virtuales creando espacios de la exposición y asignando objetos culturales y plantillas de visualización a los espacios.

La *visualización* de las representaciones digitales del museo son realizados por medio de interfaces de realidad virtual y aumentada. Estas interfaces combinan la web con exposiciones virtuales de realidad virtual o aumentada. La presentación mediante la web permite que los usuarios busquen y que hojeen el contenido de la base de datos por medio de una interfaz bien conocida, mientras que las exposiciones de VR y de AR dejan examinar reconstrucciones virtuales de objetos 3D.

### 2.13.2. Ventajas

Las principales ventajas son:

- Los objetos complejos se modelan mediante técnicas digitales.
- Un buen sistema de almacenamiento mediante la base de datos.
- Combina realidad aumentada y virtual.

### 2.13.3. Inconvenientes

Los principales inconvenientes son:

- No es un proyecto destinado a la representación en dispositivos móviles.
- Proyecto de investigación terminado que no se llegó a implantar.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.14. ARCHEOGUIDE

### 2.14.1. Introducción

ARCHEOGUIDE [15] construirá un sistema que proporciona nuevas maneras de acceso a la información en los sitios de patrimonio cultural, una manera de uso fácil con el uso de tecnologías de la información avanzadas incluyendo realidad aumentada, visualización 3D, computación móvil y una interacción multi-modal.

El sistema proporcionará las características siguientes a los visitantes:

- Información adaptada al contexto, es decir, dependiendo de la posición y orientación del usuario.

## 2. Estado del arte

---



Figura 2.30: Escena virtual del proyecto ARCHEOGUIDE

- Ayudas personalizadas y temáticas dependiendo de los perfiles del visitante, en función de la edad, las habilidades culturales y lingüísticas.
- Visualización en 3D de artefactos que faltan o de partes reconstruidas de sitios dañados.

ARCHEOGUIDE consiste en un servidor de información y un sistema de unidades móviles que llevan los visitantes. Una red inalámbrica permite que las unidades móviles se comuniquen con el servidor de información.

El servidor del sitio mantiene una base de datos con toda la información referente al lugar. El contenido se puede descargar a los móviles sobre la red inalámbrica. Además, el servidor de información incorpora el software que permite la creación de nuevos contenidos 3D del lugar.

Las unidades móviles disponen de un montan dispositivo para la cabeza (HMD), una cámara fotográfica, un micrófono, unos auriculares y un ordenador portátil. El portátil se equipa de los dispositivos adecuados permitiendo que se comunique con el servidor de información a través de la red.

### 2.14.2. Ventajas

Las principales ventajas son:

- La información es adaptada al contexto.
- Información en función de los perfiles del visitante.

### 2.14.3. Inconvenientes

Los principales inconvenientes son:

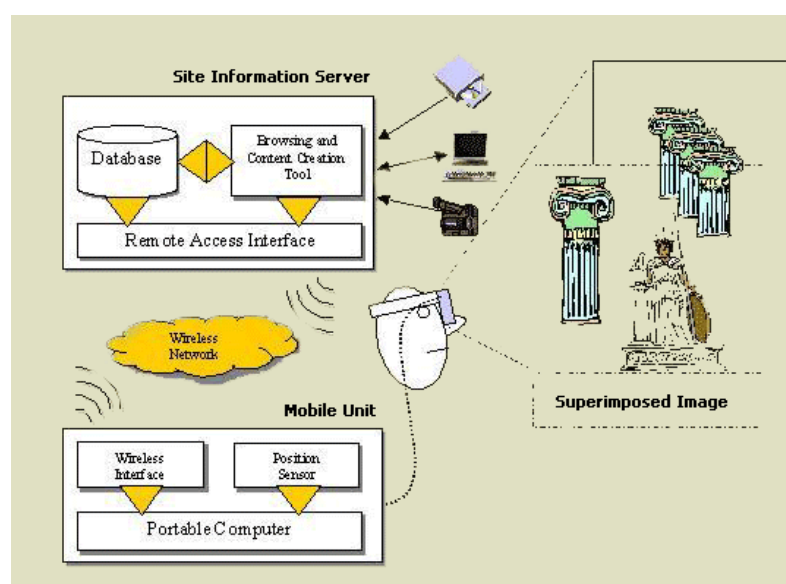


Figura 2.31: Sistema de ARCHEOGUIDE

- Proyecto piloto, no se llegó a implantar.
- La calidad de los objetos 3D no parece ser muy alta.
- Las unidades móviles no son simples dispositivos, sino un equipo especial con un artefacto para la cabeza, cámara de fotos y un portátil.

## 2.15. Ename 974

### 2.15.1. Introducción

El sitio arqueológico de Ename [16] (que cubre un área de 8 hectáreas) se ha desarrollado como museo al aire libre, a la vez que las excavaciones continúan.

En Ename, como en otras excavaciones arqueológicas, los restos son raramente reconocibles por las personas no profesionales del sector. Las paredes fragmentadas o las ruinas no pueden fascinar a la mayoría de los visitantes.

Entre los objetivos del equipo de Ename 974 está el conseguir dar al visitante una idea de cómo era la vida en ese sitio con una nueva técnica. El sistema propuesto, que todavía está en fase de desarrollo, permite que el visitante vea los edificios de los cuales solamente se preservan las fundaciones y aprenda más sobre el sitio y la gente que vivían allí.

El sistema consiste en una cámara de vídeo, un sistema informático, dos monitores, y una pantalla táctil. Un quiosco especialmente diseñado contiene el sistema y protege a visitantes contra los elementos. La cámara de vídeo se dirige hacia una sección particular



Figura 2.32: Simulación del sistema para la visualización de reconstrucciones en Ename

de los restos arqueológicos (por ejemplo las paredes de la iglesia de Santo Salvador) y transmite imágenes en tiempo real de esos restos a las pantallas del monitor en el quiosco.

Con la ayuda de una serie de iconos en la pantalla principal del monitor, los visitantes puede ver las reconstrucciones virtuales de las arquitecturas que estaban situadas en ese punto, sobrepuestas en las excavaciones.

### 2.15.2. Ventajas

Las principales ventajas son:

- La visualización de los objetos se muestra en una pantalla de gran tamaño.
- Visualización en 3D de artefactos que faltan o de partes reconstruidas de sitios dañados.

### 2.15.3. Inconvenientes

Los principales inconvenientes son:

- El usuario no puede manipular a su gusto los objetos reconstruidos.
- Se debe disponer de un espacio perfectamente equipado para poder visualizar los objetos y, por tanto, mucho más complejo que el simple uso de un dispositivo móvil.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.16. Dinosaurios en el Museo del Jurásico de Asturias

### 2.16.1. Introducción

El Museo del Jurásico de Asturias (MUJA) se encuentra situado en la Rasa de San Telmo, entre las localidades asturianas de Colunga y Lastres (España).

Bajo la forma de una gran huella de dinosaurio, el Museo acoge una de las muestras más completas y didácticas del mundo sobre estos fascinantes reptiles.

El edificio consta de tres grandes áreas, cada una de las cuales está dedicada a uno de los periodos en los que se divide el Mesozoico: Triásico, Jurásico Y Cretácico.

A lo largo del recorrido por la exposición, se ofrece una amplia información sobre distintos aspectos de la vida de los dinosaurios, grupo particular de reptiles terrestres que aparecieron hace unos 230 millones de años, extinguiéndose, en su mayoría, hace 65 millones de años.

Gracias a la colaboración entre el diario Público, el Museo Jurásico de Asturias y el Gobierno del Principado de Asturias podemos utilizar una aplicación de realidad aumentada para visualizar cuatro dinosaurios.



Figura 2.33: Dinosaurios en el MUJA

Se trata de una aplicación de escritorio que debemos instalar en el equipo. Se deben imprimir unos códigos y a través de la webcam podremos visualizar los dinosaurios.

### 2.16.2. Ventajas

Las principales ventajas son:

- Permite la visualización de dinosaurios, lo que añade valor a la visita al museo.

### 2.16.3. Inconvenientes

Los principales inconvenientes son:

- Debe utilizarse un ordenador. Por tanto, no está pensada para dispositivos móviles.
- Únicamente está pensada para realidad aumentada basada en *markers*, no en la geoposición.

## 2.17. Conclusiones

A lo largo de estas páginas se ha realizado una revisión de la situación actual de proyectos que utilicen técnicas de realidad aumentada aplicados al campo educativo y otros tipos de escenarios donde también ha demostrado su función pedagógica, como es el caso de los museos.

En los entornos educativos han empezado a aceptar, aunque lentamente, la aplicación de las tecnologías de la información y las comunicaciones en sus procesos didácticos, y han comenzado a descubrir como pueden éstas participar de forma activa en el proceso formativo que se lleva a cabo en las escuelas.

Además, las tecnologías de la información y las comunicaciones pueden contribuir al fomento del turismo, y también a la conservación de los recursos que atraen a los visitantes. Existe una parte común en las cadenas de valor de ambos propósitos: las fases de producción y gestión de contenidos resultan imprescindibles para fabricar productos de calidad. En el caso de la salvaguarda del patrimonio cultural e histórico cobra especial importancia la digitalización, existiendo en la actualidad iniciativas nacionales y europeas destinadas a este propósito.

En muchos casos vistos, la aplicación de estas nuevas tecnologías hará indispensable el uso de información virtual, que implicará la incorporación de nuevos sistemas electrónicos y salas de proyección preparadas específicamente para contener sistemas de interactividad para grupos, cuya tecnología, resulta muy costosa, de muy difícil operatividad y mantenimiento técnico.

Como se ha podido observar a través de la revisión que se ha ido ofreciendo, casi la totalidad carecen de las posibilidades que ofrecen las nuevas tecnologías de la información. Si bien es verdad, que poco a poco se van introduciendo debido a la multitud de aplicaciones que se pueden llegar a construir.

En ciertos escenarios concretos, como también hemos visto, han ido más allá y se han propuesto desarrollar sistemas que permitan una mayor interacción del usuario con los objetos. Para ello, en su mayoría utilizan técnicas ya vistas como es el caso de la realidad aumentada.

Pero todos estos sistemas tienen una serie de inconvenientes que trataremos de solventar. Muchos de los proyectos presentados están todavía en fase de análisis. Estudian cómo podría funcionar el sistema y las ventajas que este aportaría, pero no hay nada que los usuarios finales puedan probar.

Otro problema visto en estos sistemas es que crear un escenario de objetos sería una tarea que conllevaría una gran cantidad de recursos, ya que no podrían aprovechar trabajos anteriormente realizados por parte de diseñadores gráficos.

## 2. Estado del arte

---

Nuestra plataforma trata de solventar todos estos obstáculos ofreciendo una aplicación final totalmente operativa, aprovechando posibles modelos de objetos 3D que el profesor o el encargado de realizar la visita podrá configurar a su medida.

En definitiva, la utilización de las tecnologías de la información y las comunicaciones representa una manera diferente de satisfacer necesidades de siempre y puede colaborar a mejorar la calidad en los entornos expositivos, bien sean educativos o museísticos.

Además, al estar orientadas mayoritariamente a un público joven, se encuentran más familiarizados con estas tecnologías y dispositivos. Así se consigue transmitir conocimientos bajo la fórmula educación + diversión.

Por todos estos motivos, administraciones públicas y los demás entes implicados han de hacer un especial esfuerzo en conseguir que este tipo de servicios innovadores y diferenciadores sean una realidad.

# Capítulo 3

## Análisis

Las dos siguientes secciones se corresponden con la Especificación de Requisitos de Software (ERS). Ésta se realiza según el estándar de *IEEE 830*<sup>1</sup>:

### 3.1. Descripción general del sistema

#### 3.1.1. Perspectiva del producto

El sistema es un producto final totalmente independiente y autónomo. Se describe a continuación qué vamos a hacer teniendo una serie de restricciones:

##### Interfaces del sistema

El proyecto es una nueva versión de mi Proyecto Fin de Carrera de la Ingeniería Técnica en la que, para mostrar los objetos mediante técnicas de realidad aumentada, se utilizaba un portátil y una cámara web, lo que restaba movilidad y flexibilidad al usuario.

Esta nueva versión se desarrolla para dispositivos móviles. En concreto para todos aquellos con el sistema operativo Android.

Además, en esta versión nos aprovecharemos de las ventajas que ofrecen los dispositivos móviles. En concreto, utilizaremos su receptor GPS para utilizar realidad aumentada en exteriores. También, haremos uso de la cámara integrada del dispositivo para reconocer patrones en interiores.

Por último, también se podrán crear exhibiciones o visitas personalizadas en las que mostrar distintos objetos tridimensionales con su información asociada. Esta información se almacenará en un servidor externo y el dispositivo móvil se conectará mediante su conexión de datos.

---

<sup>1</sup> “IEEE STANDARD 830-1998 - IEEE Recommended Practice for Software Requirements Specifications”: <http://standards.ieee.org/findstds/standard/830-1998.html>

### 3. Análisis

---

#### Interfaces de usuario

Parece claro que el futuro de las interfaces de usuario pasa por ser táctiles. Este será el tipo de interfaces en nuestra aplicación para dispositivos móviles.

Se busca que sean fácilmente usables por cualquier tipo de usuario, independientemente de sus conocimientos o destreza en el uso de las nuevas tecnologías.

Además, las interfaces serán sencillas, en el sentido de que la aplicación trata de añadir información al mundo real y para ello simplemente reconoceremos patrones o haremos uso de nuestra posición actual. Además, el usuario apenas tendrá necesidad de introducir datos en el sistema que es una de las tareas más pesadas en las interfaces táctiles.

#### Interfaces hardware

Como hemos dicho, el sistema dispondrá de un servidor web de apoyo, en el que se almacenará la información relativa a los objetos a mostrar. Por tanto, dividiremos esta sección en dos partes:

- Servidor: la aplicación web estará almacenada en un servidor que estará escuchando por el puerto 80.
- Cliente: como hemos dicho, el sistema es una aplicación para dispositivos móviles, que dispongan de cámara, receptor GPS y conexión de datos.

#### Interfaces software

- Servidor: para la aplicación web, tendremos un servidor web Apache 2.2.\* o superior.
- Cliente: el sistema operativo del dispositivo móvil deberá ser Android 2.2 o superior

#### Interfaces de comunicaciones

Se utiliza el protocolo TCP/IP para comunicar los clientes y el servidor. Además, como hemos dicho, se utilizarán comunicaciones GPS.

#### Restricciones de memoria

En principio, no se prevé ninguna restricción de memoria ni en el servidor ni en los clientes.

#### Funcionamientos

Se realizarán copias de seguridad del servidor semanalmente. Además, en cuanto a la política de protección de datos, cumpliremos la LOPD<sup>2</sup>.

---

<sup>2</sup>Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal: [http://noticias.juridicas.com/base\\_datos/Admin/1o15-1999.html](http://noticias.juridicas.com/base_datos/Admin/1o15-1999.html)

### Requisitos de adaptación

No hay ningún requisito de adaptación de los clientes para el servidor web.

### 3.1.2. Funcionalidad del producto

Se pretende construir una plataforma para la representación de todo tipo de objetos orientada al ámbito educativo basándose en el uso de la realidad aumentada. Esta plataforma tiene como objetivo el desarrollar un conjunto de tecnologías para ayudar a que la labor docente ofrezca una visión más completa de lo que exhibe o ayude a comprender mejor el tema expuesto.

Además de mostrar los objetos, se podrán observar desde distintos puntos. Otro objetivo del sistema será la posibilidad de manipular los objetos trasladándolos e incluso haciendo zoom para acercarse o alejarse y así poder ver hasta los más mínimos detalles.

Se crea un lenguaje muy sencillo, fácil de entender, y fácil de modificar, con el fin de que posteriormente puedan realiar exposiciones de este estilo personas que no sepan nada de programación. Así se podrá añadir o modificar facilmente nuevos objetos a mostrar. De cada objeto se podrá añadir un archivo de sonido, que podrá ser una descripción del objeto; un archivo 3ds y su correspondiente textura.

La plataforma dispondrá de un sistema central que dispondrá de los catálogos y de las distintas presentaciones a mostrar.

Se ha desarrollado un software para dispositivos móviles que se descarga del servidor la presentación y los objetos mostrados.

A través de técnicas de reconocimiento de patrones se consiguen mostrar los objetos tridimensionales, especialmente útil para espacios interiores.

Además, estas técnicas son combinadas con sistemas de geoposicionamiento en las que se muestra información adaptada al contexto en función de la posición en la que se esté.

### 3.1.3. Características de los usuarios

El nivel de conocimiento de cada usuario final deberá de ser un nivel intermedio - bajo tanto en el manejo común de computadores como en el manejo de dispositivos móviles, dado que el modelo navegacional facilita el uso. Por otra parte, no podría ser demasiado complicado ya que la plataforma está inicialmente orientada a alumnos que no tienen porqué estar especializados en ninguna de estas tecnologías.

Otro usuario de la plataforma, será el administrador de sistemas. Éste, si debe de estar especializado y tener conocimientos informáticos.

### 3. Análisis

---

|                        |                                                                                                                       |
|------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Tipo de usuario</b> | Usuario de la aplicación                                                                                              |
| <b>Formación</b>       | Ninguna formación específica                                                                                          |
| <b>Habilidades</b>     | Manejo básico de su terminal móvil                                                                                    |
| <b>Actividades</b>     | El usuario final arrancará la aplicación móvil y la utilizará para recibir información añadida relativa a su entorno. |

|                        |                                                                                                                                                                                                                                |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tipo de usuario</b> | Administrador del sistema                                                                                                                                                                                                      |
| <b>Formación</b>       | Conocimientos en administración de servidores y desarrollo de aplicaciones web                                                                                                                                                 |
| <b>Habilidades</b>     | Combinará conocimientos técnicos y una visión estratégica con algo de creatividad. Debe conocer los estándares web, lenguajes de programación, conocimientos de bases de datos y experiencia en seguridad de aplicaciones web. |
| <b>Actividades</b>     | Mantener y mejorar la aplicación web y todos los sistemas integrados.                                                                                                                                                          |

## 3.2. Requisitos específicos

### 3.2.1. Requisitos comunes de los interfaces

A continuación describiremos todas las entradas y salidas del sistema:

#### Interfaces de usuario

Aquí escribiremos los requisitos del interfaz de usuario para el producto:

|                    |                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-U. 1                                                                                                                                                                        |
| <b>Nombre</b>      | Interfaz accesible e intuitiva                                                                                                                                                    |
| <b>Tipo</b>        | Requisito                                                                                                                                                                         |
| <b>Descripción</b> | El manejo de las funcionalidades de la aplicación debe ser lo más intuitivo posible, de manera que sean muy claras las posibles acciones a llevar a cabo y la manera de hacerlas. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                                                     |

|                    |                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-U. 2                                                                                                                                  |
| <b>Nombre</b>      | La interfaz será táctil                                                                                                                     |
| <b>Tipo</b>        | Requisito                                                                                                                                   |
| <b>Descripción</b> | El dispositivo móvil dispondrá de una interfaz táctil. Por tanto, deberemos diseñar nuestra aplicación para que cumpla esta característica. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                               |

#### Interfaces de hardware

Se describirán las características lógicas entre el producto y los componentes hardware del sistema.

|                    |                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-HW. 1                                                                                                                                                |
| <b>Nombre</b>      | Será necesario el uso de la cámara del dispositivo móvil                                                                                                   |
| <b>Tipo</b>        | Requisito                                                                                                                                                  |
| <b>Descripción</b> | A la hora de aumentar la información del entorno mediante técnicas de realidad aumentada, una de las opciones es reconocer patrones a través de la cámara. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                              |

|                    |                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-HW. 2                                                                                                                                                                           |
| <b>Nombre</b>      | Será necesario disponer de acceso a internet desde el dispositivo móvil                                                                                                               |
| <b>Tipo</b>        | Requisito                                                                                                                                                                             |
| <b>Descripción</b> | Los datos de cada exhibición no estarán almacenados en la aplicación, sino que serán externos a ésta. Por tanto, necesitamos un sistema de conexión para consultarlos y descargarlos. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                                                         |

|                    |                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-HW. 3                                                                                                                                                                                                                                   |
| <b>Nombre</b>      | Será necesario el uso del GPS del dispositivo móvil                                                                                                                                                                                           |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                                                     |
| <b>Descripción</b> | Otra opción a la hora de aumentar la información del entorno mediante técnicas de realidad aumentada, será obtener la posición actual mediante coordenadas GPS y en caso de encontrarse cercano a algún objeto se mostrará información de él. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                                                                                                                 |

#### Interfaces de software

Se describirán las integraciones de nuestro producto con otros productos de software:

|                    |                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-SW. 1                                                                                  |
| <b>Nombre</b>      | Reconocimiento de patrones                                                                   |
| <b>Tipo</b>        | Requisito                                                                                    |
| <b>Descripción</b> | Para reconocer los patrones, se utilizará una librería de realidad aumentada como ARToolkit. |
| <b>Prioridad</b>   | Media/Deseado                                                                                |

### 3. Análisis

---

#### Interfaces de comunicación

Se describirán las interfaces de comunicación con otros sistemas:

|                    |                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RC-IN-COM. 1                                                                                            |
| <b>Nombre</b>      | Se utilizará el protocolo TCP/IP para comunicar los clientes y el servidor                              |
| <b>Tipo</b>        | Restricción                                                                                             |
| <b>Descripción</b> | La transferencia de los ficheros y de las rutas preconfiguradas se harán a través del protocolo TCP/IP. |
| <b>Prioridad</b>   | Media/Deseado                                                                                           |

#### 3.2.2. Requisitos funcionales

##### Requisitos funcionales generales

|                    |                                                                                                                                                                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 1                                                                                                                                                                                                                                                                                                                                |
| <b>Nombre</b>      | Sistema enfocado al ámbito educativo                                                                                                                                                                                                                                                                                                |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                                                                                                                                           |
| <b>Descripción</b> | La plataforma está pensada para ofrecer una ayuda a los docentes, aunque con dicha plataforma se podrán adaptar las aplicaciones a cualquier otro ámbito como por ejemplo: arquitectura, medicina, arqueología, museos, entrenamiento militar, industria del entretenimiento, ingeniería, robótica, publicidad y un largo etcétera. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                                                                                                                                                                                                                       |

|                    |                                                                                                                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 2                                                                                                                                                                                                                                        |
| <b>Nombre</b>      | Independencia entre la plataforma y las aplicaciones                                                                                                                                                                                        |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                                                   |
| <b>Descripción</b> | Se ha creado por un lado la plataforma y por lo otro lado está el conjunto de ficheros de configuración y de datos. Así existe una independencia entre la plataforma y las aplicaciones y por lo tanto dispondremos de un sistema flexible. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                                                                                                               |

|                    |                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 3                                                                                                                                        |
| <b>Nombre</b>      | Sistema flexible                                                                                                                            |
| <b>Tipo</b>        | Requisito                                                                                                                                   |
| <b>Descripción</b> | Todos los datos serán obtenidos de ficheros. Al haber independencia entre la plataforma y las aplicaciones le otorga una gran flexibilidad. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                               |

#### Requisitos funcionales de la aplicación para dispositivos móviles

|                    |                                                                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 4                                                                                                                                                                                                                              |
| <b>Nombre</b>      | Reconocimiento de patrones y visualización de objetos                                                                                                                                                                             |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                                         |
| <b>Descripción</b> | Se comparan unos determinados patrones con cada frame que capta la cámara y si resulta que el patrón está contenido en el frame, se muestra el objeto asignado a ese patrón. Esto tiene la ventaja que se produce en tiempo real. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                                                                                                     |

|                    |                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 5                                                                                                       |
| <b>Nombre</b>      | Geoposicionar información.                                                                                 |
| <b>Tipo</b>        | Requisito                                                                                                  |
| <b>Descripción</b> | Utilizando el GPS del dispositivo, nos mostrará la información de los objetos cercanos a nuestra posición. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                              |

|                    |                                                                                                                                                       |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 6                                                                                                                                                  |
| <b>Nombre</b>      | Se podrá asignar a cada patrón un determinado objeto, así como su textura y su descripción sonora mediante un fichero de audio.                       |
| <b>Tipo</b>        | Requisito                                                                                                                                             |
| <b>Descripción</b> | Cada patrón tendrá un objeto asignado con una serie de características como la textura o la descripción modificable en los ficheros de configuración. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                                         |

|                    |                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 7                                                                                                                                                                                  |
| <b>Nombre</b>      | El audio explicativo de un objeto debe detenerse al reproducirse otro posterior.                                                                                                      |
| <b>Tipo</b>        | Requisito                                                                                                                                                                             |
| <b>Descripción</b> | Cuando se está reproduciendo un audio explicativo de un objeto, y se detecta otro patrón cuyo objeto tenga asociado otro audio, el primero deberá detenerse para dar paso al segundo. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                                                                         |

|                    |                                                                                                                |
|--------------------|----------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 8                                                                                                           |
| <b>Nombre</b>      | Desde el propio dispositivo se puede manipular el objeto                                                       |
| <b>Tipo</b>        | Requisito                                                                                                      |
| <b>Descripción</b> | Se podrá manipular a gusto del usuario el objeto aumentándolo, girándolo o moviéndolo de manera muy intuitiva. |

### 3. Análisis

---

|                  |               |
|------------------|---------------|
| <b>Prioridad</b> | Media/Deseado |
|------------------|---------------|

#### Requisitos funcionales del servidor web

|                    |                                                                                      |
|--------------------|--------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 9                                                                                 |
| <b>Nombre</b>      | Sitio web público                                                                    |
| <b>Tipo</b>        | Requisito                                                                            |
| <b>Descripción</b> | El acceso al sitio web es público. Se mostrará la información a cualquier visitante. |
| <b>Prioridad</b>   | Baja/Opcional                                                                        |

|                    |                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 10                                                                                                                                                                             |
| <b>Nombre</b>      | Configuración de rutas                                                                                                                                                            |
| <b>Tipo</b>        | Requisito                                                                                                                                                                         |
| <b>Descripción</b> | Se podrán generar rutas personalizadas que se dejarán a disposición del resto de usuarios en el que se configuren un conjunto de objetos o monumentos distribuidos por la ciudad. |
| <b>Prioridad</b>   | Baja/Opcional                                                                                                                                                                     |

#### Requisitos funcionales de los ficheros de configuración

|                    |                                                                                                                                            |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 11                                                                                                                                      |
| <b>Nombre</b>      | Deberán almacenar la definición de las exposiciones                                                                                        |
| <b>Tipo</b>        | Requisito                                                                                                                                  |
| <b>Descripción</b> | Los ficheros tendrán información tanto de los patrones de cada visita, como de cada objeto cultural y sus respectivos recursos multimedia. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                              |

|                    |                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 12                                                                                                                                                                                |
| <b>Nombre</b>      | Los ficheros de configuración utilizarán un lenguaje de marcado                                                                                                                      |
| <b>Tipo</b>        | Requisito                                                                                                                                                                            |
| <b>Descripción</b> | Para facilitar la lectura y transmisión, utilizaremos un lenguaje de marcado, ya que permite un medio homogéneo, sencillo y claro de representación e intercambio de la información. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                                                                        |

|                    |                                                                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 13                                                                                                                                                            |
| <b>Nombre</b>      | Ficheros de configuración divididos según el tipo de aspecto que abarcan                                                                                         |
| <b>Tipo</b>        | Requisito                                                                                                                                                        |
| <b>Descripción</b> | Cada fichero corresponde a un aspecto general de las exposiciones. Hay ficheros que definen los patrones y otros que definen los objetos que poseen información. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                                                    |

|                    |                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RF 14                                                                                                                                 |
| <b>Nombre</b>      | Para cada exposición habrá un conjunto de ficheros                                                                                    |
| <b>Tipo</b>        | Requisito                                                                                                                             |
| <b>Descripción</b> | Cada exposición puede ser configurada a gusto del usuario y por lo tanto sólo hay necesidad de cambiar los ficheros de configuración. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                         |

#### 3.2.3. Requisitos de rendimiento

|                    |                                                                                    |
|--------------------|------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 1                                                                              |
| <b>Nombre</b>      | Número de dispositivos conectados                                                  |
| <b>Tipo</b>        | Requisito                                                                          |
| <b>Descripción</b> | El número máximo de transacciones por segundo que soportará el sistema será de 50. |
| <b>Prioridad</b>   | Media/Deseado                                                                      |

#### 3.2.4. Restricciones de diseño

|                    |                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 2                                                                                                                                          |
| <b>Nombre</b>      | Aplicación móvil para Android                                                                                                                  |
| <b>Tipo</b>        | Restricción                                                                                                                                    |
| <b>Descripción</b> | La aplicación móvil será diseñada para el sistema operativo Android (de la versión 2.2 en adelante) y, por tanto, estará desarrollada en Java. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                  |

### 3. Análisis

---

#### 3.2.5. Atributos del software del sistema

##### Seguridad

|                    |                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 3                                                                                                                                   |
| <b>Nombre</b>      | Las preferencias del usuario se almacenarán en su propio dispositivo                                                                    |
| <b>Tipo</b>        | Requisito                                                                                                                               |
| <b>Descripción</b> | A la hora de guardar las preferencias configuradas por el usuario, éstas se guardarán en la tarjeta de memoria de su dispositivo móvil. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                           |

|                    |                                                                          |
|--------------------|--------------------------------------------------------------------------|
| <b>Número</b>      | RNF 4                                                                    |
| <b>Nombre</b>      | Ficheros de log                                                          |
| <b>Tipo</b>        | Requisito                                                                |
| <b>Descripción</b> | Se almacenarán en los servidores todos los eventos producidos en un log. |
| <b>Prioridad</b>   | Media/Deseado                                                            |

##### Fiabilidad

|                    |                                                                                                                                                                                                                                                                        |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 5                                                                                                                                                                                                                                                                  |
| <b>Nombre</b>      | Fiabilidad en los datos GPS                                                                                                                                                                                                                                            |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                                                                              |
| <b>Descripción</b> | Debido al carácter militar del sistema GPS, el Departamento de Defensa de los EE. UU. se reservaba la posibilidad de incluir un cierto grado de error aleatorio que podía variar de los 15 a los 100 m. Se deberá tener un sistema alternativo que simule la posición. |
| <b>Prioridad</b>   | Baja/Opcional                                                                                                                                                                                                                                                          |

##### Disponibilidad

|                    |                                                                    |
|--------------------|--------------------------------------------------------------------|
| <b>Número</b>      | RNF 6                                                              |
| <b>Nombre</b>      | Servidor disponible                                                |
| <b>Tipo</b>        | Requisito                                                          |
| <b>Descripción</b> | El servidor web debe mostrar, al menos, un 95 % de disponibilidad. |
| <b>Prioridad</b>   | Alta/Esencial                                                      |

|                    |                                                                                                                                                                                                             |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 7                                                                                                                                                                                                       |
| <b>Nombre</b>      | El sistema reconocerá que exista una conexión de datos activa al iniciar la aplicación                                                                                                                      |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                   |
| <b>Descripción</b> | El sistema antes de arrancar comprobará la existencia y buen funcionamiento de una conexión de datos en el dispositivo. En caso de no encontrarla nos informará de tal suceso y no proseguirá la ejecución. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                                                                                               |

#### Mantenibilidad

|                    |                                                                                                                                                                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 8                                                                                                                                                                                                                                    |
| <b>Nombre</b>      | Configuración externa                                                                                                                                                                                                                    |
| <b>Tipo</b>        | Requisito                                                                                                                                                                                                                                |
| <b>Descripción</b> | Al realizar un sistema flexible, las modificaciones o cambios de funcionamiento se harán a través de los ficheros de configuración, sin necesidad de modificar la plataforma. Estas acciones las realizará el desarrollador del sistema. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                                                                                                                                            |

|                    |                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 9                                                                                                              |
| <b>Nombre</b>      | Nuevas versiones del producto                                                                                      |
| <b>Tipo</b>        | Requisito                                                                                                          |
| <b>Descripción</b> | La instalación de las actualizaciones del software para el dispositivo móvil debe realizarse de manera automática. |
| <b>Prioridad</b>   | Media/Deseado                                                                                                      |

#### Portabilidad

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| <b>Número</b>      | RNF 10                                                                         |
| <b>Nombre</b>      | Portabilidad de la aplicación móvil                                            |
| <b>Tipo</b>        | Requisito                                                                      |
| <b>Descripción</b> | La aplicación móvil será totalmente dependiente del sistema operativo Android. |
| <b>Prioridad</b>   | Media/Deseado                                                                  |

### 3. Análisis

---

#### 3.2.6. Otros requisitos

|                    |                                                                                                                                        |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Número</b>      | OR 1                                                                                                                                   |
| <b>Nombre</b>      | Cumplimiento de la LOPD                                                                                                                |
| <b>Tipo</b>        | Requisito                                                                                                                              |
| <b>Descripción</b> | Se trata de un requisito legal que debe cumplir nuestro sitio web. En este caso deberá cumplir la Ley Orgánica de Protección de Datos. |
| <b>Prioridad</b>   | Alta/Esencial                                                                                                                          |

### 3.3. Descripción de los actores

#### 3.3.1. Actor usuario del sistema

Tal y como se había descrito en el apartado anterior, se distinguen dos roles de usuario:

- Actor usuario de la aplicación.
- Actor administrador del sistema.

##### Actor Usuario de la aplicación

Es un actor primario que podrá realizar cualquier operación disponible en la aplicación del dispositivo móvil, dado que no hay opciones a las que no pueda acceder y tampoco hay contraseñas con las que dar o quitar privilegios a los usuarios.

Principalmente, iniciará la aplicación y será el responsable de ir utilizando la cámara para capturar patrones y que éstos se conviertan en información.

##### Actor Administrador del sistema

La función de este actor primario sería la de, simplemente, cargar en el servidor los ficheros de configuración de cada exposición o zona. Además, podrá modificar las propiedades de cada patrón o de cada objeto. Esta labor se hará directamente sobre los ficheros de configuración.

### 3.4. Modelo de Casos de Uso

Mediante estos diagramas se representan gráficamente las acciones que el sistema debe soportar para ser ejecutadas por cada actor.

#### 3.4.1. Límites del sistema

A continuación se establecen los límites del sistema, así como los principales casos de uso que posteriormente serán refinados en siguientes etapas del análisis.

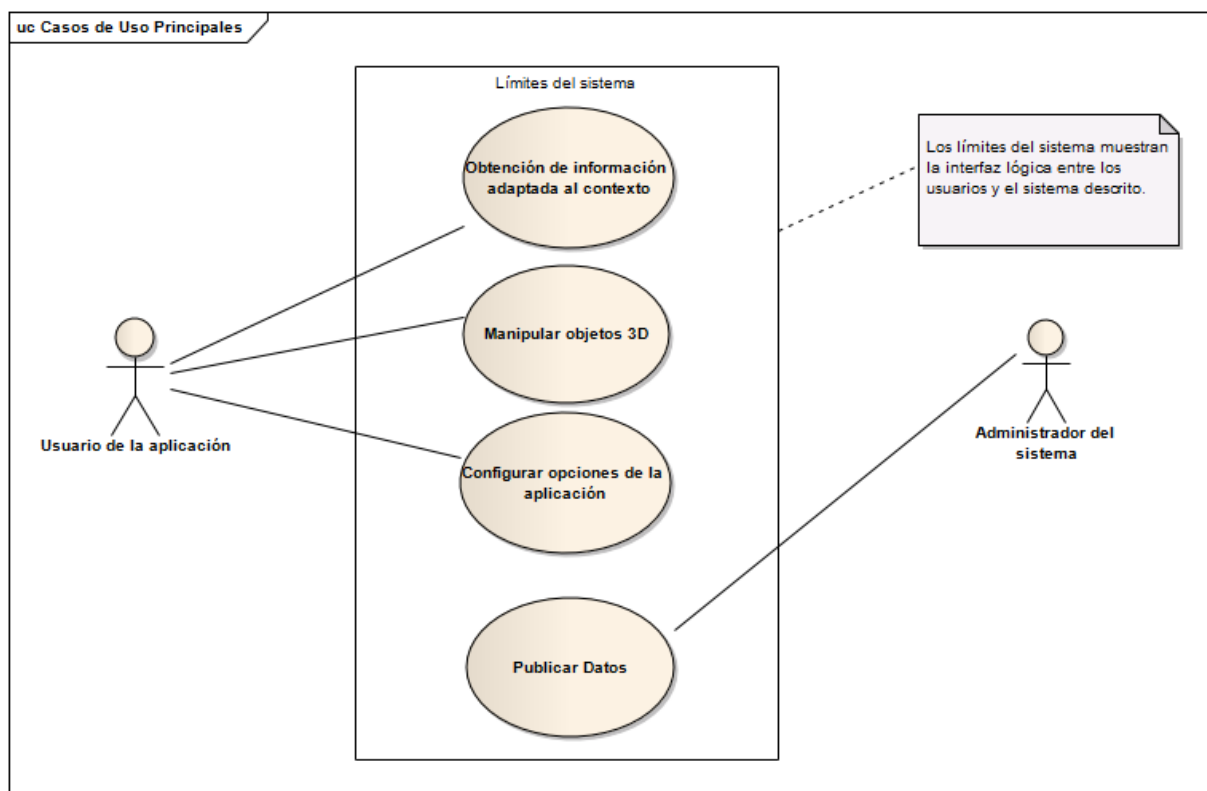


Figura 3.1: Diagrama de casos generales de uso

### 3.4.2. Casos de uso específicos

#### Obtención de información adaptada al contexto

Refinando el caso de uso general *Obtención de información adaptada al contexto* del diagrama 3.1 (página 67), identificamos dos casos más específicos como son:

- Mostrar patrón y obtener información
- Obtener geoposición e información asociada

#### Manipular objetos 3D

Refinando el caso de uso general *Manipular objetos 3D* del diagrama 3.1 (página 67), identificamos dos casos más específicos como son:

- Desplazar objeto
- Escalar objeto

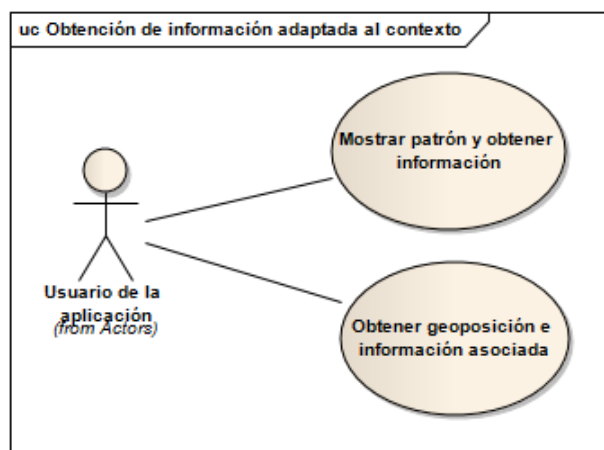


Figura 3.2: Diagrama de casos de uso específico para "Obtención de información adaptada al contexto"

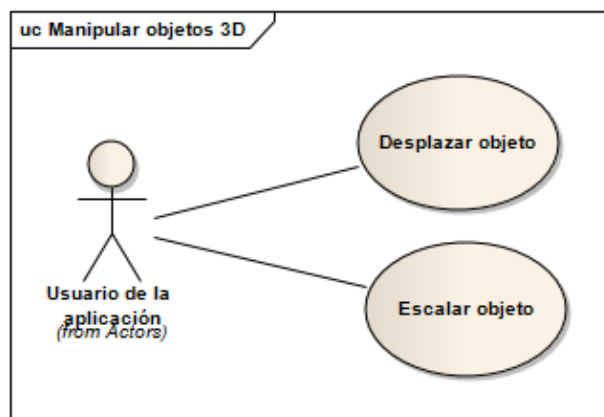


Figura 3.3: Diagrama de casos de uso específico para "Manipular objetos 3D"

#### Configurar opciones de la aplicación

Refinando el caso de uso general *Configurar opciones de la aplicación* del diagrama 3.1 (página 67), identificamos dos casos más específicos como son:

- Configuración de patrones y posiciones
- Configuración de objetos e información

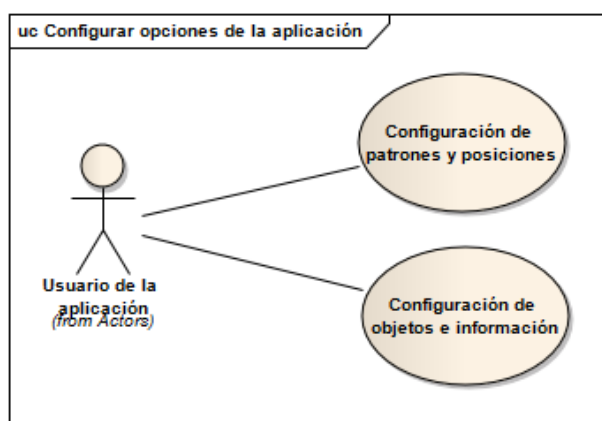


Figura 3.4: Diagrama de casos de uso específico para “Configurar opciones de la aplicación”

### Publicar datos

Refinando el caso de uso general **Publicar datos** del diagrama 3.1 (página 67), identificamos varios casos más específicos como son:

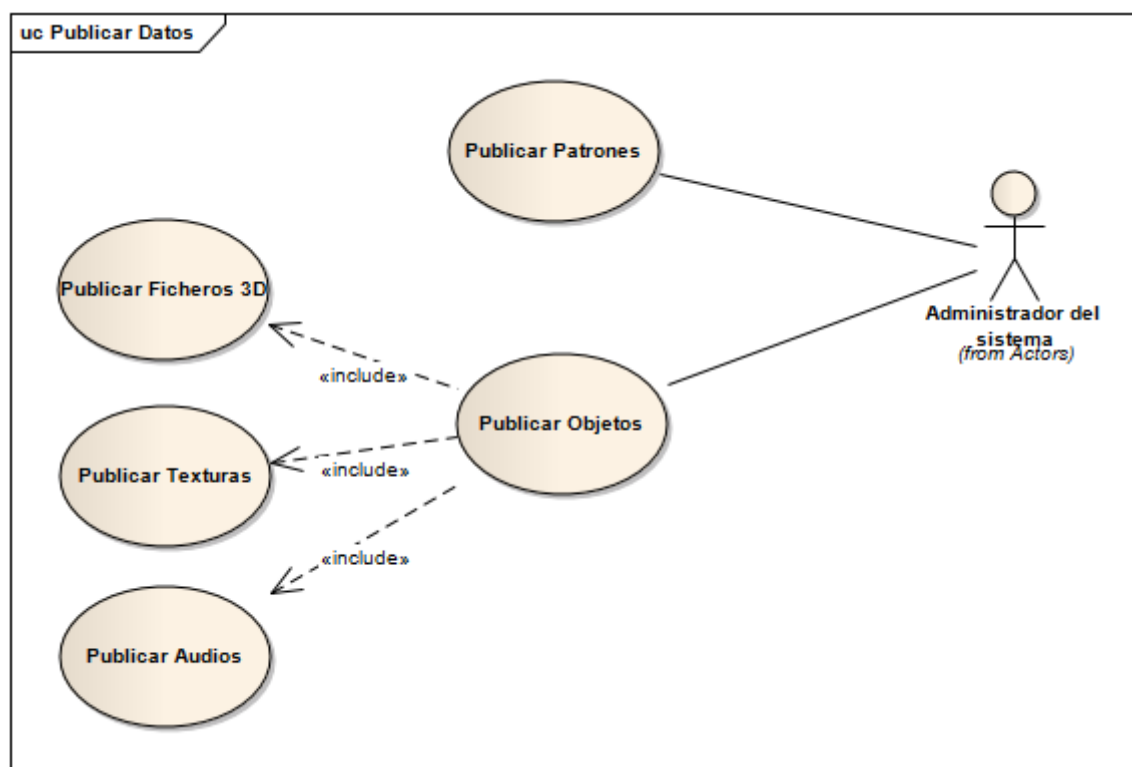


Figura 3.5: Diagrama de casos de uso específico para “Publicar datos”

- Publicar Patrones

### 3. Análisis

---

- Publicar Objetos
- Publicar Ficheros 3D
- Publicar Texturas
- Publicar Audios

## 3.5. Especificación de Casos de Uso

| Especificación de Casos de Uso |                                                                                                      |                             |
|--------------------------------|------------------------------------------------------------------------------------------------------|-----------------------------|
| Caso de Uso                    | 1.1                                                                                                  | Arrancar Aplicación Cliente |
| Objetivos                      | Antes de comenzar la navegación mediante el dispositivo móvil será necesario arrancar la aplicación. |                             |
| Actores                        | Actor Usuario de la aplicación                                                                       |                             |
| Proceso                        | El usuario ejecutará la aplicación del cliente en su dispositivo móvil.                              |                             |

Cuadro 3.35: Especificación de Caso de Uso 1.1

| Especificación de Casos de Uso |                                                                                                                                                                                                                                                           |                                      |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| Caso de Uso                    | 1.2                                                                                                                                                                                                                                                       | Mostrar patrón y obtener información |
| Objetivos                      | Se mostrará el patrón a la cámara y se transformará en un objeto informativo                                                                                                                                                                              |                                      |
| Actores                        | Actor Usuario de la aplicación                                                                                                                                                                                                                            |                                      |
| Proceso                        | El usuario mostrará el patrón en la cámara del dispositivo móvil y éste mediante reconocimiento de imágenes determinará si coincide con algunos de los patrones almacenados. En caso afirmativo, se mostrará el objeto informativo asociado a tal patrón. |                                      |

Cuadro 3.36: Especificación de Caso de Uso 1.2

| Especificación de Casos de Uso |                                                                                                                                                                                               |                                            |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| Caso de Uso                    | 1.3                                                                                                                                                                                           | Obtener geoposición e información asociada |
| Objetivos                      | El dispositivo móvil dectará nuestra posición y, si hay información asociada a ella, la mostrará                                                                                              |                                            |
| Actores                        | Actor Usuario de la aplicación                                                                                                                                                                |                                            |
| Proceso                        | El usuario dispondrá en su dispositivo móvil de unas localizaciones en las que existen información contextual. Si éste se aproxima a una de estas localizaciones, la información se mostrará. |                                            |

Cuadro 3.37: Especificación de Caso de Uso 1.3

| Especificación de Casos de Uso |                                                                                                                                                  |                  |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Caso de Uso                    | 1.4                                                                                                                                              | Desplazar Objeto |
| Objetivos                      | Una vez que se esté visualizando un objeto, el usuario podrá interactuar con él. En este caso se podrá mover de posición el objeto.              |                  |
| Actores                        | Actor Usuario de la aplicación                                                                                                                   |                  |
| Proceso                        | Al ser una interfaz táctil, el usuario deberá arrastrar el dedo sobre la pantalla hacia la posición que desee que se produzca el desplazamiento. |                  |

Cuadro 3.38: Especificación de Caso de Uso 1.4

| Especificación de Casos de Uso |                                                                                                                                                                                                                                                                                                                            |                |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Caso de Uso                    | 1.5                                                                                                                                                                                                                                                                                                                        | Escalar Objeto |
| Objetivos                      | Una vez que se esté visualizando un objeto, el usuario podrá interactuar con él. En este caso se podrá aumentar o disminuir el zoom para ver más detalles del objeto.                                                                                                                                                      |                |
| Actores                        | Actor Usuario de la aplicación                                                                                                                                                                                                                                                                                             |                |
| Proceso                        | El usuario podrá escalar un objeto. Para ello deberá hacer un gesto típico en las interfaces táctiles: <i>pinch zoom</i> . Si desea aumentar el tamaño del objeto, habrá que colocar dos dedos sobre la pantalla y sin levantarlos, alejarlos. En caso contrario, si los dedos se acercan, el objeto disminuirá de tamaño. |                |

Cuadro 3.39: Especificación de Caso de Uso 1.5

| Especificación de Casos de Uso |                                                                                                                                                                                                                                                                                                                         |                                        |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| Caso de Uso                    | 1.6                                                                                                                                                                                                                                                                                                                     | Configuración de patrones y posiciones |
| Objetivos                      | Se podrá configurar los datos relativos a los <i>markers</i> y a las posiciones GPS de la exposición.                                                                                                                                                                                                                   |                                        |
| Actores                        | Actor Usuario de la aplicación                                                                                                                                                                                                                                                                                          |                                        |
| Proceso                        | Al iniciar la aplicación, se cargarán los datos relativos a los <i>markers</i> y a las posiciones GPS. Estos datos nos vendrán en un fichero de configuración específico que estará alojado en nuestro servidor. Pero también podremos establecer manualmente la ubicación de este fichero y volver a cargar los datos. |                                        |

Cuadro 3.40: Especificación de Caso de Uso 1.6

| Especificación de Casos de Uso |                                                                                                                                                                                                                                                                                                                                                                                 |                                        |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| Caso de Uso                    | 1.7                                                                                                                                                                                                                                                                                                                                                                             | Configuración de objetos e información |
| Objetivos                      | Se podrá configurar los datos relativos a los objetos y a la información contenida en éstos.                                                                                                                                                                                                                                                                                    |                                        |
| Actores                        | Actor Usuario de la aplicación                                                                                                                                                                                                                                                                                                                                                  |                                        |
| Proceso                        | Al iniciar la aplicación, se cargarán los datos relativos a los objetos y su información. Estos datos nos vendrán en un fichero de configuración específico que estará alojado en nuestro servidor. Pero también podremos establecer manualmente la ubicación de este fichero. Una vez seleccionado el fichero, los datos serán obtenidos y cargados de nuevo en la aplicación. |                                        |

Cuadro 3.41: Especificación de Caso de Uso 1.7

| Especificación de Casos de Uso |                                                            |                          |
|--------------------------------|------------------------------------------------------------|--------------------------|
| Caso de Uso                    | 1.8                                                        | Salir Aplicación Cliente |
| Objetivos                      | Cuando se desee se puede cerrar la aplicación del cliente. |                          |
| Actores                        | Actor Usuario de la aplicación                             |                          |
| Proceso                        | El usuario finalizará la aplicación cuando lo desee.       |                          |

Cuadro 3.42: Especificación de Caso de Uso 1.8

| Especificación de Casos de Uso |                                                                                                                                                                                                                                                                                                          |                   |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Caso de Uso                    | 1.9                                                                                                                                                                                                                                                                                                      | Publicar Patrones |
| Objetivos                      | El administrador será el encargado de publicar la configuración de qué patrones reconocerá la aplicación, así como qué posiciones geográficas y los objetos asignados a estas.                                                                                                                           |                   |
| Actores                        | Actor Administrador del sistema                                                                                                                                                                                                                                                                          |                   |
| Proceso                        | Se creará un fichero de configuración para cada exposición. En este deberán figurar los patrones que el sistema reconocerá, así como los objetos informativos que se mostrarán para cada uno de ellos. Este fichero también nos proporcionará las posiciones geográficas en las que mostrar información. |                   |

Cuadro 3.43: Especificación de Caso de Uso 1.9

| Especificación de Casos de Uso |                                                                                                                                                                                                                                                                                                              |                  |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Caso de Uso                    | 1.10                                                                                                                                                                                                                                                                                                         | Publicar Objetos |
| Objetivos                      | El administrador será el encargado de publicar la información que contendrá cada uno de los objetos asociados a un <i>marker</i> o una posición geográfica.                                                                                                                                                  |                  |
| Actores                        | Actor Administrador del sistema                                                                                                                                                                                                                                                                              |                  |
| Proceso                        | Se creará un fichero de configuración para los objetos de cada exposición. En este deberán figurar cada uno de los objetos informativos que se mostrarán. De cada uno de ellos se podrá configurar una breve descripción, un fichero 3D, una textura asociada a tal fichero y un audio explicativo del mismo |                  |

Cuadro 3.44: Especificación de Caso de Uso 1.10

| Especificación de Casos de Uso |                                                                                                                                                                                                          |                      |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Caso de Uso                    | 1.11                                                                                                                                                                                                     | Publicar Ficheros 3D |
| Objetivos                      | El administrador publicará los ficheros 3D asociados a cada objeto                                                                                                                                       |                      |
| Actores                        | Actor Administrador del sistema                                                                                                                                                                          |                      |
| Proceso                        | Los objetos informativos podrán contener un fichero tridimensional. En el fichero de configuración se especificará dónde se encuentran estos objetos. El administrador será el encargado de publicarlos. |                      |

Cuadro 3.45: Especificación de Caso de Uso 1.11

| Especificación de Casos de Uso |                                                                                                                                                                                                                                            |                   |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Caso de Uso                    | 1.12                                                                                                                                                                                                                                       | Publicar Texturas |
| Objetivos                      | El administrador publicará las texturas de los ficheros 3D asociadas a cada objeto                                                                                                                                                         |                   |
| Actores                        | Actor Administrador del sistema                                                                                                                                                                                                            |                   |
| Proceso                        | Los objetos informativos podrán contener un objeto tri-dimensional, y por tanto, una textura de estos. En el fichero de configuración se especificará dónde se encuentran las texturas. El administrador será el encargado de publicarlas. |                   |

Cuadro 3.46: Especificación de Caso de Uso 1.12

| Especificación de Casos de Uso |                                                                                                                                                                                                                 |                 |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Caso de Uso                    | 1.13                                                                                                                                                                                                            | Publicar Audios |
| Objetivos                      | El administrador publicará los ficheros de audio asociados a cada objeto                                                                                                                                        |                 |
| Actores                        | Actor Administrador del sistema                                                                                                                                                                                 |                 |
| Proceso                        | Los objetos informativos podrán contener un oaudio explicativo. En el fichero de configuración se especificará dónde se encuentran estos archivos de sonido. El administrador será el encargado de publicarlos. |                 |

Cuadro 3.47: Especificación de Caso de Uso 1.13

## 3.6. Análisis de Interfaces de Usuario

A continuación se describirán los modelos y prototipos de nuestra interfaz de usuario. Ésta será táctil, con las ventajas e inconvenientes que ello supone.

Pese a que en nuestro caso en concreto, las interfaces de usuario son realmente sencillas, esta etapa supone es de una gran importancia ya que nos puede ayudar a detectar defectos tempranos respecto a la usabilidad de la aplicación.

### 3.6.1. Descripción de la Interfaz

Para realizar la estas interfaces se utiliza una técnica de prototipado en papel unido a un prototipado basado en bocetos:

- **Prototipado en papel:** Esta técnica de prototipado de baja fidelidad se basa en la utilización de materiales tan básicos como lápiz, el papel y las tijeras para la creación de prototipos simples pero enormemente versátiles.

Consiste en dibujar en un papel, sin entrar en grandes detalles estéticos, las interfaces que se van a probar y valorar. Esto permite que el usuario final interactúe con este material simulando el funcionamiento del sistema.

- **Prototipado basado en bocetos:** Los bocetos son maneras de representar “primeras ideas”. Se usan en la etapa más inicial del diseño, a menudo antes incluso de determinar muchos aspectos del análisis de requisitos, con la finalidad de recoger las primeras impresiones del espacio de trabajo de la interacción.

La clave de los bocetos es su velocidad de producción: Un boceto se realiza en unos 15 ó 20 segundos, de manera que se pueden generar gran cantidad de bocetos en muy poco tiempo.

Ambas técnicas sirven de recogida de ideas iniciales y tienen la ventaja que son muy rápidas de elaborar. Nosotros para realizarlas, hemos utilizado una herramienta para crear mockups <sup>3</sup>.

#### Interfaz principal de la aplicación

La interfaz principal de la aplicación será muy limpia. Utilizaremos todo el espacio disponible de la pantalla del dispositivo móvil para capturar imágenes provenientes de la cámara. En este espacio, es en el que se mostrará la información que obtendremos de nuestro entorno. Se puede ver un boceto en la figura 3.6

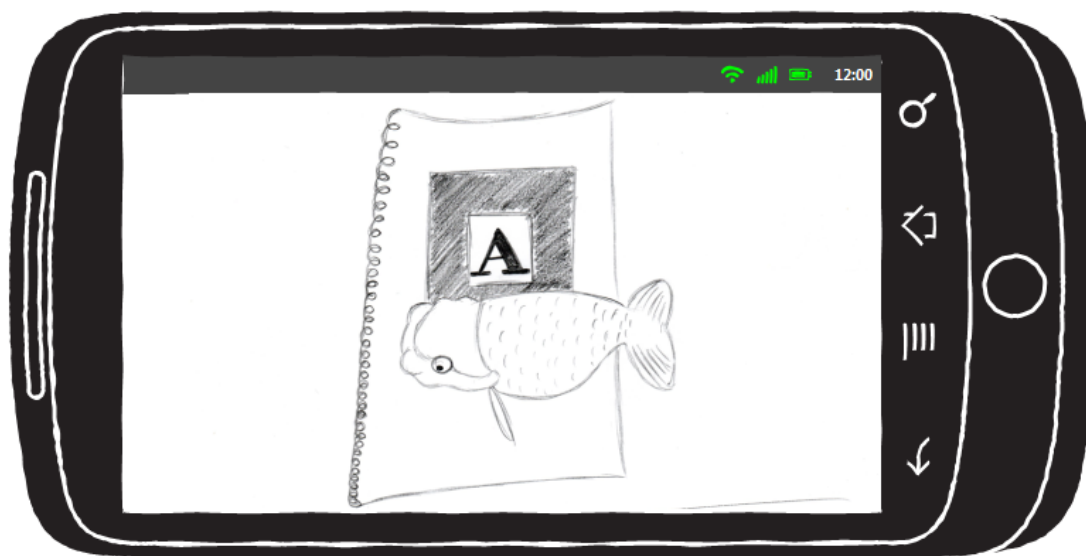


Figura 3.6: Boceto de la interfaz de la aplicación móvil

<sup>3</sup>Mockup: un modelo a escala o tamaño completo de un diseño, utilizado para ser mostrado y evaluado. Por tanto, todo error detectado aquí será más fácil y, sobre todo, más barato de subsanar que en el desarrollo final.

Existen herramientas para realizar estos bocetos. Estas herramientas disponen de un conjunto de elementos o controles como cajas de texto, combo-boxes, imágenes... con los que realizar los diseños.

#### Interfaz de la pantalla de configuración

Seleccionando el menú del teléfono nos aparecen las opciones. Nuestra única opción será entrar al menú de *Configuración* tal y como se observa en la figura 3.7

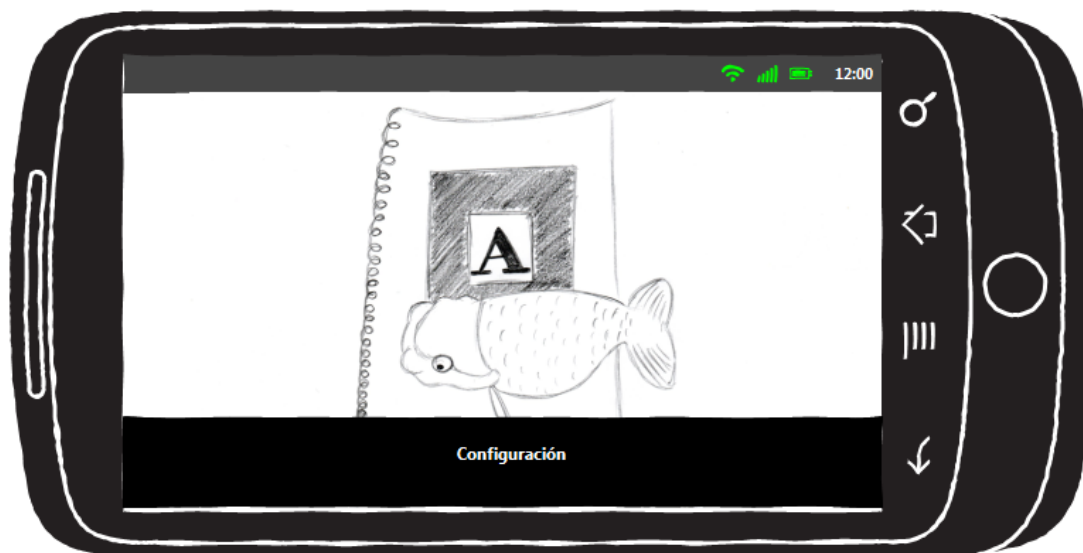


Figura 3.7: Boceto de la interfaz al mostrar las opciones

Una vez seleccionado, nos aparecerá la ventana de configuración para establecer la ubicación de los ficheros de configuración antes explicados, tal y como se observa en la figura 3.8

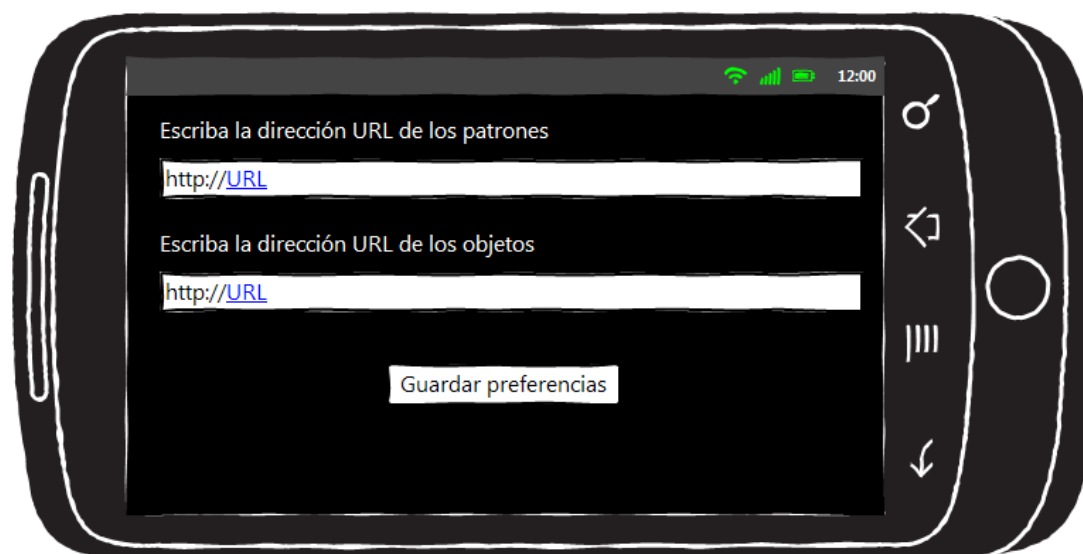


Figura 3.8: Boceto de la interfaz de configuración

# Capítulo 4

## Diseño

Puesto que el sistema a desarrollar, en cuanto a funcionalidad, proporciona servicios bien diferenciados, y para clarificar el estudio de las mismas, lo descompondremos en una serie de módulos, independientes unos de otros.

La ventaja de la descomposición del sistema en módulos, reside en que al existir la independencia de unos con respecto a otros, se puede modificar o eliminar uno de ellos, sin que el resto del sistema se vea afectado, incluso se pueden crear nuevos módulos independientes y asociarlos al sistema sin trastorno alguno a los demás módulos.

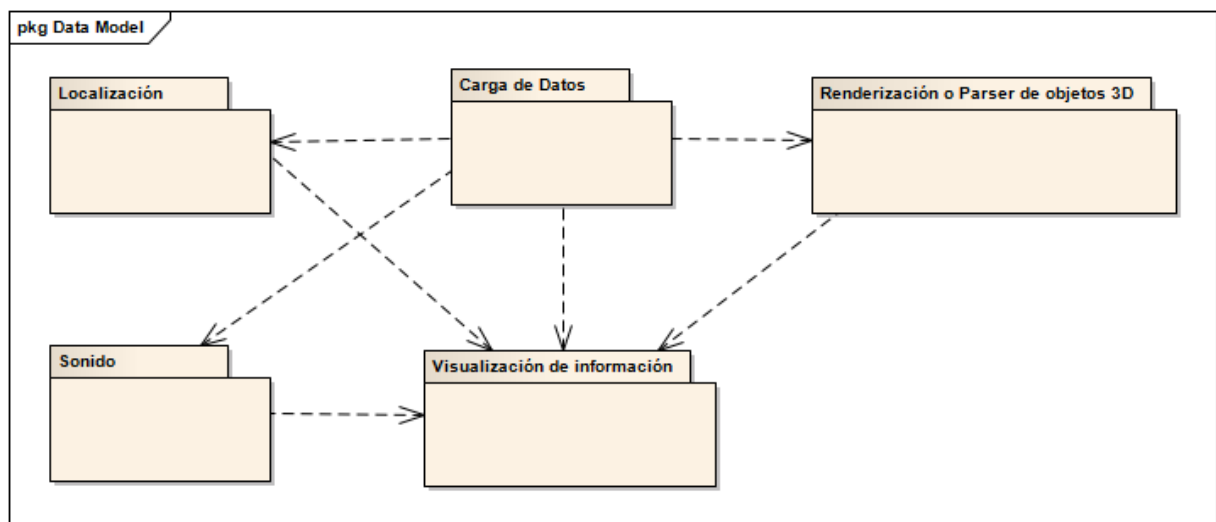


Figura 4.1: Descomposición modular del sistema

Los módulos en los que se divide el sistema son los siguientes:

- Módulo de Carga de Datos.
- Módulo de Localización.
- Módulo de Renderización o Parser de objetos 3D.

- Módulo de Sonido.
- Módulo de Visualización de información.

En la figura 4.1 se puede observar la anterior estructura de una forma gráfica.

### 4.1. Descripción de los módulos del sistema

#### 4.1.1. Módulo de Carga de Datos

Este módulo es el encargado de todo el proceso de arranque y validación de los ficheros de configuración para el buen funcionamiento de la aplicación, así como el almacenamiento de la información de cada patrón, posición GPS y de los objetos asignados a tales patrones o posiciones.

Existen varios ficheros de configuración para la exposición. Cada uno de ellos está destinado a almacenar los datos de una parte de ella. Hacemos esta división de ficheros para que nos resulte más rápido a nosotros y a la persona encargada de crear o modificar la exposición.

Se encarga de analizar los documentos de tal forma que debe comprobar tanto su sintaxis y semántica, como que los recursos incluidos existan. Para ello utilizaremos la tecnología XML[17] por varios motivos:

- Existen herramientas ampliamente contrastadas que nos ayudarán en el proceso de validación de la exposición.
- También existen herramientas que nos facilitarán mucho el acceso a los datos.
- Definiremos un lenguaje para crear los patrones o localizaciones y asignarles un objeto a través de los *XML Schemas (XSD)*<sup>1</sup> [18] de XML. A cada fichero XML, le corresponde un XSD. Gracias a esto nos va a ser muy sencillo ir añadiéndole riqueza al lenguaje.
- En un entorno Java es mucho más sencillo utilizar herramientas XML.
- XML es algo totalmente distinto a un lenguaje de programación, podremos decir que con esto diseñamos nuestra aplicación.
- De todas las formas de crear una exposición por un usuario que no tenga nociones de programación, XML es claramente la tecnología que mejor proporciona ese soporte.

---

<sup>1</sup>Existen varios métodos de validación de los ficheros XML. Los métodos más usados son la **DTD** de XML versión 1.0, el **XML Schema** de W3C, **RELAX NG** de Oasis y **Schematron** de la Academia Sinica Computing Centre.

Dentro de las tecnologías XML vamos a utilizar dos: una para analizar y validar el documento que va a ser una API J2EE, *JAXP* que nos proporciona una implementación del analizador *SAX*, y otra para acceder a su contenido el DOM (Modelo de objeto de Documento) del consorcio W3C que nos proporciona un completo estándar para acceder a los datos del documento una vez validado.

### 4.1.2. Módulo de Localización

Este módulo es el que se encarga de obtener la localización del dispositivo y comprobar si existe alguna información que deba ser mostrada para nuestra posición.

Para obtener nuestra posición utilizaremos el servicio *LOCATION\_SERVICE* que ofrece Android [19].

Para ello buscaremos el mejor proveedor para que nos ofrezca esta información o el que esté disponible. Los proveedores serán o bien mediante el GPS del dispositivo o a través de la red. Mediante cualquiera de estos dos sistemas detectará nuestra posición y nos informará sobre nuestra latitud y longitud.

Así, cada cierto periodo de tiempo comprobaremos si nuestra posición ha cambiado. En caso de que fuese así, obtendremos la nueva posición y el sistema comprobará si existe en los ficheros de configuración alguna posición cercana a la nuestra para, entonces, mostrar la información asociada a esa posición.

Y así sucesivamente cada cierta cantidad de tiempo.

En caso de hallar información asociada a esa posición se mostrará en el dispositivo hasta que finalice la exposición auditiva o, en caso de que no disponga de ella, durante un breve periodo de tiempo.

### 4.1.3. Módulo de Renderización o Parser de objetos 3D

Este módulo es el encargado del proceso de renderización de objetos 3D.

En términos generales, un gráfico 3D difiere de uno 2D principalmente por la forma en que ha sido generado. Este tipo de gráficos se origina mediante un proceso de cálculos matemáticos sobre entidades geométricas tridimensionales producidas en un ordenador, y cuyo propósito es conseguir una proyección visual en dos dimensiones para ser mostrada en una pantalla o impresa en papel.

En general, el arte de los gráficos 3D es similar a la escultura o la fotografía, mientras que el arte de los gráficos 2D es análogo a la pintura. En los programas de gráficos por computadora esta distinción es a veces difusa: algunas aplicaciones 2D utilizan técnicas 3D para alcanzar ciertos efectos como iluminación, mientras que algunas aplicaciones 3D primarias hacen uso de técnicas 2D.

La palabra **renderización** es una adaptación al castellano del vocablo inglés *rendering* y que define un proceso de cálculo complejo desarrollado por un ordenador destinado a generar una imagen 2D a partir de una escena 3D. La traducción más fidedigna es “*interpretación*”, aunque se suele usar el término inglés. Así podría decirse que en el proceso de renderización, la computadora “*interpreta*” la escena 3D y la plasma en una imagen 2D. También se emplean coloquialmente los términos “*renderizar*” y “*renderizado*”.

El proceso de rénder necesita una gran capacidad de cálculo, pues requiere simular gran cantidad de procesos físicos complejos. La capacidad de cálculo se ha incrementado rápidamente a través de los años, permitiendo un grado superior de realismo en los rénders. Estudios de cine que producen animaciones generadas por ordenador hacen uso, en general, de lo que se conoce como *render farm* (granja de rénder) para acelerar la producción de fotogramas.

En esta plataforma se utiliza una *API* especializada para facilitar los procesos en todas las etapas de la generación de gráficos. En este caso se utiliza *min3d*[20] que es una librería basada en OpenGL[21] para Android.

En la aplicación habrá distintos tipos de formatos 3d soportados. Se podrá representar tanto archivos .3ds, .obj y .md2.

En función del tipo de objetos que vayamos a mostrar, se utilizará un parser u otro.

### 4.1.4. Módulo de Sonido

Este módulo es el encargado del proceso de decodificación y reproducción de los ficheros de audio de la plataforma. La reproducción de audio se producirá cuando se visualice un objeto y siempre que éste lo tenga asociado.

Para la reproducción de los audios se utiliza la librería *MediaPlayer* [22] de Android. Pero hemos tenido que adaptarlo para poder utilizarlo en el sistema.

Este módulo permite que cuando se esté reproduciendo un archivo debido a la visualización de un objeto y en el caso que se visualice otro objeto sin haber finalizado la reproducción de ese fichero de audio, se detenga la reproducción y se reproduzca un nuevo audio si procede.

Además, cuando se esté mostrando información basada en la localización, el sistema observará cuando finaliza el audio para volver a utilizar los markers y reconocer patrones.

### 4.1.5. Módulo de Visualización de información

Éste, es el módulo que se encargará de visualizar y representar toda la información virtual superpuesta al mundo real en el que nos encontramos.

## 4.1. Descripción de los módulos del sistema

Se mostrará la información de los objetos 3D con su correspondiente textura en el dispositivo móvil. Además, en caso de disponer de fichero de audio, se reproducirá.

Para utilizar el proceso de reconocimiento de imágenes se utiliza la librería **NyAR-Toolkit** [23], que es una librería de realidad aumentada escrita en Java y específica para dispositivos móviles.

La aplicación permite manipular los objetos, es decir, moverlos virtualmente por la pantalla y escalarlo para verlos con más detalle.

Por último, este módulo también se encargará de mostrar las preferencias de la aplicación, en las que podremos modificar la dirección donde se encuentran los ficheros de configuración de la aplicación.

### 4.1.6. Internacionalización de la aplicación

La internacionalización es el proceso de diseñar una aplicación para que pueda ser adaptada a diferentes idiomas o regiones, sin necesidad de cambios de ingeniería.

Permite que las aplicaciones admitan las codificaciones de datos necesarias para procesar información multilingüe, así como los formatos de moneda, calendarios, etc.

En muchas ocasiones, por no decir la mayoría de veces, que nos compramos un software, que además es el mejor de su clase, y después de comprarlo nos encontramos algo como:

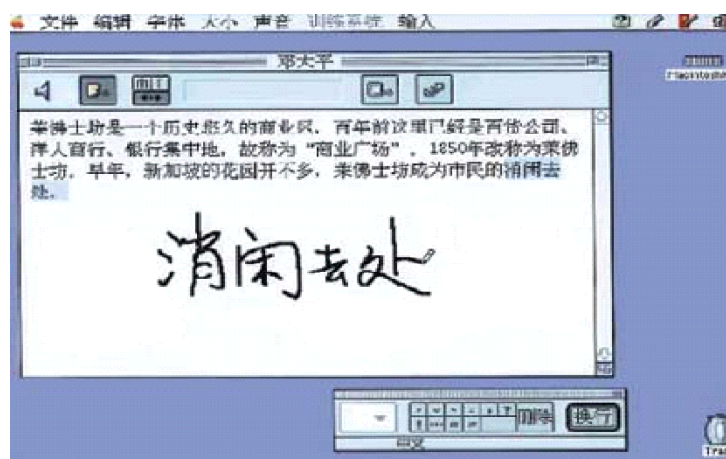


Figura 4.2: Ejemplo de programa en otro idioma

Por esto, en este subsistema nos marcamos una serie de **objetivos**, que pueden ser generales para más aplicaciones:

## 4. Diseño

---

- Asegurar la completa funcionalidad en todos los dispositivos de la plataforma, en todos los escenarios e idiomas.
- Reducir el esfuerzo, y el coste de la localización, acelerando el lanzamiento al mercado.
- Reducir el coste de la asistencia internacional y mejorar la satisfacción del cliente.

Junto a la internacionalización, va ligado el concepto de **Localización**, que es el proceso de adaptar software para una región o idioma, añadiendo componentes específicos de la localidad y traduciendo el texto.

Durante la internacionalización, los textos como las etiquetas de los componentes GUI y los mensajes de error, son almacenados fuera del código fuente, para ser recuperados en tiempo de ejecución. Y como el texto no está dentro del código fuente, el programa no requiere ninguna modificación. Así el mismo ejecutable funcionará en cualquier parte del mundo.

Algunas características de nuestra aplicación desde el punto de vista de la internacionalización son:

- El texto mostrado por el programa está en el idioma nativo del usuario final.
- Los elementos textuales, al estar almacenados fuera del código fuente, son recuperados de forma dinámica.

Y las **ventajas** que presenta esta carga dinámica de los elementos textuales con respecto a la carga estática de los datos, dentro del código son:

- El mantenimiento del código es más fácil si solo hay una versión
- El soporte de nuevos idiomas no requiere re-compilación.
- Los productos se pueden colocar en el mercado más fácilmente.

Hoy en día, todas los programas deberían de estar internacionalizados, ya que sino es así se ha demostrado que se pierde el interés por ellos, en el caso de no conocer el idioma en el que esta la ayuda, o la interfaz de dichos programas.

El proceso de internacionalización en aplicaciones Android es realmente sencillo [24].

En cualquier aplicación Android tendremos una carpeta *res*. En este directorio encontraremos los archivos que definen la interfaz gráfica y las cadenas de texto. Normalmente las cadenas de texto no se escriben en el código, sino que se ponen etiquetas que hacen referencia a dichas cadenas y éstas son escritas en el archivo *res/values/strings.xml*.

Pues bien, si queremos traducir a otro idioma estas cadenas sólo tenemos que copiar el directorio *res/values/* como *value-* y el código del idioma. En el caso de el español, sería *es*.

Dentro de este nuevo directorio tendremos el archivo **strings.xml**, que será igual que el original y lo único que tendremos que hacer es cambiar las cadenas por las de el idioma que hayamos elegido.

Esta aplicación da soporte a la internacionalización, y a poder verse el programa en múltiples idiomas diferentes. En nuestro caso, se da la opción de ver el motor en **inglés**, y **español**, pero traduciendo los ficheros en otro idioma, ya se podría ver también la aplicación en ese nuevo idioma.

## 4.2. Descripción del diseño de clases

En esta sección, se explica cuál es la estructura de cada módulo, así como se muestran las relaciones entre las clases que intervienen en cada módulo, y con los demás módulos.

### 4.2.1. Diseño del Módulo de Carga de Datos

Tal y como habíamos descrito en la sección anterior 4.1.1, este módulo se encargará de cargar en el sistema todos los datos provenientes de los ficheros de configuración.

Estos ficheros de configuración serán *XML* y tendremos distintos tipos ya sean para cargar los markers y las localizaciones o bien para cargar los objetos que contiene la información.

Debido a ello tendremos varios compiladores XML. Tendremos una clase abstracta *XMLCompiler* y subclases que redefinen ésta: *AugmentedRealityXMLCompiler* y *ObjectsXMLCompiler*.

En este módulo se dispondrá de una clase “Base de Conocimiento” (*KnowledgeBase*) que contendrá las estructuras de datos necesarias para ir almacenando todos los datos provenientes de la carga de los ficheros de configuración.

Resulta lógico pensar que sólo dispondremos de una única instancia de esta clase en toda la aplicación. Para garantizar eso utilizamos el patrón de diseño **Singleton**<sup>2</sup> [25].

---

<sup>2</sup>El patrón de diseño **singleton** (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón singleton se implementa creando en nuestra clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado).

Como hemos dicho, esta base de conocimiento contendrá los patrones, las localizaciones y los objetos 3d con información. Pero estos objetos pueden ser de diversos tipos dependiendo de su formato. Por lo tanto, hay tres tipos de objetos que heredan de la clase abstracta *Object3D*.

Para crear estos objetos se utilizará una factoría simple (**Simple Factory**)<sup>3</sup>, que es otro patrón de diseño. En nuestro caso hemos optado por una factoría simple porque simplemente esta clase creará objetos de otras clases sin delegar en ninguna subclase. Tiene unos métodos estáticos de creación (*createObject*) con los que crea los objetos en función del tipo que sean.

En la figura 4.3, se pueden ver las clases por las que está formado el módulo de carga de datos, así como los aspectos explicados en este apartado.

---

<sup>3</sup>Muchas veces, cuando se habla de un patrón factoría nos referimos a los patrones *Factory Method* o *Abstract Factory*. Sin embargo, hay casos que no son cubiertos por estos patrones como por ejemplo, clases con métodos estáticos de fabricación o fábricas concretas que tienen implementación, pero sus métodos no son redefinibles. En estos casos, estamos ante una implementación del patrón **Simple Factory**.

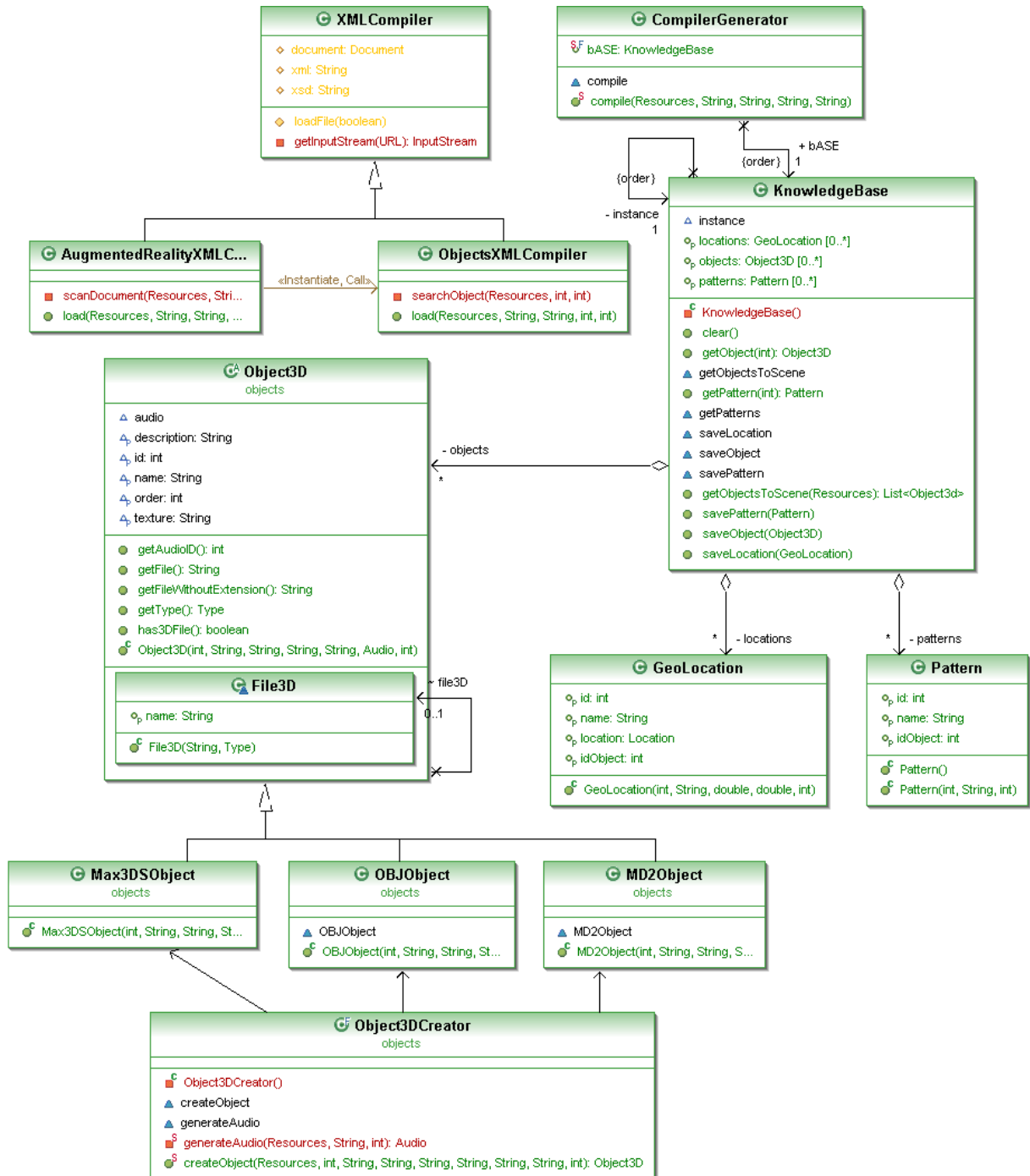


Figura 4.3: Diagrama de clases del Módulo Carga de Datos

### 4.2.2. Descripción del Módulo de Localización

Como habíamos descrito en la sección anterior 4.1.2, este módulo se encargará de obtener nuestra localización y comprobar si existe alguna información que deba ser mostrada en esa posición.

Para obtener nuestra posición utilizaremos el servicio *LOCATION\_SERVICE*. Como podemos ver en el siguiente código, la aplicación utiliza el servicio usando el mejor proveedor de localización disponible.

Se crea un *LocationListener* para recibir notificaciones del *LocationManager* cuando la localización cambie.

```
/**
 * Inicia el servicio de localización utilizando el mejor proveedor
 * , bien sea GPS o a través de la conexión a Internet.
 * La búsqueda se realiza cada 120.000 ms = 2 minutos de tiempo.
 */
private void startGPSLocation()
{
    String bestProvider;
    Criteria criteria = new Criteria();

    /* Use the LocationManager class to obtain GPS locations */
    LocationManager mlocManager = (LocationManager) getSystemService(
        Context.LOCATION_SERVICE);
    bestProvider = mlocManager.getBestProvider(criteria, false);

    LocationListener mlocListener = new ARLocationListener(this, 10);
    mlocManager.requestLocationUpdates( bestProvider, 120000, 0,
        mlocListener);
}
```

Código 4.1: TFGARActivity.java

Este *LocationListener* será una clase definida por nosotros, que será la encargada de comprobar, cuando nuestra posición cambie, si hay alguna información asociada a ese lugar.

En caso de que se descubra alguna información deberá ser representada y visualizada en el dispositivo.

Para notificar esta información utilizaremos el patrón de diseño **Observer**<sup>4</sup> [25]. Es sencillo de entender. El sujeto (*ARLocationListener*) tendrá un conjunto de observadores a los que notificará algún cambio cuando se produzca. Por tanto, nuestra interfaz de usuario será el observador al que enviaremos esta información.

---

<sup>4</sup>El patrón **Observador** (en inglés: **Observer**) también conocido como “spider” define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes.

El objetivo de este patrón es desacoplar la clase de los objetos clientes del objeto, aumentando la modularidad del lenguaje, así como evitar bucles de actualización (espera activa o polling).

Se ha construido un Observer con genericidad. Así, cualquier clase puede ser observador o sujeto implementando estas interfaces.

Se puede ver en la figura 4.4, así como las demás clases y las relaciones entre ellas.

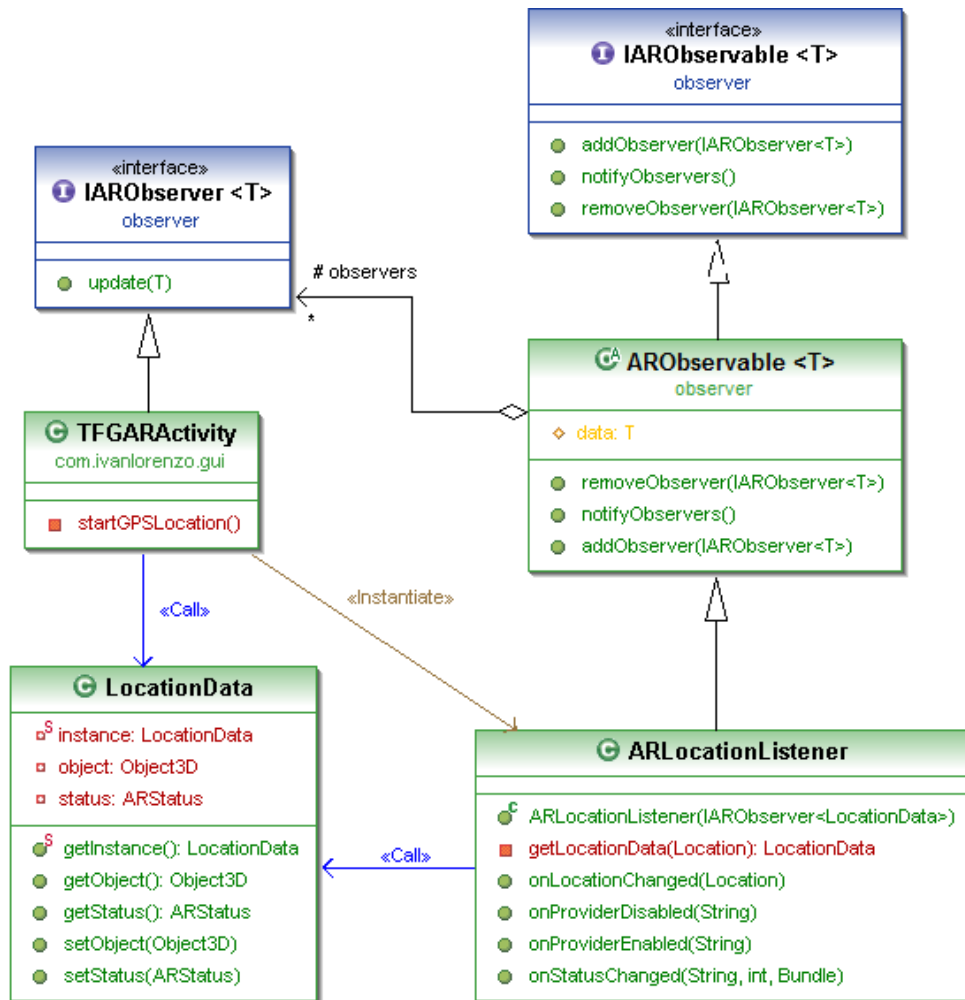


Figura 4.4: Diagrama de clases del Módulo de Localización

### 4.2.3. Descripción del Módulo de Renderización o Parser de objetos 3D

Tal y como habíamos descrito en la sección anterior 4.1.3, este módulo se encargará del proceso de renderización de objetos 3D.

Pero en nuestro sistema tenemos distintos formatos de ficheros y por tanto, la fase de renderización será diferente en función de qué tipo tenga el objeto a parsear.

Debido a esto, tendremos una jerarquía de parsers. Todos ellos derivan de la clase abstract *AParser* que a su vez implementa la interface *IParser*.

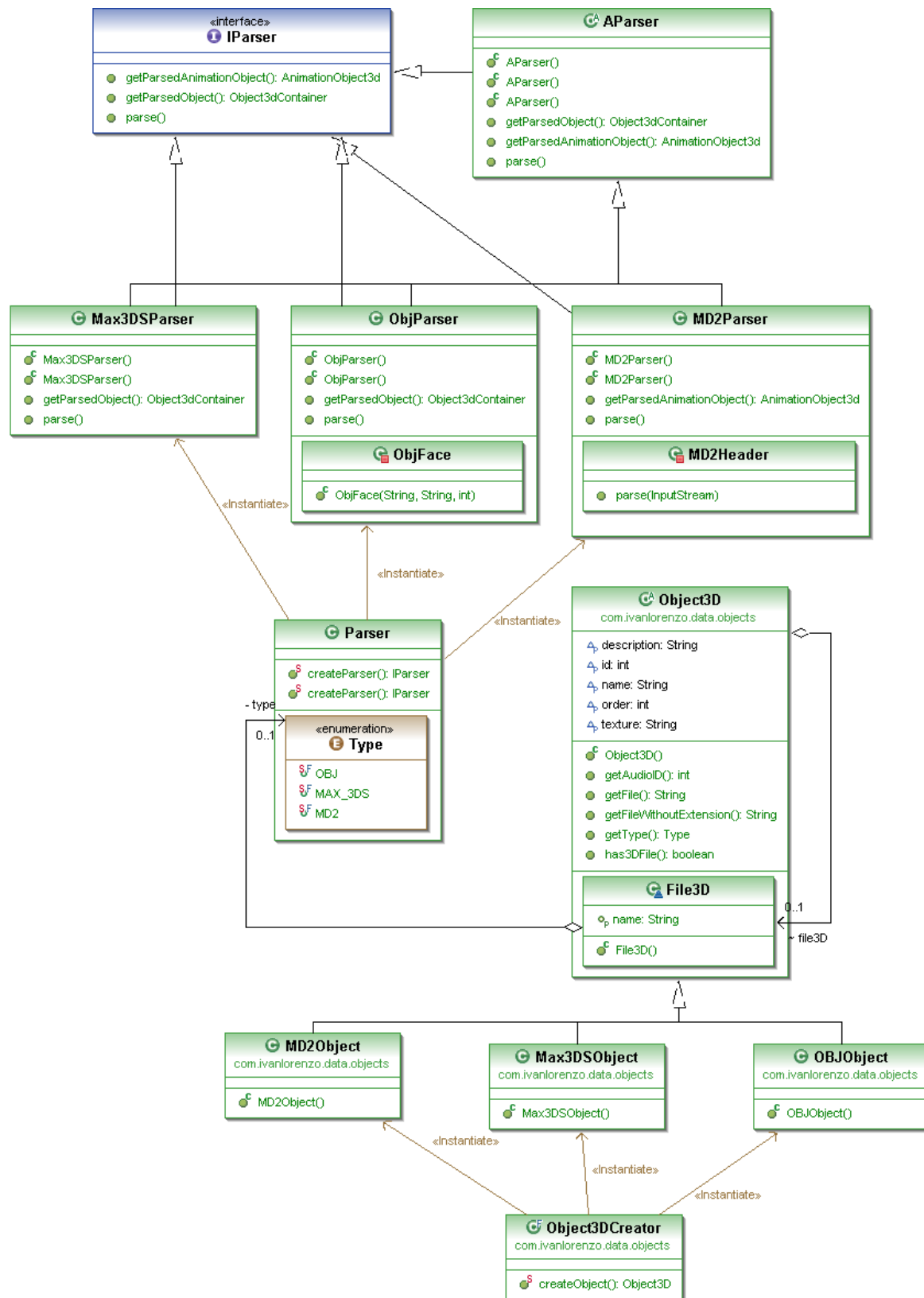


Figura 4.5: Diagrama de clases del módulo de Renderización o Parser de objetos 3D.

Cada uno de estos parsers realizará el renderizado de un tipo concreto. Para ello redefinirán el método *parse*. Los tipos que soporta el sistema estarán definidos en la enumeración *Type*

Para crear los parsers en función del tipo de archivos a procesar, usaremos un patrón de diseño factoría (**Simple Factory**). La clase *Parser* será esta factoría simple.

Todo el proceso de renderizado estará apoyado en la librería *min3d*[20] de Android.

Para utilizar *min3d* tendremos un objeto *Scene* que será algo así como un escenario en el que trabajamos y en el que iremos añadiendo los objetos a mostrar. Los parsers harán que el objeto se renderice y se convierta en un objeto procesable por esta escena.

También, como habíamos visto en el módulo de “Carga de datos” 4.2.1 hay una jerarquía de objetos que se construyen también con una factoría.

En la figura 4.5, se pueden ver las clases por las que está formado, así como las relaciones entre ellas.

### 4.2.4. Descripción del Módulo de Sonido

En el apartado 4.1.4 veíamos que este módulo se encargaba de decodificar y reproducir los ficheros de audio. Para ello utiliza la clase *MediaPlayer* [22] de Android.

Haremos que esta clase sea un sujeto observable, ya que cuando se produzca la finalización del audio de un objeto geolocalizado se deberá informar que termine la visualización de información.

Para este propósito utilizamos de nuevo el patrón **Observer** [25]. Así notificaremos a la clase encargada de representar los objetos (*ARToolkitDrawer*) cuando debemos finalizar su visualización.

Pero no podemos utilizar la clase *MediaPlayer* tal cual, sino que debemos adaptarla para que pueda implementar la interface *IARObservable*. Esto lo conseguiremos utilizando el patrón de diseño **Adapter**<sup>5</sup> [25].

Con este patrón conseguiremos justamente lo perseguido. Escribimos una nueva clase con una interfaz de programación deseada haciendo que se comunique con una clase cuya interface de programación es diferente.

Nos ofrece una gran oportunidad para la reutilización de código, permitiendo que interactúen dos o más objetos que supuestamente son incompatibles.

---

<sup>5</sup>El patrón **Adapter** (Adaptador) también conocido como *Wrapper* se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda.

## 4. Diseño

En la figura 4.6, se pueden ver las clases por las que está formado, así como las relaciones entre ellas.

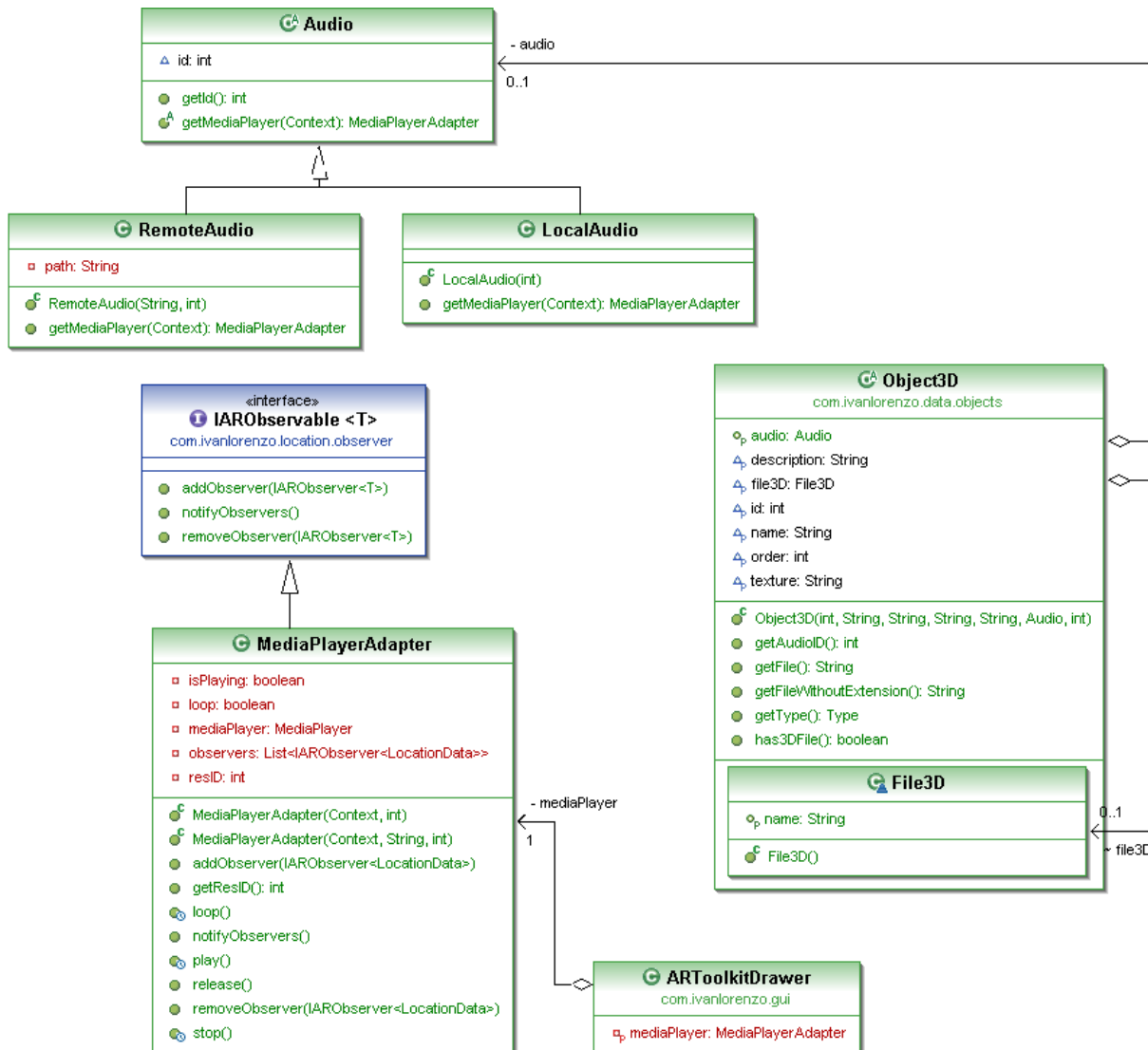


Figura 4.6: Diagrama de clases del módulo de Sonido.

### 4.2.5. Descripción del Módulo de Visualización de información

Como vimos en el apartado 4.1.5 este módulo es el encargado de visualizar y representar toda la información a través del dispositivo móvil.

En todas las aplicaciones Android, habrá unos componentes *Activity* que serán las ventanas con las que interactúa el usuario. En concreto, en nuestra aplicación habrá varias ventanas (o varias *Activity*), pero sólo una será la principal, que será la que primero se cargue.

Nosotros dispondremos de una actividad principal *TFGARActivity* que será la responsable de manejar la cámara e ir capturando las imágenes provenientes de ella.

Dispondremos de una clase *ARToolkitDrawer* que comprobará si existe algún marcador en las imágenes capturadas. En caso de ser así, pediremos a la base de conocimiento la información y la visualizará.

Esta clase tendrá un objeto *MediaPlayerAdapter*, que será el encargado de gestionar los ficheros de audio y además de informar cuando finalice el sonido para terminar la visualización de la imagen.

Para gestionar manipular los objetos debemos capturar los eventos producidos en la pantalla táctil de nuestro dispositivo móvil. Para ello redefinimos el método *boolean onTouchEvent(MotionEvent ev)*. Con esto conseguimos captar el movimiento de desplazamiento a través de la pantalla y el zoom a los objetos.

Para reconocer imágenes utilizamos la librería *NyARToolkit* [23].

Además, podemos observar como nuestra Activity principal implementa la interface *IARObserver* del patrón de diseño **Observer** [25].

En la figura 4.7, se pueden ver las clases por las que está formado este módulo, así como las relaciones entre ellas.

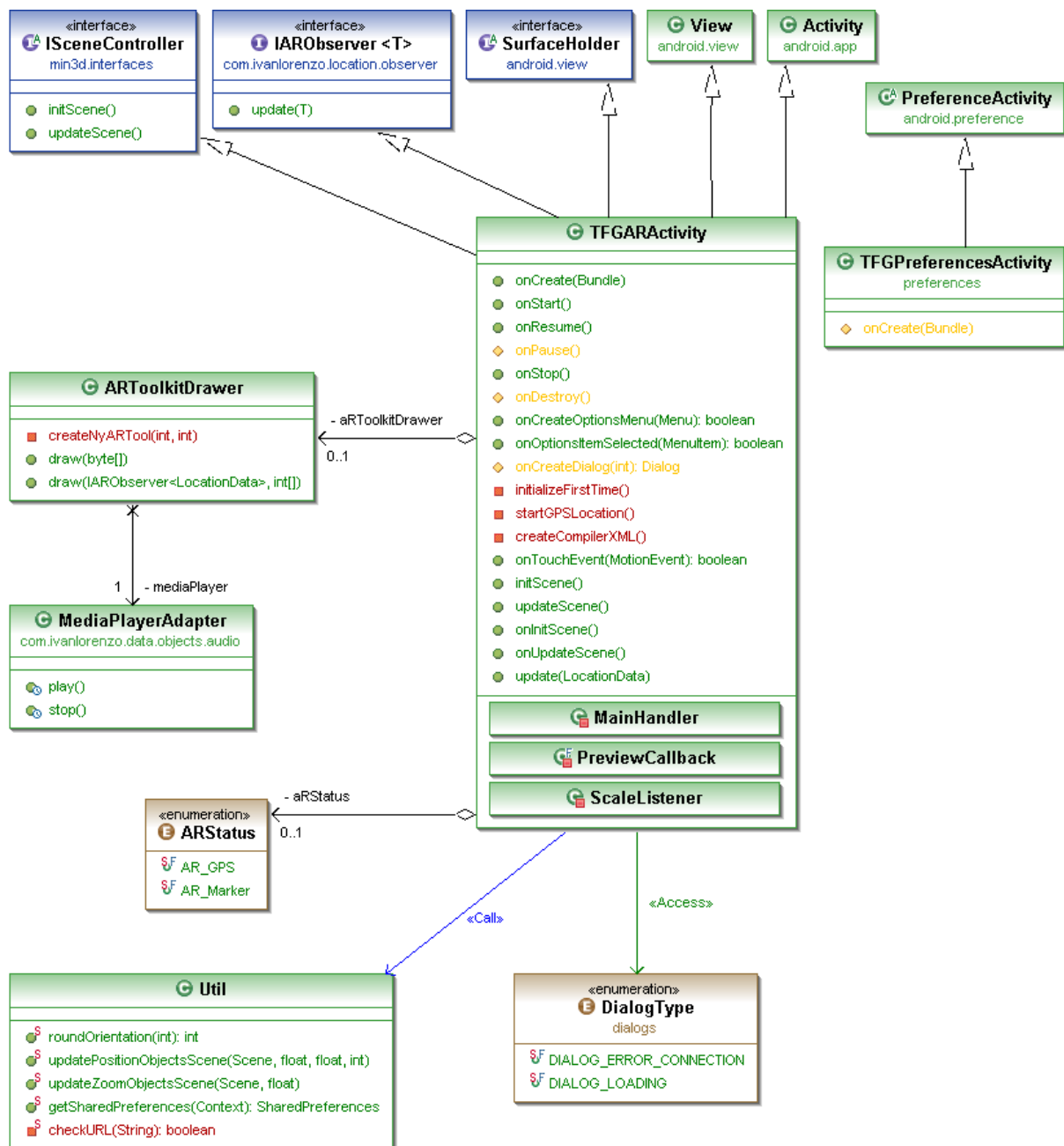


Figura 4.7: Diagrama de clases del Módulo Visualización de información.

Por último, como ya hemos indicado en la sección anterior podremos modificar las preferencias. Para ello utilizamos una clase *TFGPreferences* que hereda de *PreferenceActivity*.

Además, se definirán los componentes de la pantalla en un fichero *XML* que en nuestro caso hemos llamado *preferences.xml* y podemos ver a continuación:

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
    >

    <PreferenceCategory android:title="TFG Configuration">

        <EditTextPreference    android:dialogTitle="@string/patterns_url"
                                android:dialogMessage="@string/
                                    write_patterns_url"
                                android:key="URL_PATTERNS"
                                android:title="@string/patterns_and_locations"
                                android:summary="@string/patterns_url"
                                android:defaultValue="@string/URL_PATTERNS"/>

        <EditTextPreference    android:dialogTitle="@string/objects_url"
                                android:dialogMessage="@string/
                                    write_objects_url"
                                android:key="URL_OBJECTS"
                                android:title="@string/objects"
                                android:summary="@string/objects_url"
                                android:defaultValue="@string/URL_OBJECTS"/>

    </PreferenceCategory>
</PreferenceScreen>
```

Código 4.2: preferences.xml

### 4.3. Descripción del modelo dinámico

En esta sección nos dispondremos a realizar una descripción de cuál será el comportamiento general del sistema en aspectos dinámicos, es decir, predecir cuál será el comportamiento de éste en cada caso.

#### 4.3.1. Diagrama de actividad general de la aplicación

Como se muestra en la figura 4.8, en primer lugar ejecutaremos la aplicación en nuestro dispositivo móvil. Antes de continuar, el sistema examina si existe una cámara y si tenemos conexión a Internet. Si es así se podrá pasar al siguiente paso y sino finalizará la ejecución.

Una vez superado este proceso, se intentarán descargar los ficheros de configuración y sus datos asociados. Si no es posible, el sistema descargará los ficheros de configuración que tiene por defecto.

Después el sistema comenzará a capturar imágenes provenientes de la cámara y a obtener la posición geográfica. En el caso de que se detecte algún patrón de los que tenga cargados la plataforma o alguna de las localizaciones, se mostrará la información del objeto correspondiente a ese patrón.

Estando mostrando la información el usuario podrá manipular los objetos. Como ya habíamos dicho podrá ejecutar dos acciones:

- Mover un objeto en el espacio
- Escalarlo

En el diagrama de la figura 4.8 vemos que además podremos modificar las preferencias de la aplicación. Después de hacer estas modificaciones volveremos a recargar los datos de los ficheros de configuración que han sido cambiados.

Por último, también podremos finalizar la aplicación cuando lo deseemos.

Para una comprensión en profundidad, en la sección 4.3.2 de este mismo capítulo, se encuentran todos los diagramas de secuencia de las mismas.

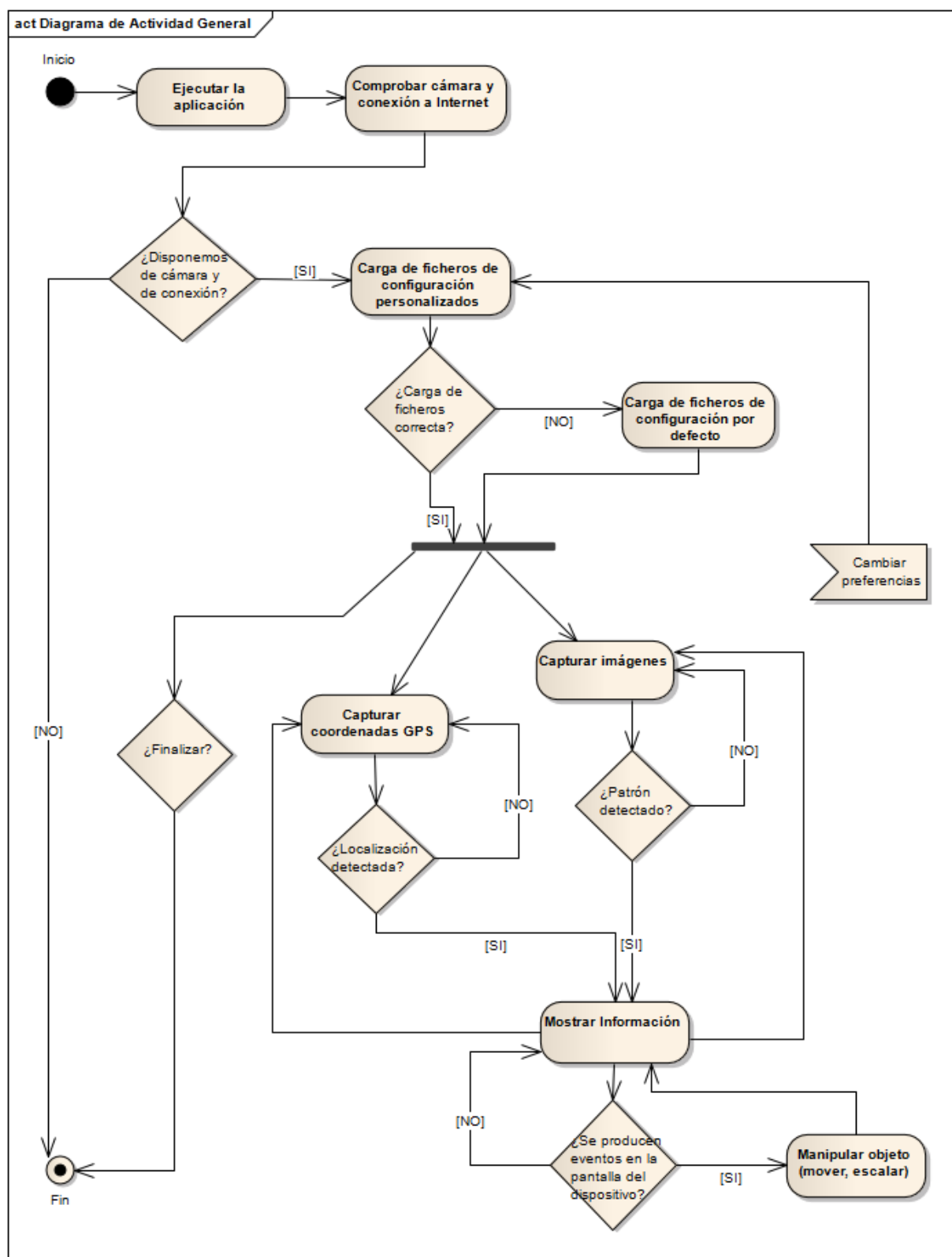


Figura 4.8: Diagrama de actividad general del sistema.

### 4.3.2. Diagramas de secuencia

En esta sección, se incluyen todos los diagramas de secuencia asociados a los casos de uso descritos en la sección 3.5 del análisis.

Para facilitar la localización y la lectura de los diagramas, éstos aparecerán por el mismo orden en el que fueron descritos en el documento de análisis.

#### Caso de uso 1.1: Arrancar Aplicación Cliente

El usuario ejecutará la aplicación del cliente en su dispositivo móvil. Para ello ejecutará su actividad principal que está definida en el *AndroidManifest.xml* de la aplicación.

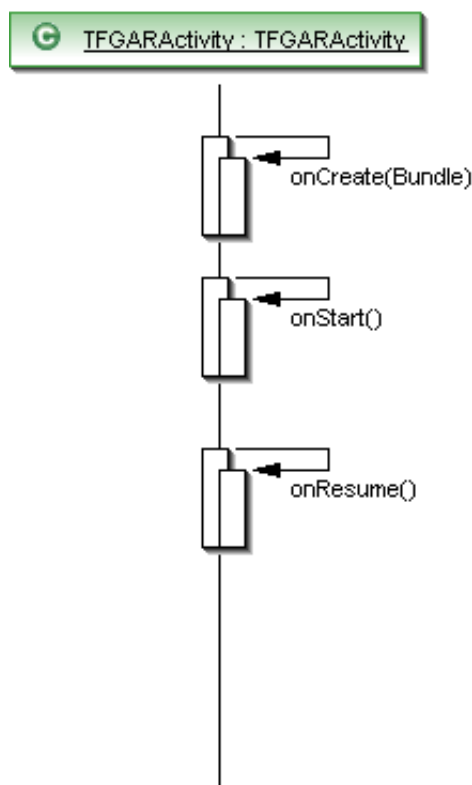


Figura 4.9: Diagrama de secuencia para el caso de uso 1.1 Arrancar Aplicación Cliente.

Vemos en la figura 4.9 que se ejecutan los métodos `onCreate`, `onStart` y `onResume` de la actividad.

Esto es debido a que nuestra clase deriva de la clase `Activity`. Una `Activity` tiene principalmente tres estados:

- **Activo:** La actividad se encuentra en primer plano e interactuando con el usuario.

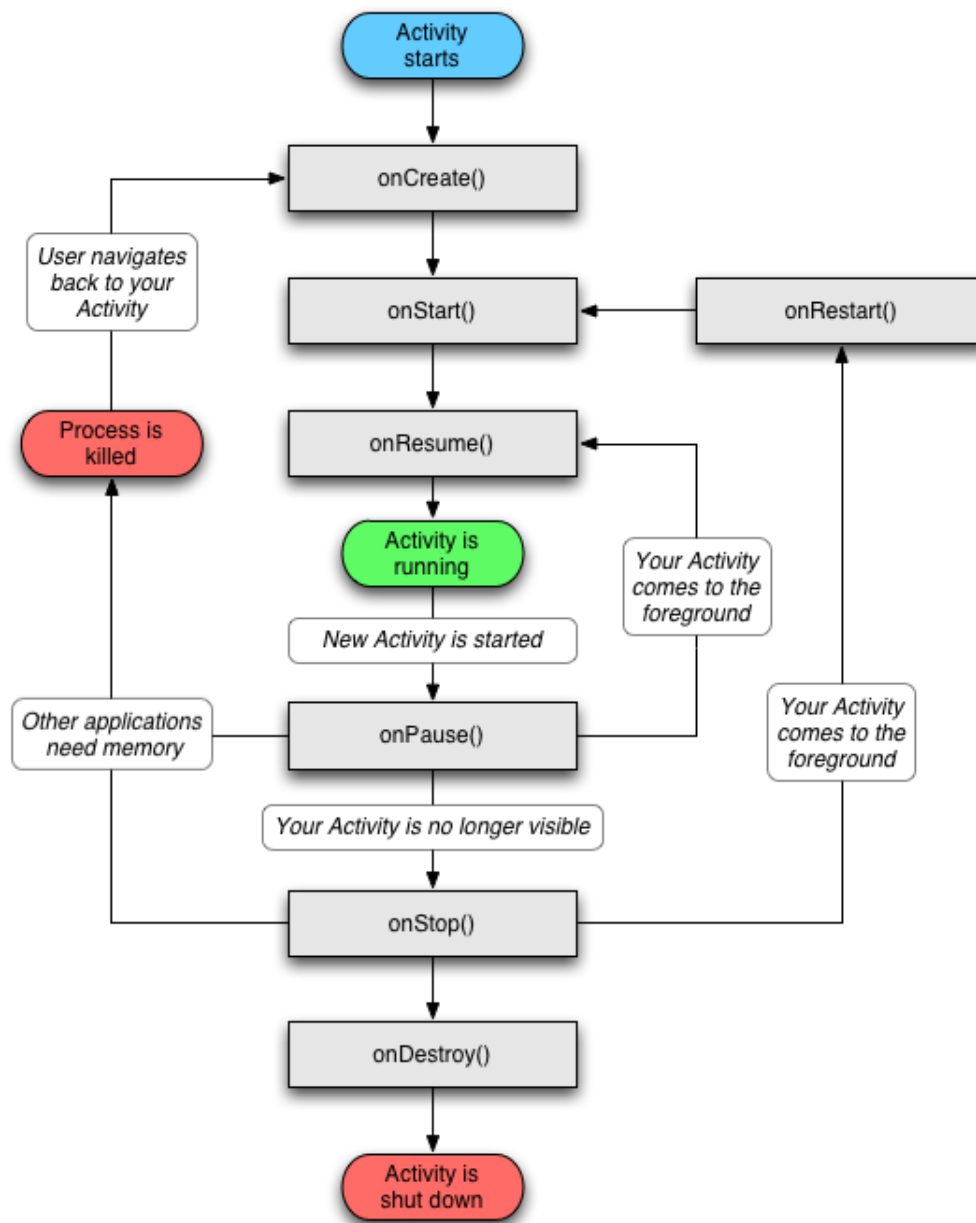


Figura 4.10: Ciclo de vida de una Activity en Android

- **Pausado:** La actividad sigue siendo visible para el usuario, pero ha perdido el foco. Por ejemplo que se haya mostrado un cuadro de diálogo delante de nuestra actividad. Debemos guardar el estado de la interfaz y los datos de esta actividad antes de entrar en este estado, ya que podríamos perderlos si el sistema necesita más recursos de memoria.
- **Parado:** La actividad no es visible para el usuario, queda a disposición del sistema para borrarla de la pila en caso de necesitar memoria.

## 4. Diseño

El ciclo de vida completo puede consultarse en la figura 4.10

### Caso de uso 1.2: Mostrar patrón y obtener información

A continuación se puede ver un diagrama de secuencia simplificado de la de obtención de imágenes y detección de markers, así como del posterior proceso de dibujo de los objetos.

Faltaría por mostrar el proceso de carga y obtención de los datos provenientes de los ficheros de configuración. Esto puede verse en la figura 4.16

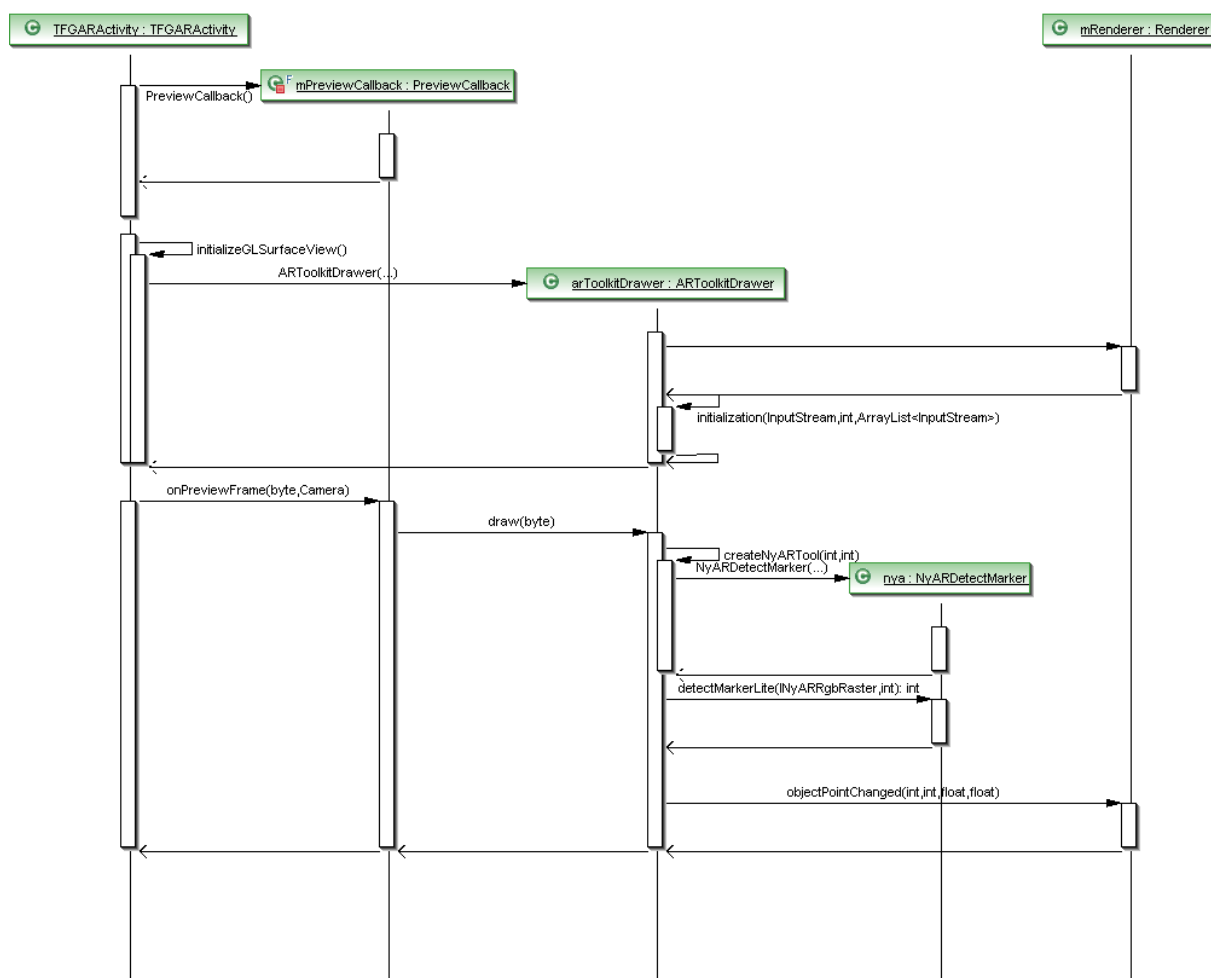


Figura 4.11: Diagrama de secuencia para el caso de uso 1.2 Mostrar patrón y obtener información.

### Caso de uso 1.3: Obtener geoposición e información asociada

En el siguiente diagrama de secuencia se obtiene la geoposición del dispositivo móvil y se comprueba si existe algún objeto alrededor a través del método *getLocationData*. En ese

caso se notifica a los observadores (*TFGARActivity*) la nueva posición y la información a mostrar a través del método *notifyObservers()* del patrón de diseño **Observer**.

A partir de estos datos, se podrá dibujar el objeto y mostrar la información al usuario.

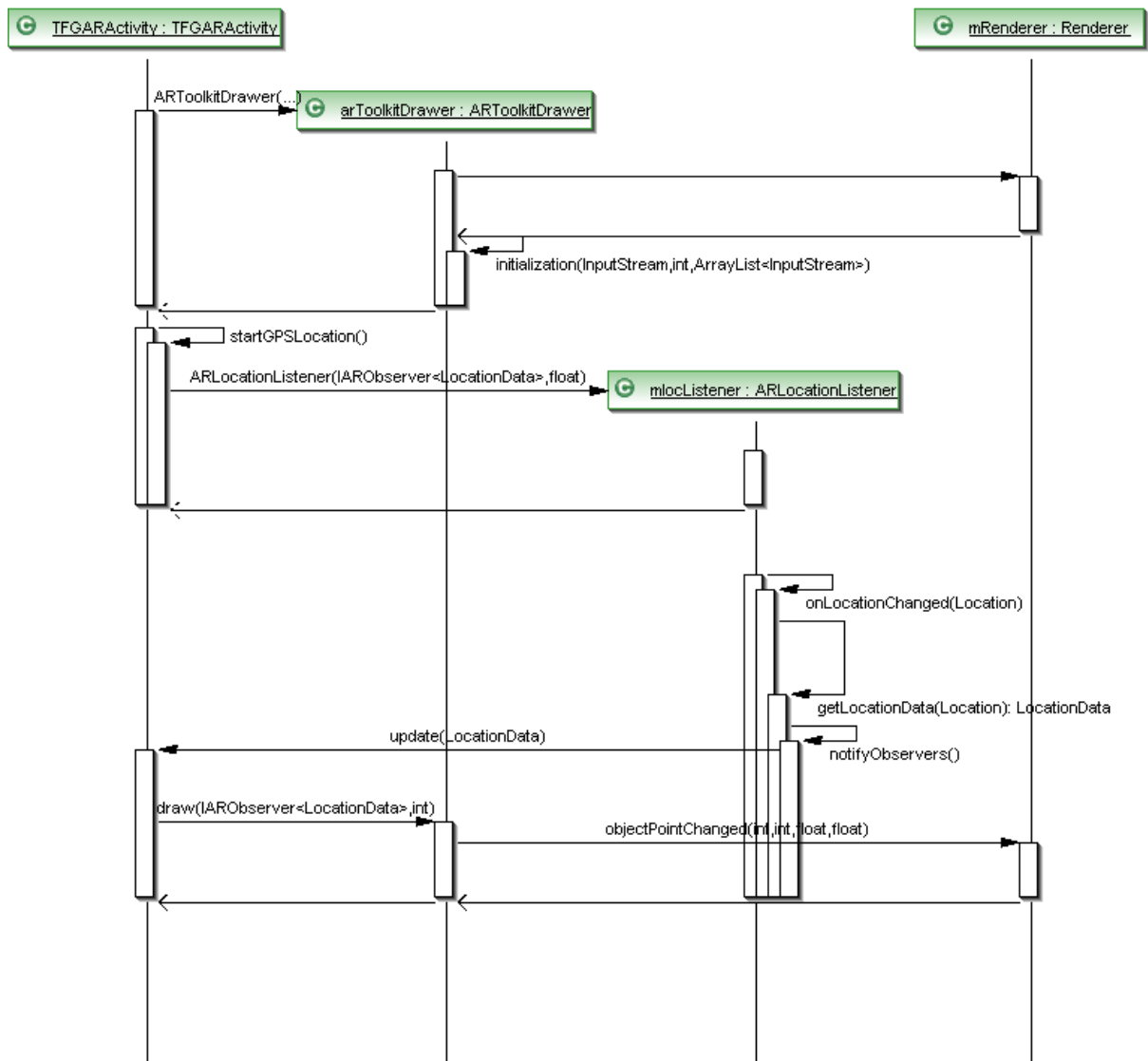


Figura 4.12: Diagrama de secuencia para el caso de uso 1.3 Obtener geoposición e información asociada.

#### Caso de uso 1.4: Desplazar Objeto

Vemos que la actividad principal de nuestra aplicación hereda de la clase *Activity*. Re-definimos, por tanto, el método *android.app.Activity.onTouchEvent(MotionEvent event)*. Este método se llama cuando se produce un evento en la pantalla táctil del dispositivo.

De este modo conseguimos capturar el desplazamiento que se realiza.

## 4. Diseño

Una vez obtenido el desplazamiento a aplicar a los objetos, debemos trasladarlo a todos los objetos de la escena. Para ello llamamos al método *updatePositionObjectsScene(Scene scene, float mPosX, float mPosY, int mLastOrientation)* de la clase Util.

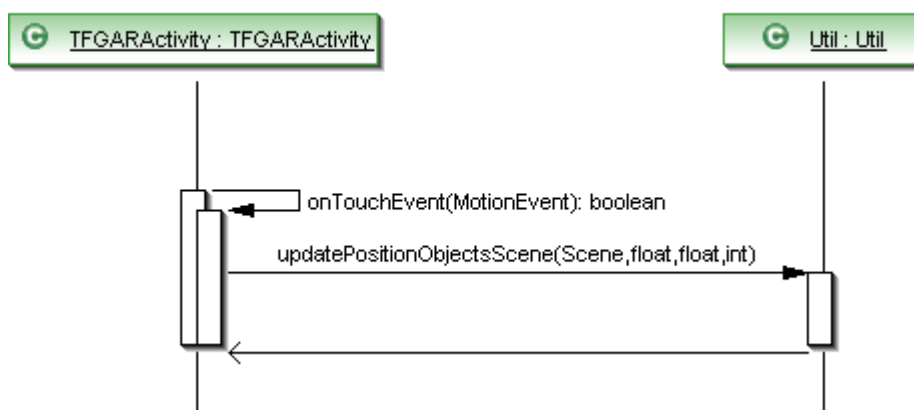


Figura 4.13: Diagrama de secuencia para el caso de uso 1.4 Desplazar Objeto.

### Caso de uso 1.5: Escalar Objeto

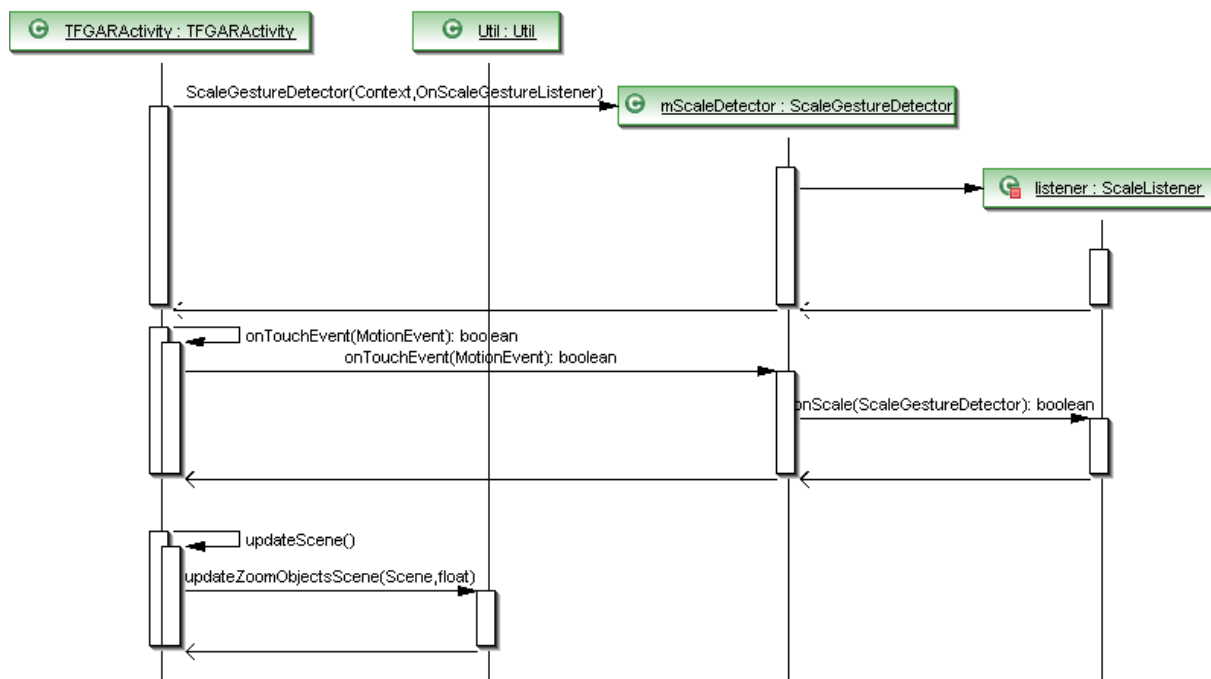


Figura 4.14: Diagrama de secuencia para el caso de uso 1.5 Escalar Objeto.

Al igual que en el caso de uso anterior, se ha redefinido el método *onTouchEvent(MotionEvent event)* de la actividad principal para capturar eventos de la pantalla. Además, creamos un objeto de la clase *ScaleGestureDetector* para capturar el gesto de escalado.

Con todo ello conseguimos cambiar la escala de los objetos.

Una vez obtenido la nueva escala a aplicar, debemos trasladarlo a todos los objetos de la escena. Para ello llamamos al método *updateZoomObjectsScene(Scene scene, float mScaleFactor)* de la clase *Util*.

#### Casos de uso 1.6 y 1.7: Configuración de patrones y posiciones y Configuración de objetos e información

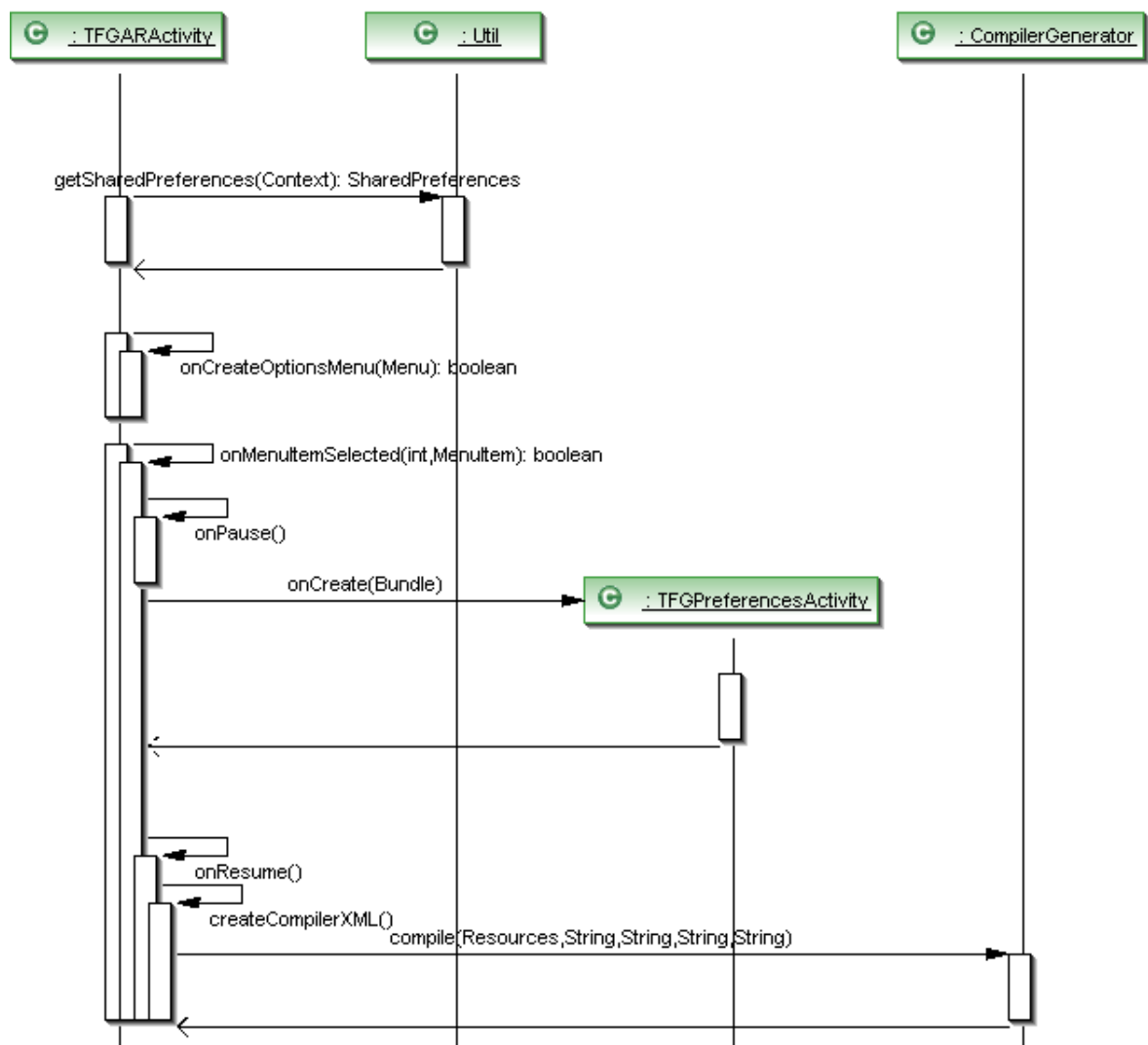


Figura 4.15: Diagrama de secuencia para los casos de uso 1.6 y 1.7.

## 4. Diseño

Como vemos en el diagrama de secuencia 4.15 se obtienen las preferencias del usuario al iniciar la aplicación, así como el menú.

Entonces el usuario selecciona el ítem del menú correspondiente a la configuración de las preferencias y se crea la actividad que las gestiona a la vez que se pausa la actividad principal.

Una vez modificadas las preferencias, se vuelve a reanudar la ejecución de la actividad principal y se recargan los datos de los nuevos ficheros de configuración.

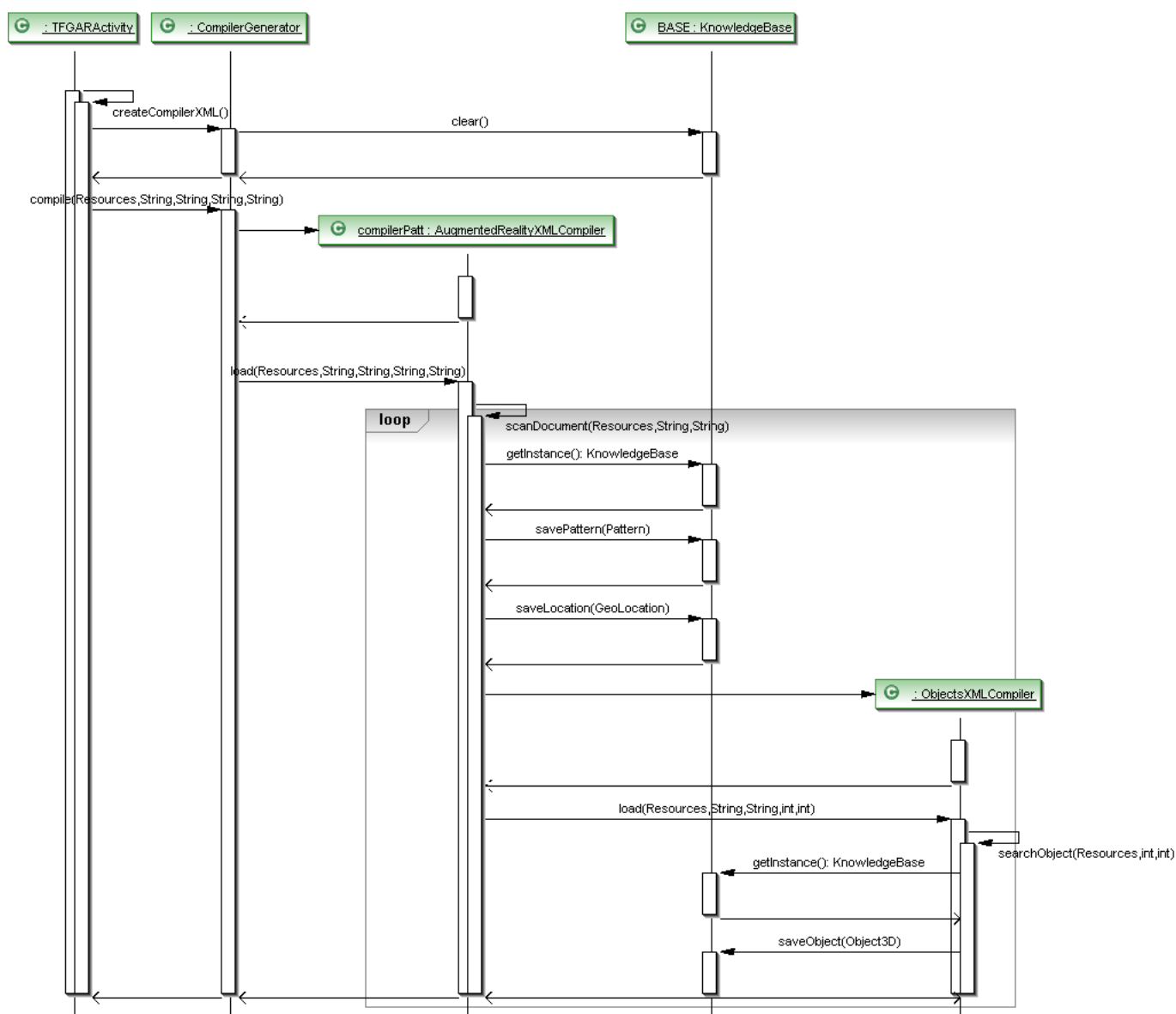


Figura 4.16: Diagrama de secuencia de la carga de datos.

Este proceso de carga de datos se detalla en el diagrama de secuencia de la figura 4.16.

En primer lugar se borran todos los datos de la base de conocimiento y se realiza el procesamiento de los ficheros *XML*.

Se crea un objeto de la clase *AugmentedRealityXMLCompiler* y se van leyendo los patrones y las localizaciones de éste para así guardarlos en la base de conocimiento.

Se crea un objeto de la clase *ObjectsXMLCompiler* y para cada una de ellas, se lee del fichero de objetos el objeto y se guarda en la base de conocimiento.

Así tendremos almacenados todos los datos necesarios en la aplicación: patrones, localizaciones y objetos.

#### Caso de uso 1.8: Salir Aplicación Cliente

El usuario decidirá salir a través del menú de la aplicación. Cuando acciona este ítem, se destruirá la actividad y finalizará la ejecución del programa.

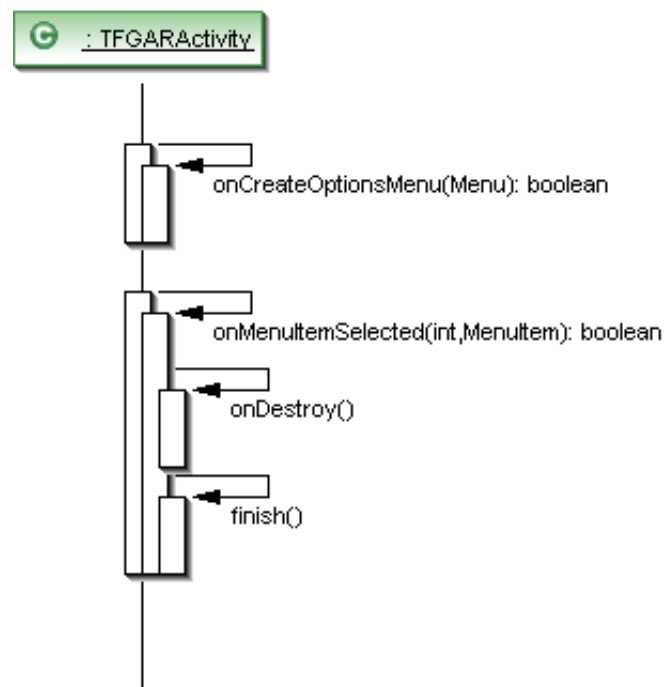


Figura 4.17: Diagrama de secuencia para el caso de uso 1.8 Salir Aplicación Cliente.

### 4.4. Validación de los requisitos

En las tablas 4.1, 4.2, 4.3, 4.4, 4.5 y 4.6 se muestran los puntos del diseño en los que son satisfechos cada uno de los requisitos documentados en el análisis.

#### 4.4.1. Requisitos comunes de los interfaces

| Requisito    | Descripción                                                                | Cumplido en                              |
|--------------|----------------------------------------------------------------------------|------------------------------------------|
| RC-IN-U. 1   | Interfaz accesible e intuitiva                                             | 4.1.5                                    |
| RC-IN-U. 2   | La interfaz será táctil                                                    | 4.1.5                                    |
| RC-IN-HW. 1  | Será necesario el uso de la cámara del dispositivo móvil                   | 4.1.5                                    |
| RC-IN-HW. 2  | Será necesario disponer de acceso a internet desde el dispositivo móvil    | 4.1.1                                    |
| RC-IN-HW. 3  | Será necesario el uso del GPS del dispositivo móvil                        | 4.1.2                                    |
| RC-IN-SW. 1  | Reconocimiento de patrones                                                 | 4.1.5 y en el diagrama de secuencia 4.11 |
| RC-IN-COM. 1 | Se utilizará el protocolo TCP/IP para comunicar los clientes y el servidor | 4.1.1                                    |

Cuadro 4.1: Validación de los requisitos comunes de los interfaces

#### 4.4.2. Requisitos funcionales generales

| Requisito | Descripción                                          | Cumplido en   |
|-----------|------------------------------------------------------|---------------|
| RF 1      | Sistema enfocado al ámbito educativo                 | 4.1.1 y 4.1.5 |
| RF 2      | Independencia entre la plataforma y las aplicaciones | 4.1.1         |
| RF 3      | Sistema flexible                                     | 4.1.1         |

Cuadro 4.2: Validación de los requisitos funcionales generales

### 4.4.3. Requisitos funcionales de la aplicación para dispositivos móviles

| Requisito | Descripción                                                                                                                    | Cumplido en                                       |
|-----------|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| RF 4      | Reconocimiento de patrones y visualización de objetos                                                                          | 4.1.3, 4.1.5 y en el diagrama de secuencia 4.11   |
| RF 5      | Geoposicionar información                                                                                                      | 4.1.2, 4.1.5 y en el diagrama de secuencia 4.12   |
| RF 6      | Se podrá asignar a cada patrón un determinado objeto, así como su textura y su descripción sonora mediante un fichero de audio | 4.1.1 y 4.1.3                                     |
| RF 7      | El audio explicativo de un objeto debe detenerse al reproducirse otro posterior                                                | 4.1.4                                             |
| RF 8      | Desde el propio dispositivo se puede manipular el objeto                                                                       | 4.1.5 y en los diagramas de secuencia 4.13 y 4.14 |

Cuadro 4.3: Validación de los requisitos funcionales de la aplicación para dispositivos móviles

### 4.4.4. Requisitos funcionales del servidor web

| Requisito | Descripción            | Cumplido en |
|-----------|------------------------|-------------|
| RF 9      | Sitio web público      | 4.1.1       |
| RF 10     | Configuración de rutas | 4.1.1       |

Cuadro 4.4: Validación de los requisitos funcionales del servidor web

### 4.4.5. Requisitos funcionales de los ficheros de configuración

| Requisito | Descripción                                                              | Cumplido en |
|-----------|--------------------------------------------------------------------------|-------------|
| RF 11     | Deberán almacenar la definición de las exposiciones                      | 4.1.1       |
| RF 12     | Los ficheros de configuración utilizarán un lenguaje de marcado          | 4.1.1       |
| RF 13     | Ficheros de configuración divididos según el tipo de aspecto que abarcan | 4.1.1       |
| RF 14     | Para cada exposición habrá un conjunto de ficheros                       | 4.1.1       |

Cuadro 4.5: Validación de los requisitos funcionales de los ficheros de configuración

### 4.4.6. Requisitos no funcionales

| Requisito | Descripción                                                                            | Cumplido en                              |
|-----------|----------------------------------------------------------------------------------------|------------------------------------------|
| RNF 1     | Número de dispositivos conectados                                                      | 4.1.1                                    |
| RNF 2     | Aplicación móvil para Android                                                          | En todo el sistema                       |
| RNF 3     | Las preferencias del usuario se almacenarán en su propio dispositivo                   | 4.1.5 y en el diagrama de secuencia 4.15 |
| RNF 4     | Ficheros de log                                                                        | 4.1.1                                    |
| RNF 5     | Fiabilidad en los datos GPS                                                            | 4.1.2                                    |
| RNF 6     | Servidor disponible                                                                    | 4.1.1                                    |
| RNF 7     | El sistema reconocerá que exista una conexión de datos activa al iniciar la aplicación | 4.1.1 y 4.1.5                            |
| RNF 8     | Configuración externa                                                                  | 4.1.1                                    |
| RNF 9     | Nuevas versiones del producto                                                          | El proceso de instalación                |
| RNF 10    | Portabilidad de la aplicación móvil                                                    | El proceso de instalación                |
| RNF OR 1  | Cumplimiento de la LOPD                                                                | 4.1.1 y 4.1.5                            |

Cuadro 4.6: Validación de los requisitos no funcionales

# Capítulo 5

## Modelo de datos

Llegados a este punto, es el momento de decidir de qué forma serán almacenados los datos que manejará la aplicación.

En primer lugar habrá que decidir de qué forma se va a guardar la información, es decir, decidir qué modelo seguirá. En este caso nos decidiremos por el modelo relacional ya que la información no se encuentra aislada entre sí y nos permite una mayor flexibilidad a la hora de manipularla.

### 5.1. Diseño conceptual: Modelo Entidad-Relación

Peter Chen publicó en 1976 el modelo entidad relación, el cual tuvo gran aceptación principalmente por su expresividad gráfica. Sobre esta primera versión han trabajado numerosos autores, generando distintas extensiones de mayor o menor utilidad de aceptación variable en el medio académico profesional.

Con el Modelo E/R se establece una visión global de los datos del sistema de información, en un nivel de abstracción próxima al usuario e independiente de las características físicas del equipo donde se vaya a instrumentar el sistema.

Consiste en describir la información del sistema mediante la definición de Entidades, atributos y asociaciones o interrelaciones entre ellas.

Teniendo en cuenta y tras analizar los requisitos anteriores, pasaremos a definir el modelo entidad-relación que se corresponde con nuestro sistema.

En esta plataforma no se usa ninguna base de datos para almacenar la información, simplemente se usan ficheros *XML* donde se guarda la misma. Estos ficheros *XML* van a ir intercambiándose entre el servidor y los dispositivos móviles de los clientes cuando los usuarios decidan obtener información dinámica.

### 5.1.1. Esquema Entidad-Relación

Ver figura 5.1 en la página 108

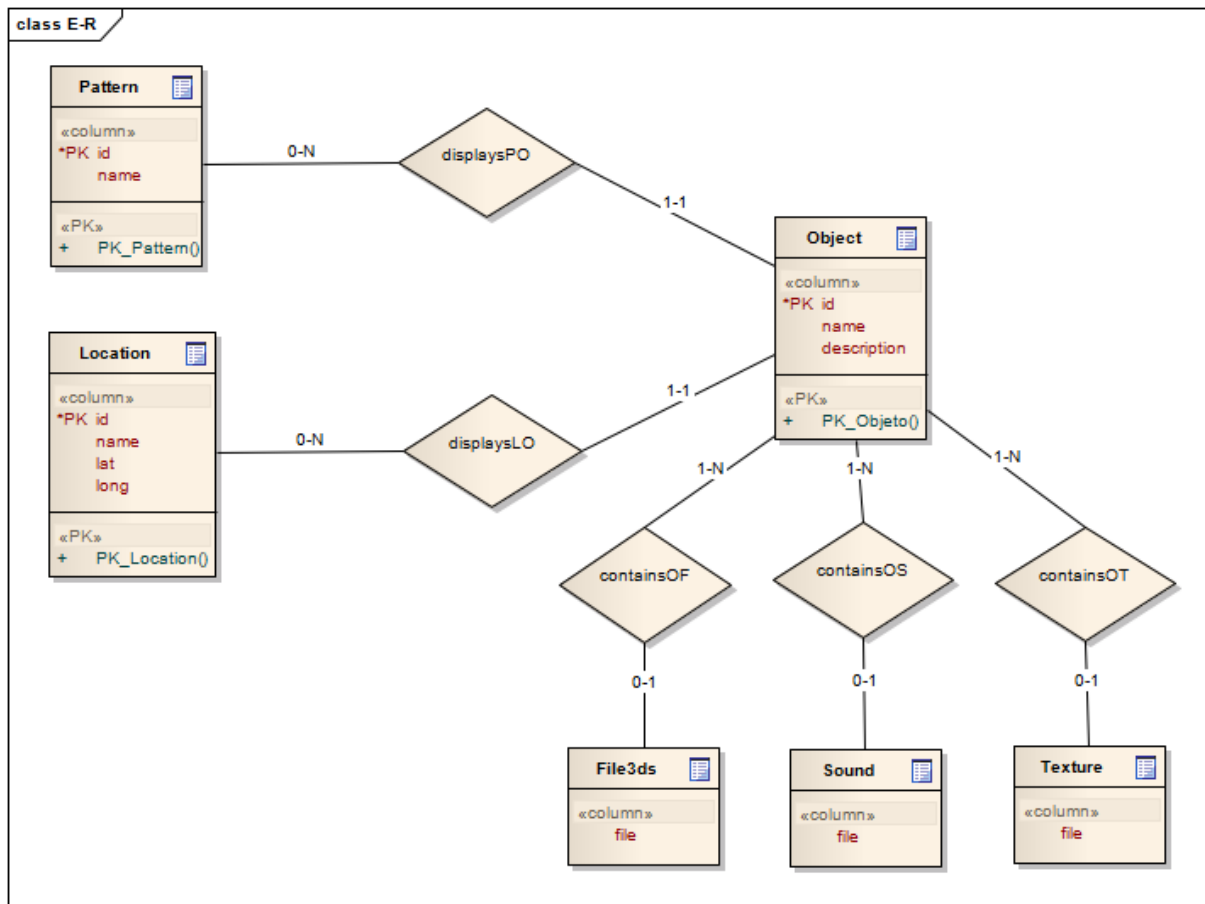


Figura 5.1: Diagrama entidad-relación del modelo de datos.

### 5.1.2. Entidades

A continuación se describen cada una de las entidades y se muestran sus atributos.

#### ■ Entidad *Pattern*

- **Descripción:** Esta entidad representa cada uno de los patrones que el sistema es capaz de detectar y transformar.
- **Atributos:** id, name

#### ■ Entidad *Location*

- **Descripción:**
- **Atributos:** id, name, lat, long

### ■ Entidad *File3D*

- **Descripción:** Esta entidad representa el fichero “3ds” que será el objeto que se mostrará.
- **Atributos:** file

### ■ Entidad *Sound*

- **Descripción:** Esta entidad representa el archivo de sonido asociado a cada uno de los objetos.
- **Atributos:** file

### ■ Entidad *Texture*

- **Descripción:** Esta entidad sirve para representar la textura asociada a cada objeto del museo.
- **Atributos:** file

### ■ Entidad *Object*

- **Descripción:** Esta entidad representa los objetos del modelo de navegación, que forman cada una de las visitas. Estos objetos serán el resultado de la transformación de los patrones.
- **Atributos:** id, name, description

## 5.1.3. Relaciones

A continuación se describen cada una de las relaciones que hay entre las entidades que forman los archivos *XML* que la aplicación utiliza para guardar la información.

### ■ Relación *displaysPO*

- **Entidades:** Pattern(0,n) - Object(1,1)

### ■ Relación *displaysLO*

- **Entidades:** Location(0,n) - Object(1,1)

### ■ Relación *containsOF*

- **Entidades:** Object(1,n) - File3ds(0,1)

### ■ Relación *containsOS*

- **Entidades:** Object(1,n) - Sound(0,1)

### ■ Relación *containsOT*

- **Entidades:** Object(1,n) - Texture(0,1)



# Capítulo 6

## Pruebas software

Las pruebas son una parte esencial del desarrollo software. El software que se desarrolla siempre va a contener errores, errores que en la fase de pruebas, debemos detectar y solucionar por completo.

A lo largo del ciclo de vida del proyecto se han ido desarrollando una serie de pruebas que se completaban al finalizar cada versión con una batería de pruebas de usabilidad.

Vamos a plantear los siguientes niveles de pruebas:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas del sistema.
- Pruebas de implantación.
- Pruebas de aceptación (usabilidad).

Además de estas pruebas, también se realizarán las “Pruebas de rendimiento”.

### 6.1. Pruebas unitarias

#### 6.1.1. Definición

“Las pruebas unitarias comprenden las verificaciones asociadas a cada componente del sistema de información. Su realización tiene como objetivo verificar la funcionalidad y estructura de cada componente individualmente”.

Constituyen un enfoque inicial de las pruebas y sirven como base a las siguientes pruebas.

Existen dos enfoques distintos de estas pruebas:

- Enfoque estructural o de caja blanca: se verifica la estructura interna del componente con independencia de la funcionalidad que debe ofrecer el mismo. Se puede comprobar los caminos de la aplicación, el código no usado, ...
- Enfoque funcional o de caja negra: se comprueba que cada componente de la aplicación cumple el papel para el que fue diseñado comprobando las entradas y las salidas obtenidas del programa. Se diseñarán tres tipos de datos de pruebas: tipo A, B y C. La primera consiste en la introducción de datos correctos a la aplicación. La segunda utiliza datos en el límite de los rangos permitidos. La tercera suministra datos no válidos a la aplicación.

### 6.1.2. Objetivos

Se trata de intentar asegurar que la implementación de una clase como ente individual es correcta. Evidentemente asumimos que hay fallos que se detectarán en fases de prueba posteriores pero es prioritario en todo desarrollo la detección de errores más temprana posible así como, también en la medida de lo posible y en los mismos términos, su solución.

### 6.1.3. Planificación

Se realizará una prueba unidad de las clases que integran el proyecto. Para esto utilizamos **JUnit 4**<sup>1</sup> [26] y **JUnit para Android** (Android Test Project) [27].

Mediante la realización de las pruebas unitarias, se detectaron pequeños errores en las unidades de software analizadas, generalmente debidos a despistes, aunque posteriormente hubo que realizar modificaciones en métodos probados por no cumplir con la funcionalidad que se le requería a la hora de integrarlas en un conjunto mayor.

Las pruebas unitarias realizadas se pueden consultar en la sección C

## 6.2. Pruebas de integración

### 6.2.1. Definición

“Las pruebas de integración comprenden verificaciones asociadas a grupos de componentes, generalmente reflejados en la definición de subsistemas de construcción o en el plan de integración del sistema de información y tienen como objetivo verificar el correcto ensamblaje entre los distintos componentes.”

---

<sup>1</sup>**JUnit** es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera.

En las pruebas de integración se examinan las interfaces entre los grupos de componentes o subsistemas. Los tipos fundamentales de integración son los siguientes:

- Integración incremental: se va combinando el componente que se desea probar con los componentes previamente probados. Se consigue así acotar los errores, ya que estos se producirán en la interfaz entre los módulos anteriores y el módulo que estamos probando. Los componentes se pueden combinar siguiendo tres técnicas:
  - De arriba a abajo (top down): se prueban desde el más genérico hasta el más específico.
  - De abajo a arriba (bottom-up): se prueban desde el más específico hasta el más genérico.
  - Estrategias combinadas: se mezclan las dos técnicas anteriores.
- Integración no incremental: se prueba cada componente por separado (pruebas unitarias), y posteriormente se ensambla todo el sistema realizando las pruebas pertinentes. Este tipo de integración se denomina Big-Bang.

La técnica aprovecha la estructura arborescente de la distribución de módulos (paquetes) del sistema para introducir el concepto en sendas modalidades de recorridos en anchura y en profundidad.

### 6.2.2. Objetivos

Se pretende comprobar la correcta colaboración establecida entre unas clases y otras así como su correcto funcionamiento. Se comprobará el comportamiento que tiene el tándem que conforman dos o más clases acopladas entre sí y contrastarlo con el que se espera de esta integración.

### 6.2.3. Planificación

Como se describió de forma exhaustiva en el diseño, la aplicación presenta una estructura modular. Estos módulos en general actúan de manera independiente del resto, y por lo tanto, el correcto funcionamiento de los servicios que cada uno de ellos brinda, se puede garantizar mediante este tipo de pruebas.

Para la realización de estas pruebas, se ejecutaron todas y cada una de las operaciones que permiten hacer cada uno de los módulos de forma independiente y el resultado de las mismas fue el que sigue:

#### Módulo Carga de Datos.

##### Prueba 1. Intentar arrancar la aplicación

- **Situación esperada:** Se espera que la aplicación detecte la presencia de una cámara y conexión a Internet en el dispositivo móvil. En caso de que se reconozca la cámara y haya conexión la aplicación comenzará. En caso contrario, se mostraría un mensaje al usuario advirtiéndole de la situación.

- **Situación obtenida:** Cuando activamos bien los datos GPRS o la Wifi la aplicación arranca y continua su ejecución. Si no es así, se muestra un mensaje de error advirtiéndonos de ello.
- **Conclusión:** En este sentido la comprobación de la conexión funciona correctamente.

### Prueba 2. Cargar ficheros de configuración

- **Situación esperada:** Al intentar cargar un fichero *XML*, se comprobará que éste sea un documento **bien formado**, es decir, que siga las reglas básicas de sintaxis para *XML*. También se deberá comprobar que sea un documento ***XML* válido**. Para ello cada tipo de fichero *XML* tiene su propio esquema definido en un fichero *XSD*. En caso que no se cumpla alguna de estos requisitos se mostrará un mensaje de alerta o error al usuario. Al cargar cada uno de los datos se irán almacenando en la *Base de Conocimiento* de la plataforma
- **Situación obtenida:** La lectura y carga de los ficheros XML es correcta. Pero detectamos que los datos almacenados en la *Base de Conocimiento* no son correctos.
- **Conclusión:** Podemos concluir que el método encargado de almacenar los objetos en la *Base de Conocimiento* estaba fallando debido a que la lectura del fichero *XML* no estaba siendo adecuada y estaba almacenando datos erróneos. Este problema se solventó sin mayores dificultades. Solucionado este inconveniente el módulo funciona correctamente.

## Módulo de Localización

### Prueba 1. Obtención de localización geográfica

- **Situación esperada:** Se deberá obtener la posición mediante algún mecanismo de localización disponible en el dispositivo móvil.
- **Situación obtenida:** Probando este punto, descubrimos que si se desactiva el GPS no obtenemos la localización.
- **Conclusión:** Para ello se modifica el módulo para que obtenga la posición buscando el mecanismo de localización mejor, es decir, aquel que nos ofrezca mayor precisión.

### Prueba 2. Detección de información en una localización concreta

- **Situación esperada:** A medida que se capturan datos acerca de nuestra posición, deberemos comprobar si existen objetos informativos cercanos a ella.
- **Situación obtenida:** Si nos desplazamos unos metros la información no es mostrada, debido a que la distancia de nuestra posición respecto al punto marcado era muy pequeña.
- **Conclusión:** Para ello se ha modificado el módulo para que detecte objetos en un radio alrededor de nuestra localización mayor.

### Prueba 3. Mecanismo de aviso de detección de un objeto informativo al módulo de visualización

- **Situación esperada:** Al detectar un objeto en una posición se debe informar al módulo de visualización.
- **Situación obtenida:** Se detecta el objeto y se envía correctamente para su posterior visualización.
- **Conclusión:** Esta prueba ha sido pasada y, por tanto, no es necesaria ninguna modificación.

### Módulo de Renderización o Parser de objetos 3D

#### Prueba 1. Renderizar un fichero .3ds

- **Situación esperada:** Se deberá renderizar el fichero 3D y sus texturas asociadas. Para eso se debe ir leyendo el fichero e interpretándolo.
- **Situación obtenida:** Se renderiza el objeto pero no es posible obtener las texturas remotas del objeto.
- **Conclusión:** Se modifica el parser específico para que obtenga la textura en la misma localización que está el fichero 3D.

#### Prueba 2. Renderizar un fichero .obj

- **Situación esperada:** Se deberá renderizar el fichero 3D y sus texturas.
- **Situación obtenida:** Se realiza la renderización de los objetos probados.
- **Conclusión:** Esta prueba ha sido pasada y, por tanto, no es necesaria ninguna modificación.

#### Prueba 3. Renderizar un fichero .md2

- **Situación esperada:** Se deberá renderizar el fichero 3D y sus texturas asociadas. Para eso se debe ir leyendo el fichero e interpretándolo.
- **Situación obtenida:** De nuevo aquí se renderiza el objeto sin obtener las texturas remotas.
- **Conclusión:** Se modifica el parser específico para que obtenga la textura en la misma localización que está el fichero 3D.

### Módulo de Sonido

#### Prueba 1. Comprobar que se reproduzca un fichero de audio al detectar un patrón

- **Situación esperada:** Al mostrarse la información de un objeto que tenga asociado un archivo de audio con información, deberá reproducirse.
- **Situación obtenida:** Al detectarse el patrón el sonido comienza a reproducirse. El problema es que si el patrón deja de detectarse durante un instante y luego vuelve a capturarlo, el audio se iniciará desde el principio.
- **Conclusión:** Se modifica el módulo de sonido para que si se deja de capturar el patrón, el sonido de éste se pause y no se pare. Por tanto, al volver a detectar el patrón anterior, el audio se iniciará en el punto en el que estaba.

#### Prueba 2. Detectar un patrón que no contiene ningún audio

- **Situación esperada:** Se espera que se muestre el objeto sin oír ninguna explicación sonora.
- **Situación obtenida:** Se reproduce un audio, concretamente el de otro objeto.
- **Conclusión:** Después de examinar la situación obtenida, se observa que se reproduce el audio correspondiente al orden en el que se cargaron los datos y no al identificador del objeto. Por tanto, se modificará este punto.

#### Prueba 3. Al finalizar el audio de un objeto situado en una localización geográfica concreta, se deberá informar que se debe dejar de visualizar el objeto

- **Situación esperada:** Se espera que al finalizar el audio se informe al módulo de visualización para detener la muestra de información.
- **Situación obtenida:** En los objetos que disponen de audio se pasa la prueba. Pero en los objetos que no tienen audio, no se deja de visualizar la información.
- **Conclusión:** Por ello se modifica el módulo para que se informe a los 10 segundos de la finalización de la visualización en los objetos que no posean fichero de sonido. Tras solucionar este problema el módulo funciona correctamente.

### Módulo de Visualización de información

#### Prueba 1. Intentar visualizar un objeto 3D

- **Situación esperada:** Al reconocer un patrón por medio de la cámara, se deberá mostrar correctamente el objeto 3D, así como su textura.
- **Situación obtenida:** En las primeras pruebas realizadas sobre este punto, sólo se cargaba una parte del objeto y por tanto la visualización no era la correcta.

- **Conclusión:** Despues de innumerables pruebas en este punto se ha solucionado el problema y se carga completamente todos los archivos 3D.

### Prueba 2. Intentar visualizar un objeto sin un archivo *3DS* asociado

- **Situación esperada:** Al no existir un objeto 3D, se pretende mostrar la textura correspondiente en plano.
- **Situación obtenida:** Se visualiza perfectamente la textura.
- **Conclusión:** Al haber realizado antes las pruebas con los objetos 3D, este punto no supuso ningún problema. En este sentido el módulo funciona correctamente.

### Prueba 3. Probar que el sistema reconozca una serie de patrones

- **Situación esperada:** El sistema deberá reconocer los patrones que estén disponibles en la plataforma.
- **Situación obtenida:** No se reconocía el patrón B.
- **Conclusión:** El fichero que representa el patrón y que contiene todos los datos de coordenadas del patrón estaba corrupto. Se vuelve a generar el patrón y todo funciona correctamente.

### Prueba 4. Una vez visualizado un objeto, desplazarlo en todas las direcciones

- **Situación esperada:** Que al utilizar la pantalla táctil, el objeto se deberá desplazar en la dirección indicada.
- **Situación obtenida:** Vemos, que funciona perfectamente manteniendo la orientación del dispositivo móvil horizontal, pero si variamos la orientación los movimientos no se corresponden.
- **Conclusión:** Se hacen los cambios oportunos para que el desplazamiento se produzca en función de la orientación del dispositivo móvil.

### Prueba 5. Escalar un objeto que se esté visualizando

- **Situación esperada:** El objeto se escalará al realizar el gesto sobre la pantalla táctil.
- **Situación obtenida:** El escalado es excesivamente grande.
- **Conclusión:** Se modifica el factor de escalado y el aumento y disminución de los objetos se realiza de forma más progresiva.

### Prueba 6. Comprobación del estado de la conexión a internet

- **Situación esperada:** Cuando el dispositivo inicie la aplicación y no tenga conexión a internet deberá mostrar un error y finalizar su ejecución.
- **Situación obtenida:** En caso de haber conexión la aplicación continua su ejecución. En caso contrario se muestra una ventana informándonos del error y, aparentemente, finaliza la aplicación. Pero nos muestra una excepción.
- **Conclusión:** Esto era debido a que se finalizaba la ejecución de la aplicación, pero no se llamaba a un método para destruir ordenadamente la aplicación y cerrar todos los recursos abiertos. Se informa del fallo y se modifica.

### Prueba 7. Comprobación de las preferencias de usuario

- **Situación esperada:** Al establecer las preferencias deberán guardarse de forma persistente. Así, al iniciar la aplicación en otro momento, deberán cargarse éstas.
- **Situación obtenida:** Al establecer alguna dirección de los ficheros de configuración incorrecta, salir de la aplicación e intentar volver a ejecutarla, nos muestra un mensaje de error y la aplicación finaliza.
- **Conclusión:** Esto se debía a que se guardaban las preferencias sin comprobar si las direcciones eran válidas o no. Por tanto, en caso de ser inválidas, al iniciar la aplicación intentaba cargar los recursos de ahí. Se modifica para que en caso de que los ficheros no estén disponibles se carguen otros por defecto.

## 6.3. Pruebas del sistema

### 6.3.1. Definición

“Las pruebas del sistema son pruebas de integración del sistema de información completo, y permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.”

### 6.3.2. Objetivos

Con este tipo de pruebas se desea asegurar que el sistema cumpla con todas las especificaciones funcionales, de rendimiento, seguridad, . . . impuestas en un principio por el cliente que solicitó la aplicación (en este caso el director del proyecto).

### 6.3.3. Planificación

#### Prueba 1. Probar el funcionamiento de visualización de objetos con patrones.

- **Situación esperada:** Se espera visualizar los objetos cuya información está establecida en los ficheros *XML* de configuración al procesar el reconocimiento de un patrón.

- **Situación obtenida:** Se realiza con éxito tanto la visualización como la reproducción sonora de los objetos.
- **Conclusión:** En este sentido los módulos de **Carga de Datos, Renderización o Parser de objetos 3D, Visualización de información y Sonido** se integran correctamente.

**Prueba 2. Probar el funcionamiento de visualización de objetos en una localización geográfica.**

- **Situación esperada:** Se espera visualizar los objetos cuya información está establecida en los ficheros *XML* de configuración al detectar una determinada posición geográfica.
- **Situación obtenida:** Se realiza con éxito tanto la visualización como la reproducción sonora de los comentarios de los objetos.
- **Conclusión:** En este sentido los módulos de **Carga de Datos, Renderización o Parser de objetos 3D, Visualización de información, Localización y Sonido** se integran correctamente.

## 6.4. Pruebas de rendimiento

Nuestra plataforma consumirá una determinada cantidad de recursos (memoria, tiempo de proceso y ancho de banda) en los dispositivos móviles en que se ejecute. Debemos procurar que sean los menores posibles.

Es por ello que debemos medir dichos consumos para intentar evitar posibles cuellos de botella en el rendimiento que se puedan presentar.

Estas mediciones las realizaremos con programas tales como Traffic Monitor, System Panel Lite o Advanced Task Manager que nos proporcionen los datos buscados.

Para realizar estas pruebas utilizaremos distintos modelos de dispositivos móviles y haremos una tabla comparativa con los resultados obtenidos.

| Dispositivo móvil                                                                       | Memoria | Tiempo CPU | Ancho de banda |
|-----------------------------------------------------------------------------------------|---------|------------|----------------|
| <i>HTC HD2</i> (Procesador Qualcomm Snapdragon a 1000 MHz y 576 Mb de RAM)              | 20.8 Mb | 48 s       | 1.6 Mb         |
| <i>Samsung Galaxy 3</i> (Procesador a 667 MHz y 256 Mb de RAM)                          | 19.3 Mb | 55 s       | 1.7 Mb         |
| <i>Samsung Galaxy Mini</i> (Procesador Qualcomm MSM7227 ARMv6 a 600MHz y 384 Mb de RAM) | 21.4 Mb | 53 s       | 1.6 Mb         |

## 6. Pruebas software

---

*NOTA 1: Los datos obtenidos se obtienen transcurridos 3 minutos desde el inicio de la aplicación.*

*NOTA 2: El consumo de red dependerá de los datos que tenga que descargar para la exposición. En las pruebas se utilizaron siempre los mismos datos.*

# Capítulo 7

## Pruebas de usabilidad

En este apartado se indicarán todas y cada una de las pruebas realizadas a la aplicación, puesto que el fin último de la aplicación es el uso por parte de personas y su opinión debe ser fundamental en la realización de la misma.

A lo largo del desarrollo del proyecto tendremos una serie de versiones y de todas ellas, salvo la final que salga de todo el proceso, habrá una serie de errores y críticas que nosotros mismos u otros detectarán y solucionarán de la manera oportuna. A fin de dejar documentado todo este proceso desarrollamos este capítulo.

El proyecto ha tenido una serie de versiones. Entre ellas distinguimos dos tipos:

- Las **versiones alpha** han sido versiones de las que sólo han tenido conocimiento y valoración el desarrollador. Probablemente este tipo de versiones entendidas así serían innumerables, sin embargo aquí nos vamos a centrar sólo en aquellas en las que se han detectado errores significativos y por tanto han marcado el rumbo del proyecto de alguna manera.
- Las **versiones beta** son aquellas que han sido publicadas con intención de que fuesen versiones finales. Han sido probadas por varias personas.

Para las versiones beta necesitamos personas que hagan pruebas y nos comuniquen errores de usabilidad, de funcionamiento o que aporten sugerencias. Para ello manejamos dos colectivos, además, claro está, de las pruebas hechas por el propio desarrollador. Estos dos grupos son:

- Profesores compañeros del departamento de Informática del centro educativo en el que trabajo que están relacionadas con el campo de la informática en general y, algunos de ellos, de la usabilidad y accesibilidad.
- Gente ajena al desarrollo informático, usuarios simples tanto de ordenadores domésticos como de dispositivos móviles.

## 7. Pruebas de usabilidad

---

Las versiones que evaluamos y documentamos aquí son:

- Versiones alpha
  - 0.1 (prototipo)
  - 0.2 (versión con visualización de archivos *3DS*)
- Versiones beta
  - 1.0 (versión con carga de datos y sonido)
  - 1.1 (versión con localización e interacción del usuario)
  - 1.2 (versión con preferencias del usuario)

### 7.1. Versiones Alpha

#### 7.1.1. Alpha 0.1 (Prototipo no funcional)

##### Descripción y alcance

Se trata de una versión de fase muy inicial. Su propósito era el de detectar una plantilla en una imagen, estudiando la orientación, posición y tamaño de la plantilla. La aplicación calcula la posición y orientación relativa de la cámara respecto a la plantilla, y usando esta información podrá pasar a dibujar el objeto correspondiente sobre la imagen capturada, de modo que el objeto aparezca sobre la plantilla en la posición, orientación y tamaño correspondiente al punto de vista de la cámara. En esta versión los objetos, no serían realmente objetos elaborados, sino primitivas básicas y polígonos.

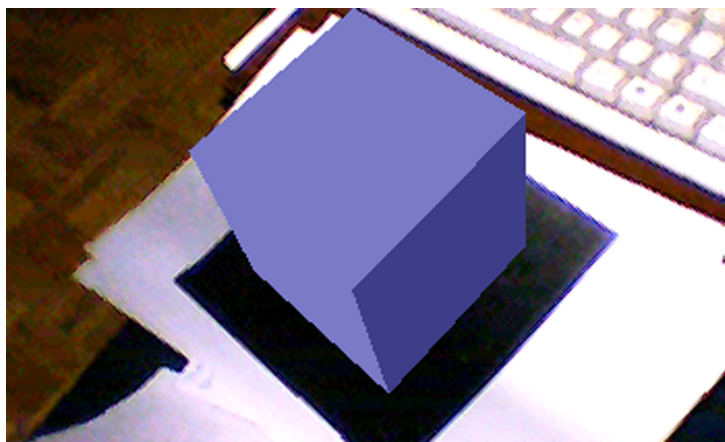


Figura 7.1: Una captura de la versión prototipo.

## Errores descubiertos y soluciones

Limitándonos al alcance del prototipo se hicieron pruebas desde el ámbito de desarrollo. En ellas se detectaron los siguientes errores:

1. Sólo se representaban primitivas estándar.
  - Descripción: La plataforma en el primer prototipo sólo mostraba primitivas, es decir, objetos predefinidos de *OpenGL*.
  - Solución: No hay ningún módulo de visualización de objetos y es el objetivo fundamental para la siguiente versión.
2. Las texturas de las primitivas están inmersas en el código fuente.
  - Descripción: Para poder mostrar las texturas de las primitivas, se deberá cargar la textura. El nombre del archivo, en esta versión, estará inmerso en el código fuente. Por tanto, para poder realizar alguna modificación sería necesario cambiar el código fuente.
  - Solución: Lógicamente, este problema es necesario subsanarlo. En las siguientes versiones se crearán archivos de configuración de la plataforma con ésta y otra información necesaria.

### 7.1.2. Alpha 0.2 (Versión servidor con visualización de archivos 3DS)

#### Descripción y alcance

Después de hacer que se detectasen los patrones y se mostrasen objetos (pese a ser primitivas), ya teníamos una versión muy precaria del proyecto, pero aún así una versión.

Lo siguiente a conseguir era que la plataforma lograra cargar y visualizar todo tipo de objetos 3D. Se cargarían ficheros *3ds*, *obj* y *md2*. Así, cualquier diseñador podría realizar su propio objeto para la plataforma y cualquier exposición puede así crearse con los objetos más apropiados.

En la figura 7.2 se puede ver cómo era la interfaz al finalizar esta versión.

#### Errores descubiertos y soluciones

1. No hay externalización en ficheros de configuración.
  - Descripción: Todos los datos correspondientes a los objetos tridimensionales a cargar, así como sus texturas siguen estando en el código fuente con lo que esto implica. Un cambio en estos datos requeriría la modificación de código, algo inadmisibile.
  - Solución: Para la siguiente versión se propondrá una solución mediante ficheros *XML*.



Figura 7.2: Interfaz de la versión Alpha 0.2

2. Se echa en falta una descripción sonora del objeto.
  - Descripción: Sólo la visualización del objeto 3D se nos queda corto para satisfacer al usuario.
  - Solución: Por tanto se propondrá una forma de aumentar la información. Se añadirán comentarios sonoros, que se reproducirán automáticamente para cada objeto al mostrar este.

## 7.2. Versiones Beta

### 7.2.1. Beta 1.0 (Versión con carga de datos y sonido)

#### Descripción y alcance

Lo siguiente a conseguir era una plataforma totalmente independiente de lo que es la aplicación y los datos cargados en ésta. Para ello se realiza una carga de datos desde la aplicación. Los datos estarán en ficheros *XML*. Cada fichero tiene asociada un esquema definido en un fichero *XSD*. Se añade así un módulo de **Carga de datos**.

Además de esto, en esta versión se ha propuesto aumentar la información que capta el visitante. Hasta el momento sólo visualizaba objetos 3D. Se pretende aumentar la información añadiendo un **módulo de Sonido**. El sonido se reproducirá al mostrar un objeto.

#### Errores descubiertos y soluciones

1. No deben ser obligatorios ni el sonido ni el objeto *3ds* de los ficheros *XML*.
  - Descripción: En los ficheros *XML* que describen cada objeto hay una serie de etiquetas que no deberán ser obligatorias, ya que muchos objetos no tienen

porque tener una descripción sonora, o simplemente es una imagen y no tienen asociados ningún archivo *3ds*.

- Solución: Se modifica el módulo **Carga de Datos** para poder establecer ciertos atributos como opcionales y así adaptarlo mucho mejor a las necesidades del visitante.

2. Detectar un patrón que contiene audio cuando se está reproduciendo el de otro objeto

- Descripción: La explicación del objeto anterior no se detiene y se lanza en paralelo otro audio con la nueva explicación. Por lo tanto suenan dos explicaciones simultáneamente lo que imposibilita el entendimiento.
- Solución: Se deberá modificar el sistema para que cuando el sistema detecte otro objeto distinto, en el que haya asociado un sonido, se detenga el primero e inmediatamente después se reproduzca el del objeto mostrado.

### 7.2.2. Beta 1.1 (Versión con localización e interacción del usuario)

#### Descripción y alcance

La segunda versión beta nace con el añadido de un **módulo de Localización** y con una mejora sustancial de la interfaz gráfica de usuario 7.3. Esta versión ya tenía una funcionalidad casi completa, solamente le faltaría todas las opciones de personalización de las opciones del usuario.



Figura 7.3: Interfaz de la versión Beta 1.1

## 7. Pruebas de usabilidad

---

Para la mejora de la interacción del usuario con la aplicación se propone un sistema para manipular los objetos a través de la pantalla táctil de los dispositivos móviles.

Además, se propone un nuevo diseño en las preferencias de usuario para la siguiente versión que se puede ver en el siguiente prototipo:

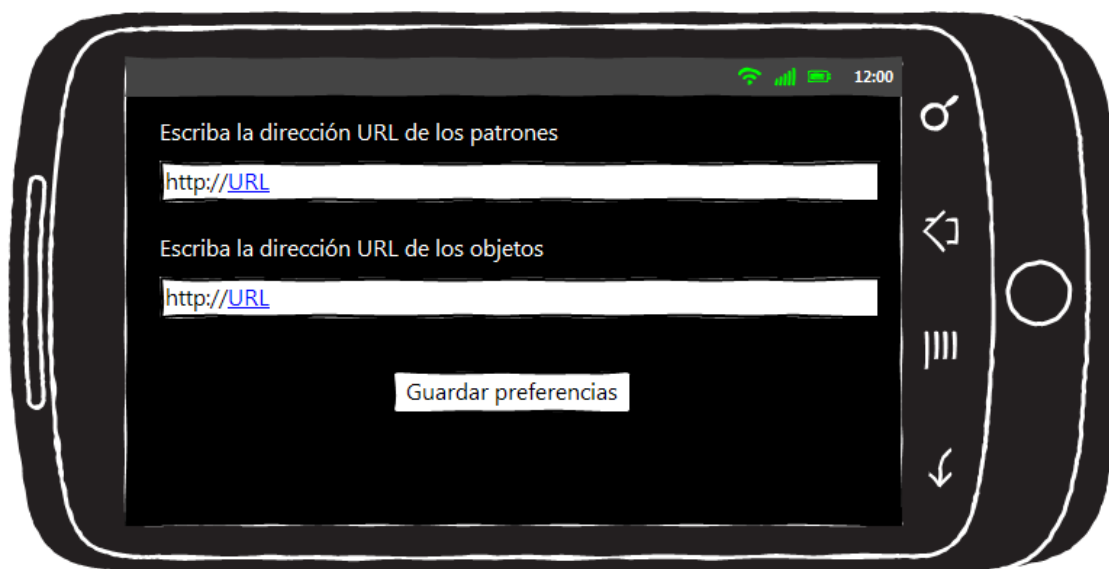


Figura 7.4: Prototipo para la versión final

### Errores descubiertos y soluciones

1. Desplazamiento de los objetos incorrecto
  - Descripción: Al interactuar con el objeto mostrado no tiene en cuenta la orientación del dispositivo. Si está en posición vertical e intentamos desplazar el objeto a la derecha, veremos que se moverá en otra dirección.
  - Solución: Deberemos provocar los desplazamientos de los objetos en función de la orientación del dispositivo móvil.
2. Escalado excesivo
  - Descripción: Al producirse un escalado de un objeto veíamos que se producía de manera excesiva.
  - Solución: La solución consistió en disminuir el factor de escalado y por tanto se realiza más lenta y progresivamente.
3. Tamaño excesivos de los objetos al mostrarse mediante localización geográfica
  - Descripción: Al mostrar un objeto cuando accedemos a una localización geográfica concreta, éste aparece a un tamaño mayor del esperado y no se muestra completo.

- Solución: Pese a que se solucionaría reduciéndolo mediante el zoom, creemos conveniente disminuir el tamaño y mostrarlo centrado en la pantalla.

### 7.2.3. Beta 1.2 (Versión con preferencias del usuario)

#### Descripción y alcance

Esta versión ya es muy cercana a la que puede ser la versión final. Con esta versión ya se consigue el 100 % de la funcionalidad, es decir, ya esta operativa la aplicación y todas las acciones que de manipulación y configuración por parte de los usuarios. Además se establece un comportamiento de mínimos para ella igual al de la aplicación final. Esto quiere decir que es la versión de partida en la última parte del desarrollo, parte en la cual nos dedicaremos a descubrir errores mediante pruebas de distinta índole hasta intentar conseguir una aplicación libre de errores.

En esta versión se siguen haciendo pruebas para seguir recibiendo, estudiando y corrigiendo los errores que detectan los probadores.

#### Errores descubiertos y soluciones

##### 1. Confusión en los ítems de configuración

- Descripción: Se detecta que con el diseño actual se producen confusiones sobre qué caja de texto corresponde a cada apartado.
- Solución: Se propone un nuevo diseño para hacer más intuitiva la pantalla de configuración. Se puede ver en las imágenes 7.5 y 7.6



Figura 7.5: Diseño de la pantalla de configuración

##### 2. Intentar cambiar el idioma



Figura 7.6: Editando en la pantalla de configuración

- Descripción: Se intenta cambiar el idioma. Al poner la aplicación en inglés había ciertos elementos en español.
- Solución: Esto era debido a que no se habían traducido ciertos ítems.

Por último, se muestran unas capturas de pantallas de la versión al finalizar la fase de pruebas con sus correspondientes correcciones:



Figura 7.7: Pantala principal final



Figura 7.8: Configuración final 1



Figura 7.9: Configuración final 2

### 7.3. Probadores : Los usuarios finales

Se ha podido reunir a unos pocos probadores (sobre 10) de diversa formación para hacer unas sencillas pruebas. Con las que se ha pretendido probar:

- La funcionalidad del sistema con todas sus restricciones y contrastar su utilidad real a nivel de usuario. Entendemos aquí que el sistema parte con unas limitaciones dadas por la tecnología que se ha empleado. Sin embargo no sería útil un sistema demasiado restrictivo, es necesario comprobar que en el grueso de las acciones, el sistema se comporta adecuadamente.
- La facilidad de uso y la adecuación de los manuales de usuario. Los usuarios simplemente dispondrán de patrones que visualizarán delante de la cámara para tratar de reconocerlo y mostrar un objeto con información y ellos mismos deberían de saber funcionar correctamente con la aplicación sin saber muy bien su total funcionalidad.
- Las más diversas cuestiones de diseño de interfaces, accesibilidad... Es muy importante que la intuitividad sea constante en el resultado final. Para ello es necesario que la interfaz sirva realmente para lo que pretende servir.
- Por último, después de valorar los puntos anteriores hemos tenido en cuenta el grado de satisfacción de los usuarios que han probado el sistema. Aunque consciente de que la validez de este punto ha de ser contrastada con el resultado de los anteriores (a una pregunta de *¿Qué te parece?* la respuesta puede que no sea todo lo objetiva que se puede desear), sí que puede ayudar a acabar de situar el resultado de la prueba.

Para realizar todas estas comprobaciones, como ya hemos dicho, hemos reunido a unas personas de diversa formación. Se pueden clasificar en tres grandes grupos:

- Compañeros del departamento de Informática del centro en el que trabajo: Ingenieros en informática muchos de los cuáles han trabajado activamente en usabilidad y temas de accesibilidad. Su papel en las pruebas se incrementa para el punto de cuestiones de diseño.
- Avanzados: Personas que con formación informática o sin ella, trabajan de manera asidua con las nuevas tecnologías. Se espera de ellos errores en puntos concretos.
- Usuarios estándar: Utilizan la informática a nivel usuario, es posiblemente el nivel que más nos interesa y en el que más se estuvo atento a la comprensión de manuales y al proceso de aprendizaje.

Aunque de cada uno de ellos se esperaría que detectasen un tipo de errores diferente, no tiene porque ser así. Cualquier usuario puede detectar cualquier error y de hecho así ocurrió:

- Gente del departamento no le ejecutaba la aplicación por problemas en la conexión de datos.
- Usuarios avanzados detectaron errores de usabilidad.

Esto sería buena señal ya que significa que los usuarios finales con el manual de usuario y la interfaz tiene un conocimiento total de lo que debe ser el sistema y lo que debe hacer. Los conocimientos sobre el tema se quedan un poco al margen, cosa que por otra parte lógica, si estamos persiguiendo.



# Capítulo 8

## Presupuesto y Plan Temporal

### 8.1. Presupuesto

En este capítulo vamos a realizar un presupuesto general para el desarrollo del proyecto. Para ello se utiliza una herramienta creada en Excel para tal fin.

#### 8.1.1. Desarrollo del proyecto

Aquí se enumeran las tareas relativas a la fase de codificación propiamente dicha y se hace una estimación de tiempos para cada una.

Se detallarán las tareas unitarias propias de la fase de codificación, pudiendo agruparse por bloques para facilitar la lectura. Para cada tarea tendremos los siguientes campos:

- *id*: es un secuencial para identificar cada entrada
- *tarea*: descripción de la tarea
- *horas*: las horas estimadas para realizar la codificación de dicha tarea (solo codificación. Se excluye todo lo demás: análisis, pruebas, manuales, etc. . .)
- *% total*: Es un campo informativo. Indica el peso de la estimación de cada tarea en relación a la estimación total del desarrollo

Como se puede ver en la figura 8.1, se especifican las tareas del proyecto agrupándolas en los módulos del proyecto:

- Módulo de Carga de Datos
- Módulo de Localización
- Módulo de Renderización o Parser de objetos 3D
- Módulo de Sonido
- Módulo de Visualización de información

## 8. Presupuesto y Plan Temporal

| TFG: Estimación de proyecto |                                                       |       |         |
|-----------------------------|-------------------------------------------------------|-------|---------|
| id                          | Tareas de desarrollo                                  | Horas | % total |
| 1                           | <b>Módulo de Carga de Datos</b>                       |       |         |
| 2                           | Diseño de ficheros de configuración                   | 10    | 2%      |
| 3                           | Gestión de patrones y localizaciones                  | 10    | 2%      |
| 4                           | Estructura de objetos informativos                    | 20    | 5%      |
| 5                           | Conectividad                                          | 5     | 1%      |
| 6                           | Lectura y almacenamiento de ficheros de configuración | 55    | 13%     |
| 7                           |                                                       |       |         |
| 8                           | <b>Módulo de Localización</b>                         |       |         |
| 9                           | Obtener datos geolocalizados                          | 10    | 2%      |
| 10                          | Observador de cambio de localización                  | 15    | 3%      |
| 11                          |                                                       |       |         |
| 12                          | <b>Módulo de Renderización o Parser de objetos 3D</b> |       |         |
| 13                          | Renderización de distintos objetos                    | 50    | 12%     |
| 14                          | Gestión de la escena que soporta los objetos          | 15    | 3%      |
| 15                          |                                                       |       |         |
| 16                          | <b>Módulo de Sonido</b>                               |       |         |
| 17                          | Estructura de los audios                              | 10    | 2%      |
| 18                          | Reproducción y gestión de eventos producidos          | 35    | 8%      |
| 19                          |                                                       |       |         |
| 20                          | <b>Módulo de Visualización de información</b>         |       |         |
| 21                          | Pantallas de las actividades                          | 20    | 5%      |
| 22                          | Reconocimiento de patrones a través de la cámara      | 65    | 15%     |
| 23                          | Visualización de la información                       | 55    | 13%     |
| 24                          | Desplazamiento de los objetos                         | 15    | 3%      |
| 25                          | Escalado de los objetos                               | 20    | 5%      |
| 26                          | Internacionalización                                  | 5     | 1%      |
| 27                          | Gestión de Preferencias                               | 15    | 3%      |
| 28                          |                                                       |       |         |
| Suma                        |                                                       | 430   | 100%    |

Figura 8.1: Estimación de horas del desarrollo del proyecto

En total, la estimación del desarrollo del proyecto será de **420 horas**.

### 8.1.2. Tareas globales del proyecto

Ahora debemos estimar las horas relativas al resto de tareas (gestión, pruebas, etc. . . ).

Podemos establecer las horas como un porcentaje de la fase de *Desarrollo* (por ejemplo 15 % de las horas del *Desarrollo*) o como un valor de horas fijo (por ejemplo 40 horas).

Se reflejan cada una de las tareas que componen el proyecto completo. Para cada fila, se tienen los siguientes campos (columnas):

- **Tarea:** Descripción de la tarea
- **h. %:** Porcentaje de horas de esta tarea en relación a las horas de la hoja *Desarrollo*
- **h.fija:** Estimación de horas fija, sin relación con las horas de *Desarrollo*
- **Horas:** Es la estimación resultado de usar una de las dos columnas anteriores. Por defecto, si se rellenan las dos, prevalece la de horas fijas

- **% total:** es un campo informativo sobre el peso de dicha tarea en relación al proyecto completo

| TFG: Estimación              |      |         |              |             |
|------------------------------|------|---------|--------------|-------------|
| Tareas globales del proyecto |      |         |              |             |
| Tarea                        | h. % | h. fija | Horas        | % total     |
| Gestión del proyecto         | 15%  |         | 64,5         | 8%          |
| Análisis Funcional           |      | 30      | 30,0         | 4%          |
| Diseño Técnico               |      | 40      | 40,0         | 5%          |
| Propuesta Gráfica            |      | 10      | 10,0         | 1%          |
| Elaboración maqueta          |      | 20      | 20,0         | 3%          |
| Elaboración Plan de Pruebas  |      | 20      | 20,0         | 3%          |
| Config. entorno desarrollo   |      | 8       | 8,0          | 1%          |
| <b>Desarrollo</b>            |      |         | <b>430,0</b> | <b>56%</b>  |
| Integración componentes      | 5%   |         | 21,5         | 3%          |
| Ejecución Plan de Pruebas    | 7%   |         | 30,1         | 4%          |
| Documentación                | 5%   |         | 21,5         | 3%          |
| Instalación                  | 2%   |         | 8,6          | 1%          |
| Pruebas TFG                  | 5%   |         | 21,5         | 3%          |
| Formación                    | 4%   |         | 17,2         | 2%          |
| Mantenimiento                |      | 30      | 30,0         | 4%          |
|                              |      |         | 0,0          |             |
| total h:                     |      |         | <b>772,9</b> | <b>100%</b> |

Figura 8.2: Estimación de horas de todas las tareas del proyecto

Las tareas del proyecto son las siguientes (se pueden consultar las horas destinadas a cada una de ellas en la figura 8.2):

- Gestión del proyecto
- Análisis Funcional
- Diseño Técnico
- Propuesta Gráfica
- Elaboración maqueta
- Elaboración Plan de Pruebas
- Config. entorno desarrollo
- Desarrollo
- Integración componentes
- Ejecución Plan de Pruebas
- Documentación
- Instalación

## 8. Presupuesto y Plan Temporal

- Formación
- Mantenimiento

### 8.1.3. Dedicación por perfiles

En este bloque, en cada columna, se refleja el grado de participación de cada perfil en la tarea correspondiente.

En nuestro caso, hemos considerado que en este proyecto participarán los perfiles siguientes:

- Gerente (G)
- Jefe de Proyecto (JP)
- Arquitecto o Analista (A)
- Programador Senior (PS)
- Programador Junior (PJ)
- Otros (O)

Cada uno de estos perfiles, tendrá un grado de participación mayor o menor en cada tarea dependiendo de sus funciones.

| TFG: Estimación              |     |        |       |         |                         |     |     |     |     |    |        |
|------------------------------|-----|--------|-------|---------|-------------------------|-----|-----|-----|-----|----|--------|
| Tareas globales del proyecto |     |        |       |         | Dedicación por perfiles |     |     |     |     |    |        |
| Tarea                        | h.% | h.fija | Horas | % total | G                       | JP  | A   | PS  | PJ  | O  | Suma % |
| Gestión del proyecto         | 15% |        | 64,5  | 8%      | 10%                     | 90% | 0%  | 0%  | 0%  | 0% | 100%   |
| Análisis Funcional           |     | 30     | 30,0  | 4%      | 0%                      | 5%  | 90% | 5%  | 0%  | 0% | 100%   |
| Diseño Técnico               |     | 40     | 40,0  | 5%      | 0%                      | 0%  | 80% | 20% | 0%  | 0% | 100%   |
| Propuesta Gráfica            |     | 10     | 10,0  | 1%      | 0%                      | 10% | 0%  | 50% | 40% | 0% | 100%   |
| Elaboración maqueta          |     | 20     | 20,0  | 3%      | 0%                      | 10% | 0%  | 50% | 40% | 0% | 100%   |
| Elaboración Plan de Pruebas  |     | 20     | 20,0  | 3%      | 0%                      | 5%  | 5%  | 90% | 0%  | 0% | 100%   |
| Config. entorno desarrollo   |     | 8      | 8,0   | 1%      | 0%                      | 0%  | 0%  | 80% | 20% | 0% | 100%   |
| Desarrollo                   |     |        | 430,0 | 56%     | 0%                      | 5%  | 5%  | 30% | 60% | 0% | 100%   |
| Integración componentes      | 5%  |        | 21,5  | 3%      | 0%                      | 0%  | 0%  | 40% | 60% | 0% | 100%   |
| Ejecución Plan de Pruebas    | 7%  |        | 30,1  | 4%      | 0%                      | 5%  | 0%  | 45% | 50% | 0% | 100%   |
| Documentación                | 5%  |        | 21,5  | 3%      | 0%                      | 5%  | 0%  | 25% | 70% | 0% | 100%   |
| Instalación                  | 2%  |        | 8,6   | 1%      | 0%                      | 10% | 0%  | 90% | 0%  | 0% | 100%   |
| Pruebas TFG                  | 5%  |        | 21,5  | 3%      | 0%                      | 5%  | 0%  | 75% | 20% | 0% | 100%   |
| Formación                    | 4%  |        | 17,2  | 2%      | 5%                      | 5%  | 0%  | 90% | 0%  | 0% | 100%   |
| Mantenimiento                |     | 30     | 30,0  | 4%      | 0%                      | 10% | 0%  | 40% | 50% | 0% | 100%   |
|                              |     |        | 0,0   |         | 0%                      | 0%  | 0%  | 0%  | 0%  | 0% | 0%     |
| total h:                     |     |        | 772,9 | 100%    |                         |     |     |     |     |    |        |

Figura 8.3: Dedicacion de cada uno de los perfiles en las tareas del proyecto

### 8.1.4. Horas y tarifas por perfiles

En este bloque se indica la tarifa (precio por hora de trabajo) correspondiente a cada perfil y se calculan los costes para cada tarea en base a su dedicación.

El contenido del bloque debe leerse en grupos de dos columnas, cada uno de grupos de dos columnas se corresponde con uno de los perfiles descritos antes.

En la cabecera de cada grupo, junto a cada perfil, se indica la tarifa asociada en euros/hora. Se han establecido unas tarifas acordes a la situación del mercado.

En la primera columna de cada grupo, se muestran las horas de dedicación de ese perfil a cada tarea. Es el resultado del porcentaje de dedicación indicado antes.

En la segunda, se muestra el resultado, en euros, de multiplicar esas horas de dedicación, por la tarifa asociada a ese perfil.

En la penúltima fila de este bloque se tienen sumatorios de horas y euros que indican la dedicación y coste total por perfil.

En la última se muestra información sobre el peso de dicha dedicación y coste en relación al total.

| TFG: Estimación              |      |         |              |             | Horas y tarifas por perfiles |                 |             |                   |             |                   |              |                    |              |                    |               |               |
|------------------------------|------|---------|--------------|-------------|------------------------------|-----------------|-------------|-------------------|-------------|-------------------|--------------|--------------------|--------------|--------------------|---------------|---------------|
| Tareas globales del proyecto |      |         |              |             |                              |                 |             |                   |             |                   |              |                    |              |                    |               |               |
| Tarea                        | h. % | h. fija | Horas        | % total     | G                            | 80,00 € JP      | 60,00 € A   | 50,00 € PS        | 40,00 € PJ  | 30,00 € O         |              |                    |              |                    |               |               |
| Gestión del proyecto         | 15%  |         | 64,5         | 8%          | 6,5                          | 516,00 €        | 58,1        | 3.483,00 €        | 0,0         | 0,00 €            | 0,0          | 0,00 €             | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| Análisis Funcional           |      | 30      | 30,0         | 4%          | 0,0                          | 0,00 €          | 1,5         | 90,00 €           | 27,0        | 1.350,00 €        | 1,5          | 60,00 €            | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| Diseño Técnico               |      | 40      | 40,0         | 5%          | 0,0                          | 0,00 €          | 0,0         | 0,00 €            | 32,0        | 1.600,00 €        | 8,0          | 320,00 €           | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| Propuesta Gráfica            |      | 10      | 10,0         | 1%          | 0,0                          | 0,00 €          | 1,0         | 60,00 €           | 0,0         | 0,00 €            | 5,0          | 200,00 €           | 4,0          | 120,00 €           | 0,00 €        | 0,00 €        |
| Elaboración maqueta          |      | 20      | 20,0         | 3%          | 0,0                          | 0,00 €          | 2,0         | 120,00 €          | 0,0         | 0,00 €            | 10,0         | 400,00 €           | 8,0          | 240,00 €           | 0,00 €        | 0,00 €        |
| Elaboración Plan de Pruebas  |      | 20      | 20,0         | 3%          | 0,0                          | 0,00 €          | 1,0         | 60,00 €           | 1,0         | 50,00 €           | 18,0         | 720,00 €           | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| Config. entorno desarrollo   |      | 8       | 8,0          | 1%          | 0,0                          | 0,00 €          | 0,0         | 0,00 €            | 0,0         | 0,00 €            | 6,4          | 256,00 €           | 1,6          | 48,00 €            | 0,00 €        | 0,00 €        |
| <b>Desarrollo</b>            |      |         | <b>430,0</b> | <b>56%</b>  | 0,0                          | 0,00 €          | 21,5        | 1.290,00 €        | 21,5        | 1.075,00 €        | 129,0        | 5.160,00 €         | 258,0        | 7.740,00 €         | 0,00 €        | 0,00 €        |
| Integración componentes      | 5%   |         | 21,5         | 3%          | 0,0                          | 0,00 €          | 0,0         | 0,00 €            | 0,0         | 0,00 €            | 8,6          | 344,00 €           | 12,9         | 387,00 €           | 0,00 €        | 0,00 €        |
| Ejecución Plan de Pruebas    | 7%   |         | 30,1         | 4%          | 0,0                          | 0,00 €          | 1,5         | 90,30 €           | 0,0         | 0,00 €            | 13,5         | 541,80 €           | 15,1         | 451,50 €           | 0,00 €        | 0,00 €        |
| Documentación                | 5%   |         | 21,5         | 3%          | 0,0                          | 0,00 €          | 1,1         | 64,50 €           | 0,0         | 0,00 €            | 5,4          | 215,00 €           | 15,1         | 451,50 €           | 0,00 €        | 0,00 €        |
| Instalación                  | 2%   |         | 8,6          | 1%          | 0,0                          | 0,00 €          | 0,9         | 51,60 €           | 0,0         | 0,00 €            | 7,7          | 309,60 €           | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| Pruebas TFG                  | 5%   |         | 21,5         | 3%          | 0,0                          | 0,00 €          | 1,1         | 64,50 €           | 0,0         | 0,00 €            | 16,1         | 645,00 €           | 4,3          | 129,00 €           | 0,00 €        | 0,00 €        |
| Formación                    | 4%   |         | 17,2         | 2%          | 0,9                          | 68,80 €         | 0,9         | 51,60 €           | 0,0         | 0,00 €            | 15,5         | 619,20 €           | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| Mantenimiento                |      | 30      | 30,0         | 4%          | 0,0                          | 0,00 €          | 3,0         | 180,00 €          | 0,0         | 0,00 €            | 12,0         | 480,00 €           | 15,0         | 450,00 €           | 0,00 €        | 0,00 €        |
|                              |      |         | 0,0          |             | 0,0                          | 0,00 €          | 0,0         | 0,00 €            | 0,0         | 0,00 €            | 0,0          | 0,00 €             | 0,0          | 0,00 €             | 0,00 €        | 0,00 €        |
| <b>total h:</b>              |      |         | <b>772,9</b> | <b>100%</b> | <b>7,3</b>                   | <b>584,80 €</b> | <b>93,4</b> | <b>5.605,50 €</b> | <b>81,5</b> | <b>4.075,00 €</b> | <b>256,8</b> | <b>10.270,60 €</b> | <b>333,9</b> | <b>10.017,00 €</b> | <b>0,00 €</b> | <b>0,00 €</b> |
|                              |      |         |              |             | <b>1%</b>                    | <b>2%</b>       | <b>12%</b>  | <b>18%</b>        | <b>11%</b>  | <b>13%</b>        | <b>33%</b>   | <b>34%</b>         | <b>43%</b>   | <b>33%</b>         | <b>0%</b>     | <b>0%</b>     |

Figura 8.4: Horas y tarifas para cada perfil

### 8.1.5. Costes finales y presupuesto total del proyecto

Por último mostraremos el coste total del proyecto en euros, en base a las horas estimadas, los perfiles, la dedicación y las tarifas de cada perfil.

- **Suma:** Indica el coste total de la tarea, en base a los perfiles implicados en ella, su dedicación y su tarifa.

## 8. Presupuesto y Plan Temporal

- **% total:** Es un campo informativo sobre el peso de dicho coste en relación al coste del proyecto completo

| TFG: Estimación              |     |        |       |         |             |         |
|------------------------------|-----|--------|-------|---------|-------------|---------|
| Tareas globales del proyecto |     |        |       |         | Costes      |         |
| Tarea                        | h.% | h.fija | Horas | % total | Suma        | % total |
| Gestión del proyecto         | 15% |        | 64,5  | 8%      | 3.999,00 €  | 13%     |
| Análisis Funcional           |     | 30     | 30,0  | 4%      | 1.500,00 €  | 5%      |
| Diseño Técnico               |     | 40     | 40,0  | 5%      | 1.920,00 €  | 6%      |
| Propuesta Gráfica            |     | 10     | 10,0  | 1%      | 380,00 €    | 1%      |
| Elaboración maqueta          |     | 20     | 20,0  | 3%      | 760,00 €    | 2%      |
| Elaboración Plan de Pruebas  |     | 20     | 20,0  | 3%      | 830,00 €    | 3%      |
| Config. entorno desarrollo   |     | 8      | 8,0   | 1%      | 304,00 €    | 1%      |
| Desarrollo                   |     |        | 430,0 | 56%     | 15.265,00 € | 50%     |
| Integración componentes      | 5%  |        | 21,5  | 3%      | 731,00 €    | 2%      |
| Ejecución Plan de Pruebas    | 7%  |        | 30,1  | 4%      | 1.083,60 €  | 4%      |
| Documentación                | 5%  |        | 21,5  | 3%      | 731,00 €    | 2%      |
| Instalación                  | 2%  |        | 8,6   | 1%      | 361,20 €    | 1%      |
| Pruebas TFG                  | 5%  |        | 21,5  | 3%      | 838,50 €    | 3%      |
| Formación                    | 4%  |        | 17,2  | 2%      | 739,60 €    | 2%      |
| Mantenimiento                |     | 30     | 30,0  | 4%      | 1.110,00 €  | 4%      |
|                              |     |        | 0,0   |         | 0,00 €      | 0%      |
| total h:                     |     |        | 772,9 | 100%    | 30.552,90 € | 100%    |
|                              |     |        |       |         | 39,53 €     | Mix     |

Figura 8.5: Costes finales del presupuesto

Este presupuesto se ha realizado para un equipo ya conformado, por tanto, los recursos materiales como pueden ser los equipos de desarrollo no se incluyen independientemente, sino que ya forman parte de las tarifas.

Así mismo, estas tarifas por horas ya incluyen un porcentaje de posibles imprevistos.

Visto todo esto, podemos concluir que nuestro presupuesto total será de **30.552,90 euros**.

### 8.2. Plan Temporal

En la figura 8.6 se puede ver el Plan temporal que realizamos para el desarrollo del Proyecto.

Este plan temporal está realizado para que el proyecto sea elaborado por el equipo de 5 personas antes mencionado. Vemos que en total, el proyecto se realizaría en 772,9 horas. Esto equivale a un plazo de 50 días (a media jornada). En caso de que el proyecto sea desarrollado por una sola persona, como es el caso del desarrollo de este proyecto, serían unos 193 días (unos 9 meses de desarrollo a media jornada).

Por último podemos ver en la figura 8.7 podemos ver las tareas y sus asignaciones a cada uno de los recursos, así como el costo de las fases.

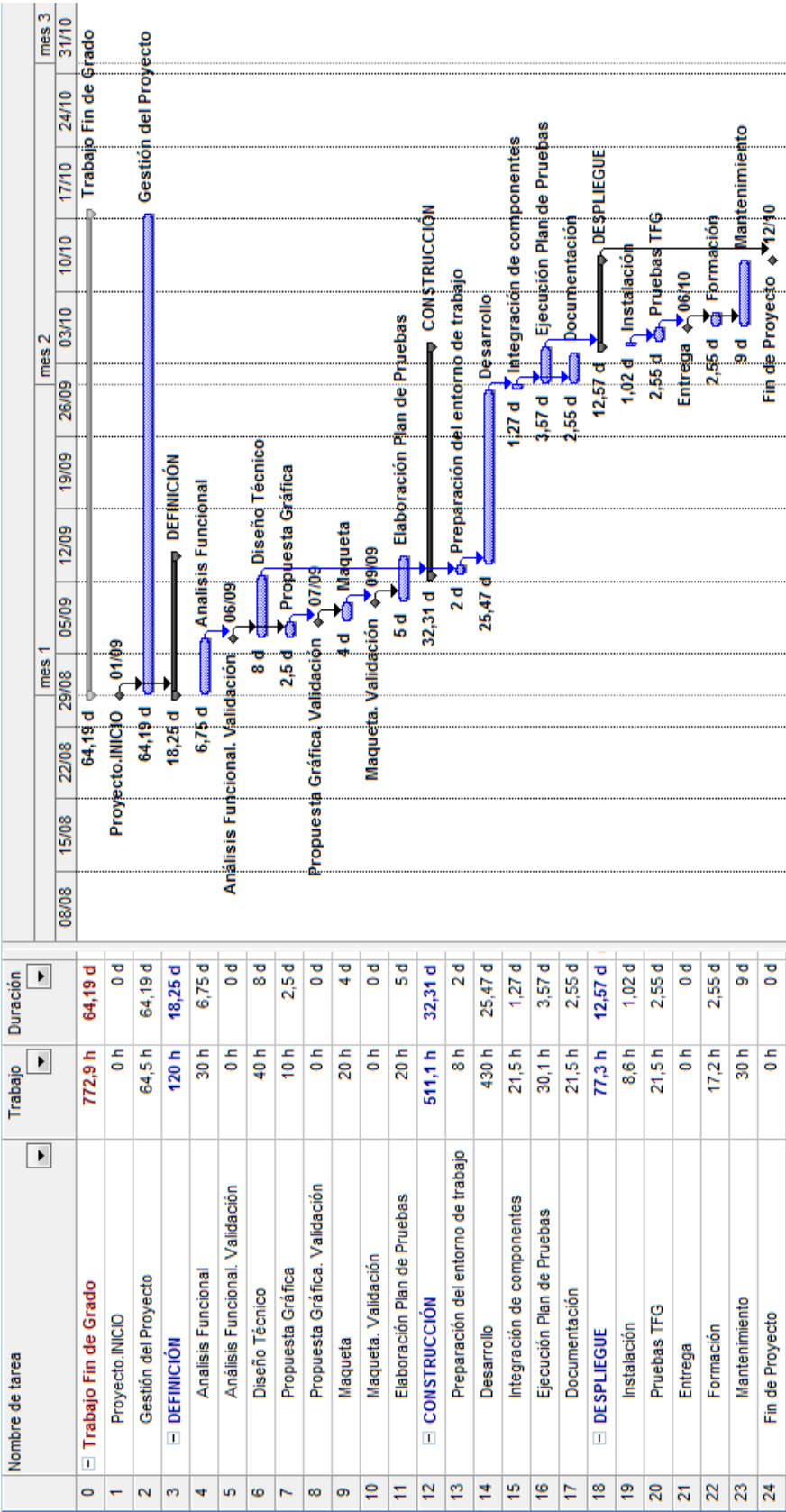


Figura 8.6: Diagrama de Gannt

## 8. Presupuesto y Plan Temporal

| Id | Nombre de tarea                    | Trabajo        | Duración       | Jornadas     | Costo              |
|----|------------------------------------|----------------|----------------|--------------|--------------------|
| 0  | <b>Trabajo Fin de Grado</b>        | <b>772,9 h</b> | <b>64,19 d</b> | <b>96,61</b> | <b>30.552,74 €</b> |
| 1  | Proyecto INICIO                    | 0 h            | 0 d            | 0            | 0,00 €             |
| 2  | Gestión del Proyecto               | 64,5 h         | 64,19 d        | 8,06         | 3.998,90 €         |
|    | G                                  | 0,45 h         |                | 0            | 515,50 €           |
|    | JP                                 | 58,05 h        |                | 0            | 3.483,31 €         |
| 3  | <b>DEFINICIÓN</b>                  | <b>120 h</b>   | <b>18,25 d</b> | <b>15</b>    | <b>5.390,00 €</b>  |
| 4  | Análisis Funcional                 | 30 h           | 6,75 d         | 3,75         | 1.500,00 €         |
|    | JP                                 | 1,5 h          |                | 0            | 90,00 €            |
|    | A                                  | 27 h           |                | 0            | 1.350,00 €         |
|    | PS                                 | 1,5 h          |                | 0            | 60,00 €            |
| 5  | Análisis Funcional. Validación     | 0 h            | 0 d            | 0            | 0,00 €             |
| 6  | Diseño Técnico                     | 40 h           | 8 d            | 5            | 1.920,00 €         |
|    | A                                  | 32 h           |                | 0            | 1.600,00 €         |
|    | PS                                 | 8 h            |                | 0            | 320,00 €           |
| 7  | Propuesta Gráfica                  | 10 h           | 2,5 d          | 1,25         | 380,00 €           |
|    | JP                                 | 1 h            |                | 0            | 60,00 €            |
|    | PS                                 | 5 h            |                | 0            | 200,00 €           |
|    | PJ                                 | 4 h            |                | 0            | 120,00 €           |
| 8  | Propuesta Gráfica. Validación      | 0 h            | 0 d            | 0            | 0,00 €             |
| 9  | Maqueta                            | 20 h           | 4 d            | 2,5          | 760,00 €           |
|    | JP                                 | 2 h            |                | 0            | 120,00 €           |
|    | PS                                 | 10 h           |                | 0            | 400,00 €           |
|    | PJ                                 | 8 h            |                | 0            | 240,00 €           |
| 10 | Maqueta. Validación                | 0 h            | 0 d            | 0            | 0,00 €             |
| 11 | Elaboración Plan de Pruebas        | 20 h           | 5 d            | 2,5          | 830,00 €           |
|    | JP                                 | 1 h            |                | 0            | 60,00 €            |
|    | A                                  | 1 h            |                | 0            | 50,00 €            |
|    | PS                                 | 18 h           |                | 0            | 720,00 €           |
| 12 | <b>CONSTRUCCIÓN</b>                | <b>511,1 h</b> | <b>32,31 d</b> | <b>63,89</b> | <b>18.114,28 €</b> |
| 13 | Preparación del entorno de trabajo | 8 h            | 2 d            | 1            | 304,00 €           |
|    | PS                                 | 0,4 h          |                | 0            | 250,00 €           |
|    | PJ                                 | 1,0 h          |                | 0            | 48,00 €            |
| 14 | Desarrollo                         | 430 h          | 25,47 d        | 53,75        | 15.265,01 €        |
|    | JP                                 | 21,5 h         |                | 0            | 1.200,01 €         |
|    | A                                  | 21,5 h         |                | 0            | 1.075,01 €         |
|    | PS                                 | 120 h          |                | 0            | 5.100,00 €         |
|    | PJ                                 | 258 h          |                | 0            | 7.740,00 €         |
| 15 | Integración de componentes         | 21,5 h         | 1,27 d         | 2,69         | 730,97 €           |
|    | PS                                 | 0,0 h          |                | 0            | 343,00 €           |
|    | PJ                                 | 12,0 h         |                | 0            | 387,08 €           |
| 16 | Ejecución Plan de Pruebas          | 30,1 h         | 3,57 d         | 3,76         | 1.083,46 €         |
|    | JP                                 | 1,5 h          |                | 0            | 80,87 €            |
|    | PS                                 | 13,55 h        |                | 0            | 542,00 €           |
|    | PJ                                 | 15,05 h        |                | 0            | 451,50 €           |
| 17 | Documentación                      | 21,5 h         | 2,55 d         | 2,69         | 730,85 €           |
|    | JP                                 | 1,07 h         |                | 0            | 64,10 €            |
|    | PS                                 | 5,38 h         |                | 0            | 215,00 €           |
|    | PJ                                 | 15,05 h        |                | 0            | 451,55 €           |
| 18 | <b>DESPLIEGUE</b>                  | <b>77,3 h</b>  | <b>12,57 d</b> | <b>9,66</b>  | <b>3.049,55 €</b>  |
| 19 | Instalación                        | 8,6 h          | 1,02 d         | 1,08         | 361,30 €           |
|    | JP                                 | 0,87 h         |                | 0            | 51,01 €            |
|    | PS                                 | 7,73 h         |                | 0            | 309,40 €           |
| 20 | Pruebas TFG                        | 21,5 h         | 2,55 d         | 2,69         | 838,35 €           |
|    | JP                                 | 1,07 h         |                | 0            | 64,10 €            |
|    | PS                                 | 10,13 h        |                | 0            | 645,00 €           |
|    | PJ                                 | 4,3 h          |                | 0            | 129,15 €           |
| 21 | Entrega                            | 0 h            | 0 d            | 0            | 0,00 €             |
| 22 | Formación                          | 17,2 h         | 2,55 d         | 2,15         | 739,91 €           |
|    | G                                  | 0,87 h         |                | 0            | 60,21 €            |
|    | JP                                 | 0,87 h         |                | 0            | 51,01 €            |
|    | PS                                 | 15,47 h        |                | 0            | 618,70 €           |
| 23 | Mantenimiento                      | 30 h           | 9 d            | 3,75         | 1.110,00 €         |
|    | JP                                 | 3 h            |                | 0            | 180,00 €           |
|    | PS                                 | 12 h           |                | 0            | 480,00 €           |
|    | PJ                                 | 15 h           |                | 0            | 450,00 €           |
| 24 | Fin de Proyecto                    | 0 h            | 0 d            | 0            | 0,00 €             |

Figura 8.7: Tareas y sus recursos asignados

# Capítulo 9

## Manual del programador

En este capítulo, nos centraremos en la descripción de la estructura con la cual ha sido implementada la aplicación, así como en la de todos los componentes utilizados. Todas las clases se encuentran almacenadas en los siguientes paquetes:

- `com.ivanlorenzo.connection`
- `com.ivanlorenzo.data`
- `com.ivanlorenzo.gui`
- `com.ivanlorenzo.location`
- `com.ivanlorenzo.parser`

### 9.1. Paquete `com.ivanlorenzo.connection`

En este paquete, es el que se utiliza para comprobar el estado de la conexión en el dispositivo móvil. A continuación se muestra una tabla con la descripción de las diferentes clases de este paquete:

- **`ConnectionStatus.java`** : Clase que utilizaremos para comprobar el estado de la conexión a Internet. Tiene métodos estáticos que comprobarán si tenemos conexión o no

### 9.2. Paquete `com.ivanlorenzo.data`

En este paquete, es en el cual se engloban todas las clases que son utilizadas para la carga de los datos necesarios para realizar la exposición. Se deberán cargar los datos de los patrones y localizaciones así como los objetos que tengan asignados.

Las clases que nos encontramos en este paquete, organizadas por subpaquetes son las explicadas en los cuadros que siguen:

- **`XMLCompiler.java`** : Clase genérica encargada de la compilación de ficheros XML.

- **AugmentedRealityXMLCompiler.java** : Clase hija de XMLCompiler encargada de compilar el fichero XML que contiene la información sobre los patrones y localizaciones.
- **ObjectsXMLCompiler.java** : Clase hija de XMLCompiler encargada de compilar el fichero XML que contiene la información sobre los objetos.
- **KnowledgeBase.java** : Base de conocimiento que almacena toda la información sobre los objetos y los patrones que se va cargando
- **CompilerGenerator.java** : Clase que es la encargada de iniciar la carga de ficheros
- **Pattern.java** : Clase que contiene las propiedades del patrón al que le irá asignado un objeto.
- **GeoLocation.java** : Clase que representa una localización al que le irá asignado un objeto.

### 9.2.1. Paquete com.ivanlorenzo.data.exceptions

- **LoadXMLException.java** : Excepción en el caso que se produzca algún error en la carga de datos.
- **NoSuchExtensionObject3DException.java** : Excepción que se lanzará cuando intentemos cargar un fichero 3D cuya extensión

### 9.2.2. Paquete com.ivanlorenzo.data.objects

Este paquete contendrá las clases que gestionan todo lo relacionado con la gestión y creación de los distintos tipos de objetos que soporta la plataforma.

- **Object3D.java** : Clase que representa un objeto 3D.
- **Max3DSObject.java** : Clase que representa un objeto con formato 3ds. Esta clase hereda de *Object3D*
- **MD2Object.java** : Clase que representa un objeto con formato md2. Esta clase hereda de *Object3D*
- **OBJObject.java** : Clase que representa un objeto con formato obj. Esta clase hereda de *Object3D*
- **Object3DCreator.java** : Clase **factoría** que será la encargada de crear un objeto determinado en función de su tipo. Para esto utiliza alguno de sus métodos estáticos.

### 9.2.3. Paquete com.ivanlorenzo.data.objects.audio

Este paquete contendrá las clases que gestionan todo lo relacionado con la obtención y reproducción de ficheros de audio.

- **Audio.java** : Clase abstracta que representa un objeto de sonido que contendrán los objetos.
- **LocalAudio.java** : Clase que hereda de *Audio*. Se utilizará para representar audios que estén en el dispositivo.
- **RemoteAudio.java** : Representa un audio localizado en una ubicación remota. Hereda de la clase *Audio*
- **MediaPlayerAdapter.java** : Clase adaptadora del patrón **Adapter** [25] que adapta la clase *MediaPlayer* y además es un sujeto observable ya que cuando finalice el audio deberá avisar a sus observadores para que realicen las acciones oportunas.

## 9.3. Paquete com.ivanlorenzo.gui

En este paquete, es en el cual se engloban todas las clases que son utilizadas para gestionar la interfaz gráfica de usuario. A continuación se muestran las descripciones de las diferentes clases de este paquete:

- **TFGARActivity.java** : Clase Activity principal de la aplicación. Será la encargada de manejar la recepción de datos de la cámara. Además, recibirá los eventos táctiles producidos en la pantalla y gestionará la escena en la que se encontrarán todos los objetos.
- **ARToolkitDrawer.java** : Clase responsable de dibujar los objetos. Está basada en min3d.
- **Util.java** : Clase de utilidades que contendrá un conjunto de métodos estáticos de ayuda a otras clases
- **ARStatus.java** : Enumeración de los estados en los que puede estar funcionando la aplicación.
- **MessageWhat.java** : Enumeración con los mensajes mostrados por la aplicación.

Las clases que nos encontramos en este paquete, organizadas por subpaquetes son las explicadas en los cuadros que siguen:

### 9.3.1. Paquete com.ivanlorenzo.gui.dialogs

- **DialogType.java** : Enumeración de los tipos de diálogos de la aplicación.

### 9.3.2. Paquete com.ivanlorenzo.gui.preferences

- **TFGPreferencesActivity.java** : Activity que gestiona las preferencias del usuario. Carga las preferencias del recurso *preferences.xml*.

## 9.4. Paquete com.ivanlorenzo.location

Esta paquete es el encargado de obtener nuestra localización actual y comprobar si existe alguna información que deba ser mostrada.

Las clases que nos encontramos en este paquete, organizadas por subpaquetes son las explicadas en los cuadros que siguen:

- **ARLocationListener.java** : Clase “sujeto observable” del patrón observer que, a su vez, implemente *LocationListener* y recibe cuándo se produce un cambio en la posición GPS y en caso de que esta posición contenga cercano una localización de los ficheros de configuración, notificará a todos los observadores que debe mostrarse la información asociada a dicha posición.
- **LocationData.java** : Clase que representa el estado de la localización. Se implementa como un Singleton ya que sólo queremos tener una única instancia.

### 9.4.1. Paquete com.ivanlorenzo.location.observer

- **IARObserver.java** : Interface genérica Observadora del patrón **Observer** [25]. Este patrón define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes.
- **IARObservable.java** : Interface genérica Observable del patrón **Observer** que tendrá un conjunto de observadores y notificará a éstos cuando se produzca algún cambio de estado.
- **ARObservable.java** : Clase abstracta que implementa la interface IARObservable. Contiene un elemento genérico y una lista de observadores genéricos.

## 9.5. Paquete com.ivanlorenzo.parser

Esta paquete es el encargado del proceso de renderización de objetos 3D.

- **IParser.java** : Interface para los parsers de objetos 3D basada en min3d.
- **AParser.java** : Clase abstracta Parser con la funcionalidad básica del proceso de parseo que implementa IParser.
- **Max3DSParser.java** : Parser específico para los objetos Max3DS.
- **MD2Parser.java** : Parser específico de los objetos MD2.

- **ObjParser.java** : Parser específico de los objetos obj.
- **ParseObjectData.java** : Clase que contiene la información de un objeto parseado.
- **Parser.java** : Clase factoría de Parsers. Especifica el tipo de parser y devuelve la clase especializada de parser.



# Capítulo 10

## Manual de internacionalización

Mediante el siguiente manual, lo que se pretende es aportar una explicación detallada de cuál ha de ser el proceso adecuado para la internacionalización del sistema.

Este manual estará dividido en dos apartados bien diferenciados:

- Internacionalización de la interfaz. 10.1.
- Internacionalización de los contenidos del modelo de navegación. 10.2.

### 10.1. Internacionalización de la interfaz

La internacionalización es el proceso de diseñar una aplicación para que pueda ser adaptada a diferentes idiomas o regiones, sin necesidad de cambios de ingeniería.

Permite que las aplicaciones admitan las codificaciones de datos necesarias para procesar información multilingüe, así como los formatos de moneda, calendarios, etc.

El nombre es una combinación de un código de referencia cultural de dos letras en minúscula asociado con un idioma, y un código de referencia cultural secundaria de dos letras en mayúscula asociado con un país o región. El código de referencia cultural secundaria sigue al código de referencia cultural, separados por un guión (-). Los ejemplos incluyen “ja-JP” para el japonés de Japón, “en-US” para el inglés de EE.UU. y “de-DE” para el alemán de Alemania (a diferencia de “de-AT” para el alemán de Austria). En los casos en los que no halla un código de idioma de dos letras disponible, se usa el código de tres letras que se deriva de ISO 639-2; por ejemplo, se usa el código de tres letras “div” para referencias culturales que utilizan el idioma dhivehi. Algunos nombres de referencias culturales tienen sufijos que especifican la escritura; por ejemplo, “-Cyril” especifica las letras del cirílico, “-Latn” especifica las letras del latín.

Ejemplos de referencias culturales asociados a unos idiomas:

- ar : Arabic

## 10. Manual de internacionalización

---

- bg : Bulgarian
- ca : Catalan
- zh-CHS : Chinese (Simplified)
- zh-CHT : Chinese (Traditional)
- cs : Czech
- da : Danish
- de : German
- el : Greek
- en : English
- es : Spanish
- fi : Finnish
- fr : French

En muchos entornos de desarrollo resulta ser una tarea “tediosa” porque muchas veces necesitamos utilizar librerías, scripts extras o archivos dedicados a indicar cuáles cadenas de texto serán las que se traducirán según la configuración de idioma que defina el usuario.

En Android es algo muy sencillo de implementar. Así que vamos explicar la internacionalización mediante un ejemplo: introducir un nuevo idioma.

En toda aplicación en Android deberá existir un directorio *res / values*. En este directorio habrá un fichero *strings.xml* que será el que contendrá las cadenas de texto.

Pero si queremos manejar distintos idiomas será necesario agregar un directorio llamado *res / values-[códigoDelIdioma]* con un archivo llamado *strings.xml* en el que se incluyen las mismas cadenas del otro archivo pero esta vez con sus traducciones respectivas al nuevo idioma.

Por tanto, si queremos añadir el idioma francés haremos lo siguiente:

- Crear la carpeta *res / values-fr*
- Crear un fichero *strings.xml*

Con estos pasos la interfaz de usuario ya soportaría un nuevo lenguaje. Tan sólo deberíamos cambiar el idioma en las propiedades del dispositivo móvil.

Ahora quedaría cambiar de lenguaje el contenido de la información.

## 10.2. Internacionalización de los contenidos

Para traducir los contenidos de la información habría que seguir los siguientes pasos:

- Vamos al archivo `strings.xml` creado y copiamos el contenido de otra carpeta *values*.
- Una vez copiado, sólo deberemos hacer la traducción de los textos al nuevo idioma, en nuestro ejemplo, el francés.

Con estos pasos ya estaría añadido un nuevo lenguaje y listo para su uso.



# Capítulo 11

## Manual del instalador

Mediante el siguiente, lo que se pretende es aportar una explicación detallada de cual ha de ser el proceso adecuado para la instalación y posterior explotación del sistema.

### 11.1. Requisitos

Para instalar la aplicación, deben cumplirse los siguientes requisitos:

- Disponer de un sistema operativo Android en el dispositivo móvil.
- Poseer conexión a Internet desde el teléfono para descargarnos la aplicación.

### 11.2. Instalación

En líneas generales para instalar una aplicación en Android se pueden seguir dos métodos:

1. Desde **Android Market** [28] o alguna tienda de aplicaciones similar: buscaremos las aplicaciones que queremos instalar en nuestro teléfono, para lo cual es necesario conectarse a Internet a través de él.

En el market encontraremos infinidad de aplicaciones y juegos, la mayoría gratuitas. Simplemente elegimos la que queramos instalar. Antes de instalarla podemos obtener más información sobre ella, los permisos que necesita e incluso comentarios y valoraciones de otros usuarios.

Nosotros descartamos esta opción ya que para tener una cuenta hay que registrarse y pagar una inscripción en Google. Usaremos, por tanto, la siguiente opción.

2. Otra opción es instalar directamente una aplicación desde un fichero **.apk**.
  - Para ello deberemos activar la casilla “Orígenes Desconocidos” en Ajustes - Aplicaciones tal como puede verse en la figura 11.1
  - El siguiente paso será descargar el fichero o bien pasarlo al fichero a través del ordenador o cual otro medio. El fichero puede encontrarse en <http://tfg.ivanlorenzo.es/apk/TFG.apk>

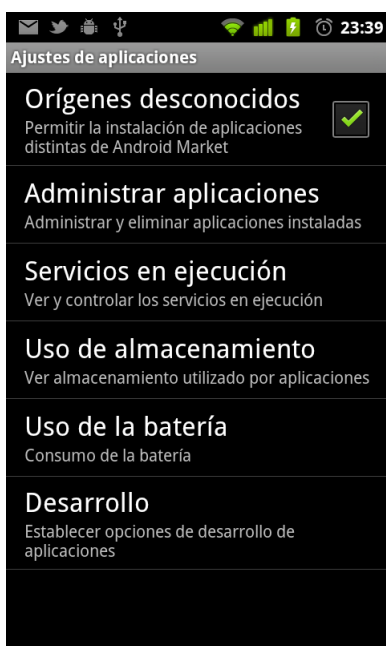


Figura 11.1: Activar “Orígenes desconocidos” en Ajustes del teléfono

- Una vez que disponemos del fichero en nuestro dispositivo móvil procederemos a ejecutarlo y se iniciará el instalador como vemos en la figura 11.2.

Como podemos observar nos pide consentimiento para instalarla y nos muestra los permisos que debemos aceptar para continuar.

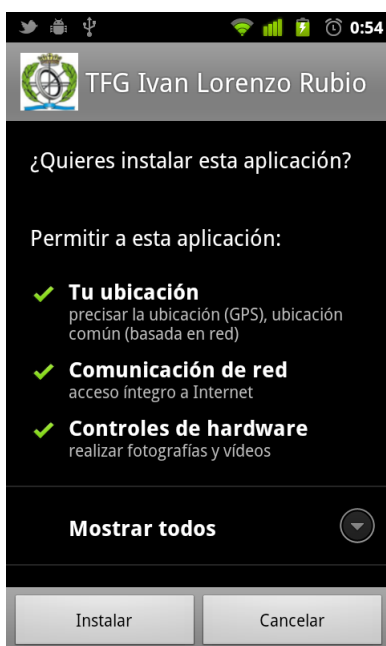


Figura 11.2: Instalador de la aplicación

- El instalador finalizará y con esto tendremos nuestra aplicación correctamente instalada en el dispositivo móvil y podremos ejecutarla 11.3.

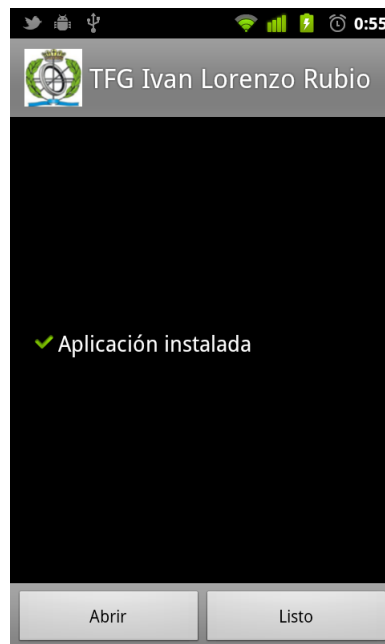


Figura 11.3: Instalación finalizada



# Capítulo 12

## Manual de usuario

### 12.1. Introducción

Llegados a este punto, es el momento de marcar un punto de inflexión y abandonar los tecnicismos, para centrarnos en lo que será una descripción analítica y totalmente práctica del funcionamiento general de la plataforma desde el punto de vista del usuario.

En las secciones que siguen, se irán describiendo una por una, todas las acciones que los usuarios pueden realizar con la herramienta.

### 12.2. Guía de uso

Para poder llevar a cabo todas las funciones de la plataforma describiremos las opciones que se nos ofrece.



Figura 12.1: Pantalla principal de la aplicación

## 12. Manual de usuario

---

Nada más concluir la instalación y arrancar la aplicación la cámara comienza a capturar imágenes. Se muestra en la figura 13.1

En la pantalla del dispositivo se muestran las imágenes capturadas por la cámara en tiempo real.

Si la cámara detecta algún patrón de los que están cargados o si nos encontramos en una posición geográfica cercana a una establecida en los ficheros de configuración, le superpone el objeto con la información asociada.

En la parte inferior de la pantalla podemos ver un espacio destinado a mostrar los comentarios de los objetos. Estos comentarios se visualizarán en el momento que se muestre dicho objeto.

Se podrán hacer una serie de transformaciones sobre el estado del objeto (ya sea moviéndolo o escalándolo) y los resultados se verán en dicha pantalla.

### 12.2.1. Acciones del usuario sobre los objetos

El usuario podrá realizar una serie de acciones sobre los objetos que esté visualizando:

- Mover un objeto
- Escalar un objeto

#### Mover un objeto

Para mover un objeto, el usuario deberá arrastrar el dedo sobre la pantalla hacia la posición que desee que se produzca el desplazamiento.



Figura 12.2: Desplazamiento de un objeto

### Escalar un objeto

El usuario también podrá escalar un objeto. Para ello deberá hacer un gesto típico en las interfaces táctiles: *pinch zoom*. Si desea aumentar el tamaño del objeto, habrá que colocar dos dedos sobre la pantalla y sin levantarlos, alejarlos. En caso contrario, si los dedos se acercan, el objeto disminuirá de tamaño.



Figura 12.3: Escalado de un objeto

#### 12.2.2. Opciones del menú de la aplicación

La aplicación dispondrá también de un menú en el que se podrán realizar una serie de acciones:



Figura 12.4: Opciones del menú de aplicación

- Establecer las preferencias de configuración

- Salir de la aplicación

Para lanzar este menú, casi todos los dispositivos móviles *Android* tienen una tecla específica. Estos elementos normalmente se muestran como un pop-up en la parte inferior de la pantalla cuando se presiona dicha tecla. En nuestra aplicación así sucederá.

### Establecer las preferencias de configuración

Los datos que se cargarán en la aplicación serán externos a ésta y, por tanto, configurables. Para ello el usuario puede modificarlos a través de la opción “*Configuración*”.

Habrán dos ficheros de configuración:

- **Fichero de patrones y localizaciones:** en él se especifican los patrones que reconocerá la cámara, así como las localizaciones que detectará la aplicación. Cada una de ellas tendrá asociada un objeto.
- **Fichero de objetos:** datos de los ficheros como son una descripción, el fichero del objeto tridimensional, una textura y un sonido.

En la aplicación podremos procesar otros ficheros de configuración y así cargar otros datos.

Para cada uno de ellos simplemente introduciremos la dirección en la que estén accesibles. La aplicación automáticamente buscará estos archivos e intentará cargar los datos contenidos en ellos.



Figura 12.5: Opciones de configuración

### Salir de la aplicación

El usuario para cerrar la aplicación deberá ir al Menú y elegir la opción “**Salir**”. Antes de realizarlo le preguntará si realmente quiere o no abandonar la aplicación.

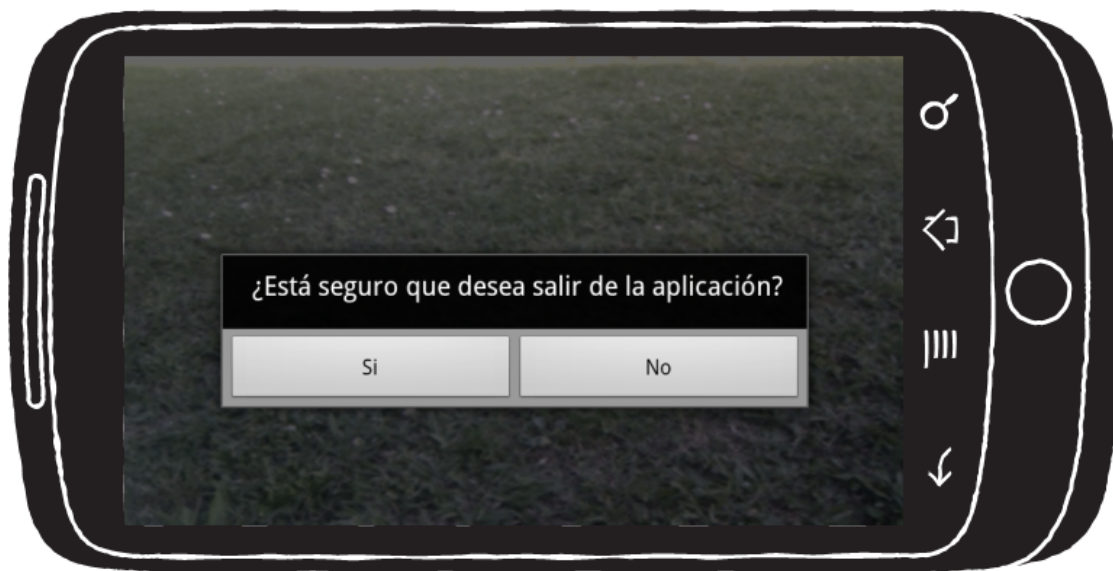


Figura 12.6: Salir de la aplicación



# Capítulo 13

## User's manual

### 13.1. Introduction

Having reached this point, it is now the moment to mark an inflection point and to abandon technical terms and concentrate on what will be an analytic and totally practical description of the general functioning of the platform from the user's point of view.

In the following sections, all the actions that can be done by the users with the tools will be described one by one.

### 13.2. Use guide

To carry out all functions of the platform we will describe the options that we offer.



Figura 13.1: Main application screen

## 13. User's manual

---

After completing the installation and start the application, the camera starts capturing images. Shown in figure 13.1

The display of device shows the images captured by the camera in real time.

If the camera detects a pattern of those charged in the application or whether we are in a geographic location close to some set in the configuration files, it will overlap the object with the associated information.

At the bottom of the screen can see a display space for comments of the objects. These comments will be displayed when the object is shown

It may make a series of transformations on the object's state (either moving or doing zoom) and the results will be displayed on that screen.

### 13.2.1. User actions on objects

The user can perform a number of actions on objects you are viewing:

- Move an object
- Scale an object

#### Move an object

To move an object, you must drag your finger across the screen to the desired position.



Figura 13.2: Moving an object

#### Scale an object

You can also scale an object. You should make a gesture typical of tactile interfaces: *pinch zoom*. To increase the size of the object, we must place two fingers on the screen without lifting, driving them away. Otherwise, if the fingers are close, the object will decrease in size.



Figura 13.3: Scaling an object

### 13.2.2. Menu options of the application

The application will also have a menu where you can perform a number of actions:



Figura 13.4: Menu options of the application

- Set configuration preferences
- Exit of the application

To launch the menu, almost all *Android* mobile devices have a dedicated key. These items are normally displayed as a pop-up in the bottom of the screen when you press that key.

### Set configuration preferences

The data loaded into the application shall be external to it, and therefore configurables. To do this the user can modify through the *Settings* option.

There are two configuration files:

- **File of patterns and locations:** it specifies patterns which will recognize the camera as well as the locations that will detect the application. Each have an associated object.
- **Objects'file:** data files such as a description, a three-dimensional object file, a texture and a sound.

The application can process other configuration files and so, load other data.

For each, simply must introduce an accessible web direction. automatically the application will search for these files and attempt to load the data contained in them.

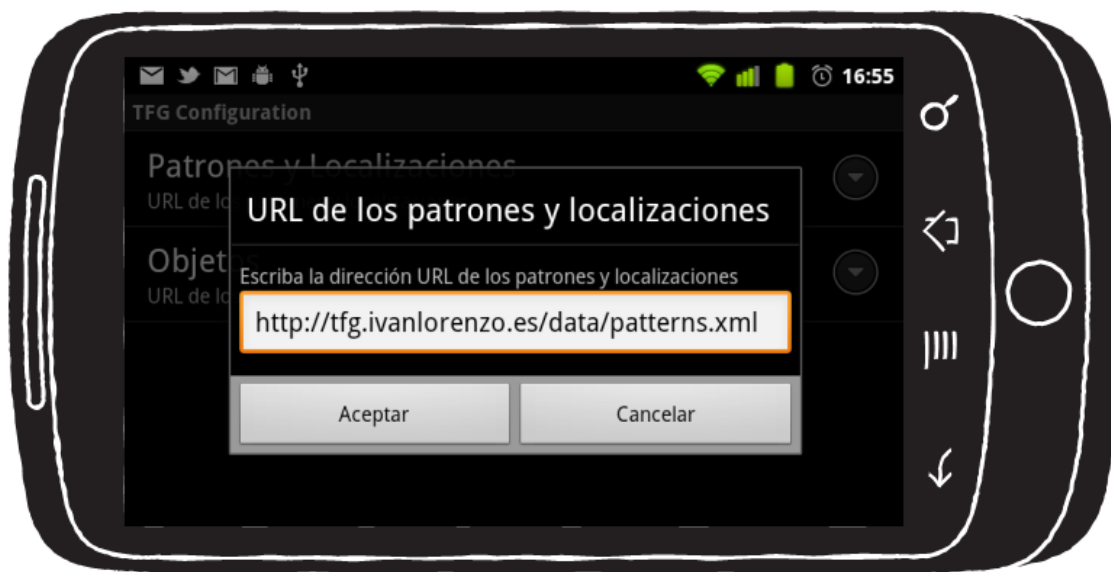


Figura 13.5: Configuration options

### Exit of the application

The user to close the application should go to Menu and choose “*Exit*”. Before, it ask if you really do or not to leave the application.

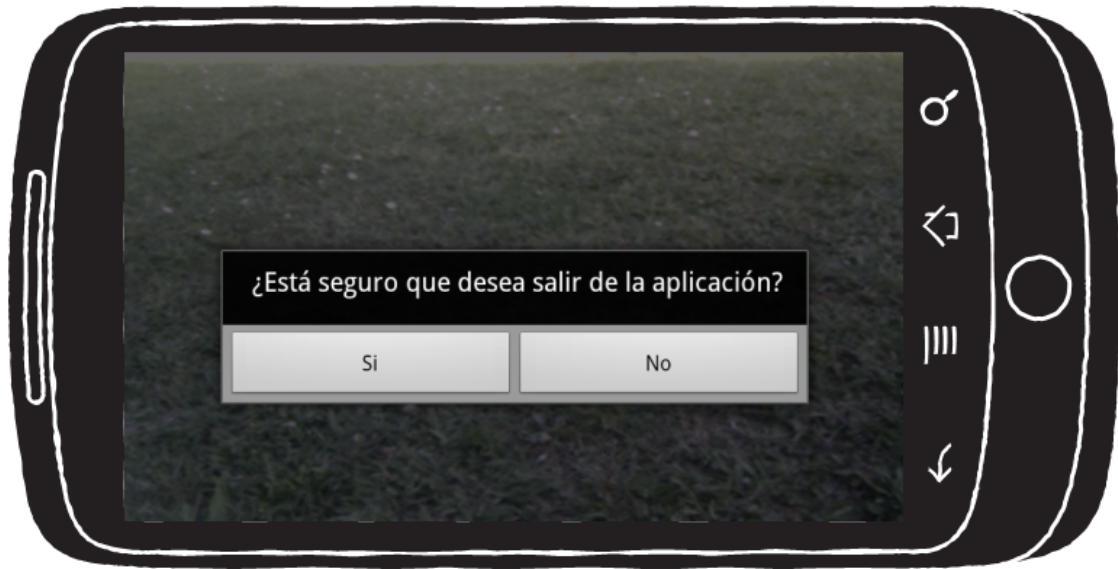


Figura 13.6: Exit of the application



## Capítulo 14

# Conclusiones y Líneas de Investigación Futuras

Al comienzo del proyecto, si bien el cómo se iba a conseguir no estaba del todo definido, el qué se pretendía alcanzar estaba claro. Se quería una plataforma que permitiera realizar una exposición visual, centrada en el ámbito educativo aunque no centrado exclusivamente en éste, ya que se podría aplicar en muchos otros escenarios.

Con este proyecto nace una plataforma basada en realidad aumentada, una tecnología que añade información extra al mundo en el que vivimos; no nos aísla del mundo y nos sumerge en otro como en el caso de la realidad virtual.

La plataforma proporciona una solución completa en la que se puede decidir qué objetos, cómo y donde mostrarlos. La información que se obtiene de cada objeto aumentará considerablemente a la que teníamos inicialmente.

En nuestras exposiciones, a la vez que se visualiza el objeto en 3D, se recibe un comentario de manera visual y sonora.

Otra de las ventajas que ofrece la plataforma es que en innumerables ocasiones muchos de los objetos no se pueden ver físicamente. Con esta plataforma se podrá paliar la falta de objetos físicos en un lugar o incluso se podrá mostrar cualquier otro que se estime oportuno.

Todas las exposiciones se generan dinámicamente debido a la independencia de la plataforma con los datos que componen cada una de ellas.

De este modo, crear un recorrido mostrando ciertos objetos se haría de forma muy sencilla modificando los ficheros de configuración.

Gracias a nuestro sistema se puede disponer de un sistema interactivo de aprendizaje que transformará a los usuarios de meros espectadores pasivos en actores y jugadores activos. Los usuarios se involucrarán más en la exposición haciéndola más entretenida y más fructífera.

## 14. Conclusiones y Líneas de Investigación Futuras

---

La forma de interacción de los visitantes con las obras se realizará a través de dispositivos móviles que se comunicarán con un servidor por medio de redes inalámbricas.

Se podrá, entre otras cosas, modificar ciertas propiedades de los objetos como son la posición o la escala..

Esta información, cuando se recibe en los dispositivos móviles, se recibe de forma visual y sonora.

La solución basada en las técnicas de Realidad Aumentada permite evitar el uso de hardware costoso que podría ser un obstáculo serio para la mayoría de los usuarios. La gran mayoría de la población ya tiene un dispositivo móvil con estas características o acceso a uno de ellos.

Además, al utilizar dispositivos móviles se permite una mayor movilidad e independencia a los usuarios.

Por lo tanto, cualquier persona, institución o empresa (incluso con recursos financieros limitados) podrían adaptar la plataforma y aprovecharse de los logros de la Realidad Aumentada.

La mayor dificultad en la investigación para la realización del proyecto se debe al uso de una tecnología nueva, poco usual y de la que hasta el momento no se ha investigado todo lo suficiente para las grandes posibilidades que ofrece.

Éste es el momento de reflexionar sobre las tecnologías que se aplicaron para la consecución de los objetivos, todas ellas totalmente actuales y, de las que o bien se carece o es muy escasa la formación académica de la que se dispone, habiendo por tanto detrás del proyecto una labor importante tanto de investigación como de aprendizaje.

En cuanto a los problemas encontrados en el desarrollo cabe destacar la renderización de archivos con modelos 3D en un dispositivo móvil con *Android*. Para solventar este problema se han utilizado librerías que hemos modificado para conseguir nuestro objetivo. Algunas de las librerías son *NyARToolkit* y *min3d*.

Otro asunto complicado en el desarrollo ha sido la comunicación de los clientes con los servidores en los que se encuentren los archivos de configuración.

Y por último, también ha sido laborioso implementar los gestos para manejar los objetos a través de la interfaz táctil.

Una vez concluido este proyecto, quedan abiertas varias vías para seguir trabajando sobre el mismo, con el objetivo de mejorarlo.

---

A continuación, se detalla una lista de mejoras que se podrían incorporar en futuras versiones de la plataforma:

1. **Portal web:** En este portal los usuarios podrían subir exposiciones completas, es decir, todos los elementos necesarios para realizarlas, como son los patrones o las posiciones, los objetos tridimensionales, la información de los mismos, etc. . . De esta forma los visitantes podrán elegir qué ver desde cualquier lugar.
2. **Un editor para las visitas:** En nuestra aplicación para hacer un cambio sobre la exposición, se debería de editar el fichero XML correspondiente, para luego ver la información cambiada. Una buena ampliación, sería la posibilidad de tener un editor para generar estos cambios. En este editor se podría modificar los patrones y localizaciones, los objetos, así como la información de éstos.
3. **Gestor de comentarios:** Una posible ampliación sería que de cada objeto, los usuarios pudiesen dejar comentarios opinando sobre ellos. Así los futuros usuarios podrían ver los comentarios depositados en cada objeto hasta el momento.
4. **Investigar en otros tipos de reconocimiento de imágenes:** Se ha pensado en aplicar en un futuro otro tipo de reconocimiento de imágenes como pueden ser el reconocimiento facial o reconocimiento de gestos.



# Bibliografía

- [1] Woodrow Barfield and Thomas Caudell. *Fundamentals of Wearable Computers and Augmented Reality*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 2000.
- [2] Jerome S Bruner. *Desarrollo cognitivo y educación*. Morata, Madrid, 1988.
- [3] P. Abrahamsson. Keynote: Mobile software development - the business opportunity of today. 2005.
- [4] Tecnologías 3g. <http://www.malavida.com/blog/70/que-es-3g-tecnologia-movil-3g>.
- [5] Magicbook. <http://www.hitlabnz.org/index.php/research/augmented-reality?view=project&task=show&id=54>.
- [6] Arise. <http://www.arise-project.org/>.
- [7] Learnar. <http://www.learnar.org/>.
- [8] Scimorph. <http://scimorph.greatfridays.com/>.
- [9] Zooburst. <http://www.zooburst.com/>.
- [10] Arsights. <http://www.arsights.com/>.
- [11] Wikitude. <http://www.wikitude.com/en/>.
- [12] Layar. <http://www.layar.com/>.
- [13] Mtra. Rocío Ruiz Rodarte. Museografía virtual para el museo de sitio de calakmul. Día Cudi de Mundos Virtuales.
- [14] Rafal Wojciechowski, Krzysztof Walczak, Martin White, and Wojciech Cellary. Building virtual and augmented reality museum exhibitions. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, pages 135–144, New York, NY, USA, 2004. ACM Press.
- [15] Vassilios Vlahakis, Nikolaos Ioannidis, John Karigiannis, Manolis Tsotros, Michael Gounaris, Didier Stricker, Tim Gleue, Patrick Daehne, and Luis Almeida. Archeo-guide: An augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications*, 22(5):52–60, 2002.

- [16] Daniel Pletinckx, Dirk Callebaut, Ann E. Killebrew, and Neil A. Silberman. Virtual-reality heritage presentation at ename. *IEEE MultiMedia*, 7(2):45–48, 2000.
- [17] W. Scott Means Elliotte Rusty Harold. *XML imprescindible*. Anaya, 2004.
- [18] Xml schema por el w3c. <http://www.w3.org/XML/Schema>.
- [19] Obtaining user location in android. <http://developer.android.com/guide/topics/location/obtaining-user-location.html>.
- [20] Librería min3d de android. <http://code.google.com/p/min3d/>.
- [21] Dave Shreiner. *OpenGL programming guide*. Addison-Wesley, Upper Saddle River, NJ, 2006.
- [22] Mediaplayer de android. <http://developer.android.com/reference/android/media/MediaPlayer.html>.
- [23] Nyartoolkit project. <http://nyatla.jp/nyartoolkit/wp/>.
- [24] Application resources en android. <http://developer.android.com/guide/topics/resources/index.html>.
- [25] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [26] Junit 4. <http://www.junit.org/>.
- [27] Junit en android. [http://developer.android.com/guide/topics/testing/testing\\_android.html](http://developer.android.com/guide/topics/testing/testing_android.html).
- [28] Android market. <https://market.android.com/?hl=es>.
- [29] Code examples for working with touch input in android. <http://code.google.com/p/android-touchexample/source/browse/trunk/src/com/example/android/touchexample/TouchExampleView.java>.
- [30] Manual de latex. [http://es.wikibooks.org/wiki/Manual\\_de\\_LaTeX](http://es.wikibooks.org/wiki/Manual_de_LaTeX).
- [31] Tratamiento de XML en Android con DOM. <http://www.sgoliver.net/blog/?p=1588>.

# Apéndice A

## Tecnologías

En este resumen, se realizará una breve descripción de las tecnologías utilizadas para el desarrollo de éste proyecto.

### A.1. Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, **un núcleo de sistema operativo libre, gratuito y multiplataforma.**

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.

Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hacen que una de las cosas más importantes de este sistema operativo sea la cantidad de aplicaciones disponibles, que extienden casi sin límites la experiencia del usuario.

#### A.1.1. Historia

Android era un sistema operativo para móviles prácticamente desconocido hasta que en 2005 Google lo compró. Hasta noviembre de 2007 sólo hubo rumores, pero en esa fecha se lanzó la **Open Handset Alliance**, que agrupaba a muchos fabricantes de teléfonos móviles, chipsets y Google y se proporcionó la primera versión de Android, junto con el SDK para que los programadores empezaran a crear sus aplicaciones para este sistema.

Aunque los inicios fueran un poco lentos, debido a que se lanzó antes el sistema operativo que el primer móvil, rápidamente se ha colocado como el sistema operativo de móviles más vendido del mundo, situación que se alcanzó en el último trimestre de 2010.

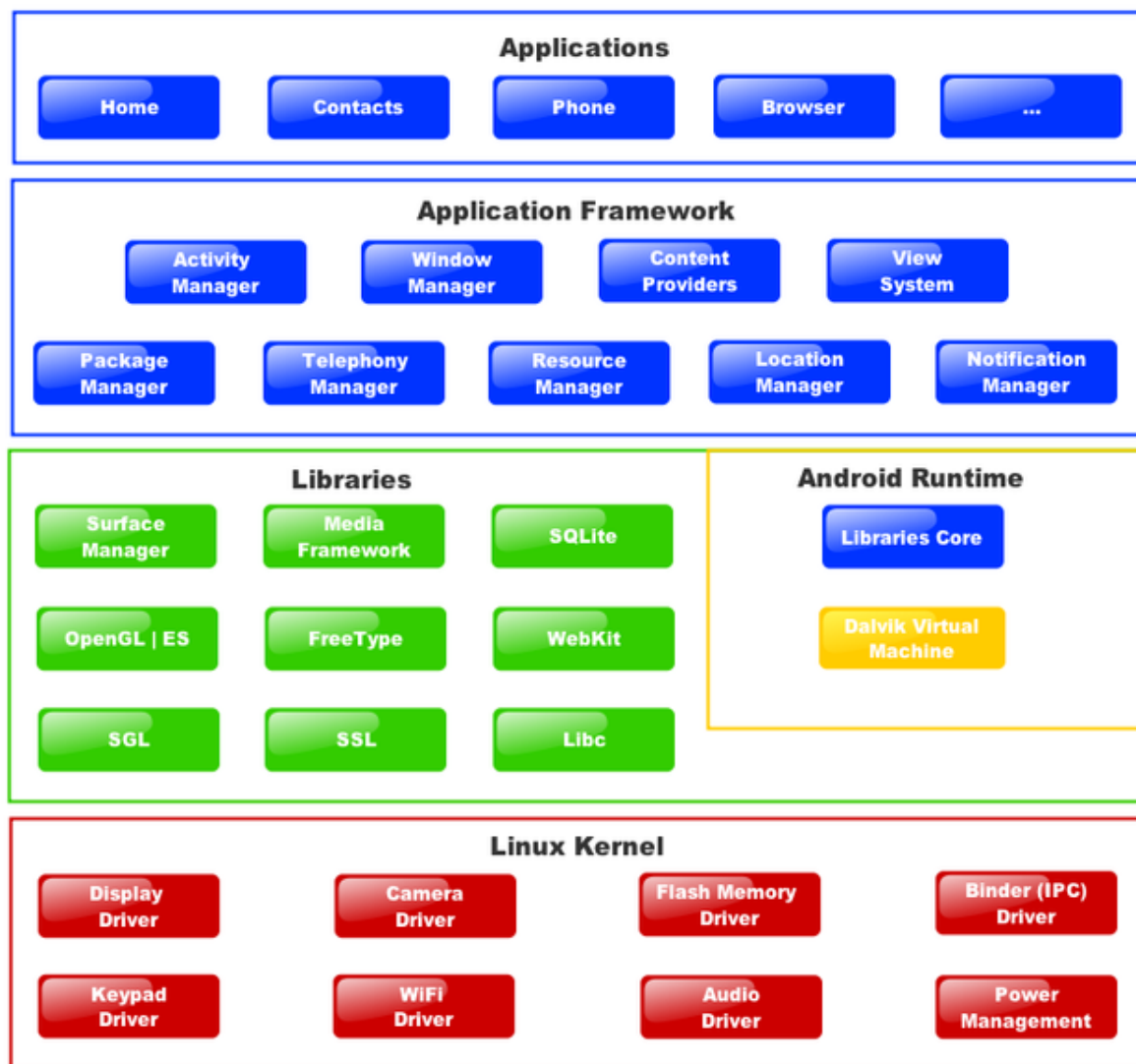


Figura A.1: Sistema operativo Android

En febrero de 2011 se anunció la versión 3.0 de Android, llamada con nombre en clave *Honeycomb*, que está optimizado para tabletas en lugar de teléfonos móviles. Por tanto Android ha trascendido los teléfonos móviles para trascender a dispositivos más grandes.

Una evolución de esta versión ha sido la 4.0 de Android lanzada en marzo de 2012, llamada *Ice Cream Sandwich* que unifica el uso en cualquier dispositivo, tanto en teléfonos, tablets, televisores, netbooks, etc. . .

Al final podemos resumir que Android es la forma de afrontar la telefonía móvil por parte de Google, que entiende estos dispositivos como una forma de estar conectado constantemente a Internet. Las aplicaciones de Google que vienen preinstaladas en el teléfono, permiten acceder a los servicios de Google de forma muy integrada, aparte de la

tienda de aplicaciones **Google Play** que posee más de 400.000 (de las cuales, dos tercios son gratuitas) y permite instalarlas de una forma muy sencilla.

## A.2. JAVA

**Java** es un lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems* a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

*Sun Microsystems* proporciona una implementación *GNU General Public License* de un compilador Java y una máquina virtual Java, conforme a las especificaciones del *Java Community Process*, aunque la biblioteca de clases que se requiere para ejecutar los programas Java no es software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia *GNU GPL*, de acuerdo con las especificaciones del *Java Community Process*, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

## A.3. NyArtoolkit

NyArtoolkit [23] es una variante de ARToolKit. Es una biblioteca que permite la creación de aplicaciones de realidad aumentada, en las que se sobrepone imágenes virtuales al mundo real.

Para ello, utiliza las capacidades de seguimiento de vídeo, con el fin de calcular, en tiempo real, la posición de la cámara y la orientación relativa a la posición de los marcadores físicos. Una vez que la posición de la cámara real se sabe, la cámara virtual se pueden colocar en el mismo punto y modelos 3d son sobrepuestos exactamente sobre el marcador real. Así ARToolKit resuelve dos de los principales problemas en la realidad aumentada, el seguimiento de punto de vista y la interacción objeto virtual.

## A.4. min3D

min3D [20] es una librería para Android que usa Java y OpenGL y es compatible con dispositivos que tengan Android v1.5/OpenGL ES 1.0 y superior.

Se asimila mucho a la API de OpenGL así que es ideal para conocer ésta y a la vez proporcionar una funcionalidad similar en una librería de clases orientada a objetos.

En la plataforma se utiliza para la representación de los objetos tridimensionales.

### A.5. Tecnologías XML

#### A.5.1. XML

Esta tecnología [17] basada en etiquetas permite un medio homogéneo, sencillo y claro de representación e intercambio de la información. Entorno a esta tecnología se desarrollan otras muchas, de tal forma que resulta totalmente viable la construcción de sitios web potentes y totalmente funcionales utilizando únicamente este lenguaje de etiquetado.

Utilizamos XML por su simplicidad, potencia, herramientas existentes en Java para procesar el documento, extensibilidad de cara a nuevas versiones, el soporte de las herramientas para crear una interfaz gráfica, etcétera...

Utilizando esta tecnología, la plataforma se convierte en un sistema fácil de entender, fácil de modificar y fácil de recordar. En los ficheros de xml, se almacena toda la información sobre la configuración y las imágenes de la exposición.

#### A.5.2. XML Schema

XML Schema es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el World Wide Web Consortium (W3C) y alcanzó el nivel de recomendación en mayo de 2001.

No nos bastará un documento bien construido en XML para nuestro propósito, ya que este podría tener diferentes estructuras, queremos que nuestros ficheros tengan una estructura perfectamente definida. Para ello utilizaremos un XSD para validar los ficheros.

#### A.5.3. DOM

El DOM [31] (Modelo de Objeto de Documento) es el medio estándar desarrollador por el W3C por el cual acceder y manipular un documento XML. Un buen DOM debe de ser completo, permitiendo la reconstrucción del documento completo desde el propio modelo y permitir el acceso a cualquiera de sus partes, además de métodos que permitan manipular, añadir y eliminar en el documento original.

Las APIs DOM ya sean escritas en Visual Basic, C o Java expondrán al programador una misma serie de métodos y atributos. De las opciones posibles nosotros vamos a utilizar la implementación Java del propio consorcio W3C.

En nuestro caso vamos a poblar esa implementación DOM con un analizador SAX, y luego solamente accederemos al contenido del documento para crear los objetos que definen la exposición.

## A.6. L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X[30] es un sistema pensado para la edición de textos científicos. Concretamente se trata de un paquete de macros que permite componer e imprimir un documento con una gran calidad basándose, para ello, en una serie de patrones previamente definidos. Además permite su transformación al formato PDF, formato destinado a la impresión por antonomasia.

L<sup>A</sup>T<sub>E</sub>X, una vez dominado, nos permite realizar publicaciones con un acabado final espectacular.



# Documentación

## Classes

- `com.ivanlorenzo.gui.preferences.TFGPreferencesActivity` (in B.11.1, page 272)
- `com.ivanlorenzo.gui.TFGARActivity` (in B.4.13, page 227)
  - `android.os.Handler`
    - `com.ivanlorenzo.gui.TFGARActivity.MainHandler` (in B.4.14, page 241)
  - `android.view.ScaleGestureDetector.SimpleOnScaleGestureListener`
    - `com.ivanlorenzo.gui.TFGARActivity.ScaleListener` (in B.4.16, page 243)
  - `com.ivanlorenzo.connection.ConnectionStatus` (in B.10.1, page 271)
  - `com.ivanlorenzo.data.CompilerGenerator` (in B.9.2, page 260)
  - `com.ivanlorenzo.data.GeoLocation` (in B.9.3, page 261)
  - `com.ivanlorenzo.data.KnowledgeBase` (in B.9.4, page 263)
  - `com.ivanlorenzo.data.Pattern` (in B.9.6, page 268)
  - `com.ivanlorenzo.data.XMLCompiler` (in B.9.7, page 270)
    - `com.ivanlorenzo.data.AugmentedRealityXMLCompiler` (in B.9.1, page 258)
    - `com.ivanlorenzo.data.ObjectsXMLCompiler` (in B.9.5, page 266)
  - `com.ivanlorenzo.data.objects.Object3D` (in B.5.3, page 247)
    - `com.ivanlorenzo.data.objects.MD2Object` (in B.5.2, page 246)
    - `com.ivanlorenzo.data.objects.Max3DSObject` (in B.5.1, page 245)
    - `com.ivanlorenzo.data.objects.OBJObject` (in B.5.6, page 252)
  - `com.ivanlorenzo.data.objects.Object3D.File3D` (in B.5.4, page 250)
  - `com.ivanlorenzo.data.objects.Object3DCreator` (in B.5.5, page 251)

- `com.ivanlorenzo.data.objects.audio.Audio` (in B.1.1, page 181)
  - `com.ivanlorenzo.data.objects.audio.LocalAudio` (in B.1.2, page 182)
  - `com.ivanlorenzo.data.objects.audio.RemoteAudio` (in B.1.4, page 185)
- `com.ivanlorenzo.gui.ARToolkitDrawer` (in B.4.2, page 216)
- `com.ivanlorenzo.gui.R` (in B.4.4, page 220)
- `com.ivanlorenzo.gui.R.attr` (in B.4.5, page 220)
- `com.ivanlorenzo.gui.R.drawable` (in B.4.6, page 221)
- `com.ivanlorenzo.gui.R.id` (in B.4.7, page 221)
- `com.ivanlorenzo.gui.R.layout` (in B.4.8, page 222)
- `com.ivanlorenzo.gui.R.menu` (in B.4.9, page 223)
- `com.ivanlorenzo.gui.R.raw` (in B.4.10, page 223)
- `com.ivanlorenzo.gui.R.string` (in B.4.11, page 224)
- `com.ivanlorenzo.gui.R.xml` (in B.4.12, page 226)
- `com.ivanlorenzo.gui.TFGARActivity.PreviewCallback` (in B.4.15, page 242)
- `com.ivanlorenzo.gui.Util` (in B.4.17, page 243)
- `com.ivanlorenzo.location.LocationData` (in B.7.2, page 255)
- `com.ivanlorenzo.location.observer.ARObservable` (in B.3.3, page 213)
  - `com.ivanlorenzo.data.objects.audio.MediaPlayerAdapter` (in B.1.3, page 183)
  - `com.ivanlorenzo.location.ARLocationListener` (in B.7.1, page 254)
- `com.ivanlorenzo.parser.AParser` (in B.2.2, page 187)
  - `com.ivanlorenzo.parser.MD2Parser` (in B.2.8, page 200)
  - `com.ivanlorenzo.parser.Max3DSParser` (in B.2.7, page 196)
  - `com.ivanlorenzo.parser.ObjParser` (in B.2.10, page 204)
- `com.ivanlorenzo.parser.AParser.BitmapAsset` (in B.2.3, page 191)
- `com.ivanlorenzo.parser.AParser.Material` (in B.2.4, page 192)
- `com.ivanlorenzo.parser.AParser.TextureAtlas` (in B.2.5, page 193)
- `com.ivanlorenzo.parser.AParser.TextureAtlas.BitmapHeightComparer` (in B.2.6, page 195)
- `com.ivanlorenzo.parser.MD2Parser.MD2Header` (in B.2.9, page 202)
- `com.ivanlorenzo.parser.ParseObjectData` (in B.2.12, page 207)
- `com.ivanlorenzo.parser.Parser` (in B.2.13, page 209)
- `java.lang.Enum`
  - `com.ivanlorenzo.gui.ARStatus` (in B.4.1, page 214)
  - `com.ivanlorenzo.gui.MessageWhat` (in B.4.3, page 219)
  - `com.ivanlorenzo.gui.dialogs.DialogType` (in B.8.1, page 257)
  - `com.ivanlorenzo.parser.Parser.Type` (in B.2.14, page 210)
- `min3d.parser.ParseObjectFace`
  - `com.ivanlorenzo.parser.ObjParser.ObjFace` (in B.2.11, page 206)

## Interfaces

- `com.ivanlorenzo.location.observer.IARObservable` (in B.3.1, page 212)
- `com.ivanlorenzo.location.observer.IARObserver` (in B.3.2, page 212)
- `com.ivanlorenzo.parser.IParser` (in B.2.1, page 187)

## Excepciones

- java.lang.Object
  - java.lang.Throwable
    - java.lang.Exception
      - com.ivanlorenzo.data.exceptions.LoadXMLException
      -

com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException

## B.1. Paquete com.ivanlorenzo.data.objects.audio

### B.1.1. Clase Audio

---

Clase abstracta Audio que tendrá un método abstracto que devolverá una objeto de la clase MediaPlayerAdapter

#### Declaración

```
public abstract class Audio
extends java.lang.Object
```

#### Field summary

**id** Identificador del objeto al que pertenece el audio

#### Constructor summary

**Audio()**

#### Method summary

**getId()** Devuelve el identificador del objeto

**getMediaPlayer(Context)** Método que devuelve la instancia de la clase MediaPlayerAdapter

#### Atributos

---

- **int id**
  - Identificador del objeto al que pertenece el audio

#### Constructores

---

- *Audio*  
**public Audio( )**

## Métodos

---

- *getId*  
`public int getId( )`
  - **Description**  
Devuelve el identificador del objeto
  - **Returns** – El identificador del objeto al que pertenece el audio
- *getMediaPlayer*  
`public abstract MediaPlayerAdapter getMediaPlayer( android.content.Context ctx )`
  - **Description**  
Método que devuelve la instancia de la clase MediaPlayerAdapter
  - **Parameters**
    - `ctx` – Contexto de la aplicación
  - **Returns** – Un objeto MediaPlayerAdapter

### B.1.2. Clase LocalAudio

---

Clase que hereda de Audio. Se utilizará para representar audios que estén en el dispositivo

#### Declaración

```
public class LocalAudio
extends com.ivanlorenzo.data.objects.audio.Audio (in B.1.1, page 181)
```

#### Constructor summary

**LocalAudio(int)** Constructor

#### Method summary

**getMediaPlayer(Context)**

#### Constructores

---

- *LocalAudio*  
`public LocalAudio( int id )`
  - **Description**  
Constructor
  - **Parameters**
    - `id` – Identificador del objeto

---

## Métodos

---

- *getMediaPlayer*  
`public abstract MediaPlayerAdapter getMediaPlayer( android.content.Context ctx )`
  - **Description copied from Audio (in B.1.1, page 181)**  
Método que devuelve la instancia de la clase MediaPlayerAdapter
  - **Parameters**
    - `ctx` – Contexto de la aplicación
  - **Returns** – Un objeto MediaPlayerAdapter

Miembros heredados de la clase `com.ivanlorenzo.data.objects.audio.Audio` (in B.1.1, page 181)

---

- `public int getId( )`
- `public abstract MediaPlayerAdapter getMediaPlayer( android.content.Context ctx )`
- `id`

### B.1.3. Clase MediaPlayerAdapter

---

Clase adaptadora del patrón Adapter que adapta la clase MediaPlayer y además es un sujeto observable ya que cuando finalice el audio deberá avisar a sus observadores para que realicen las acciones oportunas

#### Declaración

```
public class MediaPlayerAdapter
extends com.ivanlorenzo.location.observer.ARObservable (in B.3.3, page 213)
```

#### Field summary

**isPlaying** Variable que nos indicará si está sonando o no  
**loop** Variable que nos indica si está marcado como repetición  
**mediaPlayer** MediaPlayer que vamos a adaptar  
**resID** Identificador del player

#### Constructor summary

**MediaPlayerAdapter(Context, int)** Constructor de la clase en el que se crea un objeto MediaPlayer y gestiona el evento onCompletion, que informará cuando finalice el audio  
**MediaPlayerAdapter(Context, String, int)** Constructor de la clase usado para reproducir un audio remoto, en el que se crea un objeto de la clase MediaPlayer y se le establece un datasource que será la url con el fichero a reproducir.

## Method summary

**getResID()** Devuelve el identificador del audio  
**loop()** Método que pone el reproductor en modo repetición  
**notifyObservers()**  
**play()** Método que inicia la reproducción del audio  
**release()** Libera el reproductor  
**stop()** Método que detiene la reproducción del audio

## Atributos

---

- `private android.media.MediaPlayer mediaPlayer`
  - MediaPlayer que vamos a adaptar
- `private int resID`
  - Identificador del player
- `private boolean isPlaying`
  - Variable que nos indicará si está sonando o no
- `private boolean loop`
  - Variable que nos indica si está marcado como repetición

## Constructores

---

- *MediaPlayerAdapter*  
`public MediaPlayerAdapter( android.content.Context ctx, int resID )`
  - **Description**  
Constructor de la clase en el que se crea un objeto MediaPlayer y gestiona el evento `onCompletion`, que informará cuando finalice el audio
  - **Parameters**
    - `ctx` – Contexto de la aplicación
    - `resID` – Identificador del audio
- *MediaPlayerAdapter*  
`public MediaPlayerAdapter( android.content.Context ctx, java.lang.String url, int id )`
  - **Description**  
Constructor de la clase usado para reproducir un audio remoto, en el que se crea un objeto de la clase MediaPlayer y se le establece un datasource que será la url con el fichero a reproducir. También gestiona el evento `onCompletion` que informará cuando finalice el audio.
  - **Parameters**
    - `ctx` – Contexto de la aplicación
    - `url` – Dirección del fichero a reproducir
    - `id` – Identificador del audio

---

## Métodos

---

- *getResID*  
`public int getResID( )`
  - **Description**  
Devuelve el identificador del audio
  - **Returns** – El identificador
- *loop*  
`public synchronized void loop( )`
  - **Description**  
Método que pone el reproductor en modo repetición
- *notifyObservers*  
`void notifyObservers( )`
  - **Description copied from com.ivanlorenzo.location.observer.IARObservable (in B.3.1, page 212)**  
Cuando se produzca un cambio en él, notificará a todos los observadores a través de este método
- *play*  
`public synchronized void play( )`
  - **Description**  
Método que inicia la reproducción del audio
- *release*  
`public void release( )`
  - **Description**  
Libera el reproductor
- *stop*  
`public synchronized void stop( )`
  - **Description**  
Método que detiene la reproducción del audio

Miembros heredados de la clase `com.ivanlorenzo.location.observer.ARObservable` (in B.3.3, page 213)

---

- `public void addObserver( IARObserver observer )`
- `protected data`
- `public void notifyObservers( )`
- `protected observers`
- `public void removeObserver( IARObserver observer )`

### B.1.4. Clase RemoteAudio

---

Representa un audio localizado en una ubicación remota. Hereda de la clase Audio

## Declaración

```
public class RemoteAudio
extends com.ivanlorenzo.data.objects.audio.Audio (in B.1.1, page 181)
```

## Field summary

**path** URL donde se encuentra el fichero remotamente

## Constructor summary

**RemoteAudio(String, int)** Constructor

## Method summary

**getMediaPlayer(Context)**

## Atributos

---

- `private java.lang.String path`
  - URL donde se encuentra el fichero remotamente

## Constructores

---

- *RemoteAudio*  
`public RemoteAudio( java.lang.String path, int id )`
  - **Description**  
Constructor
  - **Parameters**
    - `path` – URL remota
    - `id` – Identificador del objeto que tiene asociado el audio

## Métodos

---

- *getMediaPlayer*  
`public abstract MediaPlayerAdapter getMediaPlayer( android.content.Context ctx )`
  - **Description copied from Audio (in B.1.1, page 181)**  
Método que devuelve la instancia de la clase MediaPlayerAdapter
  - **Parameters**
    - `ctx` – Contexto de la aplicación
  - **Returns** – Un objeto MediaPlayerAdapter

Miembros heredados de la clase `com.ivanlorenzo.data.objects.audio.Audio` (in B.1.1, page 181)

---

- `public int getId( )`
- `public abstract MediaPlayerAdapter getMediaPlayer( android.content.Context ctx )`
- `id`

## B.2. Paquete com.ivanlorenzo.parser

### B.2.1. *Interfaz* IParser

---

Interface para los parsers de objetos 3D basada en min3d

#### Declaración

```
public interface IParser
```

#### Method summary

- `getParsedAnimationObject()` Devuelve el objeto animado parseado
- `getParsedObject()` Devuelve el objeto parseado
- `parse()` Empieza a parsear el objeto 3d

#### Métodos

---

- *getParsedAnimationObject*  
`min3d.animation.AnimationObject3d getParsedAnimationObject( )`
  - **Description**  
Devuelve el objeto animado parseado
  - **Returns** – El objeto animado
- *getParsedObject*  
`min3d.core.Object3dContainer getParsedObject( )`
  - **Description**  
Devuelve el objeto parseado
  - **Returns** – El objeto parseado
- *parse*  
`void parse( )`
  - **Description**  
Empieza a parsear el objeto 3d

### B.2.2. *Clase* AParser

---

Clase abstracta Parser con la funcionalidad básica del proceso de parseo.

## Declaración

```
public abstract class AParser
extends java.lang.Object
implements IParser
```

## Field summary

**co**  
**currentMaterialKey**  
**fileIn**  
**firstObject**  
**generateMipMap**  
**materialMap**  
**normals**  
**packageID**  
**parseObjects**  
**remoteLocation**  
**resourceID**  
**resources**  
**texCoords**  
**textureAtlas**  
**url**  
**vertices**

## Constructor summary

**AParser()** Constructor básico  
**AParser(Resources, String, Boolean)** Constructor del parser que recibe el recurso específico  
**AParser(String)** Constructor en el que se le pasa la dirección url del objeto a parsear

## Method summary

**cleanup()** Borra todas las estructuras de datos  
**getParsedAnimationObject()** Devuelve el objeto animado parseado.  
**getParsedObject()** Devuelve el objeto parseado.  
**parse()** Realiza el parseo del objeto.  
**readFloat(InputStream)** Lee un número flotante de caracteres de un InputStream  
**readInt(InputStream)** Lee un entero de un InputStream  
**readShort(InputStream)** Lee un short de caracteres de un InputStream  
**readString(InputStream)** Lee una cadena de caracteres de un InputStream

## Atributos

---

- protected android.content.res.Resources **resources**
- protected java.lang.String **resourceID**
- protected java.lang.String **packageID**
- protected java.lang.String **currentMaterialKey**
- protected java.util.ArrayList **parseObjects**
- protected ParseObjectData **co**
- protected boolean **firstObject**
- protected AParser.TextureAtlas **textureAtlas**
- protected java.util.ArrayList **vertices**
- protected java.util.ArrayList **texCoords**
- protected java.util.ArrayList **normals**
- protected boolean **generateMipMap**
- protected java.util.HashMap **materialMap**
- protected java.io.InputStream **fileIn**
- protected boolean **remoteLocation**
- protected java.lang.String **url**

## Constructores

---

- *AParser*  
**public AParser( )**
  - **Description**  
Constructor básico
- *AParser*  
**public AParser( android.content.res.Resources resources, java.lang.String resourceID, java.lang.Boolean generateMipMap )**
  - **Description**  
Constructor del parser que recibe el recurso específico
  - **Parameters**
    - **resources** – Los recursos de la aplicación
    - **resourceID** – El identificador del recurso a cargar
    - **generateMipMap** –
- *AParser*  
**public AParser( java.lang.String url )**
  - **Description**  
Constructor en el que se le pasa la dirección url del objeto a parsear
  - **Parameters**
    - **url** – Dirección del objeto a parsear

## Métodos

---

- *cleanup*  
`protected void cleanup( )`
  - **Description**  
Borra todas las estructuras de datos
- *getParsedAnimationObject*  
`public min3d.animation.AnimationObject3d getParsedAnimationObject( )`
  - **Description**  
Devuelve el objeto animado parseado. Será sobrescrito en el parseador concreto.
- *getParsedObject*  
`public min3d.core.Object3dContainer getParsedObject( )`
  - **Description**  
Devuelve el objeto parseado. Será sobrescrito en el parseador concreto.
- *parse*  
`public void parse( )`
  - **Description**  
Realiza el parseo del objeto. Se sobrescribe en el parseador concreto
- *readFloat*  
`protected float readFloat( java.io.InputStream stream ) throws java.io.IOException`
  - **Description**  
Lee un número flotante de caracteres de un InputStream
  - **Parameters**
    - `stream` – Flujo de bytes a leer
  - **Returns** – El float leído
  - **Throws**
    - `java.io.IOException` – En caso de que se produzca alguna excepción
- *readInt*  
`protected int readInt( java.io.InputStream stream ) throws java.io.IOException`
  - **Description**  
Lee un entero de un InputStream
  - **Parameters**
    - `stream` – Flujo de bytes a leer
  - **Returns** – El entero leído
  - **Throws**
    - `java.io.IOException` – En caso de que se produzca alguna excepción
- *readShort*  
`protected int readShort( java.io.InputStream stream ) throws java.io.IOException`
  - **Description**  
Lee un short de caracteres de un InputStream
  - **Parameters**
    - `stream` – Flujo de bytes a leer

- **Returns** – El short leído
  - **Throws**
    - `java.io.IOException` – En caso de que se produzca alguna excepción
- *readString*
- `protected java.lang.String readString( java.io.InputStream stream ) throws java.io.IOException`
- **Description**  
Lee una cadena de caracteres de un `InputStream`
  - **Parameters**
    - `stream` – Flujo de bytes a leer
  - **Returns** – El string leído
  - **Throws**
    - `java.io.IOException` – En caso de que se produzca alguna excepción

### B.2.3. Clase AParser.BitmapAsset

Contiene información sobre la textura.

#### Declaración

```
protected class AParser.BitmapAsset
extends java.lang.Object
```

#### Field summary

**bitmap** El bitmap de la textura  
**key** El identificador de la textura  
**resourceID** ID del recurso  
**uOffset** U-coordinate offset  
**uScale** U-coordinate scaling value  
**useForAtlasDimensions**  
**vOffset** V-coordinate offset  
**vScale** V-coordinate scaling value

#### Constructor summary

**AParser.BitmapAsset(String, String)** Crear un objeto BitmapAsset

#### Atributos

- `public android.graphics.Bitmap bitmap`
  - El bitmap de la textura
- `public java.lang.String key`
  - El identificador de la textura

- public java.lang.String **resourceID**
  - ID del recurso
- public float **uOffset**
  - U-coordinate offset
- public float **vOffset**
  - V-coordinate offset
- public float **uScale**
  - U-coordinate scaling value
- public float **vScale**
  - V-coordinate scaling value
- public boolean **useForAtlasDimensions**

## Constructores

---

- *AParser.BitmapAsset*  
public **AParser.BitmapAsset**( java.lang.String key, java.lang.String resourceID )
  - **Description**  
Crear un objeto BitmapAsset
  - **Parameters**
    - **key** – Identificador de la textura
    - **resourceID** – Identificador del recurso

### B.2.4. Clase AParser.Material

---

Clase que representa un material para una textura

#### Declaración

```
protected class AParser.Material
extends java.lang.Object
```

#### Field summary

**diffuseColor**  
**diffuseTextureMap**  
**name**

#### Constructor summary

**AParser.Material(String)**

## Atributos

---

- public java.lang.String **name**
- public java.lang.String **diffuseTextureMap**
- public min3d.vos.Color4 **diffuseColor**

## Constructores

---

- *AParser.Material*  
public **AParser.Material**( java.lang.String name )

### B.2.5. Clase AParser.TextureAtlas

---

Cuando un modelo contiene una textura por cara, se crea un TextureAtlas. Así combinamos varias texturas

#### Declaración

```
protected class AParser.TextureAtlas
extends java.lang.Object
```

#### Field summary

**atlas** El bitmap de la textura  
**atlasId** Identificador del atlas  
**bitmaps** El conjunto de textura combinadas en una

#### Constructor summary

**AParser.TextureAtlas()** Crear una instancia de una TextureAtlas

#### Method summary

**addBitmapAsset(AParser.BitmapAsset)** Añade un bitmap al atlas  
**addBitmapAssetRemote(AParser.BitmapAsset)** Añade un  
 BitmapAsset  
**cleanup()**  
**generate()** Genera una nueva textureAtlas  
**getBitmap()** Devuelve la textura generada  
**getBitmapAssetByName(String)** Devuelve un bitmap con una textura  
 determinada  
**getBitmapAssetByResourceID(String)**  
**getId()**  
**hasBitmaps()** Nos indica si el textureAtlas tiene algún bitmap  
**setId(String)**

## Atributos

---

- private java.util.ArrayList **bitmaps**
  - El conjunto de textura combinadas en una
- private android.graphics.Bitmap **atlas**
  - El bitmap de la textura
- private java.lang.String **atlasId**
  - Identificador del atlas

## Constructores

---

- *AParser.TextureAtlas*  
public **AParser.TextureAtlas**( )
  - **Description**  
Crear una instancia de una TextureAtlas

## Métodos

---

- *addBitmapAsset*  
public void **addBitmapAsset**( AParser.BitmapAsset ba )
  - **Description**  
Añade un bitmap al atlas
  - **Parameters**
    - **bitmap** – Bitmap añadido
- *addBitmapAssetRemote*  
public void **addBitmapAssetRemote**( AParser.BitmapAsset ba )
  - **Description**  
Añade un BitmapAsset
  - **Parameters**
    - **ba** –
- *cleanup*  
public void **cleanup**( )
- *generate*  
public void **generate**( )
  - **Description**  
Genera una nueva textureAtlas
- *getBitmap*  
public android.graphics.Bitmap **getBitmap**( )
  - **Description**  
Devuelve la textura generada
  - **Returns** – El atlas generado

- *getBitmapAssetByName*  
`public AParser.BitmapAsset getBitmapAssetByName( java.lang.String materialKey )`
  - **Description**  
Devuelve un bitmap con una textura determinada
  - **Parameters**
    - `materialKey` – El nombre de la textura
  - **Returns** – El bitmap devuelto
- *getBitmapAssetByResourceID*  
`public AParser.BitmapAsset getBitmapAssetByResourceID( java.lang.String resourceID )`
- *getId*  
`public java.lang.String getId( )`
- *hasBitmaps*  
`public boolean hasBitmaps( )`
  - **Description**  
Nos indica si el textureAtlas tiene algún bitmap
  - **Returns** – Verdadero si tiene algún bitmap, falso en caso contrario
- *setId*  
`public void setId( java.lang.String newAtlasId )`

### B.2.6. Clase AParser.TextureAtlas.BitmapHeightComparer

---

Clase que compara el peso de dos bitmaps

#### Declaración

```
private class AParser.TextureAtlas.BitmapHeightComparer
extends java.lang.Object
implements java.util.Comparator
```

#### Constructor summary

`AParser.TextureAtlas.BitmapHeightComparer()`

#### Method summary

`compare(AParser.BitmapAsset, AParser.BitmapAsset)`

#### Constructores

---

- *AParser.TextureAtlas.BitmapHeightComparer*  
`private AParser.TextureAtlas.BitmapHeightComparer( )`

## Métodos

---

- *compare*  
`public int compare( AParser.BitmapAsset b1, AParser.BitmapAsset b2 )`

### B.2.7. Clase Max3DSParser

---

Parser específico para los objetos Max3DS

#### Declaración

```
public class Max3DSParser
extends com.ivanlorenzo.parser.AParser (in B.2.2, page 187)
implements IParser
```

#### Field summary

`chunkEndOffset`  
`chunkID`  
`currentObjName`  
`endReached`  
`FACES`  
`IDENTIFIER_3DS`  
`MATERIAL`  
`MESH_BLOCK`  
`OBJECT_BLOCK`  
`TEX_FILENAME`  
`TEX_MAP`  
`TEX_NAME`  
`TEXCOORD`  
`texture`  
`TRI_MATERIAL`  
`TRIMESH`  
`VERTICES`

#### Constructor summary

`Max3DSParser(Resources, String, boolean)` Constructor del parser  
`Max3DSParser(String, String)` Constructor de la clase en el que se le pasa la dirección del objeto y la textura

#### Method summary

`getParsedObject()`  
`getRemoteTexture(String)` Método que devuelve la textura.  
`parse()`

**readChunk(InputStream)** Lee un pedazo del fichero  
**readFaces(InputStream)** Lee las caras del fichero 3ds  
**readHeader(InputStream)** Lee la cabecera del fichero 3ds  
**readTexCoords(InputStream)** Lee las texturas del fichero 3ds  
**readTexture(InputStream)** Lee una textura del fichero 3ds  
**readVertices(InputStream)** Lee los vértices del fichero 3ds  
**skipRead(InputStream)** Finaliza la lectura del fichero

## Atributos

---

- private final int **IDENTIFIER\_3DS**
- private final int **MESH\_BLOCK**
- private final int **OBJECT\_BLOCK**
- private final int **TRIMESH**
- private final int **TRI\_MATERIAL**
- private final int **VERTICES**
- private final int **FACES**
- private final int **TEXCOORD**
- private final int **TEX\_MAP**
- private final int **TEX\_NAME**
- private final int **TEX\_FILENAME**
- private final int **MATERIAL**
- private int **chunkID**
- private int **chunkEndOffset**
- private boolean **endReached**
- private java.lang.String **currentObjName**
- private java.lang.String **texture**

## Constructores

---

- *Max3DSParser*  
public **Max3DSParser**( android.content.res.Resources resources, java.lang.String resourceID, boolean generateMipMap )
  - **Description**  
Constructor del parser
  - **Parameters**
    - **resources** – Recursos de la aplicación
    - **resourceID** – Identificador del recurso
    - **generateMipMap** –
- *Max3DSParser*  
public **Max3DSParser**( java.lang.String url, java.lang.String texture )

- **Description**

Constructor de la clase en el que se le pasa la dirección del objeto y la textura

- **Parameters**

- `url` –
- `texture` –

## Métodos

---

- *getParsedObject*

```
public min3d.core.Object3dContainer getParsedObject( )
```

- **Description copied from AParser (in B.2.2, page 187)**

Devuelve el objeto parseado. Será sobrescrito en el parseador concreto.

- *getRemoteTexture*

```
private java.lang.String getRemoteTexture( java.lang.String objectTexture )
```

- **Description**

Método que devuelve la textura. Si en la configuración ya dispone explícitamente de una, se devolverá esta, en caso contrario, se devolverá la que tiene asociada el objeto 3d

- **Parameters**

- `objectTexture` – El nombre de la textura contenida en el fichero 3d

- **Returns** – La textura remota

- *parse*

```
public void parse( )
```

- **Description copied from AParser (in B.2.2, page 187)**

Realiza el parseo del objeto. Se sobrescribe en el parseador concreto

- *readChunk*

```
private void readChunk( java.io.InputStream stream ) throws java.io.IOException
```

- **Description**

Lee un pedazo del fichero

- **Parameters**

- `stream` – Stream con los datos a leer

- **Throws**

- `java.io.IOException` –

- *readFaces*

```
private void readFaces( java.io.InputStream buffer ) throws java.io.IOException
```

- **Description**

Lee las caras del fichero 3ds

- **Parameters**

- `buffer` – Stream con los datos a leer

- **Throws**

- `java.io.IOException` –

- *readHeader*

```
private void readHeader( java.io.InputStream stream ) throws  
java.io.IOException
```

- **Description**  
Lee la cabecera del fichero 3ds
  - **Parameters**
    - `stream` – Stream con los datos a leer
  - **Throws**
    - `java.io.IOException` –
- *readTexCoords*  
`private void readTexCoords( java.io.InputStream buffer ) throws java.io.IOException`
- **Description**  
Lee las texturas del fichero 3ds
  - **Parameters**
    - `buffer` – Stream con los datos a leer
  - **Throws**
    - `java.io.IOException` –
- *readTexture*  
`private void readTexture( java.io.InputStream stream ) throws java.io.IOException`
- **Description**  
Lee una textura del fichero 3ds
  - **Parameters**
    - `stream` – Stream con los datos a leer
  - **Throws**
    - `java.io.IOException` –
- *readVertices*  
`private void readVertices( java.io.InputStream buffer ) throws java.io.IOException`
- **Description**  
Lee los vértices del fichero 3ds
  - **Parameters**
    - `buffer` – Stream con los datos a leer
  - **Throws**
    - `java.io.IOException` –
- *skipRead*  
`private void skipRead( java.io.InputStream stream ) throws java.io.IOException`
- **Description**  
Finaliza la lectura del fichero
  - **Parameters**
    - `stream` – Stream con los datos a leer
  - **Throws**
    - `java.io.IOException` –

Miembros heredados de la clase `com.ivanlorenzo.parser.AParser` (in B.2.2, page 187)

---

- `protected void cleanup( )`

- protected `co`
- protected `currentMaterialKey`
- protected `fileIn`
- protected `firstObject`
- protected `generateMipMap`
- public `AnimationObject3d getParsedAnimationObject( )`
- public `Object3dContainer getParsedObject( )`
- protected `materialMap`
- protected `normals`
- protected `packageID`
- public void `parse( )`
- protected `parseObjects`
- protected float `readFloat( java.io.InputStream stream )` throws `java.io.IOException`
- protected int `readInt( java.io.InputStream stream )` throws `java.io.IOException`
- protected int `readShort( java.io.InputStream stream )` throws `java.io.IOException`
- protected String `readString( java.io.InputStream stream )` throws `java.io.IOException`
- protected `remoteLocation`
- protected `resourceID`
- protected `resources`
- protected `texCoords`
- protected `textureAtlas`
- protected `url`
- protected `vertices`

### B.2.8. Clase MD2Parser

---

Parser específico de los objetos MD2

#### Declaración

```
public class MD2Parser
extends com.ivanlorenzo.parser.AParser (in B.2.2, page 187)
implements IParser
```

#### Field summary

`currentTextureName`  
`frames`  
`header`

#### Constructor summary

**MD2Parser(Resources, String, boolean)** Constructor del parser MD2  
**MD2Parser(String)** Constructor del parser en el que se le pasa la dirección donde se encontrará el objeto a parsear

#### Method summary

**getFrames(BufferedInputStream, byte[])**  
**getMaterials(BufferedInputStream, byte[])**  
**getParsedAnimationObject()**  
**getTexCoords(BufferedInputStream, byte[])**  
**getTriangles(BufferedInputStream, byte[])**  
**parse()**

## Atributos

---

- private MD2Parser.MD2Header **header**
- private java.lang.String **currentTextureName**
- private min3d.animation.KeyFrame **frames**

## Constructores

---

- *MD2Parser*  
public MD2Parser( android.content.res.Resources resources, java.lang.String resourceID, boolean generateMipMap )
  - **Description**  
Constructor del parser MD2
  - **Parameters**
    - resources –
    - resourceID –
    - generateMipMap –
- *MD2Parser*  
public MD2Parser( java.lang.String url )
  - **Description**  
Constructor del parser en el que se le pasa la dirección donde se encontrará el objeto a parsear
  - **Parameters**
    - url –

## Métodos

---

- *getFrames*  
private void getFrames( java.io.BufferedReader stream, byte[] bytes )  
throws java.io.IOException
- *getMaterials*  
private void getMaterials( java.io.BufferedReader stream, byte[] bytes )  
throws java.io.IOException
- *getParsedAnimationObject*  
public min3d.animation.AnimationObject3d getParsedAnimationObject( )
  - **Description copied from AParser (in B.2.2, page 187)**  
Devuelve el objeto animado parseado. Será sobrescrito en el parseador concreto.
- *getTexCoords*  
private void getTexCoords( java.io.BufferedReader stream, byte[] bytes )  
throws java.io.IOException
- *getTriangles*  
private void getTriangles( java.io.BufferedReader stream, byte[] bytes )  
throws java.io.IOException
- *parse*  
public void parse( )
  - **Description copied from AParser (in B.2.2, page 187)**  
Realiza el parseo del objeto. Se sobrescribe en el parseador concreto

Miembros heredados de la clase `com.ivanlorenzo.parser.AParser` (in B.2.2, page 187)

---

- `protected void cleanup( )`
- `protected co`
- `protected currentMaterialKey`
- `protected fileIn`
- `protected firstObject`
- `protected generateMipMap`
- `public AnimationObject3d getParsedAnimationObject( )`
- `public Object3dContainer getParsedObject( )`
- `protected materialMap`
- `protected normals`
- `protected packageID`
- `public void parse( )`
- `protected parseObjects`
- `protected float readFloat( java.io.InputStream stream ) throws java.io.IOException`
- `protected int readInt( java.io.InputStream stream ) throws java.io.IOException`
- `protected int readShort( java.io.InputStream stream ) throws java.io.IOException`
- `protected String readString( java.io.InputStream stream ) throws java.io.IOException`
- `protected remoteLocation`
- `protected resourceID`
- `protected resources`
- `protected texCoords`
- `protected textureAtlas`
- `protected url`
- `protected vertices`

### B.2.9. Clase MD2Parser.MD2Header

---

Clase que representa la cabecera de los ficheros MD2

#### Declaración

```
private class MD2Parser.MD2Header
extends java.lang.Object
```

#### Field summary

`frameSize`  
`id`  
`numFrames`  
`numGLCommands`  
`numSkins`  
`numTexCoord`  
`numTriangles`  
`numVerts`  
`offsetEnd`  
`offsetFrames`  
`offsetGLCommands`  
`offsetSkins`

`offsetTexCoord`  
`offsetTriangles`  
`skinHeight`  
`skinWidth`  
`version`

### Constructor summary

`MD2Parser.MD2Header()`

### Method summary

`parse(InputStream)`

### Atributos

---

- `public int id`
- `public int version`
- `public int skinWidth`
- `public int skinHeight`
- `public int frameSize`
- `public int numSkins`
- `public int numVerts`
- `public int numTexCoord`
- `public int numTriangles`
- `public int numGLCommands`
- `public int numFrames`
- `public int offsetSkins`
- `public int offsetTexCoord`
- `public int offsetTriangles`
- `public int offsetFrames`
- `public int offsetGLCommands`
- `public int offsetEnd`

### Constructores

---

- *MD2Parser.MD2Header*  
`private MD2Parser.MD2Header( )`

### Métodos

---

- *parse*  
`public void parse( java.io.InputStream stream ) throws java.lang.Exception`

## B.2.10. *Clase* ObjParser

---

Parser específico de los objetos obj

### Declaración

```
public class ObjParser
extends com.ivanlorenzo.parser.AParser (in B.2.2, page 187)
implements IParser
```

### Field summary

**DIFFUSE\_COLOR**  
**DIFFUSE\_TEX\_MAP**  
**FACE**  
**MATERIAL\_LIB**  
**NEW\_MATERIAL**  
**NORMAL**  
**OBJECT**  
**TEXCOORD**  
**USE\_MATERIAL**  
**VERTEX**

### Constructor summary

**ObjParser(Resources, String, boolean)** Crea una nueva instancia de un parser OBJ  
**ObjParser(String)** Crea una nueva instancia de un parser OBJ

### Method summary

**cleanup()**  
**getParsedObject()**  
**parse()**  
**readMaterialLib(String)** Lee el material

### Atributos

---

- `private final java.lang.String` **VERTEX**
- `private final java.lang.String` **FACE**
- `private final java.lang.String` **TEXCOORD**
- `private final java.lang.String` **NORMAL**
- `private final java.lang.String` **OBJECT**
- `private final java.lang.String` **MATERIAL\_LIB**

- `private final java.lang.String USE_MATERIAL`
- `private final java.lang.String NEW_MATERIAL`
- `private final java.lang.String DIFFUSE_COLOR`
- `private final java.lang.String DIFFUSE_TEX_MAP`

## Constructores

---

- *ObjParser*  
`public ObjParser( android.content.res.Resources resources, java.lang.String resourceID, boolean generateMipMap )`
  - **Description**  
Crea una nueva instancia de un parser OBJ
  - **Parameters**
    - `resources` – Recursos de la aplicación
    - `resourceID` – Identificador del recurso
    - `generateMipMap` –
- *ObjParser*  
`public ObjParser( java.lang.String url )`
  - **Description**  
Crea una nueva instancia de un parser OBJ
  - **Parameters**
    - `url` – Dirección en la que se encuentra el objeto a parsear

## Métodos

---

- *cleanup*  
`protected void cleanup( )`
  - **Description copied from AParser (in B.2.2, page 187)**  
Borra todas las estructuras de datos
- *getParsedObject*  
`public min3d.core.Object3dContainer getParsedObject( )`
  - **Description copied from AParser (in B.2.2, page 187)**  
Devuelve el objeto parseado. Será sobrescrito en el parseador concreto.
- *parse*  
`public void parse( )`
  - **Description copied from AParser (in B.2.2, page 187)**  
Realiza el parseo del objeto. Se sobrescribe en el parseador concreto
- *readMaterialLib*  
`private void readMaterialLib( java.lang.String libID )`
  - **Description**  
Lee el material
  - **Parameters**
    - `libID` – Identificador del material

Miembros heredados de la clase `com.ivanlorenzo.parser.AParser` (in B.2.2, page 187)

---

- `protected void cleanup( )`
- `protected co`
- `protected currentMaterialKey`
- `protected fileIn`
- `protected firstObject`
- `protected generateMipMap`
- `public AnimationObject3d getParsedAnimationObject( )`
- `public Object3dContainer getParsedObject( )`
- `protected materialMap`
- `protected normals`
- `protected packageID`
- `public void parse( )`
- `protected parseObjects`
- `protected float readFloat( java.io.InputStream stream ) throws java.io.IOException`
- `protected int readInt( java.io.InputStream stream ) throws java.io.IOException`
- `protected int readShort( java.io.InputStream stream ) throws java.io.IOException`
- `protected String readString( java.io.InputStream stream ) throws java.io.IOException`
- `protected remoteLocation`
- `protected resourceID`
- `protected resources`
- `protected texCoords`
- `protected textureAtlas`
- `protected url`
- `protected vertices`

### B.2.11. Clase `ObjParser.ObjFace`

---

Clase privada que representa una cara del objeto

#### Declaración

```
private class ObjParser.ObjFace
extends min3d.parser.ParseObjectFace
```

#### Constructor summary

`ObjParser.ObjFace(String, String, int)`

#### Constructores

---

- *ObjParser.ObjFace*  
`public ObjParser.ObjFace( java.lang.String line, java.lang.String materialKey, int faceLength )`

---

## Miembros heredados de la clase `min3d.parser.ParseObjectFace`

---

- `public faceLength`
- `public hasn`
- `public hasuv`
- `public materialKey`
- `public n`
- `public uv`
- `public v`

### B.2.12. Clase `ParseObjectData`

---

Clase que contiene la información de un objeto parseado

#### Declaración

```
public class ParseObjectData
    extends java.lang.Object
```

#### Field summary

**faces** Caras del objeto  
**name** Nombre del objeto  
**normals** Las normales del objeto  
**numFaces** Número de caras  
**texCoords** Coordenadas de las texturas del objeto  
**vertices** Vértices del objeto

#### Constructor summary

**ParseObjectData()** Constructor sin parámetros  
**ParseObjectData(ArrayList, ArrayList, ArrayList)** Constructor que inicializa un objeto parseado

#### Method summary

**calculateFaceNormal(ParseObjectFace)** Calcula la normal de un objeto  
**cleanup()** Borra las caras  
**getParsedObject(AParser.TextureAtlas, HashMap, KeyFrame[])**  
Devuelve un objeto `AnimationObject3d` parseado  
**getParsedObject(HashMap, AParser.TextureAtlas)** Devuelve un `Object3d` parseado  
**parseObject(Object3d, HashMap, AParser.TextureAtlas)** Parsea un objeto

## Atributos

---

- protected java.util.ArrayList **faces**
  - Caras del objeto
- protected int **numFaces**
  - Número de caras
- protected java.util.ArrayList **vertices**
  - Vértices del objeto
- protected java.util.ArrayList **texCoords**
  - Coordenadas de las texturas del objeto
- protected java.util.ArrayList **normals**
  - Las normales del objeto
- public java.lang.String **name**
  - Nombre del objeto

## Constructores

---

- *ParseObjectData*  
public **ParseObjectData**( )
  - **Description**  
Constructor sin parámetros
- *ParseObjectData*  
public **ParseObjectData**( java.util.ArrayList vertices, java.util.ArrayList texCoords, java.util.ArrayList normals )
  - **Description**  
Constructor que inicializa un objeto parseado
  - **Parameters**
    - **vertices** – ArrayList con los vértices
    - **texCoords** – ArrayList con las coordenadas de las texturas
    - **normals** – ArrayList con las normales del objeto

## Métodos

---

- *calculateFaceNormal*  
public void **calculateFaceNormal**( min3d.parser.ParseObjectFace face )
  - **Description**  
Calcula la normal de un objeto
  - **Parameters**
    - **face** –
- *cleanup*  
protected void **cleanup**( )

- **Description**  
Borra las caras
- *getParsedObject*  

```
public min3d.animation.AnimationObject3d getParsedObject( AParser.TextureAtlas textureAtlas, java.util.HashMap materialMap, min3d.animation.KeyFrame[] frames )
```

  - **Description**  
Devuelve un objeto AnimationObject3d parseado
  - **Parameters**
    - textureAtlas –
    - materialMap –
    - frames –
  - **Returns** – Objeto AnimationObject3d parseado
- *getParsedObject*  

```
public min3d.core.Object3d getParsedObject( java.util.HashMap materialMap, AParser.TextureAtlas textureAtlas )
```

  - **Description**  
Devuelve un Object3d parseado
  - **Parameters**
    - materialMap –
    - textureAtlas –
  - **Returns** – Objeto Object3d parseado
- *parseObject*  

```
private void parseObject( min3d.core.Object3d obj, java.util.HashMap materialMap, AParser.TextureAtlas textureAtlas )
```

  - **Description**  
Parsea un objeto
  - **Parameters**
    - obj –
    - materialMap –
    - textureAtlas –

### B.2.13. Clase Parser

Clase factoría de Parsers. Especifica el tipo de parser y devuelve la clase especializada de parser.

#### Declaración

```
public class Parser
extends java.lang.Object
```

#### Constructor summary

**Parser()**

## Method summary

**createParser(Parser.Type, Resources, String, boolean)** Crea un parser de un tipo determinado

**createParser(Parser.Type, String, String)** Crea un parser de un tipo determinado

## Constructores

---

- *Parser*  
`public Parser( )`

## Métodos

---

- *createParser*  
`public static IParse createParser( Parser.Type type, android.content.res.Resources resources, java.lang.String resourceID, boolean generateMipMap )`
  - **Description**  
Crea un parser de un tipo determinado
  - **Parameters**
    - `type` – Tipo de parser
    - `resources` – Recursos de la aplicación
    - `resourceID` – Identificador del recurso
    - `generateMipMap` –
  - **Returns** – Parser del tipo correspondiente
- *createParser*  
`public static IParse createParser( Parser.Type type, java.lang.String url, java.lang.String texture )`
  - **Description**  
Crea un parser de un tipo determinado
  - **Parameters**
    - `type` – Tipo de parser
    - `url` – La dirección url del objeto
    - `texture` – La dirección de la textura
  - **Returns** – Parser del tipo correspondiente

## B.2.14. Clase Parser.Type

---

Enumeración de los tipos de parsers de la aplicación

### Declaración

```
public static final class Parser.Type
extends java.lang.Enum
```

## Field summary

**MAX\_3DS**  
**MD2**  
**OBJ**

## Constructor summary

**Parser.Type()**

## Method summary

**valueOf(String)**  
**values()**

## Atributos

---

- public static final Parser.Type **OBJ**
- public static final Parser.Type **MAX\_3DS**
- public static final Parser.Type **MD2**

## Constructores

---

- *Parser.Type*  
private **Parser.Type**( )

## Métodos

---

- *valueOf*  
public static Parser.Type **valueOf**( java.lang.String name )
- *values*  
public static Parser.Type[] **values**( )

## Miembros heredados de la clase java.lang.Enum

---

- protected final Object **clone**( ) throws CloneNotSupportedException
- public final int **compareTo**( Enum arg0 )
- public final boolean **equals**( Object arg0 )
- protected final void **finalize**( )
- public final Class **getDeclaringClass**( )
- public final int **hashCode**( )
- private final **name**
- public final String **name**( )
- private final **ordinal**
- public final int **ordinal**( )
- private void **readObject**( java.io.ObjectInputStream arg0 ) throws java.io.IOException, java.lang.ClassNotFoundException
- private void **readObjectNoData**( ) throws java.io.ObjectStreamException
- public String **toString**( )
- public static Enum **valueOf**( Class arg0, String arg1 )

## B.3. Paquete com.ivanlorenzo.location.observer

### B.3.1. *Interfaz IARObservable*

---

Interface genérica Observable del patrón Observer que tendrá un conjunto de observadores y notificará a estos cuando se produzca algún cambio de estado

#### Declaración

```
public interface IARObservable
```

#### Method summary

**addObserver(IARObserver)** Método que añade un observador a la lista  
**notifyObservers()** Cuando se produzca un cambio en él, notificará a todos los observadores a través de este método  
**removeObserver(IARObserver)** Borra un observador de la lista

#### Métodos

---

- *addObserver*  
void **addObserver**( IARObserver observer )
  - **Description**  
Método que añade un observador a la lista
  - **Parameters**
    - **observer** – Observador a añadir
- *notifyObservers*  
void **notifyObservers**( )
  - **Description**  
Cuando se produzca un cambio en él, notificará a todos los observadores a través de este método
- *removeObserver*  
void **removeObserver**( IARObserver observer )
  - **Description**  
Borra un observador de la lista
  - **Parameters**
    - **observer** – Observador a borrar

### B.3.2. *Interfaz IARObserver*

---

Interface genérica Observadora del patrón Observer. Este patrón define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes.

## Declaración

```
public interface IARObserver
```

## Method summary

**update(T)** Realiza los cambios oportunos en el observador pasándole un parámetro genérico

## Métodos

---

- *update*  
`void update( java.lang.Object subject )`
  - **Description**  
Realiza los cambios oportunos en el observador pasándole un parámetro genérico
  - **Parameters**
    - `subject` – Se podrá corresponder con el estado del sujeto

### B.3.3. Clase ARObservable

---

Clase abstracta que implementa la interface IARObservable. Contiene un elemento genérico y una lista de observadores genéricos

## Declaración

```
public abstract class ARObservable  
extends java.lang.Object  
implements IARObservable
```

## Field summary

**data** Elemento genérico  
**observers** Lista de observadores genéricos

## Constructor summary

**ARObservable()**

## Method summary

**addObserver(IARObserver)**  
**notifyObservers()**  
**removeObserver(IARObserver)**

## Atributos

---

- `protected java.lang.Object data`
  - Elemento genérico
- `protected java.util.List observers`
  - Lista de observadores genéricos

## Constructores

---

- *ARObservable*  
`public ARObservable( )`

## Métodos

---

- *addObserver*  
`void addObserver( IARObserver observer )`
  - **Description copied from IARObservable (in B.3.1, page 212)**  
Método que añade un observador a la lista
  - **Parameters**
    - `observer` – Observador a añadir
- *notifyObservers*  
`void notifyObservers( )`
  - **Description copied from IARObservable (in B.3.1, page 212)**  
Cuando se produzca un cambio en él, notificará a todos los observadores a través de este método
- *removeObserver*  
`void removeObserver( IARObserver observer )`
  - **Description copied from IARObservable (in B.3.1, page 212)**  
Borra un observador de la lista
  - **Parameters**
    - `observer` – Observador a borrar

## B.4. Paquete com.ivanlorenzo.gui

### B.4.1. Clase ARStatus

---

Enumeración de los estados en los que puede estar funcinando la aplicación

#### Declaración

```
public final class ARStatus
extends java.lang.Enum
```

## Field summary

**AR\_GPS** Estado en el que se está mostrando información proveniente de una localización

**AR\_Marker** Estado en el que se están buscando markers en las imágenes obtenidas por la cámara

## Constructor summary

**ARStatus()**

## Method summary

**valueOf(String)**

**values()**

## Atributos

---

- public static final ARStatus **AR\_Marker**
  - Estado en el que se están buscando markers en las imágenes obtenidas por la cámara
- public static final ARStatus **AR\_GPS**
  - Estado en el que se está mostrando información proveniente de una localización

## Constructores

---

- *ARStatus*  
private **ARStatus()**

## Métodos

---

- *valueOf*  
public static ARStatus **valueOf**( java.lang.String name )
- *values*  
public static ARStatus[] **values**( )

## Miembros heredados de la clase java.lang.Enum

---

- protected final Object **clone**( ) throws CloneNotSupportedException
- public final int **compareTo**( Enum arg0 )
- public final boolean **equals**( Object arg0 )
- protected final void **finalize**( )
- public final Class **getDeclaringClass**( )
- public final int **hashCode**( )
- private final **name**
- public final String **name**( )
- private final **ordinal**
- public final int **ordinal**( )
- private void **readObject**( java.io.ObjectInputStream arg0 ) throws java.io.IOException, java.lang.ClassNotFoundException
- private void **readObjectNoData**( ) throws java.io.ObjectStreamException
- public String **toString**( )
- public static Enum **valueOf**( Class arg0, String arg1 )

### B.4.2. Clase ARToolkitDrawer

---

Clase responsable de dibujar los objetos. Está basada en min3d

#### Declaración

```
public class ARToolkitDrawer
extends java.lang.Object
```

#### Field summary

**ar\_code**  
**ar\_param**  
**ar\_transmat\_result**  
**context**  
**dialog**  
**MARKER\_MAX**  
**marker\_width**  
**mediaPlayer**  
**mNumPatt**  
**mRenderer** Render de min3d  
**nya**  
**raster**

#### Constructor summary

**ARToolkitDrawer(InputStream, int[], ArrayList, Renderer)**  
Constructor

#### Method summary

**createNyARTool(int, int)** Se crean los recursos  
**decodeYUV420SP(byte[], byte[], int, int, int)**  
**decodeYUV420SP(int[], byte[], int, int, int)**  
**draw(byte[])** Método que detecta los patrones y si los encuentra pinta los objetos  
**draw(IARObserver, int[])** Método para pintar un objeto en el centro de la pantalla sin falta de detectar ningún patrón  
**getDescriptionDialog(String)**  
**initialization(InputStream, int[], ArrayList)** Inicializa los elementos de la clase  
**setContext(Context)** Establece el contexto  
**toCameraFrustumRHf(NyARParam, float[])**  
**toCameraViewRHf(NyARDoubleMatrix44, float[])**

---

## Atributos

---

- private int **mNumPatt**
- private static final int **MARKER\_MAX**
- private jp.nyatla.nyartoolkit.detector.NyARDetectMarker **nya**
- private jp.nyatla.nyartoolkit.core.raster.rgb.NyARRgbRaster\_RGB **raster**
- private jp.nyatla.nyartoolkit.core.param.NyARParam **ar\_param**
- private jp.nyatla.nyartoolkit.core.NyARCode **ar\_code**
- private double **marker\_width**
- private jp.nyatla.nyartoolkit.core.transmat.NyARTransMatResult **ar\_transmat\_result**
- private min3d.core.Renderer **mRenderer**
  - Render de min3d
- private com.ivanlorenzo.data.objects.audio.MediaPlayerAdapter **mediaPlayer**
- private android.content.Context **context**
- private android.app.Dialog **dialog**

---

## Constructores

---

- *ARToolkitDrawer*  
public **ARToolkitDrawer**( java.io.InputStream camePara, int[] width, java.util.ArrayList patt, min3d.core.Renderer mRenderer )
  - **Description**  
Constructor
  - **Parameters**
    - **camePara** – Parámetros de la cámara
    - **width** – Ancho de los patrones
    - **patt** – Patrones de la aplicación
    - **mRenderer** – El render

---

## Métodos

---

- *createNyARTool*  
private void **createNyARTool**( int w, int h )
  - **Description**  
Se crean los recursos
  - **Parameters**
    - **w** – Ancho de la pantalla
    - **h** – Altura de la pantalla
- *decodeYUV420SP*  
public static native void **decodeYUV420SP**( byte[] rgb, byte[] yuv420sp, int width, int height, int type )

■ *decodeYUV420SP*

```
public static native void decodeYUV420SP( int[] rgb, byte[] yuv420sp, int
width, int height, int type )
```

■ *draw*

```
public void draw( byte[] data )
```

- **Description**

Método que detecta los patrones y si los encuentra pinta los objetos

- **Parameters**

- *data* – Datos enviados para pintar

■ *draw*

```
public void draw( com.ivanlorenzo.location.observer.IARObserver observer, int[]
objects )
```

- **Description**

Método para pintar un objeto en el centro de la pantalla sin falta de detectar ningún patrón

- **Parameters**

- *observer* – Datos acerca de la localización actual
- *objects* – Objetos a pintar

■ *getDescriptionDialog*

```
private android.app.Dialog getDescriptionDialog( java.lang.String text )
```

■ *initialization*

```
private void initialization( java.io.InputStream camePara, int[] width,
java.util.ArrayList patt )
```

- **Description**

Inicializa los elementos de la clase

- **Parameters**

- *camePara* – Parámetros de la cámara
- *width* – Ancho de los patrones
- *patt* – Lista con los patrones

■ *setContext*

```
public void setContext( android.content.Context context )
```

- **Description**

Establece el contexto

- **Parameters**

- *context* – Contexto de la aplicación

■ *toCameraFrustumRHf*

```
public static void toCameraFrustumRHf(
jp.nyatla.nyartoolkit.core.param.NyARParam i_arparam, float[] o_gl_projection )
```

- **Parameters**

- *i\_arparam* –
- *o\_gl\_projection* –

■ *toCameraViewRHf*

```
public static void toCameraViewRHf(
jp.nyatla.nyartoolkit.core.types.matrix.NyARDoubleMatrix44 i_ny_result, float[]
o_gl_result )
```

- **Parameters**

- *i\_ny\_result* –
- *o\_gl\_result* –

### B.4.3. Clase MessageWhat

---

Enumeración con los mensajes mostrados por la aplicación

#### Declaración

```
public final class MessageWhat
    extends java.lang.Enum
```

#### Field summary

**CLEAR\_SCREEN\_DELAY**  
**CROP\_MSG**  
**FIRST\_TIME\_INIT**  
**HIDE\_LOADING**  
**NONE**  
**RESTART\_PREVIEW**  
**SET\_CAMERA\_PARAMETERS\_WHEN\_IDLE**  
**SHOW\_LOADING**

#### Constructor summary

**MessageWhat()**

#### Method summary

**valueOf(String)**  
**values()**

#### Atributos

---

- public static final MessageWhat **NONE**
- public static final MessageWhat **CROP\_MSG**
- public static final MessageWhat **FIRST\_TIME\_INIT**
- public static final MessageWhat **RESTART\_PREVIEW**
- public static final MessageWhat **CLEAR\_SCREEN\_DELAY**
- public static final MessageWhat **SET\_CAMERA\_PARAMETERS\_WHEN\_IDLE**
- public static final MessageWhat **SHOW\_LOADING**
- public static final MessageWhat **HIDE\_LOADING**

#### Constructores

---

- *MessageWhat*  
private MessageWhat( )

## Métodos

---

- *valueOf*  
`public static MessageWhat valueOf( java.lang.String name )`
- *values*  
`public static MessageWhat[] values( )`

## Miembros heredados de la clase `java.lang.Enum`

---

- `protected final Object clone( )` throws `CloneNotSupportedException`
- `public final int compareTo( Enum arg0 )`
- `public final boolean equals( Object arg0 )`
- `protected final void finalize( )`
- `public final Class getDeclaringClass( )`
- `public final int hashCode( )`
- `private final name`
- `public final String name( )`
- `private final ordinal`
- `public final int ordinal( )`
- `private void readObject( java.io.ObjectInputStream arg0 )` throws `java.io.IOException`, `java.lang.ClassNotFoundException`
- `private void readObjectNoData( )` throws `java.io.ObjectStreamException`
- `public String toString( )`
- `public static Enum valueOf( Class arg0, String arg1 )`

### B.4.4. Clase *R*

---

#### Declaración

```
public final class R
extends java.lang.Object
```

#### Constructor summary

**R()**

#### Constructores

---

- *R*  
`public R( )`

### B.4.5. Clase *R.attr*

---

#### Declaración

```
public static final class R.attr
extends java.lang.Object
```

## Constructor summary

**R.attr()**

## Constructores

---

- *R.attr*  
`public R.attr( )`

## B.4.6. Clase R.drawable

---

### Declaración

```
public static final class R.drawable  
extends java.lang.Object
```

### Field summary

**icon**

### Constructor summary

**R.drawable()**

### Atributos

---

- `public static final int icon`

### Constructores

---

- *R.drawable*  
`public R.drawable( )`

## B.4.7. Clase R.id

---

### Declaración

```
public static final class R.id  
extends java.lang.Object
```

## Field summary

**camera\_preview**  
**configuration**  
**exit**  
**frame**  
**main**

## Constructor summary

**R.id()**

## Atributos

---

- public static final int **camera\_preview**
- public static final int **configuration**
- public static final int **exit**
- public static final int **frame**
- public static final int **main**

## Constructores

---

- *R.id*  
public **R.id**( )

## B.4.8. Clase R.layout

---

### Declaración

```
public static final class R.layout  
extends java.lang.Object
```

## Field summary

**main**

## Constructor summary

**R.layout()**

## Atributos

---

- public static final int **main**

## Constructores

---

- *R.layout*  
`public R.layout( )`

### B.4.9. Clase R.menu

---

#### Declaración

```
public static final class R.menu
extends java.lang.Object
```

#### Field summary

**menu**

#### Constructor summary

**R.menu()**

#### Atributos

---

- `public static final int menu`

## Constructores

---

- *R.menu*  
`public R.menu( )`

### B.4.10. Clase R.raw

---

#### Declaración

```
public static final class R.raw
extends java.lang.Object
```

## Field summary

**camera\_para**  
**marker**  
**markerd**  
**multi**  
**patta**  
**pattb**  
**pattc**  
**pattd**  
**pattf**  
**pattg**  
**patthiro**  
**pattkanji**

## Constructor summary

**R.raw()**

## Atributos

---

- public static final int **camera\_para**
- public static final int **marker**
- public static final int **markerd**
- public static final int **multi**
- public static final int **patta**
- public static final int **pattb**
- public static final int **pattc**
- public static final int **pattd**
- public static final int **pattf**
- public static final int **pattg**
- public static final int **patthiro**
- public static final int **pattkanji**

## Constructores

---

- *R.raw*  
public **R.raw**( )

## B.4.11. Clase R.string

---

## Declaración

```
public static final class R.string
extends java.lang.Object
```

## Field summary

**app\_name**

**camera\_error\_title** title for the dialog showing the error of camera  
hardware title for the dialog showing the error of camera hardware title  
for the dialog showing the error of camera hardware

**cannot\_connect\_camera** message for the dialog showing the error of  
camera hardware message for the dialog showing the error of camera  
hardware message for the dialog showing the error of camera hardware

**configuration**

**connection\_error**

**details\_ok** Details dialog .°K” button.

**exit**

**FILE\_PREFERENCES**

**internet\_connection**

**loading**

**no**

**NO\_OBJECT\_3D**

**objects**

**objects\_url**

**patterns\_and\_locations**

**patterns\_url**

**save\_preferences**

**sure\_exit**

**URL\_OBJECTS**

**URL\_PATTERNS**

**write\_objects\_url**

**write\_patterns\_url**

**yes**

## Constructor summary

**R.string()**

## Atributos

---

- public static final int **FILE\_PREFERENCES**
- public static final int **NO\_OBJECT\_3D**
- public static final int **URL\_OBJECTS**
- public static final int **URL\_PATTERNS**

- public static final int **app\_name**
- public static final int **camera\_error\_title**
  - title for the dialog showing the error of camera hardware title for the dialog showing the error of camera hardware title for the dialog showing the error of camera hardware
- public static final int **cannot\_connect\_camera**
  - message for the dialog showing the error of camera hardware message for the dialog showing the error of camera hardware message for the dialog showing the error of camera hardware
- public static final int **configuration**
- public static final int **connection\_error**
- public static final int **details\_ok**
  - Details dialog .°K”button. Dismisses dialog. Details dialog .°K”button. Dismisses dialog. Details dialog .°K”button. Dismisses dialog.
- public static final int **exit**
- public static final int **internet\_connection**
- public static final int **loading**
- public static final int **no**
- public static final int **objects**
- public static final int **objects\_url**
- public static final int **patterns\_and\_locations**
- public static final int **patterns\_url**
- public static final int **save\_preferences**
- public static final int **sure\_exit**
- public static final int **write\_objects\_url**
- public static final int **write\_patterns\_url**
- public static final int **yes**

## Constructores

---

- *R.string*  
public **R.string**( )

## B.4.12. Clase R.xml

---

### Declaración

```
public static final class R.xml
extends java.lang.Object
```

### Field summary

**preferences**

## Constructor summary

**R.xml()**

## Atributos

---

- `public static final int preferences`

## Constructores

---

- *R.xml*  
`public R.xml( )`

## B.4.13. Clase TFGARActivity

---

Clase Activity principal de la aplicación. Será la encargada de manejar la recepción de datos de la cámara. Además, recibirá los eventos táctiles producidos en la pantalla y gestionará la escena en la que se encontrarán todos los objetos.

## Declaración

```
public class TFGARActivity
extends android.app.Activity
implements android.view.View.OnClickListener, android.view.SurfaceHolder.Callback,
min3d.interfaces.ISceneController, com.ivanlorenzo.location.observer.IARObserver
```

## Field summary

**\_initSceneHandler** Menajador de inicio de la escena  
**\_initSceneRunnable**  
**\_updateSceneHandler** Manejador de actualización de la escena  
**\_updateSceneRunnable**  
**arStatus** Estado de la aplicación  
**arToolkitDrawer** Instancia encargada de dibujar los objetos  
**INVALID\_POINTER\_ID**  
**mActivePointerId**  
**mCameraDevice** Dispositivo de cámara  
**messageWhat** Array con los mensajes de la aplicación  
**mFirstTimeInitialized** Nos indica si ya está inicializada la aplicación  
**mGLSurfaceView**  
**mHandler** Manejador principal  
**mLastOrientation** Última orientación del dispositivo  
**mLastTouchX**  
**mLastTouchY**  
**mMediaPlayer** MediaPlayer

**mOrientationListener** Manejador de eventos de la orientación  
**mParameters** Parámetros de la cámara  
**mPausing** Nos indica si está pausado  
**mPreferences** Preferencias compartidas de la aplicación  
**mPreviewCallback** Llamada a la vista previa  
**mPreviewing** Nos indica si estamos capturando imágenes previas  
**mRenderer** Renderizador de min3d  
**mScaleDetector**  
**mScaleFactor**  
**mStartPreviewFail**  
**mSurfaceHolder**  
**mSurfaceView**  
**scene** Escena que mantendrá todos los objetos  
**SCREEN\_DELAY** Tiempo de delay de la pantalla

### Constructor summary

**TFGARActivity()**

### Method summary

**changeGLSurfaceViewState()** Cambia la vista  
**closeCamera()** Apagamos la cámara  
**createCompilerXML()** Crea el compilador XML pasándole las URLs de los archivos de configuración y se muestra un diálogo informado de que ha terminado la carga  
**ensureCameraDevice()** Nos aseguramos de que la cámara está disponible  
**finalizeGLSurfaceView()** Finaliza la vista  
**getInitSceneHandler()**  
**getInitSceneRunnable()**  
**getUpdateSceneHandler()**  
**getUpdateSceneRunnable()**  
**initializeFirstTime()** Empezaremos a tomar capturas de la cámara.  
**initializeGLSurfaceView()** Inicializa la vista  
**initializeSecondTime()** Si la actividad es pausada y luego se continúa, se llamará a este método en onResume  
**initScene()** Inicializamos los objetos 3d, establecemos sus propiedades y añadimos los objetos a la escena  
**keepScreenOnAwhile()** Mantenemos la pantalla en espera  
**onActivityResult(int, int, Intent)**  
**onBackPressed()**  
**onClick(View)**  
**onCreate(Bundle)** Se llama cuando se crea la actividad  
**onCreateDialog(int)**  
**onCreateOptionsMenu(Menu)**  
**onDestroy()**  
**onInitScene()** Llamado justo después de iniciarse la escena

**onOptionsItemSelected(MenuItem)**  
**onPause()**  
**onResume()**  
**onStart()**  
**onStop()**  
**onTouchEvent(MotionEvent)**  
**onUpdateScene()** Llamado justo después de actualizarse la escena  
**resetScreenOn()** Reseteamos la pantalla  
**restartPreview()** Se restaura la preview  
**setCameraParameters()** Se establecen los parámetros de la cámara  
**setPreviewDisplay(SurfaceHolder)** Establecemos la vista  
**showCameraErrorAndFinish()** Mostrar el error de la cámara y finalizar  
**startGPSLocation()** Inicia el servicio de localización utilizando el mejor proveedor, bien sea GPS o a través de la conexión a Internet.  
**startPreview()** Iniciamos la vista  
**stopPreview()** Paramos la vista  
**surfaceChanged(SurfaceHolder, int, int, int)**  
**surfaceCreated(SurfaceHolder)**  
**surfaceDestroyed(SurfaceHolder)**  
**update(LocationData)** Establecemos el estado de la aplicación: Marker o GPS y en caso de ser localización, lo mandamos dibujar  
**updateScene()** Cualquier manipulación de la escena y de los objetos 3d irá aquí.

## Atributos

---

- private static MessageWhat **messageWhat**
  - Array con los mensajes de la aplicación
- private static final int **SCREEN\_DELAY**
  - Tiempo de delay de la pantalla
- private android.hardware.Camera.Parameters **mParameters**
  - Parámetros de la cámara
- private android.view.OrientationEventListener **mOrientationListener**
  - Manejador de eventos de la orientación
- private int **mLastOrientation**
  - Última orientación del dispositivo
- private android.content.SharedPreferences **mPreferences**
  - Preferencias compartidas de la aplicación
- private android.hardware.Camera **mCameraDevice**
  - Dispositivo de cámara
- private android.view.SurfaceView **mSurfaceView**
- private android.view.SurfaceHolder **mSurfaceHolder**

- private boolean **mStartPreviewFail**
- private android.opengl.GLSurfaceView **mGLSurfaceView**
- private min3d.core.Renderer **mRenderer**
  - Renderizador de min3d
- private boolean **mPreviewing**
  - Nos indica si estamos capturando imágenes previas
- private boolean **mPausing**
  - Nos indica si está pausado
- private boolean **mFirstTimeInitialized**
  - Nos indica si ya está inicializada la aplicación
- private android.os.Handler **mHandler**
  - Manejador principal
- private TFGARActivity.PreviewCallback **mPreviewCallback**
  - Llamada a la vista previa
- private ARToolkitDrawer **arToolkitDrawer**
  - Instancia encargada de dibujar los objetos
- private android.media.MediaPlayer **mMediaPlayer**
  - MediaPlayer
- private ARStatus **arStatus**
  - Estado de la aplicación
- private static final int **INVALID\_POINTER\_ID**
- private float **mLastTouchX**
- private float **mLastTouchY**
- private int **mActivePointerId**
- private android.view.ScaleGestureDetector **mScaleDetector**
- private float **mScaleFactor**
- public min3d.core.Scene **scene**
  - Escena que mantendrá todos los objetos
- protected android.os.Handler **\_initSceneHandler**
  - Menajador de inicio de la escena
- protected android.os.Handler **\_updateSceneHandler**
  - Manejador de actualización de la escena
- final java.lang.Runnable **\_initSceneRunnable**
- final java.lang.Runnable **\_updateSceneRunnable**

## Constructores

---

- *TFGARActivity*  
public **TFGARActivity**( )

---

## Métodos

---

- *changeGLSurfaceViewState*  
`private void changeGLSurfaceViewState( )`
  - **Description**  
Cambia la vista
- *closeCamera*  
`private void closeCamera( )`
  - **Description**  
Apagamos la cámara
- *createCompilerXML*  
`private void createCompilerXML( )`
  - **Description**  
Crea el compilador XML pasándole las URLs de los archivos de configuración y se muestra un diálogo informado de que ha terminado la carga
- *ensureCameraDevice*  
`private void ensureCameraDevice( ) throws  
com.android.camera.CameraHardwareException`
  - **Description**  
Nos aseguramos de que la cámara está disponible
  - **Throws**
    - `com.android.camera.CameraHardwareException` – En caso de que el hardware de la cámara nos dé algún error
- *finalizeGLSurfaceView*  
`private void finalizeGLSurfaceView( )`
  - **Description**  
Finaliza la vista
- *getInitSceneHandler*  
`public android.os.Handler getInitSceneHandler( )`
  - **See also**
    - `min3d.interfaces.ISceneController.getInitSceneHandler()`
- *getInitSceneRunnable*  
`public java.lang.Runnable getInitSceneRunnable( )`
  - **See also**
    - `min3d.interfaces.ISceneController.getInitSceneRunnable()`
- *getUpdateSceneHandler*  
`public android.os.Handler getUpdateSceneHandler( )`
  - **See also**
    - `min3d.interfaces.ISceneController.getUpdateSceneHandler()`
- *getUpdateSceneRunnable*  
`public java.lang.Runnable getUpdateSceneRunnable( )`
  - **See also**

◦ `min3d.interfaces.ISceneController.getUpdateSceneRunnable()`

■ *initializeFirstTime*

`private void initializeFirstTime( )`

• **Description**

Empezaremos a tomar capturas de la cámara. Debería ser llamado una sola vez. Esto podría haberse hecho en el método `onCreate()`, pero queremos así logramos que aparezca la pantalla de la cámara tan pronto como sea posible.

■ *initializeGLSurfaceView*

`private void initializeGLSurfaceView( )`

• **Description**

Inicializa la vista

■ *initializeSecondTime*

`private void initializeSecondTime( )`

• **Description**

Si la actividad es pausada y luego se continúa, se llamará a este método en `onResume`

■ *initScene*

`public void initScene( )`

• **Description**

Inicializamos los objetos 3d, establecemos sus propiedades y añadimos los objetos a la escena

■ *keepScreenOnAwhile*

`private void keepScreenOnAwhile( )`

• **Description**

Mantenemos la pantalla en espera

■ *onActivityResult*

`protected void onActivityResult( int arg0, int arg1, android.content.Intent arg2 )`

■ *onBackPressed*

`public void onBackPressed( )`

■ *onClick*

`void onClick( android.view.View arg0 )`

■ *onCreate*

`public void onCreate( android.os.Bundle icle )`

• **Description**

Se llama cuando se crea la actividad

■ *onCreateDialog*

`protected android.app.Dialog onCreateDialog( int arg0 )`

■ *onCreateOptionsMenu*

`public boolean onCreateOptionsMenu( android.view.Menu arg0 )`

■ *onDestroy*

`protected void onDestroy( )`

■ *onInitScene*

`public void onInitScene( )`

- **Description**

Llamado justo después de iniciarse la escena

- *onOptionsItemSelected*  
public boolean **onOptionsItemSelected**( android.view.MenuItem **arg0** )
- *onPause*  
protected void **onPause**( )
- *onResume*  
protected void **onResume**( )
- *onStart*  
protected void **onStart**( )
- *onStop*  
protected void **onStop**( )
- *onTouchEvent*  
public boolean **onTouchEvent**( android.view.MotionEvent **arg0** )
- *onUpdateScene*  
public void **onUpdateScene**( )

- **Description**

Llamado justo después de actualizarse la escena

- *resetScreenOn*  
private void **resetScreenOn**( )

- **Description**

Reseteamos la pantalla

- *restartPreview*  
public void **restartPreview**( )

- **Description**

Se restaura la preview

- *setCameraParameters*  
private void **setCameraParameters**( )

- **Description**

Se establecen los parámetros de la cámara

- *setPreviewDisplay*  
private void **setPreviewDisplay**( android.view.SurfaceHolder **holder** )

- **Description**

Establecemos la vista

- **Parameters**

- **holder** – Holder a establecer

- *showCameraErrorAndFinish*  
private void **showCameraErrorAndFinish**( )

- **Description**

Mostrar el error de la cámara y finalizar

- *startGPSLocation*  
private void **startGPSLocation**( )

- **Description**  
Inicia el servicio de localización utilizando el mejor proveedor, bien sea GPS o a través de la conexión a Internet. La búsqueda se realiza cada 120.000 ms = 2 minutos de tiempo.
- *startPreview*  
`private void startPreview( ) throws com.android.camera.CameraHardwareException`
  - **Description**  
Iniciamos la vista
  - **Throws**
    - `com.android.camera.CameraHardwareException` – En caso de encontrarse algún problema con la cámara del dispositivo
- *stopPreview*  
`private void stopPreview( )`
  - **Description**  
Paramos la vista
- *surfaceChanged*  
`void surfaceChanged( android.view.SurfaceHolder arg0, int arg1, int arg2, int arg3 )`
- *surfaceCreated*  
`void surfaceCreated( android.view.SurfaceHolder arg0 )`
- *surfaceDestroyed*  
`void surfaceDestroyed( android.view.SurfaceHolder arg0 )`
- *update*  
`public void update( com.ivanlorenzo.location.LocationData data )`
  - **Description**  
Establecemos el estado de la aplicación: Marker o GPS y en caso de ser localización, lo mandamos dibujar
- *updateScene*  
`public void updateScene( )`
  - **Description**  
Cualquier manipulación de la escena y de los objetos 3d irá aquí. En nuestro caso tan solo modificaremos la escala por aquella que tengan en cada momento.

## Miembros heredados de la clase `android.app.Activity`

---

- `static void ( )`
- `public void addContentView( android.view.View arg0, android.view.ViewGroup.LayoutParams arg1 )`
- `public void closeContextMenu( )`
- `public void closeOptionsMenu( )`
- `public PendingIntent createPendingResult( int arg0, android.content.Intent arg1, int arg2 )`
- `public static final DEFAULT_KEYS_DIALER`
- `public static final DEFAULT_KEYS_DISABLE`
- `public static final DEFAULT_KEYS_SEARCH_GLOBAL`
- `public static final DEFAULT_KEYS_SEARCH_LOCAL`
- `public static final DEFAULT_KEYS_SHORTCUT`
- `public final void dismissDialog( int arg0 )`
- `public boolean dispatchKeyEvent( android.view.KeyEvent arg0 )`
- `public boolean dispatchPopulateAccessibilityEvent( android.view.accessibility.AccessibilityEvent arg0 )`

```

▪ public boolean dispatchTouchEvent( android.view.MotionEvent arg0 )
▪ public boolean dispatchTrackballEvent( android.view.MotionEvent arg0 )
▪ public View findViewById( int arg0 )
▪ public void finish( )
▪ public void finishActivity( int arg0 )
▪ public void finishActivityFromChild( Activity arg0, int arg1 )
▪ public void finishFromChild( Activity arg0 )
▪ protected static final FOCUSED_STATE_SET
▪ public final Application getApplication( )
▪ public ComponentName getCallingActivity( )
▪ public String getCallingPackage( )
▪ public int getChangingConfigurations( )
▪ public ComponentName getComponentName( )
▪ public View getCurrentFocus( )
▪ public static long getInstanceCount( )
▪ public Intent getIntent( )
▪ public Object getLastNonConfigurationInstance( )
▪ public LayoutInflater getLayoutInflater( )
▪ public String getLocalClassName( )
▪ public MenuInflater getMenuInflater( )
▪ public final Activity getParent( )
▪ public SharedPreferences getPreferences( int arg0 )
▪ public int getRequestedOrientation( )
▪ public Object getSystemService( java.lang.String arg0 )
▪ public int getTaskId( )
▪ public final CharSequence getTitle( )
▪ public final int getTitleColor( )
▪ public final int getVolumeControlStream( )
▪ public int getWallpaperDesiredMinimumHeight( )
▪ public int getWallpaperDesiredMinimumWidth( )
▪ public Window getWindow( )
▪ public WindowManager getWindowManager( )
▪ public boolean hasWindowFocus( )
▪ public final boolean isChild( )
▪ public boolean isFinishing( )
▪ public boolean isTaskRoot( )
▪ public final Cursor managedQuery( android.net.Uri arg0, java.lang.String[]
    arg1, java.lang.String arg2, java.lang.String[] arg3, java.lang.String arg4 )
▪ public boolean moveTaskToBack( boolean arg0 )
▪ protected void onActivityResult( int arg0, int arg1, android.content.Intent arg2
    )
▪ protected void onApplyThemeResource( android.content.res.Resources.Theme
    arg0, int arg1, boolean arg2 )
▪ public void onAttachedToWindow( )
▪ public void onBackPressed( )
▪ protected void onChildTitleChanged( Activity arg0, java.lang.CharSequence arg1
    )
▪ public void onConfigurationChanged( android.content.res.Configuration arg0 )
▪ public void onContentChanged( )
▪ public boolean onContextItemSelected( android.view.MenuItem arg0 )
▪ public void onContextMenuClosed( android.view.Menu arg0 )
▪ protected void onCreate( android.os.Bundle arg0 )
▪ public void onCreateContextMenu( android.view.ContextMenu arg0,
    android.view.View arg1, android.view.ContextMenu.ContextMenuInfo arg2 )
▪ public CharSequence onCreateDescription( )
▪ protected Dialog onCreateDialog( int arg0 )
▪ protected Dialog onCreateDialog( int arg0, android.os.Bundle arg1 )
▪ public boolean onCreateOptionsMenu( android.view.Menu arg0 )
▪ public boolean onCreatePanelMenu( int arg0, android.view.Menu arg1 )
▪ public View onCreatePanelView( int arg0 )
▪ public boolean onCreateThumbnail( android.graphics.Bitmap arg0,
    android.graphics.Canvas arg1 )
▪ public View onCreateView( java.lang.String arg0, android.content.Context arg1,
    android.util.AttributeSet arg2 )
▪ protected void onDestroy( )
▪ public void onDetachedFromWindow( )
▪ public boolean onKeyDown( int arg0, android.view.KeyEvent arg1 )
▪ public boolean onKeyLongPress( int arg0, android.view.KeyEvent arg1 )
▪ public boolean onKeyMultiple( int arg0, int arg1, android.view.KeyEvent arg2 )

```

- public boolean onKeyUp( int arg0, android.view.KeyEvent arg1 )
- public void onLowMemory( )
- public boolean onOptionsItemSelected( int arg0, android.view.MenuItem arg1 )
- public boolean onMenuOpened( int arg0, android.view.Menu arg1 )
- protected void onNewIntent( android.content.Intent arg0 )
- public boolean onOptionsItemSelected( android.view.MenuItem arg0 )
- public void onOptionsMenuClosed( android.view.Menu arg0 )
- public void onPanelClosed( int arg0, android.view.Menu arg1 )
- protected void onPause( )
- protected void onCreate( android.os.Bundle arg0 )
- protected void onResume( )
- protected void onPrepareDialog( int arg0, Dialog arg1 )
- protected void onPrepareDialog( int arg0, Dialog arg1, android.os.Bundle arg2 )
- public boolean onPrepareOptionsMenu( android.view.Menu arg0 )
- public boolean onPreparePanel( int arg0, android.view.View arg1, android.view.Menu arg2 )
- protected void onRestart( )
- protected void onRestoreInstanceState( android.os.Bundle arg0 )
- protected void onResume( )
- public Object onRetainNonConfigurationInstance( )
- protected void onSaveInstanceState( android.os.Bundle arg0 )
- public boolean onSearchRequested( )
- protected void onStart( )
- protected void onStop( )
- protected void onTitleChanged( java.lang.CharSequence arg0, int arg1 )
- public boolean onTouchEvent( android.view.MotionEvent arg0 )
- public boolean onTrackballEvent( android.view.MotionEvent arg0 )
- public void onUserInteraction( )
- protected void onUserLeaveHint( )
- public void onWindowAttributesChanged( android.view.WindowManager.LayoutParams arg0 )
- public void onWindowFocusChanged( boolean arg0 )
- public void openContextMenu( android.view.View arg0 )
- public void openOptionsMenu( )
- public void overridePendingTransition( int arg0, int arg1 )
- public void registerForContextMenu( android.view.View arg0 )
- public final void removeDialog( int arg0 )
- public final boolean requestWindowFeature( int arg0 )
- public static final RESULT\_CANCELED
- public static final RESULT\_FIRST\_USER
- public static final RESULT\_OK
- public final void runOnUiThread( java.lang.Runnable arg0 )
- public void setContentView( int arg0 )
- public void setContentView( android.view.View arg0 )
- public void setContentView( android.view.View arg0, android.view.ViewGroup.LayoutParams arg1 )
- public final void setDefaultKeyMode( int arg0 )
- public final void setFeatureDrawable( int arg0, android.graphics.drawable.Drawable arg1 )
- public final void setFeatureDrawableAlpha( int arg0, int arg1 )
- public final void setFeatureDrawableResource( int arg0, int arg1 )
- public final void setFeatureDrawableUri( int arg0, android.net.Uri arg1 )
- public void setIntent( android.content.Intent arg0 )
- public void setPersistent( boolean arg0 )
- public final void setProgress( int arg0 )
- public final void setProgressBarIndeterminate( boolean arg0 )
- public final void setProgressBarIndeterminateVisibility( boolean arg0 )
- public final void setProgressBarVisibility( boolean arg0 )
- public void setRequestedOrientation( int arg0 )
- public final void setResult( int arg0 )
- public final void setResult( int arg0, android.content.Intent arg1 )
- public final void setSecondaryProgress( int arg0 )
- public void setTitle( java.lang.CharSequence arg0 )
- public void setTitle( int arg0 )
- public void setTitleColor( int arg0 )
- public void setVisible( boolean arg0 )
- public final void setVolumeControlStream( int arg0 )
- public final void showDialog( int arg0 )
- public final boolean showDialog( int arg0, android.os.Bundle arg1 )
- public void startActivity( android.content.Intent arg0 )

- `public void startActivityResult( android.content.Intent arg0, int arg1 )`
- `public void startActivityFromChild( Activity arg0, android.content.Intent arg1, int arg2 )`
- `public boolean startActivityIfNeeded( android.content.Intent arg0, int arg1 )`
- `public void startIntentSender( android.content.IntentSender arg0, android.content.Intent arg1, int arg2, int arg3, int arg4 ) throws android.content.IntentSender.SendIntentException`
- `public void startIntentSenderForResult( android.content.IntentSender arg0, int arg1, android.content.Intent arg2, int arg3, int arg4, int arg5 ) throws android.content.IntentSender.SendIntentException`
- `public void startIntentSenderFromChild( Activity arg0, android.content.IntentSender arg1, int arg2, android.content.Intent arg3, int arg4, int arg5, int arg6 ) throws android.content.IntentSender.SendIntentException`
- `public void startManagingCursor( android.database.Cursor arg0 )`
- `public boolean startNextMatchingActivity( android.content.Intent arg0 )`
- `public void startSearch( java.lang.String arg0, boolean arg1, android.os.Bundle arg2, boolean arg3 )`
- `public void stopManagingCursor( android.database.Cursor arg0 )`
- `public void takeKeyEvents( boolean arg0 )`
- `public void triggerSearch( java.lang.String arg0, android.os.Bundle arg1 )`
- `public void unregisterForContextMenu( android.view.View arg0 )`

### Miembros heredados de la clase `android.view.ContextThemeWrapper`

---

- `protected void attachBaseContext( android.content.Context arg0 )`
- `public Object getSystemService( java.lang.String arg0 )`
- `public Resources.Theme getTheme( )`
- `protected void onApplyThemeResource( android.content.res.Resources.Theme arg0, int arg1, boolean arg2 )`
- `public void setTheme( int arg0 )`

### Miembros heredados de la clase `android.content.ContextWrapper`

---

- `protected void attachBaseContext( Context arg0 )`
- `public boolean bindService( Intent arg0, ServiceConnection arg1, int arg2 )`
- `public int checkCallingOrSelfPermission( java.lang.String arg0 )`
- `public int checkCallingOrSelfUriPermission( android.net.Uri arg0, int arg1 )`
- `public int checkCallingPermission( java.lang.String arg0 )`
- `public int checkCallingUriPermission( android.net.Uri arg0, int arg1 )`
- `public int checkPermission( java.lang.String arg0, int arg1, int arg2 )`
- `public int checkUriPermission( android.net.Uri arg0, int arg1, int arg2, int arg3 )`
- `public int checkUriPermission( android.net.Uri arg0, java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5 )`
- `public void clearWallpaper( ) throws java.io.IOException`
- `public Context createPackageContext( java.lang.String arg0, int arg1 ) throws pm.PackageManager.NameNotFoundException`
- `public String databaseList( )`
- `public boolean deleteDatabase( java.lang.String arg0 )`
- `public boolean deleteFile( java.lang.String arg0 )`
- `public void enforceCallingOrSelfPermission( java.lang.String arg0, java.lang.String arg1 )`
- `public void enforceCallingOrSelfUriPermission( android.net.Uri arg0, int arg1, java.lang.String arg2 )`
- `public void enforceCallingPermission( java.lang.String arg0, java.lang.String arg1 )`
- `public void enforceCallingUriPermission( android.net.Uri arg0, int arg1, java.lang.String arg2 )`

- `public void enforcePermission( java.lang.String arg0, int arg1, int arg2, java.lang.String arg3 )`
- `public void enforceUriPermission( android.net.Uri arg0, int arg1, int arg2, int arg3, java.lang.String arg4 )`
- `public void enforceUriPermission( android.net.Uri arg0, java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5, java.lang.String arg6 )`
- `public String fileList( )`
- `public Context getApplicationContext( )`
- `public ApplicationInfo getApplicationInfo( )`
- `public AssetManager getAssets( )`
- `public Context getBaseContext( )`
- `public File getCacheDir( )`
- `public ClassLoader getClassLoader( )`
- `public ContentResolver getContentResolver( )`
- `public File getDatabasePath( java.lang.String arg0 )`
- `public File getDir( java.lang.String arg0, int arg1 )`
- `public File getExternalCacheDir( )`
- `public File getExternalFilesDir( java.lang.String arg0 )`
- `public File getFilesDir( )`
- `public File getFileStreamPath( java.lang.String arg0 )`
- `public Looper getMainLooper( )`
- `public String getPackageCodePath( )`
- `public PackageManager getPackageManager( )`
- `public String getPackageName( )`
- `public String getPackageResourcePath( )`
- `public Resources getResources( )`
- `public SharedPreferences getSharedPreferences( java.lang.String arg0, int arg1 )`
- `public Object getSystemService( java.lang.String arg0 )`
- `public Resources.Theme getTheme( )`
- `public Drawable getWallpaper( )`
- `public int getWallpaperDesiredMinimumHeight( )`
- `public int getWallpaperDesiredMinimumWidth( )`
- `public void grantUriPermission( java.lang.String arg0, android.net.Uri arg1, int arg2 )`
- `public boolean isRestricted( )`
- `public FileInputStream openFileInput( java.lang.String arg0 ) throws java.io.FileNotFoundException`
- `public FileOutputStream openFileOutput( java.lang.String arg0, int arg1 ) throws java.io.FileNotFoundException`
- `public SQLiteDatabase openOrCreateDatabase( java.lang.String arg0, int arg1, android.database.sqlite.SQLiteDatabase.CursorFactory arg2 )`
- `public Drawable peekWallpaper( )`
- `public Intent registerReceiver( BroadcastReceiver arg0, IntentFilter arg1 )`
- `public Intent registerReceiver( BroadcastReceiver arg0, IntentFilter arg1, java.lang.String arg2, android.os.Handler arg3 )`
- `public void removeStickyBroadcast( Intent arg0 )`
- `public void revokeUriPermission( android.net.Uri arg0, int arg1 )`
- `public void sendBroadcast( Intent arg0 )`
- `public void sendBroadcast( Intent arg0, java.lang.String arg1 )`
- `public void sendOrderedBroadcast( Intent arg0, java.lang.String arg1 )`
- `public void sendOrderedBroadcast( Intent arg0, java.lang.String arg1, BroadcastReceiver arg2, android.os.Handler arg3, int arg4, java.lang.String arg5, android.os.Bundle arg6 )`
- `public void sendStickyBroadcast( Intent arg0 )`
- `public void sendStickyOrderedBroadcast( Intent arg0, BroadcastReceiver arg1, android.os.Handler arg2, int arg3, java.lang.String arg4, android.os.Bundle arg5 )`
- `public void setTheme( int arg0 )`
- `public void setWallpaper( android.graphics.Bitmap arg0 ) throws java.io.IOException`
- `public void setWallpaper( java.io.InputStream arg0 ) throws java.io.IOException`
- `public void startActivity( Intent arg0 )`
- `public boolean startInstrumentation( ComponentName arg0, java.lang.String arg1, android.os.Bundle arg2 )`
- `public void startIntentSender( IntentSender arg0, Intent arg1, int arg2, int arg3, int arg4 ) throws IntentSender.SendIntentException`
- `public ComponentName startService( Intent arg0 )`
- `public boolean stopService( Intent arg0 )`
- `public void unbindService( ServiceConnection arg0 )`
- `public void unregisterReceiver( BroadcastReceiver arg0 )`

---

## Miembros heredados de la clase android.content.Context

---

```

▪ public static final ACCESSIBILITY_SERVICE
▪ public static final ACCOUNT_SERVICE
▪ public static final ACTIVITY_SERVICE
▪ public static final ALARM_SERVICE
▪ public static final AUDIO_SERVICE
▪ public static final BIND_AUTO_CREATE
▪ public static final BIND_DEBUG_UNBIND
▪ public static final BIND_NOT_FOREGROUND
▪ public abstract boolean bindService( Intent arg0, ServiceConnection arg1, int
  arg2 )
▪ public abstract int checkCallingOrSelfPermission( java.lang.String arg0 )
▪ public abstract int checkCallingOrSelfUriPermission( android.net.Uri arg0, int
  arg1 )
▪ public abstract int checkCallingPermission( java.lang.String arg0 )
▪ public abstract int checkCallingUriPermission( android.net.Uri arg0, int arg1 )
▪ public abstract int checkPermission( java.lang.String arg0, int arg1, int arg2 )
▪ public abstract int checkUriPermission( android.net.Uri arg0, int arg1, int
  arg2, int arg3 )
▪ public abstract int checkUriPermission( android.net.Uri arg0, java.lang.String
  arg1, java.lang.String arg2, int arg3, int arg4, int arg5 )
▪ public abstract void clearWallpaper( ) throws java.io.IOException
▪ public static final CLIPBOARD_SERVICE
▪ public static final CONNECTIVITY_SERVICE
▪ public static final CONTEXT_IGNORE_SECURITY
▪ public static final CONTEXT_INCLUDE_CODE
▪ public static final CONTEXT_RESTRICTED
▪ public abstract Context createPackageContext( java.lang.String arg0, int arg1 )
  throws pm.PackageManager.NameNotFoundException
▪ public abstract String databaseList( )
▪ public abstract boolean deleteDatabase( java.lang.String arg0 )
▪ public abstract boolean deleteFile( java.lang.String arg0 )
▪ public static final DEVICE_POLICY_SERVICE
▪ public static final DROPBOX_SERVICE
▪ public abstract void enforceCallingOrSelfPermission( java.lang.String arg0,
  java.lang.String arg1 )
▪ public abstract void enforceCallingOrSelfUriPermission( android.net.Uri arg0,
  int arg1, java.lang.String arg2 )
▪ public abstract void enforceCallingPermission( java.lang.String arg0,
  java.lang.String arg1 )
▪ public abstract void enforceCallingUriPermission( android.net.Uri arg0, int
  arg1, java.lang.String arg2 )
▪ public abstract void enforcePermission( java.lang.String arg0, int arg1, int
  arg2, java.lang.String arg3 )
▪ public abstract void enforceUriPermission( android.net.Uri arg0, int arg1, int
  arg2, int arg3, java.lang.String arg4 )
▪ public abstract void enforceUriPermission( android.net.Uri arg0,
  java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5,
  java.lang.String arg6 )
▪ public abstract String fileList( )
▪ public abstract Context getApplicationContext( )
▪ public abstract ApplicationInfo getApplicationInfo( )
▪ public abstract AssetManager getAssets( )
▪ public abstract File getCacheDir( )
▪ public abstract ClassLoader getClassLoader( )
▪ public abstract ContentResolver getContentResolver( )
▪ public abstract File getDatabasePath( java.lang.String arg0 )
▪ public abstract File getDir( java.lang.String arg0, int arg1 )
▪ public abstract File getExternalCacheDir( )
▪ public abstract File getExternalFilesDir( java.lang.String arg0 )
▪ public abstract File getFilesDir( )
▪ public abstract File getFileStreamPath( java.lang.String arg0 )
▪ public abstract Looper getMainLooper( )
▪ public abstract String getPackageCodePath( )
▪ public abstract PackageManager getPackageManager( )

```

```
▪ public abstract String getPackageName( )
▪ public abstract String getPackageResourcePath( )
▪ public abstract Resources getResources( )
▪ public abstract SharedPreferences getSharedPreferences( java.lang.String arg0,
int arg1 )
▪ public final String getString( int arg0 )
▪ public final String getString( int arg0, java.lang.Object[] arg1 )
▪ public abstract Object getSystemService( java.lang.String arg0 )
▪ public final CharSequence getText( int arg0 )
▪ public abstract Resources.Theme getTheme( )
▪ public abstract Drawable getWallpaper( )
▪ public abstract int getWallpaperDesiredMinimumHeight( )
▪ public abstract int getWallpaperDesiredMinimumWidth( )
▪ public abstract void grantUriPermission( java.lang.String arg0, android.net.Uri
arg1, int arg2 )
▪ public static final INPUT_METHOD_SERVICE
▪ public boolean isRestricted( )
▪ public static final KEYGUARD_SERVICE
▪ public static final LAYOUT_INFLATER_SERVICE
▪ public static final LOCATION_SERVICE
▪ public static final MODE_APPEND
▪ public static final MODE_PRIVATE
▪ public static final MODE_WORLD_READABLE
▪ public static final MODE_WORLD_WRITEABLE
▪ public static final NOTIFICATION_SERVICE
▪ public final TypedArray obtainStyledAttributes( android.util.AttributeSet arg0,
int[] arg1 )
▪ public final TypedArray obtainStyledAttributes( android.util.AttributeSet arg0,
int[] arg1, int arg2, int arg3 )
▪ public final TypedArray obtainStyledAttributes( int[] arg0 )
▪ public final TypedArray obtainStyledAttributes( int arg0, int[] arg1 ) throws
res.Resources.NotFoundException
▪ public abstract FileInputStream openFileInput( java.lang.String arg0 ) throws
java.io.FileNotFoundException
▪ public abstract FileOutputStream openFileOutput( java.lang.String arg0, int
arg1 ) throws java.io.FileNotFoundException
▪ public abstract SQLiteDatabase openOrCreateDatabase( java.lang.String arg0,
int arg1, android.database.sqlite.SQLiteDatabase.CursorFactory arg2 )
▪ public abstract Drawable peekWallpaper( )
▪ public static final POWER_SERVICE
▪ public abstract Intent registerReceiver( BroadcastReceiver arg0, IntentFilter
arg1 )
▪ public abstract Intent registerReceiver( BroadcastReceiver arg0, IntentFilter
arg1, java.lang.String arg2, android.os.Handler arg3 )
▪ public abstract void removeStickyBroadcast( Intent arg0 )
▪ public abstract void revokeUriPermission( android.net.Uri arg0, int arg1 )
▪ public static final SEARCH_SERVICE
▪ public abstract void sendBroadcast( Intent arg0 )
▪ public abstract void sendBroadcast( Intent arg0, java.lang.String arg1 )
▪ public abstract void sendOrderedBroadcast( Intent arg0, java.lang.String arg1 )
▪ public abstract void sendOrderedBroadcast( Intent arg0, java.lang.String arg1,
BroadcastReceiver arg2, android.os.Handler arg3, int arg4, java.lang.String
arg5, android.os.Bundle arg6 )
▪ public abstract void sendStickyBroadcast( Intent arg0 )
▪ public abstract void sendStickyOrderedBroadcast( Intent arg0, BroadcastReceiver
arg1, android.os.Handler arg2, int arg3, java.lang.String arg4,
android.os.Bundle arg5 )
▪ public static final SENSOR_SERVICE
▪ public abstract void setTheme( int arg0 )
▪ public abstract void setWallpaper( android.graphics.Bitmap arg0 ) throws
java.io.IOException
▪ public abstract void setWallpaper( java.io.InputStream arg0 ) throws
java.io.IOException
▪ public abstract void startActivity( Intent arg0 )
▪ public abstract boolean startInstrumentation( ComponentName arg0,
java.lang.String arg1, android.os.Bundle arg2 )
```

- `public abstract void startIntentSender( IntentSender arg0, Intent arg1, int arg2, int arg3, int arg4 ) throws IntentSender.SendIntentException`
- `public abstract ComponentName startService( Intent arg0 )`
- `public abstract boolean stopService( Intent arg0 )`
- `public static final TELEPHONY_SERVICE`
- `public static final UI_MODE_SERVICE`
- `public abstract void unbindService( ServiceConnection arg0 )`
- `public abstract void unregisterReceiver( BroadcastReceiver arg0 )`
- `public static final VIBRATOR_SERVICE`
- `public static final WALLPAPER_SERVICE`
- `public static final WIFI_SERVICE`
- `public static final WINDOW_SERVICE`

#### B.4.14. Clase TFGARActivity.MainHandler

---

Clase manejadora que se utiliza para enviar mensajes al hilo principal de la aplicación

##### Declaración

```
private class TFGARActivity.MainHandler
extends android.os.Handler
```

##### Constructor summary

`TFGARActivity.MainHandler()`

##### Method summary

`handleMessage(Message)`

##### Constructores

---

- *TFGARActivity.MainHandler*  
`private TFGARActivity.MainHandler( )`

##### Métodos

---

- *handleMessage*  
`public void handleMessage( android.os.Message arg0 )`

##### Miembros heredados de la clase android.os.Handler

---

- `public void dispatchMessage( Message arg0 )`
- `public final void dump( android.util.Printer arg0, java.lang.String arg1 )`
- `public final Looper getLooper( )`
- `public void handleMessage( Message arg0 )`
- `public final boolean hasMessages( int arg0 )`
- `public final boolean hasMessages( int arg0, java.lang.Object arg1 )`
- `public final Message obtainMessage( )`
- `public final Message obtainMessage( int arg0 )`
- `public final Message obtainMessage( int arg0, int arg1, int arg2 )`

- `public final Message obtainMessage( int arg0, int arg1, int arg2, java.lang.Object arg3 )`
- `public final Message obtainMessage( int arg0, java.lang.Object arg1 )`
- `public final boolean post( java.lang.Runnable arg0 )`
- `public final boolean postAtFrontOfQueue( java.lang.Runnable arg0 )`
- `public final boolean postAtTime( java.lang.Runnable arg0, long arg1 )`
- `public final boolean postAtTime( java.lang.Runnable arg0, java.lang.Object arg1, long arg2 )`
- `public final boolean postDelayed( java.lang.Runnable arg0, long arg1 )`
- `public final void removeCallbacks( java.lang.Runnable arg0 )`
- `public final void removeCallbacks( java.lang.Runnable arg0, java.lang.Object arg1 )`
- `public final void removeCallbacksAndMessages( java.lang.Object arg0 )`
- `public final void removeMessages( int arg0 )`
- `public final void removeMessages( int arg0, java.lang.Object arg1 )`
- `public final boolean sendEmptyMessage( int arg0 )`
- `public final boolean sendEmptyMessageAtTime( int arg0, long arg1 )`
- `public final boolean sendEmptyMessageDelayed( int arg0, long arg1 )`
- `public final boolean sendMessage( Message arg0 )`
- `public final boolean sendMessageAtFrontOfQueue( Message arg0 )`
- `public boolean sendMessageAtTime( Message arg0, long arg1 )`
- `public final boolean sendMessageDelayed( Message arg0, long arg1 )`
- `public String toString( )`

### B.4.15. Clase TFGARActivity.PreviewCallback

---

Clase utilizada para enviar y recibir copias de los frames mostrados

#### Declaración

```
private final class TFGARActivity.PreviewCallback
extends java.lang.Object
implements android.hardware.Camera.PreviewCallback
```

#### Constructor summary

**TFGARActivity.PreviewCallback()**

#### Method summary

**onPreviewFrame(byte[], Camera)**

#### Constructores

---

- *TFGARActivity.PreviewCallback*  
`private TFGARActivity.PreviewCallback( )`

#### Métodos

---

- *onPreviewFrame*  
`void onPreviewFrame( byte[] arg0, android.hardware.Camera arg1 )`

### B.4.16. Clase `TFGARActivity.ScaleListener`

---

Clase que gestiona el gesto de escalar un objeto

#### Declaración

```
private class TFGARActivity.ScaleListener
extends android.view.ScaleGestureDetector.SimpleOnScaleGestureListener
```

#### Constructor summary

**TFGARActivity.ScaleListener()**

#### Method summary

**onScale(ScaleGestureDetector)**

#### Constructores

---

- *TFGARActivity.ScaleListener*  
`private TFGARActivity.ScaleListener( )`

#### Métodos

---

- *onScale*  
`boolean onScale( android.view.ScaleGestureDetector arg0 )`

Miembros heredados de la clase `android.view.ScaleGestureDetector.SimpleOnScaleGestureListener`

- 
- `public boolean onScale( ScaleGestureDetector arg0 )`
  - `public boolean onScaleBegin( ScaleGestureDetector arg0 )`
  - `public void onScaleEnd( ScaleGestureDetector arg0 )`

### B.4.17. Clase `Util`

---

Clase de utilidades que contendrá un conjunto de métodos estáticos de ayuda a otras clases

#### Declaración

```
public class Util
extends java.lang.Object
```

## Constructor summary

**Util()**

## Method summary

**checkURL(String)** Método que comprueba si una determinada URL está disponible

**getSharedPreferences(Context)** Obtiene las preferencias.

**roundOrientation(int)** Método que cambia la orientación de la pantalla redondeándola

**updatePositionObjectsScene(Scene, float, float, int)** Actualiza las posiciones de todos los objetos de la escena

**updateZoomObjectsScene(Scene, float)** Actualiza la escala de todos los objetos de la escena

## Constructores

---

- *Util*  
`public Util( )`

## Métodos

---

- *checkURL*  
`private static boolean checkURL( java.lang.String url )`
  - **Description**  
Método que comprueba si una determinada URL está disponible
  - **Parameters**
    - `url` – Dirección a comprobar si está activa
  - **Returns** – Verdadero si está disponible, falso en caso contrario
- *getSharedPreferences*  
`public static android.content.SharedPreferences getSharedPreferences( android.content.Context ctx )`
  - **Description**  
Obtiene las preferencias. O bien las definidas y guardadas por el usuario o en caso de no estar disponibles, carga unas por defecto
  - **Parameters**
    - `ctx` – El contexto de la aplicación
  - **Returns** – Las preferencias del usuario
- *roundOrientation*  
`public static int roundOrientation( int orientationInput )`
  - **Description**  
Método que cambia la orientación de la pantalla redondeándola
  - **Parameters**
    - `orientationInput` – Orientación de la pantalla
  - **Returns** – La orientación redondeada

■ *updatePositionObjectsScene*

```
public static void updatePositionObjectsScene( min3d.core.Scene scene, float
mPosX, float mPosY, int mLastOrientation )
```

● **Description**

Actualiza las posiciones de todos los objetos de la escena

● **Parameters**

- `scene` – Escena que contendrá los objetos
- `mPosX` – El desplazamiento en el eje de las X
- `mPosY` – El desplazamiento en el eje de las Y
- `mLastOrientation` – La orientación de la pantalla

■ *updateZoomObjectsScene*

```
public static void updateZoomObjectsScene( min3d.core.Scene scene, float
mScaleFactor )
```

● **Description**

Actualiza la escala de todos los objetos de la escena

● **Parameters**

- `scene` – Escena
- `mScaleFactor` – El factor de escala a aplicar

## B.5. Paquete com.ivanlorenzo.data.objects

### B.5.1. Clase Max3DSObject

---

Clase que representa un objeto con formato 3ds. Esta clase hereda de Object3D

#### Declaración

```
public class Max3DSObject
extends com.ivanlorenzo.data.objects.Object3D (in B.5.3, page 247)
```

#### Constructor summary

**Max3DSObject(int, String, String, String, String, Audio, int)**

Constructor

#### Constructores

---

■ *Max3DSObject*

```
public Max3DSObject( int id, java.lang.String name, java.lang.String
description, java.lang.String file, java.lang.String texture, audio.Audio audio,
int order )
```

● **Description**

Constructor

● **Parameters**

- `id` – Identificador del objeto
- `name` – Nombre del objeto

- **description** – Descripción del objeto
- **file** – Fichero 3d
- **texture** – Imagen de textura
- **audio** – Audio del objeto
- **order** – Número de orden en el que almacenará

**Miembros heredados de la clase `com.ivanlorenzo.data.objects.Object3D`** (in B.5.3, page 247)

---

- **audio**
- **description**
- **file3D**
- `public Audio getAudio( )`
- `public int getAudioID( )`
- `public String getDescription( )`
- `public String getFile( )`
- `public String getFileWithoutExtension( )`
- `public int getId( )`
- `public String getName( )`
- `public int getOrder( )`
- `public String getTexture( )`
- `public Parser.Type getType( )`
- `public boolean has3DFile( )`
- **id**
- **name**
- **order**
- **texture**

## B.5.2. Clase MD2Object

---

Clase que representa un objeto con formato md2. Esta clase hereda de `Object3D`

### Declaración

```
public class MD2Object
extends com.ivanlorenzo.data.objects.Object3D (in B.5.3, page 247)
```

### Constructor summary

**MD2Object(int, String, String, String, String, Audio, int)**  
Constructor

### Constructores

---

- *MD2Object*  
`public MD2Object( int id, java.lang.String name, java.lang.String description, java.lang.String file, java.lang.String texture, audio.Audio audio, int order )`
  - **Description**  
Constructor
  - **Parameters**
    - **id** – Identificador del objeto
    - **name** – Nombre del objeto

- **description** – Descripción del objeto
- **file** – Fichero 3d
- **texture** – Imagen de textura
- **audio** – Audio del objeto
- **order** – Número de orden en el que almacenará

**Miembros heredados de la clase `com.ivanlorenzo.data.objects.Object3D`** (in B.5.3, page 247)

---

- **audio**
- **description**
- **file3D**
- `public Audio getAudio( )`
- `public int getAudioID( )`
- `public String getDescription( )`
- `public String getFile( )`
- `public String getFileWithoutExtension( )`
- `public int getId( )`
- `public String getName( )`
- `public int getOrder( )`
- `public String getTexture( )`
- `public Parser.Type getType( )`
- `public boolean has3DFile( )`
- **id**
- **name**
- **order**
- **texture**

### B.5.3. Clase Object3D

---

Clase abstracta que representa un objeto 3d proveniente de los ficheros de configuración

#### Declaración

```
public abstract class Object3D
extends java.lang.Object
```

#### Field summary

**audio** Audio  
**description** Descripción sobre el objeto  
**file3D** Fichero 3D  
**id** Identificador del objeto  
**name** Nombre del objeto  
**order** Orden en que el objeto se almacena  
**texture** Textura del objeto

#### Constructor summary

**Object3D(int, String, String, String, String, Audio, int)** Constructor de la clase

## Method summary

**getAudio()** Devuelve el audio  
**getAudioID()** Devuelve el identificador del audio  
**getDescription()** Devuelve la descripción  
**getFile()** Devuelve el nombre del fichero  
**getFileWithoutExtension()** Devuelve el nombre del fichero sin extensión  
**getId()** Devuelve el identificador  
**getName()** Devuelve el nombre  
**getOrder()** Devuelve el orden del objeto  
**getTexture()** Devuelve la textura  
**getType()** Tipo del fichero  
**has3DFile()** Comprueba si el objeto tiene un fichero 3d asociado

## Atributos

---

- **int id**
  - Identificador del objeto
- **java.lang.String name**
  - Nombre del objeto
- **Object3D.File3D file3D**
  - Fichero 3D
- **java.lang.String texture**
  - Textura del objeto
- **java.lang.String description**
  - Descripción sobre el objeto
- **audio.Audio audio**
  - Audio
- **int order**
  - Orden en que el objeto se almacena

## Constructores

---

- *Object3D*
  - public Object3D( int id, java.lang.String name, java.lang.String description, java.lang.String file, java.lang.String texture, audio.Audio audio, int order )**
  - **Description**  
Constructor de la clase
  - **Parameters**
    - **id** – Identificador del objeto
    - **name** – Nombre del objeto
    - **description** – Descripción del objeto
    - **file** – Nombre del fichero
    - **texture** – Textura
    - **audio** – Audio del objeto
    - **order** – Orden en el que se almacena

## Métodos

---

- *getAudio*  
`public audio.Audio getAudio( )`
  - **Description**  
Devuelve el audio
  - **Returns** – Audio del objeto
- *getAudioID*  
`public int getAudioID( )`
  - **Description**  
Devuelve el identificador del audio
  - **Returns** – Identificador del audio
- *getDescription*  
`public java.lang.String getDescription( )`
  - **Description**  
Devuelve la descripción
  - **Returns** – La descripción del objeto
- *getFile*  
`public java.lang.String getFile( )`
  - **Description**  
Devuelve el nombre del fichero
  - **Returns** – Nombre del fichero 3D
- *getFileWithoutExtension*  
`public java.lang.String getFileWithoutExtension( )`
  - **Description**  
Devuelve el nombre del fichero sin extensión
  - **Returns** – Nombre del fichero suprimiendo la extensión
- *getId*  
`public int getId( )`
  - **Description**  
Devuelve el identificador
  - **Returns** – Identificador del objeto
- *getName*  
`public java.lang.String getName( )`
  - **Description**  
Devuelve el nombre
  - **Returns** – Nombre del objeto
- *getOrder*  
`public int getOrder( )`
  - **Description**  
Devuelve el orden del objeto
  - **Returns** – Orden del objeto

- *getTexture*  
`public java.lang.String getTexture( )`
  - **Description**  
Devuelve la textura
  - **Returns** – La textura del objeto
- *getType*  
`public com.ivanlorenzo.parser.Parser.Type getType( )`
  - **Description**  
Tipo del fichero
  - **Returns** – El tipo del fichero
- *has3DFile*  
`public boolean has3DFile( )`
  - **Description**  
Comprueba si el objeto tiene un fichero 3d asociado
  - **Returns** – Verdadero si dispone de un fichero 3d, falso en caso contrario

#### B.5.4. Clase Object3D.File3D

---

Clase que representa un fichero 3D

##### Declaración

```
class Object3D.File3D
extends java.lang.Object
```

##### Field summary

**name** Nombre del fichero 3D  
**type** Tipo del fichero 3D

##### Constructor summary

**Object3D.File3D(String, Parser.Type)** Constructor

##### Method summary

**getName()**  
**getType()**

##### Atributos

---

- `private java.lang.String name`
  - Nombre del fichero 3D
- `private com.ivanlorenzo.parser.Parser.Type type`
  - Tipo del fichero 3D

---

## Constructores

---

- *Object3D.File3D*  
`public Object3D.File3D( java.lang.String name,  
com.ivanlorenzo.parser.Parser.Type type )`
  - **Description**  
Constructor
  - **Parameters**
    - `name` – Nombre del fichero
    - `type` – Extensión del fichero

---

## Métodos

---

- *getName*  
`public java.lang.String getName( )`
- *getType*  
`public com.ivanlorenzo.parser.Parser.Type getType( )`

### B.5.5. Clase Object3DCreator

---

Clase factoría que será la encargada de crear un objeto determinado en función de su tipo. Para esto utiliza alguno de sus métodos estáticos

#### Declaración

```
public final class Object3DCreator  
extends java.lang.Object
```

#### Constructor summary

**Object3DCreator()** Constructor privado para que no se puedan crear instancias de esta clase

#### Method summary

**createObject(Resources, int, String, String, String, String, String, String, int)** Método encargado de crear un objeto determinado en función del tipo del fichero leído

**generateAudio(Resources, String, int)** Método que genera un Audio concreto

## Constructores

---

### ■ *Object3DCreator*

`private Object3DCreator( )`

- **Description**

Constructor privado para que no se puedan crear instancias de esta clase

## Métodos

---

### ■ *createObject*

```
public static Object3D createObject( android.content.res.Resources res, int id,
java.lang.String name, java.lang.String description, java.lang.String file,
java.lang.String type, java.lang.String texture, java.lang.String mediaPlayer,
int order ) throws
com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException
```

- **Description**

Método encargado de crear un objeto determinado en función del tipo del fichero leído

- **Parameters**

- `res` – Recursos de la aplicación
- `id` – Identificador del objeto
- `name` – Nombre del objeto
- `description` – Descripción del objeto
- `file` – Archivo leído
- `type` – Tipo del fichero
- `texture` – Imagen de textura
- `mediaPlayer` – Audio asignado al objeto
- `order` – Orden en que se almacenará

- **Returns** – El objeto creado

- **Throws**

- `com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException` – Se lanzará una excepción si no coincide con ninguno de los tipos de objetos soportados en la aplicación

### ■ *generateAudio*

```
private static audio.Audio generateAudio( android.content.res.Resources res,
java.lang.String audio, int id )
```

- **Description**

Método que genera un Audio concreto

- **Parameters**

- `res` – Recursos de la aplicación
- `audio` – URL del fichero de sonido
- `id` – Identificador del objeto al que se asociará el audio

- **Returns** – Una instancia concreta de Audio

## B.5.6. Clase OBJObject

---

Clase que representa un objeto con formato obj. Esta clase hereda de Ojbect3D

## Declaración

```
public class OBJObject
extends com.ivanlorenzo.data.objects.Object3D (in B.5.3, page 247)
```

## Constructor summary

**OBJObject(int, String, String, String, String, Audio, int)**  
Constructor

## Constructores

---

- *OBJObject*  
`public OBJObject( int id, java.lang.String name, java.lang.String description, java.lang.String file, java.lang.String texture, audio.Audio audio, int order )`
  - **Description**  
Constructor
  - **Parameters**
    - `id` – Identificador del objeto
    - `name` – Nombre del objeto
    - `description` – Descripción del objeto
    - `file` – Fichero 3d
    - `texture` – Imagen de textura
    - `audio` – Audio del objeto
    - `order` – Número de orden en el que almacenará

Miembros heredados de la clase `com.ivanlorenzo.data.objects.Object3D` (in B.5.3, page 247)

---

- `audio`
- `description`
- `file3D`
- `public Audio getAudio( )`
- `public int getAudioID( )`
- `public String getDescription( )`
- `public String getFile( )`
- `public String getFileWithoutExtension( )`
- `public int getId( )`
- `public String getName( )`
- `public int getOrder( )`
- `public String getTexture( )`
- `public Parser.Type getType( )`
- `public boolean has3DFile( )`
- `id`
- `name`
- `order`
- `texture`

## B.6. Paquete com.ivanlorenzo.data.exceptions

## B.7. Paquete com.ivanlorenzo.location

### B.7.1. Clase ARLocationListener

---

Clase "sujeto observable" del patrón observer que, a su vez, implemente `LocationListener` y recibe cuándo se produce un cambio en la posición GPS y en caso de que esta posición contenga cercano una localización de los ficheros de configuración, notificará a todos los observadores que debe mostrarse la información asociada a dicha posición.

#### Declaración

```
public class ARLocationListener
extends com.ivanlorenzo.location.observer.ARObservable (in B.3.3, page 213)
implements android.location.LocationListener
```

#### Field summary

**distance** Distancia máxima a la que mostraremos la información de las posiciones

#### Constructor summary

**ARLocationListener(IARObserver, float)** Constructor de la clase que recibe un observador y la distancia máxima de la nuestra a la que mostrar información

#### Method summary

**getLocationData(Location)** Devuelve el estado de la localización.  
**onLocationChanged(Location)**  
**onProviderDisabled(String)**  
**onProviderEnabled(String)**  
**onStatusChanged(String, int, Bundle)**

#### Atributos

---

- private float **distance**
  - Distancia máxima a la que mostraremos la información de las posiciones

---

## Constructores

---

- *ARLocationListener*

```
public ARLocationListener( observer.IARObserver observer, float distance )
```

- **Description**

Constructor de la clase que recibe un observador y la distancia máxima de la nuestra a la que mostrar información

- **Parameters**

- **observer** – Observador
    - **distance** – Distancia máxima de la nuestra a la que mostrar información

---

## Métodos

---

- *getLocationData*

```
private LocationData getLocationData( android.location.Location currentLocation )
```

- **Description**

Devuelve el estado de la localización. En caso que exista alguna localización cercana, devolverá el objeto asociado junto con el estado.

- **Parameters**

- **currentLocation** – Nuestra posición actual

- **Returns** – El estado de la localización

- *onLocationChanged*

```
void onLocationChanged( android.location.Location arg0 )
```

- *onProviderDisabled*

```
void onProviderDisabled( java.lang.String arg0 )
```

- *onProviderEnabled*

```
void onProviderEnabled( java.lang.String arg0 )
```

- *onStatusChanged*

```
void onStatusChanged( java.lang.String arg0, int arg1, android.os.Bundle arg2 )
```

Miembros heredados de la clase `com.ivanlorenzo.location.observer.ARObservable` (in B.3.3, page 213)

---

- `public void addObserver( IARObserver observer )`
- `protected data`
- `public void notifyObservers( )`
- `protected observers`
- `public void removeObserver( IARObserver observer )`

### B.7.2. Clase LocationData

---

Clase que representa el estado de la localización. Se implementa como un Singleton ya que sólo queremos tener una única instancia

## Declaración

```
public class LocationData
extends java.lang.Object
```

## Field summary

**instance** Instancia de la clase  
**object** Objeto 3D que se está mostrando  
**status** Estado del sistema de realidad aumentada

## Constructor summary

**LocationData()**

## Method summary

**getInstance()** Devuelve la única instancia de la clase  
**getObject()** Devuelve el objeto 3d que se mostrará  
**getStatus()** Devuelve el estado del sistema  
**setObject(Object3D)** Establece el objeto 3D que se muestra  
**setStatus(ARStatus)** Establece el estado del sistema

## Atributos

---

- private static LocationData **instance**
  - Instancia de la clase
- private com.ivanlorenzo.gui.ARStatus **status**
  - Estado del sistema de realidad aumentada
- private com.ivanlorenzo.data.objects.Object3D **object**
  - Objeto 3D que se está mostrando

## Constructores

---

- *LocationData*  
public **LocationData**( )

## Métodos

---

- *getInstance*  
public static LocationData **getInstance**( )
  - **Description**  
Devuelve la única instancia de la clase
  - **Returns** – Instancia

- *getObject*  
`public com.ivanlorenzo.data.objects.Object3D getObject( )`
  - **Description**  
Devuelve el objeto 3d que se mostrará
  - **Returns** – El objeto a mostrar
- *getStatus*  
`public com.ivanlorenzo.gui.ARStatus getStatus( )`
  - **Description**  
Devuelve el estado del sistema
  - **Returns** – Estado
- *setObject*  
`public void setObject( com.ivanlorenzo.data.objects.Object3D object )`
  - **Description**  
Establece el objeto 3D que se muestra
  - **Parameters**
    - `object` – Objeto 3D a mostrar
- *setStatus*  
`public void setStatus( com.ivanlorenzo.gui.ARStatus status )`
  - **Description**  
Establece el estado del sistema
  - **Parameters**
    - `status` – Estado establecido

## B.8. Paquete com.ivanlorenzo.gui.dialogs

### B.8.1. Clase DialogType

---

Enumeración de los tipos de diálogos de la aplicación

#### Declaración

```
public final class DialogType
extends java.lang.Enum
```

#### Field summary

**DIALOG\_ERROR\_CONNECTION**  
**DIALOG\_EXIT**  
**DIALOG\_LOADING**

#### Constructor summary

**DialogType()**

## Method summary

**valueOf(String)**  
**values()**

## Atributos

---

- public static final DialogType **DIALOG\_LOADING**
- public static final DialogType **DIALOG\_ERROR\_CONNECTION**
- public static final DialogType **DIALOG\_EXIT**

## Constructores

---

- *DialogType*  
private DialogType( )

## Métodos

---

- *valueOf*  
public static DialogType valueOf( java.lang.String name )
- *values*  
public static DialogType[] values( )

## Miembros heredados de la clase java.lang.Enum

---

- protected final Object clone( ) throws CloneNotSupportedException
- public final int compareTo( Enum arg0 )
- public final boolean equals( Object arg0 )
- protected final void finalize( )
- public final Class getDeclaringClass( )
- public final int hashCode( )
- private final name
- public final String name( )
- private final ordinal
- public final int ordinal( )
- private void readObject( java.io.ObjectInputStream arg0 ) throws java.io.IOException, java.lang.ClassNotFoundException
- private void readObjectNoData( ) throws java.io.ObjectStreamException
- public String toString( )
- public static Enum valueOf( Class arg0, String arg1 )

## B.9. Paquete com.ivanlorenzo.data

### B.9.1. Clase AugmentedRealityXMLCompiler

---

Clase hija de XMLCompiler encargada de compilar el fichero XML que contiene la información sobre los patrones (markers) y las posiciones GPS

## Declaración

```
public class AugmentedRealityXMLCompiler
extends com.ivanlorenzo.data.XMLCompiler (in B.9.7, page 270)
```

## Constructor summary

### AugmentedRealityXMLCompiler()

## Method summary

**load(Resources, String, String, String, String)** Método que carga los patrones

**scanDocument(Resources, String, String)** Busca todos los patrones y localizaciones del fichero XML y los guarda en la base de conocimiento

## Constructores

---

- *AugmentedRealityXMLCompiler*  
`public AugmentedRealityXMLCompiler( )`

## Métodos

---

- *load*  
`public void load( android.content.res.Resources res, java.lang.String urlXMLPatterns, java.lang.String urlXSDPatterns, java.lang.String urlXMLObjects, java.lang.String urlXSDObjects ) throws com.ivanlorenzo.data.exceptions.LoadXMLException, javax.xml.xpath.XPathExpressionException, com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException, java.io.IOException`
  - **Description**  
Método que carga los patrones
  - **Parameters**
    - `res` – Recursos de la aplicación
    - `urlXMLPatterns` – Ruta del fichero XML de patrones
    - `urlXSDPatterns` – Ruta del fichero XSD de patrones
    - `urlXMLObjects` – Ruta del fichero XML de objetos
    - `urlXSDObjects` – Ruta del fichero XSD de objetos
  - **Throws**
    - `com.ivanlorenzo.data.exceptions.LoadXMLException` – Si hay errores en la carga
    - `javax.xml.xpath.XPathExpressionException` – Si se produce un error en una expresión XPath
    - `com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException` – Si el objeto no tiene una extensión 3D válida
    - `java.io.IOException` – En caso de que no se pueda acceder a la URL lanzaría una `IOException`

■ *scanDocument*

```
private void scanDocument( android.content.res.Resources res, java.lang.String
urlXMLObjects, java.lang.String urlXSDOBJECTS ) throws
com.ivanlorenzo.data.exceptions.LoadXMLException,
javax.xml.xpath.XPathExpressionException,
com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException,
java.io.IOException
```

• **Description**

Busca todos los patrones y localizaciones del fichero XML y los guarda en la base de conocimiento

• **Parameters**

- **res** – Recursos de la aplicación
- **urlXMLObjects** – Ruta del fichero XML de objetos
- **urlXSDOBJECTS** – Ruta del fichero XSD de objetos

• **Throws**

- **com.ivanlorenzo.data.exceptions.LoadXMLException** – Si hay errores en la carga
- **javax.xml.xpath.XPathExpressionException** – Si se produce un error en una expresión XPath
- **com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException** – Si el objeto no tiene una extensión 3D válida
- **java.io.IOException** – En caso de que no se pueda acceder a la URL lanzaría una IOException

Miembros heredados de la clase `com.ivanlorenzo.data.XMLCompiler` (in B.9.7, page 270)

---

- **protected document**
- **private** `InputStream` **getInputStream**( `java.net.URL url` ) **throws** `java.io.IOException`
- **protected** `void` **loadFile**( `boolean validate` ) **throws** `exceptions.LoadXMLException`, `java.io.IOException`
- **protected xml**
- **protected xsd**

## B.9.2. *Clase* CompilerGenerator

---

Clase que es la encargada de iniciar la carga de ficheros

### Declaración

```
public class CompilerGenerator
extends java.lang.Object
```

### Field summary

**BASE** Instancia del gestor de objetos

### Constructor summary

**CompilerGenerator()**

## Method summary

**compile(Resources, String, String, String, String)** Método que se llamará desde para cargar los datos

## Atributos

---

- public static final KnowledgeBase **BASE**
  - Instancia del gestor de objetos

## Constructores

---

- *CompilerGenerator*  
public **CompilerGenerator**( )

## Métodos

---

- *compile*  
public static void **compile**( android.content.res.Resources res, java.lang.String urlXMLPatterns, java.lang.String urlXSDPatterns, java.lang.String urlXMLObjects, java.lang.String urlXSDObjets ) throws com.ivanlorenzo.data.exceptions.LoadXMLException, javax.xml.xpath.XPathExpressionException, com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException, java.io.IOException
  - **Description**  
Método que se llamará desde para cargar los datos
  - **Parameters**
    - res – Recursos de la aplicación
    - urlXMLPatterns – Fichero XML con los patrones
    - urlXSDPatterns – Fichero XSD de los patrones
    - urlXMLObjects – Fichero XML con los objetos
    - urlXSDObjets – Fichero XSD de los objetos
  - **Throws**
    - com.ivanlorenzo.data.exceptions.LoadXMLException – Si hay algún error en la carga
    - javax.xml.xpath.XPathExpressionException – Si se produce un error en una expresión XPath
    - com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException – Si el objeto no tiene una extensión 3D válida
    - java.io.IOException – En caso de que no se pueda acceder a la URL lanzaría una IOException

### B.9.3. Clase GeoLocation

---

Clase que se corresponde con una localización de los ficheros de configuración de la aplicación. Tendrá un identificador, un nombre, una localización GPS y un objeto asociado

## Declaración

```
public class GeoLocation
extends java.lang.Object
```

## Field summary

**id** Identificador de la localización  
**idObject** Identificador del objeto asociado  
**location** Localización GPS  
**name** Nombre del lugar

## Constructor summary

**GeoLocation(int, String, double, double, int)** Constructor en el que se establecen los atributos

## Method summary

**getId()** Devuelve el identificador de la localización  
**getIdObject()** Devuelve el identificador del objeto asociado  
**getLocation()** Devuelve la localización  
**getName()** Devuelve el nombre del lugar

## Atributos

---

- `private int id`
  - Identificador de la localización
- `private java.lang.String name`
  - Nombre del lugar
- `private android.location.Location location`
  - Localización GPS
- `private int idObject`
  - Identificador del objeto asociado

## Constructores

---

- *GeoLocation*

```
public GeoLocation( int id, java.lang.String name, double latitude, double longitude, int idObject )
```

  - **Description**

Constructor en el que se establecen los atributos
  - **Parameters**
    - **id** – Identificador de la localización
    - **name** – Nombre del lugar

- **latitude** – Latitud geográfica
- **longitude** – Longitud geográfica
- **idObject** – Identificador del objeto asociado

## Métodos

---

- *getId*  
`public int getId( )`
  - **Description**  
Devuelve el identificador de la localización
  - **Returns** – Identificador de la localización
- *getIdObject*  
`public int getIdObject( )`
  - **Description**  
Devuelve el identificador del objeto asociado
  - **Returns** – Identificador del objeto asociado
- *getLocation*  
`public android.location.Location getLocation( )`
  - **Description**  
Devuelve la localización
  - **Returns** – Localización
- *getName*  
`public java.lang.String getName( )`
  - **Description**  
Devuelve el nombre del lugar
  - **Returns** – Nombre del lugar

### B.9.4. Clase KnowledgeBase

---

Clase Singleton que contiene la base de conocimiento de la aplicación. Contendrá tanto la lista de patrones (markers) como la lista de localizaciones y de objetos.

#### Declaración

```
public class KnowledgeBase
extends java.lang.Object
```

#### Field summary

**instance** Única instancia de la base de conocimiento  
**locations** Lista de geolocalizaciones de la aplicación  
**objects** Lista de objetos cargados en la aplicación  
**patterns** Lista de patrones (markers) de la aplicación

## Constructor summary

**KnowledgeBase()** Constructor sin parámetros que simplemente crea las estructuras de datos para almacenar la información

## Method summary

**clear()** Método que borra las listas de objetos y patrones  
**getInstance()** Método estático que devuelve la única instancia de la clase  
**getLocations()** Listado de localizaciones  
**getObject(int)** Devuelve un objeto buscado por su identificador  
**getObjects()** Devuelve la lista de objetos de la base de conocimiento  
**getObjectsToScene(Resources)** Devuelve un listado de objetos que se deben visualizar en la aplicación  
**getPattern(int)** Devuelve el patrón que se corresponde con un identificador  
**getPatterns()** Obtiene la lista de patrones de la base de conocimiento  
**getPatterns(Resources)** Devuelve una lista de flujo de bytes correspondiente a los patrones  
**saveLocation(GeoLocation)** Guarda una localización GPS  
**saveObject(Object3D)** Guarda un objeto 3d  
**savePattern(Pattern)** Guarda un patrón

## Atributos

---

- private static KnowledgeBase **instance**
  - Única instancia de la base de conocimiento
- private java.util.List **patterns**
  - Lista de patrones (markers) de la aplicación
- private java.util.List **locations**
  - Lista de geolocalizaciones de la aplicación
- private java.util.List **objects**
  - Lista de objetos cargados en la aplicación

## Constructores

---

- *KnowledgeBase*  
private **KnowledgeBase( )**
  - **Description**  
Constructor sin parámetros que simplemente crea las estructuras de datos para almacenar la información

---

## Métodos

---

- *clear*  
`public void clear( )`
  - **Description**  
Método que borra las listas de objetos y patrones
- *getInstance*  
`public static KnowledgeBase getInstance( )`
  - **Description**  
Método estático que devuelve la única instancia de la clase
  - **Returns** – La instancia de la clase
- *getLocations*  
`public java.util.List getLocation( )`
  - **Description**  
Listado de localizaciones
  - **Returns** – Lista de geolocalizaciones de la base de conocimiento
- *getObject*  
`public objects.Object3D getObject( int id )`
  - **Description**  
Devuelve un objeto buscado por su identificador
  - **Parameters**
    - `id` – Identificador del objeto a buscar
  - **Returns** – Objeto buscado
- *getObjects*  
`public java.util.List getObjects( )`
  - **Description**  
Devuelve la lista de objetos de la base de conocimiento
  - **Returns** – Lista de objetos
- *getObjectsToScene*  
`public java.util.List getObjectsToScene( android.content.res.Resources res )`
  - **Description**  
Devuelve un listado de objetos que se deben visualizar en la aplicación
  - **Parameters**
    - `res` – Recursos de la aplicación
  - **Returns** – Listado de objetos para ser visualizados
- *getPattern*  
`public Pattern getPattern( int id )`
  - **Description**  
Devuelve el patrón que se corresponde con un identificador
  - **Parameters**
    - `id` – Identificador del patrón
  - **Returns** – Patrón buscado

- *getPatterns*  
`public java.util.List getPatterns( )`
  - **Description**  
Obtiene la lista de patrones de la base de conocimiento
  - **Returns** – Lista de patrones
- *getPatterns*  
`public java.util.ArrayList getPatterns( android.content.res.Resources res )`
  - **Description**  
Devuelve una lista de flujo de bytes correspondiente a los patrones
  - **Parameters**
    - `res` – Recursos de la aplicación
  - **Returns** – Lista de flujo de bytes correspondientes a los patrones
- *saveLocation*  
`public void saveLocation( GeoLocation location )`
  - **Description**  
Guarda una localización GPS
  - **Parameters**
    - `location` – Localización a guardar
- *saveObject*  
`public void saveObject( objects.Object3D objData )`
  - **Description**  
Guarda un objeto 3d
  - **Parameters**
    - `objData` – Objeto a guardar
- *savePattern*  
`public void savePattern( Pattern patData )`
  - **Description**  
Guarda un patrón
  - **Parameters**
    - `patData` – Patrón a guardar

### B.9.5. Clase ObjectsXMLCompiler

---

Clase hija de XMLCompiler encargada de compilar el fichero XML que contiene la información sobre los objetos

#### Declaración

```
public class ObjectsXMLCompiler
extends com.ivanlorenzo.data.XMLCompiler (in B.9.7, page 270)
```

#### Constructor summary

**ObjectsXMLCompiler()**

## Method summary

**load(Resources, String, String, int, int)** Método que carga la información de un objeto

**searchObject(Resources, int, int)** Busca todos un objeto determinado del fichero XML y lo guarda en la base de conocimiento.

## Constructores

---

- *ObjectsXMLCompiler*  
`public ObjectsXMLCompiler( )`

## Métodos

---

- *load*  

```
public void load( android.content.res.Resources res, java.lang.String
urlXMLObjects, java.lang.String urlXSDObjects, int idObject, int order )
throws com.ivanlorenzo.data.exceptions.LoadXMLException,
javax.xml.xpath.XPathExpressionException,
com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException,
java.io.IOException
```

  - **Description**  
Método que carga la información de un objeto
  - **Parameters**
    - **res** – Recursos de la aplicación
    - **urlXMLObjects** – Ruta del fichero XML de objetos
    - **urlXSDObjects** – Ruta del fichero XSD de objetos
    - **idObject** – Identificador del objeto a buscar
    - **order** – Orden en que se cargará el objeto
  - **Throws**
    - **com.ivanlorenzo.data.exceptions.LoadXMLException** – Si hay errores en la carga
    - **javax.xml.xpath.XPathExpressionException** – Si se produce un error en una expresión XPath
    - **com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException** – Si el objeto no tiene una extensión 3D válida
    - **java.io.IOException** – En caso de que no se pueda acceder a la URL lanzaría una **IOException**
- *searchObject*  

```
private void searchObject( android.content.res.Resources res, int id, int order
) throws javax.xml.xpath.XPathExpressionException,
com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException
```

  - **Description**  
Busca todos un objeto determinado del fichero XML y lo guarda en la base de conocimiento.
  - **Parameters**
    - **res** – Recursos de la aplicación
    - **id** – Identificador del objeto a buscar
    - **order** – Orden en que se cargará el objeto
  - **Throws**

- `javax.xml.xpath.XPathExpressionException` – Si se produce un error en una expresión XPath
- `com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException` – Si el objeto no tiene una extensión 3D válida

**Miembros heredados de la clase `com.ivanlorenzo.data.XMLCompiler`** (in B.9.7, page 270)

---

- `protected document`
- `private InputStream getInputStream( java.net.URL url ) throws java.io.IOException`
- `protected void loadFile( boolean validate ) throws exceptions.LoadXMLException, java.io.IOException`
- `protected xml`
- `protected xsd`

### B.9.6. *Clase Pattern*

---

Clase que contiene las propiedades del patrón al que le irá asignado un objeto

#### Declaración

```
public class Pattern
extends java.lang.Object
```

#### Field summary

**id** Identificador del patrón  
**idObject** Identificador del objeto que tiene asignado  
**name** Nombre del patrón

#### Constructor summary

**Pattern()** Constructor sin parámetros  
**Pattern(int, String, int)** Constructor

#### Method summary

**getId()** Devuelve el identificador del patrón  
**getIdObject()** Devuelve el identificador del objeto asignado al patrón  
**getName()** Devuelve el nombre del patrón  
**setId(int)** Establece el identificador del patrón  
**setIdObject(int)** Establece el identificador del objeto asignado al patrón  
**setName(String)** Establece el nombre del patrón

## Atributos

---

- `private int id`
  - Identificador del patrón
- `private java.lang.String name`
  - Nombre del patrón
- `private int idObject`
  - Identificador del objeto que tiene asignado

## Constructores

---

- *Pattern*  
`public Pattern( )`
  - **Description**  
Constructor sin parámetros
- *Pattern*  
`public Pattern( int id, java.lang.String name, int idObject )`
  - **Description**  
Constructor
  - **Parameters**
    - `id` – Identificador del patrón
    - `name` – Nombre del patrón
    - `idObject` – Identificador del objeto que tiene asignado

## Métodos

---

- *getId*  
`public int getId( )`
  - **Description**  
Devuelve el identificador del patrón
  - **Returns** – Identificador del patrón
- *getIdObject*  
`public int getIdObject( )`
  - **Description**  
Devuelve el identificador del objeto asignado al patrón
  - **Returns** – Identificador del objeto asignado al patrón
- *getName*  
`public java.lang.String getName( )`
  - **Description**  
Devuelve el nombre del patrón
  - **Returns** – El nombre del patrón

- *setId*  
`public void setId( int id )`
  - **Description**  
Establece el identificador del patrón
  - **Parameters**
    - `id` – Identificador del patrón
- *setIdObject*  
`public void setIdObject( int idObject )`
  - **Description**  
Establece el identificador del objeto asignado al patrón
  - **Parameters**
    - `idObject` – Identificador del objeto asignado al patrón
- *setName*  
`public void setName( java.lang.String name )`
  - **Description**  
Establece el nombre del patrón
  - **Parameters**
    - `name` – El nombre del patrón

### B.9.7. Clase XMLCompiler

---

Clase genérica abstracta encargada de la compilación de ficheros XML

#### Declaración

```
public abstract class XMLCompiler
extends java.lang.Object
```

#### Field summary

**document** Documento DOM que representa el juego  
**xml** Url del fichero XML a cargar  
**xsd** Url del fichero XSD a cargar

#### Constructor summary

**XMLCompiler()**

#### Method summary

**getInputStream(URL)** Devuelve un flujo de bytes del recurso que se encuentra en la url pasada  
**loadFile(boolean)** Carga en el documento DOM el fichero XML

---

## Atributos

---

- protected java.lang.String **xml**
  - Url del fichero XML a cargar
- protected java.lang.String **xsd**
  - URL del fichero XSD a cargar
- protected org.w3c.dom.Document **document**
  - Documento DOM que representa el juego

---

## Constructores

---

- *XMLCompiler*  
public **XMLCompiler**( )

---

## Métodos

---

- *getInputStream*  
private java.io.InputStream **getInputStream**( java.net.URL url ) throws java.io.IOException
  - **Description**  
Devuelve un flujo de bytes del recurso que se encuentra en la url pasada
  - **Parameters**
    - url – La dirección donde se cuenta el recurso a abrir
  - **Returns** – Flujo de bytes del recurso
  - **Throws**
    - java.io.IOException – En caso de que no se pueda acceder a la URL lanzaría una IOException
- *loadFile*  
protected void **loadFile**( boolean validate ) throws com.ivanlorenzo.data.exceptions.LoadXMLException, java.io.IOException
  - **Description**  
Carga en el documento DOM el fichero XML
  - **Parameters**
    - validate – Si el fichero XML se validará con un DTD
  - **Throws**
    - com.ivanlorenzo.data.exceptions.LoadXMLException – Si hay errores en la carga
    - java.io.IOException – Se puede producir una excepción si no encuentra el recurso en la URL

## B.10. Paquete com.ivanlorenzo.connection

### B.10.1. Clase ConnectionStatus

---

Clase que utilizaremos para comprobar el estado de la conexión a Internet

## Declaración

```
public class ConnectionStatus
extends java.lang.Object
```

## Constructor summary

**ConnectionStatus()**

## Method summary

**checkInternetConnection(Context)** Comprueba si tenemos o no  
conexión a Internet

## Constructores

---

- *ConnectionStatus*  
`public ConnectionStatus( )`

## Métodos

---

- *checkInternetConnection*  
`public static boolean checkInternetConnection( android.content.Context context )`
  - **Description**  
Comprueba si tenemos o no conexión a Internet
  - **Parameters**
    - `context` – El contexto de la aplicación
  - **Returns** – Devuelve verdadero en caso de tener conexión y falso en caso contrario

## B.11. Paquete com.ivanlorenzo.gui.preferences

### B.11.1. Clase TFGPreferencesActivity

---

Activity que gestiona las preferencias del usuario. Carga las preferencias del recurso preferences.xml

## Declaración

```
public class TFGPreferencesActivity
extends android.preference.PreferenceActivity
```

## Constructor summary

**TFGPreferencesActivity()**

## Method summary

**onCreate(Bundle)**

## Constructores

---

- *TFGPreferencesActivity*  
`public TFGPreferencesActivity( )`

## Métodos

---

- *onCreate*  
`protected void onCreate( android.os.Bundle arg0 )`

## Miembros heredados de la clase android.preference.PreferenceActivity

---

- `public void addPreferencesFromIntent( android.content.Intent arg0 )`
- `public void addPreferencesFromResource( int arg0 )`
- `public Preference findPreference( java.lang.CharSequence arg0 )`
- `public PreferenceManager getPreferenceManager( )`
- `public PreferenceScreen getPreferenceScreen( )`
- `protected void onActivityResult( int arg0, int arg1, android.content.Intent arg2 )`
- `public void onContentChanged( )`
- `protected void onCreate( android.os.Bundle arg0 )`
- `protected void onDestroy( )`
- `protected void onNewIntent( android.content.Intent arg0 )`
- `public boolean onPreferenceTreeClick( PreferenceScreen arg0, Preference arg1 )`
- `protected void onRestoreInstanceState( android.os.Bundle arg0 )`
- `protected void onSaveInstanceState( android.os.Bundle arg0 )`
- `protected void onStop( )`
- `public void setPreferenceScreen( PreferenceScreen arg0 )`

## Miembros heredados de la clase android.app.ListActivity

---

- `public ListAdapter getListAdapter( )`
- `public ListView getListView( )`
- `public long getSelectedItemId( )`
- `public int getSelectedItemPosition( )`
- `public void onContentChanged( )`
- `protected void onDestroy( )`
- `protected void onItemClick( android.widget.ListView arg0, android.view.View arg1, int arg2, long arg3 )`
- `protected void onRestoreInstanceState( android.os.Bundle arg0 )`
- `public void setListAdapter( android.widget.ListAdapter arg0 )`
- `public void setSelection( int arg0 )`

## Miembros heredados de la clase android.app.Activity

---

- static void ( )
- public void addContentView( android.view.View arg0, android.view.ViewGroup.LayoutParams arg1 )
- public void closeContextMenu( )
- public void closeOptionsMenu( )
- public PendingIntent createPendingResult( int arg0, android.content.Intent arg1, int arg2 )
- public static final DEFAULT\_KEYS\_DIALER
- public static final DEFAULT\_KEYS\_DISABLE
- public static final DEFAULT\_KEYS\_SEARCH\_GLOBAL
- public static final DEFAULT\_KEYS\_SEARCH\_LOCAL
- public static final DEFAULT\_KEYS\_SHORTCUT
- public final void dismissDialog( int arg0 )
- public boolean dispatchKeyEvent( android.view.KeyEvent arg0 )
- public boolean dispatchPopulateAccessibilityEvent( android.view.accessibility.AccessibilityEvent arg0 )
- public boolean dispatchTouchEvent( android.view.MotionEvent arg0 )
- public boolean dispatchTrackballEvent( android.view.MotionEvent arg0 )
- public View findViewById( int arg0 )
- public void finish( )
- public void finishActivity( int arg0 )
- public void finishActivityFromChild( Activity arg0, int arg1 )
- public void finishFromChild( Activity arg0 )
- protected static final FOCUSED\_STATE\_SET
- public final Application getApplication( )
- public ComponentName getCallingActivity( )
- public String getCallingPackage( )
- public int getChangingConfigurations( )
- public ComponentName getComponentName( )
- public View getCurrentFocus( )
- public static long getInstanceCount( )
- public Intent getIntent( )
- public Object getLastNonConfigurationInstance( )
- public LayoutInflater getLayoutInflater( )
- public String getLocalClassName( )
- public MenuInflater getMenuInflater( )
- public final Activity getParent( )
- public SharedPreferences getPreferences( int arg0 )
- public int getRequestedOrientation( )
- public Object getSystemService( java.lang.String arg0 )
- public int getTaskId( )
- public final CharSequence getTitle( )
- public final int getTitleColor( )
- public final int getVolumeControlStream( )
- public int getWallpaperDesiredMinimumHeight( )
- public int getWallpaperDesiredMinimumWidth( )
- public Window getWindow( )
- public WindowManager getWindowManager( )
- public boolean hasWindowFocus( )
- public final boolean isChild( )
- public boolean isFinishing( )
- public boolean isTaskRoot( )
- public final Cursor managedQuery( android.net.Uri arg0, java.lang.String[] arg1, java.lang.String arg2, java.lang.String[] arg3, java.lang.String arg4 )
- public boolean moveTaskToBack( boolean arg0 )
- protected void onActivityResult( int arg0, int arg1, android.content.Intent arg2 )
- protected void onApplyThemeResource( android.content.res.Resources.Theme arg0, int arg1, boolean arg2 )
- public void onAttachedToWindow( )
- public void onBackPressed( )
- protected void onChildTitleChanged( Activity arg0, java.lang.CharSequence arg1 )
- public void onConfigurationChanged( android.content.res.Configuration arg0 )
- public void onContentChanged( )

```

▪ public boolean onContextItemSelected( android.view.MenuItem arg0 )
▪ public void onContextMenuClosed( android.view.Menu arg0 )
▪ protected void onCreate( android.os.Bundle arg0 )
▪ public void onCreateContextMenu( android.view.ContextMenu arg0,
    android.view.View arg1, android.view.ContextMenu.ContextMenuInfo arg2 )
▪ public CharSequence onCreateDescription( )
▪ protected Dialog onCreateDialog( int arg0 )
▪ protected Dialog onCreateDialog( int arg0, android.os.Bundle arg1 )
▪ public boolean onCreateOptionsMenu( android.view.Menu arg0 )
▪ public boolean onCreatePanelMenu( int arg0, android.view.Menu arg1 )
▪ public View onCreatePanelView( int arg0 )
▪ public boolean onCreateThumbnail( android.graphics.Bitmap arg0,
    android.graphics.Canvas arg1 )
▪ public View onCreateView( java.lang.String arg0, android.content.Context arg1,
    android.util.AttributeSet arg2 )
▪ protected void onDestroy( )
▪ public void onDetachedFromWindow( )
▪ public boolean onKeyDown( int arg0, android.view.KeyEvent arg1 )
▪ public boolean onKeyLongPress( int arg0, android.view.KeyEvent arg1 )
▪ public boolean onKeyMultiple( int arg0, int arg1, android.view.KeyEvent arg2 )
▪ public boolean onKeyUp( int arg0, android.view.KeyEvent arg1 )
▪ public void onLowMemory( )
▪ public boolean onMenuItemSelected( int arg0, android.view.MenuItem arg1 )
▪ public boolean onMenuOpened( int arg0, android.view.Menu arg1 )
▪ protected void onNewIntent( android.content.Intent arg0 )
▪ public boolean onOptionsItemSelected( android.view.MenuItem arg0 )
▪ public void onOptionsMenuClosed( android.view.Menu arg0 )
▪ public void onPanelClosed( int arg0, android.view.Menu arg1 )
▪ protected void onPause( )
▪ protected void onCreate( android.os.Bundle arg0 )
▪ protected void onResume( )
▪ protected void onPrepareDialog( int arg0, Dialog arg1 )
▪ protected void onPrepareDialog( int arg0, Dialog arg1, android.os.Bundle arg2 )
▪ public boolean onPrepareOptionsMenu( android.view.Menu arg0 )
▪ public boolean onPreparePanel( int arg0, android.view.View arg1,
    android.view.Menu arg2 )
▪ protected void onRestart( )
▪ protected void onRestoreInstanceState( android.os.Bundle arg0 )
▪ protected void onResume( )
▪ public Object onRetainNonConfigurationInstance( )
▪ protected void onSaveInstanceState( android.os.Bundle arg0 )
▪ public boolean onSearchRequested( )
▪ protected void onStart( )
▪ protected void onStop( )
▪ protected void onTitleChanged( java.lang.CharSequence arg0, int arg1 )
▪ public boolean onTouchEvent( android.view.MotionEvent arg0 )
▪ public boolean onTrackballEvent( android.view.MotionEvent arg0 )
▪ public void onUserInteraction( )
▪ protected void onUserLeaveHint( )
▪ public void onWindowAttributesChanged(
    android.view.WindowManager.LayoutParams arg0 )
▪ public void onWindowFocusChanged( boolean arg0 )
▪ public void openContextMenu( android.view.View arg0 )
▪ public void openOptionsMenu( )
▪ public void overridePendingTransition( int arg0, int arg1 )
▪ public void registerForContextMenu( android.view.View arg0 )
▪ public final void removeDialog( int arg0 )
▪ public final boolean requestWindowFeature( int arg0 )
▪ public static final RESULT_CANCELED
▪ public static final RESULT_FIRST_USER
▪ public static final RESULT_OK
▪ public final void runOnUiThread( java.lang.Runnable arg0 )
▪ public void setContentView( int arg0 )
▪ public void setContentView( android.view.View arg0 )
▪ public void setContentView( android.view.View arg0,
    android.view.ViewGroup.LayoutParams arg1 )
▪ public final void setDefaultKeyMode( int arg0 )
▪ public final void setFeatureDrawable( int arg0,
    android.graphics.drawable.Drawable arg1 )

```

- `public final void setFeatureDrawableAlpha( int arg0, int arg1 )`
- `public final void setFeatureDrawableResource( int arg0, int arg1 )`
- `public final void setFeatureDrawableUri( int arg0, android.net.Uri arg1 )`
- `public void setIntent( android.content.Intent arg0 )`
- `public void setPersistent( boolean arg0 )`
- `public final void setProgress( int arg0 )`
- `public final void setProgressBarIndeterminate( boolean arg0 )`
- `public final void setProgressBarIndeterminateVisibility( boolean arg0 )`
- `public final void setProgressBarVisibility( boolean arg0 )`
- `public void setRequestedOrientation( int arg0 )`
- `public final void setResult( int arg0 )`
- `public final void setResult( int arg0, android.content.Intent arg1 )`
- `public final void setSecondaryProgress( int arg0 )`
- `public void setTitle( java.lang.CharSequence arg0 )`
- `public void setTitle( int arg0 )`
- `public void setTitleColor( int arg0 )`
- `public void setVisible( boolean arg0 )`
- `public final void setVolumeControlStream( int arg0 )`
- `public final void showDialog( int arg0 )`
- `public final boolean showDialog( int arg0, android.os.Bundle arg1 )`
- `public void startActivity( android.content.Intent arg0 )`
- `public void startActivityForResult( android.content.Intent arg0, int arg1 )`
- `public void startActivityFromChild( Activity arg0, android.content.Intent arg1, int arg2 )`
- `public boolean startActivityIfNeeded( android.content.Intent arg0, int arg1 )`
- `public void startIntentSender( android.content.IntentSender arg0, android.content.Intent arg1, int arg2, int arg3, int arg4 ) throws android.content.IntentSender.SendIntentException`
- `public void startIntentSenderForResult( android.content.IntentSender arg0, int arg1, android.content.Intent arg2, int arg3, int arg4, int arg5 ) throws android.content.IntentSender.SendIntentException`
- `public void startIntentSenderFromChild( Activity arg0, android.content.IntentSender arg1, int arg2, android.content.Intent arg3, int arg4, int arg5, int arg6 ) throws android.content.IntentSender.SendIntentException`
- `public void startManagingCursor( android.database.Cursor arg0 )`
- `public boolean startNextMatchingActivity( android.content.Intent arg0 )`
- `public void startSearch( java.lang.String arg0, boolean arg1, android.os.Bundle arg2, boolean arg3 )`
- `public void stopManagingCursor( android.database.Cursor arg0 )`
- `public void takeKeyEvents( boolean arg0 )`
- `public void triggerSearch( java.lang.String arg0, android.os.Bundle arg1 )`
- `public void unregisterForContextMenu( android.view.View arg0 )`

## Miembros heredados de la clase `android.view.ContextThemeWrapper`

---

- `protected void attachBaseContext( android.content.Context arg0 )`
- `public Object getSystemService( java.lang.String arg0 )`
- `public Resources.Theme getTheme( )`
- `protected void onApplyThemeResource( android.content.res.Resources.Theme arg0, int arg1, boolean arg2 )`
- `public void setTheme( int arg0 )`

---

## Miembros heredados de la clase android.content.ContextWrapper

---

- protected void attachBaseContext( Context arg0 )
- public boolean bindService( Intent arg0, ServiceConnection arg1, int arg2 )
- public int checkCallingOrSelfPermission( java.lang.String arg0 )
- public int checkCallingOrSelfUriPermission( android.net.Uri arg0, int arg1 )
- public int checkCallingPermission( java.lang.String arg0 )
- public int checkCallingUriPermission( android.net.Uri arg0, int arg1 )
- public int checkPermission( java.lang.String arg0, int arg1, int arg2 )
- public int checkUriPermission( android.net.Uri arg0, int arg1, int arg2, int arg3 )
- public int checkUriPermission( android.net.Uri arg0, java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5 )
- public void clearWallpaper( ) throws java.io.IOException
- public Context createPackageContext( java.lang.String arg0, int arg1 ) throws pm.PackageManager.NameNotFoundException
- public String databaseList( )
- public boolean deleteDatabase( java.lang.String arg0 )
- public boolean deleteFile( java.lang.String arg0 )
- public void enforceCallingOrSelfPermission( java.lang.String arg0, java.lang.String arg1 )
- public void enforceCallingOrSelfUriPermission( android.net.Uri arg0, int arg1, java.lang.String arg2 )
- public void enforceCallingPermission( java.lang.String arg0, java.lang.String arg1 )
- public void enforceCallingUriPermission( android.net.Uri arg0, int arg1, java.lang.String arg2 )
- public void enforcePermission( java.lang.String arg0, int arg1, int arg2, java.lang.String arg3 )
- public void enforceUriPermission( android.net.Uri arg0, int arg1, int arg2, int arg3, java.lang.String arg4 )
- public void enforceUriPermission( android.net.Uri arg0, java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5, java.lang.String arg6 )
- public String fileList( )
- public Context getApplicationContext( )
- public ApplicationInfo getApplicationInfo( )
- public AssetManager getAssets( )
- public Context getBaseContext( )
- public File getCacheDir( )
- public ClassLoader getClassLoader( )
- public ContentResolver getContentResolver( )
- public File getDatabasePath( java.lang.String arg0 )
- public File getDir( java.lang.String arg0, int arg1 )
- public File getExternalCacheDir( )
- public File getExternalFilesDir( java.lang.String arg0 )
- public File getFilesDir( )
- public File getFileStreamPath( java.lang.String arg0 )
- public Looper getMainLooper( )
- public String getPackageCodePath( )
- public PackageManager getPackageManager( )
- public String getPackageName( )
- public String getPackageResourcePath( )
- public Resources getResources( )
- public SharedPreferences getSharedPreferences( java.lang.String arg0, int arg1 )
- public Object getSystemService( java.lang.String arg0 )
- public Resources.Theme getTheme( )
- public Drawable getWallpaper( )
- public int getWallpaperDesiredMinimumHeight( )
- public int getWallpaperDesiredMinimumWidth( )
- public void grantUriPermission( java.lang.String arg0, android.net.Uri arg1, int arg2 )
- public boolean isRestricted( )
- public FileInputStream openFileInput( java.lang.String arg0 ) throws java.io.FileNotFoundException
- public FileOutputStream openFileOutput( java.lang.String arg0, int arg1 ) throws java.io.FileNotFoundException

- `public SQLiteDatabase openOrCreateDatabase( java.lang.String arg0, int arg1, android.database.sqlite.SQLiteDatabase.CursorFactory arg2 )`
- `public Drawable peekWallpaper( )`
- `public Intent registerReceiver( BroadcastReceiver arg0, IntentFilter arg1 )`
- `public Intent registerReceiver( BroadcastReceiver arg0, IntentFilter arg1, java.lang.String arg2, android.os.Handler arg3 )`
- `public void removeStickyBroadcast( Intent arg0 )`
- `public void revokeUriPermission( android.net.Uri arg0, int arg1 )`
- `public void sendBroadcast( Intent arg0 )`
- `public void sendBroadcast( Intent arg0, java.lang.String arg1 )`
- `public void sendOrderedBroadcast( Intent arg0, java.lang.String arg1 )`
- `public void sendOrderedBroadcast( Intent arg0, java.lang.String arg1, BroadcastReceiver arg2, android.os.Handler arg3, int arg4, java.lang.String arg5, android.os.Bundle arg6 )`
- `public void sendStickyBroadcast( Intent arg0 )`
- `public void sendStickyOrderedBroadcast( Intent arg0, BroadcastReceiver arg1, android.os.Handler arg2, int arg3, java.lang.String arg4, android.os.Bundle arg5 )`
- `public void setTheme( int arg0 )`
- `public void setWallpaper( android.graphics.Bitmap arg0 ) throws java.io.IOException`
- `public void setWallpaper( java.io.InputStream arg0 ) throws java.io.IOException`
- `public void startActivity( Intent arg0 )`
- `public boolean startInstrumentation( ComponentName arg0, java.lang.String arg1, android.os.Bundle arg2 )`
- `public void startIntentSender( IntentSender arg0, Intent arg1, int arg2, int arg3, int arg4 ) throws IntentSender.SendIntentException`
- `public ComponentName startService( Intent arg0 )`
- `public boolean stopService( Intent arg0 )`
- `public void unbindService( ServiceConnection arg0 )`
- `public void unregisterReceiver( BroadcastReceiver arg0 )`

## Miembros heredados de la clase `android.content.Context`

---

- `public static final ACCESSIBILITY_SERVICE`
- `public static final ACCOUNT_SERVICE`
- `public static final ACTIVITY_SERVICE`
- `public static final ALARM_SERVICE`
- `public static final AUDIO_SERVICE`
- `public static final BIND_AUTO_CREATE`
- `public static final BIND_DEBUG_UNBIND`
- `public static final BIND_NOT_FOREGROUND`
- `public abstract boolean bindService( Intent arg0, ServiceConnection arg1, int arg2 )`
- `public abstract int checkCallingOrSelfPermission( java.lang.String arg0 )`
- `public abstract int checkCallingOrSelfUriPermission( android.net.Uri arg0, int arg1 )`
- `public abstract int checkCallingPermission( java.lang.String arg0 )`
- `public abstract int checkCallingUriPermission( android.net.Uri arg0, int arg1 )`
- `public abstract int checkPermission( java.lang.String arg0, int arg1, int arg2 )`
- `public abstract int checkUriPermission( android.net.Uri arg0, int arg1, int arg2, int arg3 )`
- `public abstract int checkUriPermission( android.net.Uri arg0, java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5 )`
- `public abstract void clearWallpaper( ) throws java.io.IOException`
- `public static final CLIPBOARD_SERVICE`
- `public static final CONNECTIVITY_SERVICE`
- `public static final CONTEXT_IGNORE_SECURITY`
- `public static final CONTEXT_INCLUDE_CODE`
- `public static final CONTEXT_RESTRICTED`
- `public abstract Context createPackageContext( java.lang.String arg0, int arg1 ) throws pm.PackageManager.NameNotFoundException`
- `public abstract String databaseList( )`
- `public abstract boolean deleteDatabase( java.lang.String arg0 )`

```

▪ public abstract boolean deleteFile( java.lang.String arg0 )
▪ public static final DEVICE_POLICY_SERVICE
▪ public static final DROPBOX_SERVICE
▪ public abstract void enforceCallingOrSelfPermission( java.lang.String arg0,
    java.lang.String arg1 )
▪ public abstract void enforceCallingOrSelfUriPermission( android.net.Uri arg0,
    int arg1, java.lang.String arg2 )
▪ public abstract void enforceCallingPermission( java.lang.String arg0,
    java.lang.String arg1 )
▪ public abstract void enforceCallingUriPermission( android.net.Uri arg0, int
    arg1, java.lang.String arg2 )
▪ public abstract void enforcePermission( java.lang.String arg0, int arg1, int
    arg2, java.lang.String arg3 )
▪ public abstract void enforceUriPermission( android.net.Uri arg0, int arg1, int
    arg2, int arg3, java.lang.String arg4 )
▪ public abstract void enforceUriPermission( android.net.Uri arg0,
    java.lang.String arg1, java.lang.String arg2, int arg3, int arg4, int arg5,
    java.lang.String arg6 )
▪ public abstract String fileList( )
▪ public abstract Context getApplicationContext( )
▪ public abstract ApplicationInfo getApplicationInfo( )
▪ public abstract AssetManager getAssets( )
▪ public abstract File getCacheDir( )
▪ public abstract ClassLoader getClassLoader( )
▪ public abstract ContentResolver getContentResolver( )
▪ public abstract File getDatabasePath( java.lang.String arg0 )
▪ public abstract File getDir( java.lang.String arg0, int arg1 )
▪ public abstract File getExternalCacheDir( )
▪ public abstract File getExternalFilesDir( java.lang.String arg0 )
▪ public abstract File getFilesDir( )
▪ public abstract File getFileStreamPath( java.lang.String arg0 )
▪ public abstract Looper getMainLooper( )
▪ public abstract String getPackageCodePath( )
▪ public abstract PackageManager getPackageManager( )
▪ public abstract String getPackageName( )
▪ public abstract String getPackageResourcePath( )
▪ public abstract Resources getResources( )
▪ public abstract SharedPreferences getSharedPreferences( java.lang.String arg0,
    int arg1 )
▪ public final String getString( int arg0 )
▪ public final String getString( int arg0, java.lang.Object[] arg1 )
▪ public abstract Object getSystemService( java.lang.String arg0 )
▪ public final CharSequence getText( int arg0 )
▪ public abstract Resources.Theme getTheme( )
▪ public abstract Drawable getWallpaper( )
▪ public abstract int getWallpaperDesiredMinimumHeight( )
▪ public abstract int getWallpaperDesiredMinimumWidth( )
▪ public abstract void grantUriPermission( java.lang.String arg0, android.net.Uri
    arg1, int arg2 )
▪ public static final INPUT_METHOD_SERVICE
▪ public boolean isRestricted( )
▪ public static final KEYGUARD_SERVICE
▪ public static final LAYOUT_INFLATER_SERVICE
▪ public static final LOCATION_SERVICE
▪ public static final MODE_APPEND
▪ public static final MODE_PRIVATE
▪ public static final MODE_WORLD_READABLE
▪ public static final MODE_WORLD_WRITEABLE
▪ public static final NOTIFICATION_SERVICE
▪ public final TypedArray obtainStyledAttributes( android.util.AttributeSet arg0,
    int[] arg1 )
▪ public final TypedArray obtainStyledAttributes( android.util.AttributeSet arg0,
    int[] arg1, int arg2, int arg3 )
▪ public final TypedArray obtainStyledAttributes( int[] arg0 )
▪ public final TypedArray obtainStyledAttributes( int arg0, int[] arg1 ) throws
    res.Resources.NotFoundException

```

- public abstract FileInputStream openFileInput( java.lang.String arg0 ) throws java.io.FileNotFoundException
- public abstract FileOutputStream openFileOutput( java.lang.String arg0, int arg1 ) throws java.io.FileNotFoundException
- public abstract SQLiteDatabase openOrCreateDatabase( java.lang.String arg0, int arg1, android.database.sqlite.SQLiteDatabase.CursorFactory arg2 )
- public abstract Drawable peekWallpaper( )
- public static final POWER\_SERVICE
- public abstract Intent registerReceiver( BroadcastReceiver arg0, IntentFilter arg1 )
- public abstract Intent registerReceiver( BroadcastReceiver arg0, IntentFilter arg1, java.lang.String arg2, android.os.Handler arg3 )
- public abstract void removeStickyBroadcast( Intent arg0 )
- public abstract void revokeUriPermission( android.net.Uri arg0, int arg1 )
- public static final SEARCH\_SERVICE
- public abstract void sendBroadcast( Intent arg0 )
- public abstract void sendBroadcast( Intent arg0, java.lang.String arg1 )
- public abstract void sendOrderedBroadcast( Intent arg0, java.lang.String arg1 )
- public abstract void sendOrderedBroadcast( Intent arg0, java.lang.String arg1, BroadcastReceiver arg2, android.os.Handler arg3, int arg4, java.lang.String arg5, android.os.Bundle arg6 )
- public abstract void sendStickyBroadcast( Intent arg0 )
- public abstract void sendStickyOrderedBroadcast( Intent arg0, BroadcastReceiver arg1, android.os.Handler arg2, int arg3, java.lang.String arg4, android.os.Bundle arg5 )
- public static final SENSOR\_SERVICE
- public abstract void setTheme( int arg0 )
- public abstract void setWallpaper( android.graphics.Bitmap arg0 ) throws java.io.IOException
- public abstract void setWallpaper( java.io.InputStream arg0 ) throws java.io.IOException
- public abstract void startActivity( Intent arg0 )
- public abstract boolean startInstrumentation( ComponentName arg0, java.lang.String arg1, android.os.Bundle arg2 )
- public abstract void startIntentSender( IntentSender arg0, Intent arg1, int arg2, int arg3, int arg4 ) throws IntentSender.SendIntentException
- public abstract ComponentName startService( Intent arg0 )
- public abstract boolean stopService( Intent arg0 )
- public static final TELEPHONY\_SERVICE
- public static final UI\_MODE\_SERVICE
- public abstract void unbindService( ServiceConnection arg0 )
- public abstract void unregisterReceiver( BroadcastReceiver arg0 )
- public static final VIBRATOR\_SERVICE
- public static final WALLPAPER\_SERVICE
- public static final WIFI\_SERVICE
- public static final WINDOW\_SERVICE

# Apéndice C

## Código Fuente

En éste último apéndice, se mostrarán todos los listados de código que componen el proyecto.

### C.1. Ficheros de configuración XML

#### C.1.1. Fichero patterns.xml

Éste es el fichero que contiene información sobre los patrones y las localizaciones.

##### patterns.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<AR>

    <pattern id = "1" name="pattc" >
        <object id="5"/>
    </pattern>

    <pattern id = "2" name="pattf" >
        <object id="9"/>
    </pattern>

    <pattern id = "3" name="patta" >
        <object id="3"/>
    </pattern>

    <!--
    <pattern id = "2" name="F" path="multi/patt.f">
        <object id="1"/>
    </pattern>

    <pattern id = "5" name="C" path="multi/patt.c">
        <object id="8"/>
    </pattern>

    <pattern id = "3" name="B" path="multi/patt.b">
```

```
        <object id="4"/>
    </pattern>

    <pattern id = "4" name="G" path="multi/patt.g">
        <object id="3"/>
    </pattern>
    →

    <location id = "1" name="Mi casa" lat="43.4059" long="-5.8052" >
        <object id="1"/>
    </location>

</AR>
```

Fichero de configuración C.1: patterns.xml

### C.1.2. Fichero objects.xml

Es el fichero que contiene información sobre los objetos.

#### objects.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objects>
    <object id="1" name="Grillo">
        <description>Comentario sobre el grillo</description>
        <file3ds>http://tfg.ivanlorenzo.es/data/objects/saltamontes
            .3ds</file3ds>
        <sound>http://tfg.ivanlorenzo.es/data/sounds/saltamontes.
            mp3</sound>
    </object>

    <object id="3" name="Carpa Dorada">
        <description>Carpa Dorada</description>
        <file3ds>http://tfg.ivanlorenzo.es/data/objects/carpa.3ds</
            file3ds>
        <sound>http://tfg.ivanlorenzo.es/data/sounds/carpa.mp3</
            sound>
    </object>

    <object id="2" name="Pez Espada">
        <description>Pez Espada</description>
        <file3ds>http://tfg.ivanlorenzo.es/data/objects/espada.3ds<
            /file3ds>
        <sound>http://tfg.ivanlorenzo.es/data/sounds/espada.mp3</
            sound>
    </object>

    <object id="4" name="Pensamientos">
        <description>Pensamientos</description>
        <file3ds>http://tfg.ivanlorenzo.es/data/objects/
            pensamientos.3ds</file3ds>
```

```

</object>

<object id="5" name="Delfin">
  <description>Delfin</description>
  <file3ds>http://tfg.ivanlorenzo.es/data/objects/delfin.3ds<
    /file3ds>
</object>

<object id="6" name="Ballena Jorobada">
  <description>Ballena Jorobada</description>
  <file3ds>http://tfg.ivanlorenzo.es/data/objects/ballena.3ds
    </file3ds>
</object>

  <object id="20" name="Cuadro">
    <description>Un cuadro en el que meter varias imagenes</
      description>
    <file3ds>http://tfg.ivanlorenzo.es/data/objects/pictureIvan
      .3ds</file3ds>
  </object>

<object id="7" name="Rosa">

  <texture>YukiTuT.jpg</texture>
  <file3ds>YukiTubaki.3ds</file3ds>
</object>

<object id="9" name="Nala">
  <texture>http://tfg.ivanlorenzo.es/data/objects/Nala.png</
    texture>
</object>

<object id="8" name="Gioconda">
  <description>El cuadro La Gioconda, conocido tambien como
    la Mona Lisa, es una obra pictorica de Leonardo da Vinci
    que nunca ha sido tasada.
    Desde el siglo XVI es propiedad del Estado Frances, el cual
    la expone en el museo del Louvre.
    Su nombre oficial es Gioconda (cuya traduccion del italiano
    al castellano es alegre) en honor a la tesis mas fuerte
    acerca de la identidad de la modelo, apoyada en que era
    esposa de Francesco Bartolomeo del Giocondo
    y que su nombre era Lisa Gherardini.</description>
  <sound>http://tfg.ivanlorenzo.es/data/sounds/gio.mp3</sound
    >
  <texture>http://tfg.ivanlorenzo.es/data/objects/gio.jpg</
    texture>
</object>
</objects>

```

Fichero de configuración C.2: objects.xml

### C.1.3. Fichero patterns.xsd

Éste es el fichero que valida patterns.xml

#### patterns.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified">
  <xs:element name="AR">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="pattern"/>
        <xs:element maxOccurs="unbounded" ref="location"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="pattern">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="object"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NMTOKEN"/>
      <xs:attribute name="name" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="location">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="object"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NMTOKEN"/>
      <xs:attribute name="name" use="required"/>
      <xs:attribute name="lat" use="required"/>
      <xs:attribute name="long" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="object">
    <xs:complexType>
      <xs:attribute name="id" use="required" type="xs:NMTOKEN"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fichero de configuración C.3: patterns.xsd

### C.1.4. Fichero objects.xsd

Es el fichero que valida objects.xml

#### objects.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="
  qualified">
  <xs:element name="objects">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="object"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="object">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" ref="description"/>
        <xs:element minOccurs="0" ref="sound"/>
        <xs:element minOccurs="0" ref="texture"/>
        <xs:element minOccurs="0" ref="file3ds"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NMTOKEN"/>
      <xs:attribute name="name" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="description" type="xs:string"/>
  <xs:element name="sound" type="xs:string"/>
  <xs:element name="texture" type="xs:string"/>
  <xs:element name="file3ds" type="xs:string"/>
</xs:schema>
```

Fichero de configuración C.4: objects.xsd

## C.2. Ficheros de recursos de la aplicación Android

### C.2.1. Layout

Es el fichero de Layout de la aplicación

#### main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/main"
  android:orientation="horizontal"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <FrameLayout android:id="@+id/frame"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <SurfaceView android:id="@+id/camera_preview"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"/>
  </FrameLayout>
</LinearLayout>
```

---

Fichero de recursos C.5: main.xml

### C.2.2. Menu

Es el fichero que configura el menú de la aplicación

**menu.xml**

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/configuration"
        android:title="@string/configuration" />
    <item android:id="@+id/exit"
        android:title="@string/exit" />
</menu>
```

Fichero de recursos C.6: menu.xml

### C.2.3. Preferencias

**preferences.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
    ">

    <PreferenceCategory android:title="TFG Configuration">

        <EditTextPreference    android:dialogTitle="@string/patterns_url"
                                android:dialogMessage="@string/
                                    write_patterns_url"
                                android:key="URL_PATTERNS"
                                android:title="@string/patterns_and_locations"
                                android:summary="@string/patterns_url"
                                android:defaultValue="@string/URL_PATTERNS"/>

        <EditTextPreference    android:dialogTitle="@string/objects_url"
                                android:dialogMessage="@string/
                                    write_objects_url"
                                android:key="URL_OBJECTS"
                                android:title="@string/objects"
                                android:summary="@string/objects_url"
                                android:defaultValue="@string/URL_OBJECTS"/>

    </PreferenceCategory>

</PreferenceScreen>
```

Fichero de recursos C.7: preferences.xml

### C.2.4. Internacionalización

#### values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- title for the dialog showing the error of camera hardware -->
    <string name="camera_error_title">Error en la cámara</string>

    <!-- message for the dialog showing the error of camera hardware -->
    <string name="cannot_connect_camera">No se puede conectar la cámara</string>

    <!-- Details dialog "OK" button. Dismisses dialog. -->
    <string name="details_ok">OK</string>

    <string name="app_name">TFG Ivan Lorenzo Rubio</string>

    <string name="configuration">Configuración</string>
    <string name="exit">Salir</string>

    <string name="write_patterns_url">Escriba la dirección URL de los patrones</string>
    <string name="write_objects_url">Escriba la dirección URL de los objetos</string>
    <string name="save_preferences">Guardar preferencias</string>
    <string name="loading">Cargando</string>
    <string name="internet_connection">Por favor, revisa tu conexión de internet e inténtalo de nuevo</string>
    <string name="connection_error">Error de conexión</string>
    <string name="yes">Si</string>
    <string name="no">No</string>
    <string name="sure_exit">¿Está seguro que desea salir de la aplicación?</string>
</resources>
```

Fichero de recursos C.8: values/strings.xml

#### values-en/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- title for the dialog showing the error of camera hardware -->
    <string name="camera_error_title">Camera error</string>

    <!-- message for the dialog showing the error of camera hardware -->
    <string name="cannot_connect_camera">Cannot connect to camera.</string>

    <!-- Details dialog "OK" button. Dismisses dialog. -->
    <string name="details_ok">OK</string>

    <string name="app_name">TFG Ivan Lorenzo Rubio</string>
```

```
<string name="configuration">Configuration</string>
<string name="exit">Exit</string>

<string name="write_patterns_url">Enter the URL of the patterns</
  string>
<string name="write_objects_url">Enter the URL of the objects</
  string>
<string name="save_preferences">Save preferences</string>

<string name="loading">Loading...</string>
<string name="internet_connection">Please, check your internet
  connection and try again</string>
<string name="connection_error">Connection error</string>
<string name="yes">Yes</string>
<string name="no">No</string>
<string name="sure_exit">Are you sure you want to exit?</string>

<!--
  <string name="URL_PATTERNS">http://tfg.ivanlorenzo.es/data/patterns
    .xml</string>
  <string name="URL_OBJECTS">http://tfg.ivanlorenzo.es/data/objects.
    xml</string>
  <string name="NO_OBJECT_3D">http://tfg.ivanlorenzo.es/data/objects/
    DibujoIvan.3DS</string>
-->
</resources>
```

Fichero de recursos C.9: values-en/strings.xml

### values-es/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <!-- title for the dialog showing the error of camera hardware -->
  <string name="camera_error_title">Error de cámara</string>

  <!-- message for the dialog showing the error of camera hardware -->
  <string name="cannot_connect_camera">No se puede conectar la cámara</
    string>

  <!-- Details dialog "OK" button. Dismisses dialog. -->
  <string name="details_ok">OK</string>

  <string name="app_name">TFG Ivan Lorenzo Rubio</string>

  <string name="configuration">Configuración</string>
  <string name="exit">Salir</string>

  <string name="patterns_and_locations">Patrones y Localizaciones</
    string><string name="patterns_url">URL de los patrones y
    localizaciones</string><string name="write_patterns_url">Escriba
    la dirección URL de los patrones y localizaciones</string><
    string name="objects">Objetos</string>
```

```
<string name="objects_url" >URL de los objetos</string><string name="write_objects_url">Escriba la dirección URL de los objetos</string>

<string name="loading">Cargando...</string>
<string name="internet_connection">Por favor, revisa tu conexión de internet e inténtalo de nuevo</string>
<string name="connection_error">Error de conexión</string>
<string name="yes">Si</string>
<string name="no">No</string>
<string name="sure_exit">¿Está seguro que desea salir de la aplicación?</string>

</resources>
```

Fichero de recursos C.10: values-es/strings.xml

### C.2.5. AndroidManifest

#### AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ivanlorenzo.gui" android:versionName="3.0.0-1"
    android:versionCode="30001">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
        </uses-permission>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
        </uses-permission>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
        </uses-permission>

    <uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
        android:name="android.permission.INSTALL_PACKAGES"
        </uses-permission>
    <uses-permission android:name="android.permission.DELETE_PACKAGES"
        xmlns:android="http://schemas.android.com/apk/res/android"
        </uses-permission>
    <uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
        android:name="android.permission.RESTART_PACKAGES"
        </uses-permission>

    <uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
        android:name="android.permission.CAMERA"
        </uses-permission>
    <uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
        android:name="android.permission.ACCESS_SURFACE_FLINGER"
        </uses-permission>
```

```
<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"></uses-
-permission>
<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.BRICK"></uses-permission>
<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.FLASHLIGHT"></uses-permission>
<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.INTERNAL_SYSTEM_WINDOW"></uses-
permission>
<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.MASTER_CLEAR"></uses-permission>
<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.READ_FRAME_BUFFER"></uses-permission>

<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.INTERNET"></uses-permission>

<uses-permission xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"></uses-
-permission>
<uses-sdk android:minSdkVersion="8"></uses-sdk>

    <application android:icon="@drawable/icon" android:label="@string/
app_name" android:debuggable="true">
        <activity android:name=".TFGARActivity" android:screenOrientation="
landscape" android:theme="@android:style/Theme.NoTitleBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" /
                >
            </intent-filter>
        </activity>

        <activity android:name=".preferences.TFGPreferencesActivity"
            android:screenOrientation="landscape" android:theme="
            @android:style/Theme.NoTitleBar">

        </activity>

    </application>

</manifest>
```

Fichero de recursos C.11: AndroidManifest.xml

## C.3. JNI

### yuv420sp2rgb/yuv420sp2rgb.c

```

#include <jni.h>
#include <android/log.h>

#define LOG_TAG    "libyuv420sp2rgb"
#define LOGI(...)  __android_log_print(ANDROID_LOG_INFO, LOG_TAG,
    __VA_ARGS__)
#define LOGE(...)  __android_log_print(ANDROID_LOG_ERROR, LOG_TAG,
    __VA_ARGS__)

/*
 * derived from development/tools/yuv420sp2rgb/yuv420sp2rgb.c
 */
#include <unistd.h>

#ifndef max
#define max(a,b) ({typeof(a) _a = (a); typeof(b) _b = (b); _a > _b ? _a :
    _b; })
#define min(a,b) ({typeof(a) _a = (a); typeof(b) _b = (b); _a < _b ? _a :
    _b; })
#endif

#define CONVERT_TYPE_PPM 0
#define CONVERT_TYPE_RGB 1
#define CONVERT_TYPE_ARGB 2

/*
    YUV 4:2:0 image with a plane of 8 bit Y samples followed by an
    interleaved
    U/V plane containing 8 bit 2x2 subsampled chroma samples.
    except the interleave order of U and V is reversed.

                H V
    Y Sample Period    1 1
    U (Cb) Sample Period 2 2
    V (Cr) Sample Period 2 2
 */

typedef struct rgb_context {
    unsigned char *buffer;
    int width;
    int height;
    int rotate;
    int i;
    int j;
    int size; /* for debugging */
} rgb_context;

typedef void (*rgb_cb)(
    unsigned char r,
    unsigned char g,

```

```

    unsigned char b,
    rgb_context *ctx);

const int bytes_per_pixel = 2;

static void color_convert_common(
    unsigned char *pY, unsigned char *pUV,
    int width, int height,
    unsigned char *buffer,
    int size, /* buffer size in bytes */
    int gray,
    int rotate,
    rgb_cb cb)
{
    int i, j;
    int nR, nG, nB;
    int nY, nU, nV;
    rgb_context ctx;

    ctx.buffer = buffer;
    ctx.size = size; /* debug */
    ctx.width = width;
    ctx.height = height;
    ctx.rotate = rotate;

    if (gray) {
        for (i = 0; i < height; i++) {
            for (j = 0; j < width; j++) {
                nB = *(pY + i * width + j);
                ctx.i = i;
                ctx.j = j;
                cb(nB, nB, nB, &ctx);
            }
        }
    } else {
        // YUV 4:2:0
        for (i = 0; i < height; i++) {
            for (j = 0; j < width; j++) {
                nY = *(pY + i * width + j);
                nV = *(pUV + (i/2) * width + bytes_per_pixel * (j/2));
                nU = *(pUV + (i/2) * width + bytes_per_pixel * (j/2) + 1);

                // Yuv Convert
                nY -= 16;
                nU -= 128;
                nV -= 128;

                if (nY < 0)
                    nY = 0;

                // nR = (int)(1.164 * nY + 2.018 * nU);
                // nG = (int)(1.164 * nY - 0.813 * nV - 0.391 * nU);
                // nB = (int)(1.164 * nY + 1.596 * nV);

                nB = (int)(1192 * nY + 2066 * nU);
                nG = (int)(1192 * nY - 833 * nV - 400 * nU);
            }
        }
    }
}

```

```

        nR = (int)(1192 * nY + 1634 * nV);

        nR = min(262143, max(0, nR));
        nG = min(262143, max(0, nG));
        nB = min(262143, max(0, nB));

        nR >>= 10; nR &= 0xff;
        nG >>= 10; nG &= 0xff;
        nB >>= 10; nB &= 0xff;

        ctx.i = i;
        ctx.j = j;
        cb(nR, nG, nB, &ctx);
    }
}

static void common_rgb_cb(
    unsigned char r,
    unsigned char g,
    unsigned char b,
    rgb_context *ctx,
    int alpha)
{
    unsigned char *out = ctx->buffer;
    int offset = 0;
    int bpp;
    int i = 0;
    switch(ctx->rotate) {
        case 0: /* no rotation */
            offset = ctx->i * ctx->width + ctx->j;
            break;
        case 1: /* 90 degrees */
            offset = ctx->height * (ctx->j + 1) - ctx->i;
            break;
        case 2: /* 180 degrees */
            offset = (ctx->height - 1 - ctx->i) * ctx->width + ctx->j;
            break;
        case 3: /* 270 degrees */
            offset = (ctx->width - 1 - ctx->j) * ctx->height + ctx->i;
            break;
        default:
            //FAILIF(1, "Unexpected roation value %d\n", ctx->rotate);
            break;
    }

    bpp = 3 + !!alpha;
    offset *= bpp;
    /*FAILIF(offset < 0, "point (%d, %d) generates a negative offset.\n",
        ctx->i, ctx->j);
    FAILIF(offset + bpp > ctx->size, "point (%d, %d) at offset %d exceeds
        the size %d of the buffer.\n",
        ctx->i, ctx->j,
        offset,
        ctx->size);*/

```

```
    out += offset;

    /* if (alpha) out[i++] = 0xff;
    out[i++] = r;
    out[i++] = g;
    out[i] = b; */
    out[i++] = b;
    out[i++] = g;
    out[i++] = r;
    if (alpha) out[i] = 0xff;
}

static void rgb24_cb(
    unsigned char r,
    unsigned char g,
    unsigned char b,
    rgb_context *ctx)
{
    return common_rgb_cb(r, g, b, ctx, 0);
}

static void argb_cb(
    unsigned char r,
    unsigned char g,
    unsigned char b,
    rgb_context *ctx)
{
    return common_rgb_cb(r, g, b, ctx, 1);
}

JNIEXPORT void JNICALL
Java_com_ivanlorenzo_gui_ARToolkitDrawer_decodeYUV420SP(JNIEnv * env,
    jobject obj, jintArray rgb, jbyteArray yuv420sp, jint width, jint height,
    jint type)
{
    void *in, *out;
    int psz = getpagesize();
    int header_size;
    size_t outsize;

    int bpp = 3;
    switch (type) {
    case CONVERT_TYPE_RGB:
        header_size = 0;
        break;
    case CONVERT_TYPE_ARGB:
        header_size = 0;
        bpp = 4;
        break;
    }

    outsize = header_size + width * height * bpp;
    outsize = (outsize + psz - 1) & ~(psz - 1);

    jboolean isCopyIn, isCopyOut;
```

```

in = (*env)->GetByteArrayElements(env, yuv420sp, &isCopyIn);
out = (*env)->GetIntArrayElements(env, rgb, &isCopyOut);

color_convert_common(in, in + width * height,
                    width, height,
                    out + header_size, outsize - header_size,
                    0, 0,
                    type == CONVERT_TYPE_ARGB ? argb_cb : rgb24_cb);

//if (isCopyIn == JNI_TRUE)
    (*env)->ReleaseByteArrayElements(env, yuv420sp, in, 0);
//if (isCopyOut == JNI_TRUE)
    (*env)->ReleaseIntArrayElements(env, rgb, out, 0);
}

```

JNI C.12: yuv420sp2rgb/yuv420sp2rgb.c

## C.4. Clases Java

### C.4.1. Paquete com.ivanlorenzo.data

#### XMLCompiler.java

```

package com.ivanlorenzo.data;

import java.io.IOException;
import java.io.InputStream;
import java.net.URL;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

/*
import javax.xml.XMLConstants;
import javax.xml.transform.Source;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamSource;
import javax.xml.validation.Schema;
import javax.xml.validation.SchemaFactory;
import javax.xml.validation.Validator;
*/

import com.ivanlorenzo.data.exceptions.LoadXMLException;

import org.w3c.dom.Document;
import org.xml.sax.SAXException;

/**
 * Clase genérica abstracta encargada de la compilación de ficheros XML
 * @author Iván Lorenzo Rubio

```

```
*
*/
public abstract class XMLCompiler
{
    /**
     * Url del fichero XML a cargar
     */
    protected String xml;

    /**
     * URL del fichero XSD a cargar
     */
    protected String xsd;

    /**
     * Documento DOM que representa el juego
     */
    protected Document document;

    /**
     * Devuelve un flujo de bytes del recurso que se encuentra en la url pasada
     * @param url La dirección donde se cuenta el recurso a abrir
     * @return Flujo de bytes del recurso
     * @throws IOException En caso de que no se pueda acceder a la URL lanzaría una IOException
     */
    private InputStream getInputStream(URL url) throws IOException
    {
        return url.openConnection().getInputStream();
    }

    /**
     * Carga en el documento DOM el fichero XML
     * @param validate Si el fichero XML se validará con un DTD
     * @throws LoadXMLException Si hay errores en la carga
     * @throws IOException Se puede producir una excepción si no encuentra el recurso en la URL
     */
    protected void loadFile(boolean validate) throws LoadXMLException,
        IOException
    {
        DocumentBuilderFactory parserFactory =
            DocumentBuilderFactory.newInstance();
        parserFactory.setNamespaceAware(true);

        try
        {

```

```

        DocumentBuilder parser = parserFactory.
            newDocumentBuilder();
        document = parser.parse( getInputStream(new URL(xml
            )) );

//          SchemaFactory factory = SchemaFactory
//              .newInstance(XMLConstants.XML_DTD_NS_URI);

//          Source schemaFile = new StreamSource(
getInputStream(new URL(xsd)) );
//          Schema schema = factory.newSchema(schemaFile);
//
//          Validator validator = schema.newValidator();
//          validator.validate(new DOMSource(document));

    }
    catch (ParserConfigurationException e) {}
    catch (SAXException e) {}

}

}

```

Paquete data C.13: XMLCompiler.java

### AugmentedRealityXMLCompiler.java

```

package com.ivanlorenzo.data;

import java.io.IOException;

import javax.xml.xpath.XPathExpressionException;

import android.content.res.Resources;

import com.ivanlorenzo.data.exceptions.LoadXMLException;
import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;

import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

/**
 * Clase hija de XMLCompiler encargada de compilar el fichero XML que
 * contiene la información sobre
 * los patrones (markers) y las posiciones GPS
 * @author Iván Lorenzo Rubio
 *
 */
public class AugmentedRealityXMLCompiler extends XMLCompiler
{

```

```
/**
 * Busca todos los patrones y localizaciones del fichero XML y los
 * guarda en la base de conocimiento
 * @param res Recursos de la aplicación
 * @param urlXMLObjects Ruta del fichero XML de objetos
 * @param urlXSDObjects Ruta del fichero XSD de objetos
 * @throws LoadXMLException Si hay errores en la carga
 * @throws XPathExpressionException Si se produce un error en una
 * expresión XPath
 * @throws NoSuchExtensionObject3DException Si el objeto no tiene
 * una extensión 3D válida
 * @throws IOException En caso de que no se pueda acceder a la URL
 * lanzaría una IOException
 */
private void scanDocument(Resources res, String urlXMLObjects,
    String urlXSDObjects)
    throws LoadXMLException, XPathExpressionException,
    NoSuchExtensionObject3DException, IOException
{
    int count = 0;

    //Nos posicionamos en el elemento principal <patterns>
    Element root = document.getDocumentElement();

    NodeList items = root.getElementsByTagName("pattern");
    for (int i=0; i<items.getLength(); i++)
    {
        //Obtenemos el patrón
        Node item = items.item(i);

        int idPattern = Integer.parseInt(
            item.getAttributes().getNamedItem("id").
            getNodeValue());

        String namePattern = item.getAttributes().getNamedItem("name").
            getNodeValue();

        int idObject = 0;

        //Obtenemos la lista de objetos del patrón
        NodeList objects = item.getChildNodes();
        for (int j=0; j<objects.getLength(); j++)
        {
            Node object = objects.item(j);
            String label = object.getNodeName();

            if (label.equals("object"))
            {
                idObject = Integer.parseInt( object.
                    getAttributes().getNamedItem("id").
                    getNodeValue());
            }
        }
    }
}
```

```

        Pattern patData = new Pattern(idPattern ,
            namePattern , idObject);
        KnowledgeBase.getInstance().savePattern(patData);

        //Llamada a otro fichero XML
        new ObjectsXMLCompiler().load(res , urlXMLObjects ,
            urlXSDOObjects , idObject , count);
        count++;
    }

    /* LOCATION */

    items = root.getElementsByTagName("location");
    for (int i=0; i<items.getLength(); i++)
    {
        //Obtenemos la localización
        Node item = items.item(i);

        int idLocation = Integer.parseInt(
            item.getAttributes().getNamedItem("id").
                getNodeValue() );

        String nameLocation = item.getAttributes().getNamedItem("name")
            .getNodeValue();

        double latitude= Double.parseDouble(
            item.getAttributes().getNamedItem("lat").
                getNodeValue() );
        double longitude= Double.parseDouble(
            item.getAttributes().getNamedItem("long").
                getNodeValue() );

        int idObject = 0;

        //Obtenemos la lista de objetos de la localización
        NodeList objects = item.getChildNodes();
        for (int j=0; j<objects.getLength(); j++)
        {
            Node object = objects.item(j);
            String label = object.getNodeName();

            if(label.equals("object"))
            {
                idObject = Integer.parseInt( object.
                    getAttributes().getNamedItem("id").
                        getNodeValue() );
            }
        }

        GeoLocation location = new GeoLocation(idLocation , nameLocation
            , latitude , longitude , idObject);
        KnowledgeBase.getInstance().saveLocation(location);
    }

```

```
        //Llamada a otro fichero XML
        new ObjectsXMLCompiler().load(res, urlXMLObjects,
            urlXSDObjects, idObject, count);
        count++;
    }

}

/**
 * Método que carga los patrones
 * @param res Recursos de la aplicación
 * @param urlXMLPatterns Ruta del fichero XML de patrones
 * @param urlXSDPatterns Ruta del fichero XSD de patrones
 * @param urlXMLObjects Ruta del fichero XML de objetos
 * @param urlXSDObjects Ruta del fichero XSD de objetos
 * @throws LoadXMLException Si hay errores en la carga
 * @throws XPathExpressionException Si se produce un error en una
 *     expresión XPath
 * @throws NoSuchExtensionObject3DException Si el objeto no tiene
 *     una extensión 3D válida
 * @throws IOException En caso de que no se pueda acceder a la URL
 *     lanzaría una IOException
 */
public void load (Resources res, String urlXMLPatterns, String
    urlXSDPatterns,
    String urlXMLObjects, String urlXSDObjects)
    throws LoadXMLException, XPathExpressionException,
        NoSuchExtensionObject3DException, IOException
{
    this.xml = urlXMLPatterns;
    this.xsd = urlXSDPatterns;
    loadFile(true);
    scanDocument(res, urlXMLObjects, urlXMLObjects);
}
}
```

Paquete data C.14: AugmentedRealityXMLCompiler.java

### ObjectsXMLCompiler.java

```
package com.ivanlorenzo.data;

import java.io.IOException;

import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpression;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;
```

```

import android.content.res.Resources;

import com.ivanlorenzo.data.exceptions.LoadXMLException;
import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;
import com.ivanlorenzo.data.objects.Object3D;
import com.ivanlorenzo.data.objects.Object3DCreator;

import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

/**
 * Clase hija de XMLCompiler encargada de compilar el fichero XML que
 * contiene la información sobre los objetos
 * @author Iván Lorenzo Rubio
 *
 */
public class ObjectsXMLCompiler extends XMLCompiler
{
    /**
     * Busca todos un objeto determinado del fichero XML y lo guarda en
     * la base de conocimiento.
     * @param res Recursos de la aplicación
     * @param id Identificador del objeto a buscar
     * @param order Orden en que se cargará el objeto
     * @throws XPathExpressionException Si se produce un error en una
     * expresión XPath
     * @throws NoSuchExtensionObject3DException Si el objeto no tiene
     * una extensión 3D válida
     */
    private void searchObject (Resources res , int id , int order)
        throws XPathExpressionException ,
        NoSuchExtensionObject3DException
    {
        XPathFactory factory = XPathFactory.newInstance();
        XPath xpath = factory.newXPath();
        String ex = "//object[@id='" + id + "']";
        XPathExpression expr = xpath.compile(ex);

        Object result = expr.evaluate(document, XPathConstants.NODESET);
        ;
        NodeList nodes = (NodeList) result;
        for (int i = 0; i < nodes.getLength(); i++)
        {
            int idObject = Integer.parseInt( nodes.item(i).
                getAttributes().getNamedItem("id").getNodeValue() );
            String nameObject = nodes.item(i).getAttributes().
                getNamedItem("name").getNodeValue();

            Node object = nodes.item(i);
            NodeList children = object.getChildNodes();

            String description = "";

```

```
String sound = "";
String file3ds = "";
String extensionFile3ds = "";
String texture = "";
for(int j = 0; j<children.getLength(); j++)
{
    if (children.item(j).getNodeTypes() == Node.
        ELEMENTNODE)
    {
        String nombre = children.item(j).
            getNodeName();
        if(children.item(j).getFirstChild()!=null)
        {
            if(nombre.equals("description"))
            {
                description = children.item
                    (j).getFirstChild().
                    getNodeValue();
            }
            else if(nombre.equals("sound"))
            {
                sound = children.item(j).
                    getFirstChild().
                    getNodeValue();
            }
            else if (nombre.equals("file3ds"))
            {
                file3ds = children.item(j).
                    getFirstChild().
                    getNodeValue();
                int dotposition= file3ds.
                    lastIndexOf(".");
                extensionFile3ds = file3ds.
                    substring(dotposition +
                        1, file3ds.length());
            }
            else if(nombre.equals("texture"))
            {
                texture = children.item(j).
                    getFirstChild().
                    getNodeValue();
            }
        }
    }
}
Object3D objData = Object3DCreator.createObject(res ,
    idObject, nameObject, description ,
    file3ds , extensionFile3ds , texture , sound ,
    order);
//Object3D objData = new Object3D(idObject , nameObject ,
    description , new File3D(nameFile3ds , extensionFile3ds),
    sound);
KnowledgeBase.getInstance().saveObject(objData);
    }
}
```

```

/**
 * Método que carga la información de un objeto
 * @param res Recursos de la aplicación
 * @param urlXMLObjects Ruta del fichero XML de objetos
 * @param urlXSDObjects Ruta del fichero XSD de objetos
 * @param idObject Identificador del objeto a buscar
 * @param order Orden en que se cargará el objeto
 * @throws LoadXMLException Si hay errores en la carga
 * @throws XPathExpressionException Si se produce un error en una
 *     expresión XPath
 * @throws NoSuchExtensionObject3DException Si el objeto no tiene
 *     una extensión 3D válida
 * @throws IOException En caso de que no se pueda acceder a la URL
 *     lanzaría una IOException
 */
public void load (Resources res , String urlXMLObjects , String
    urlXSDObjects , int idObject , int order)
    throws LoadXMLException , XPathExpressionException ,
        NoSuchExtensionObject3DException , IOException
{
    this.xml = urlXMLObjects;
    this.xsd = urlXSDObjects;
    loadFile(true);
    searchObject(res , idObject , order);
}
}

```

Paquete data C.15: ObjectsXMLCompiler.java

### CompilerGenerator.java

```

package com.ivanlorenzo.data;

import java.io.IOException;

import javax.xml.xpath.XPathExpressionException;

import android.content.res.Resources;

import com.ivanlorenzo.data.exceptions.LoadXMLException;
import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;

/**
 * Clase que es la encargada de iniciar la carga de ficheros
 * @author Iván Lorenzo Rubio
 */
public class CompilerGenerator
{
    /**
     * Instancia del gestor de objetos
     */
}

```

```
public final static KnowledgeBase BASE = KnowledgeBase.getInstance
();

/**
 * Método que se llamará desde para cargar los datos
 * @param res Recursos de la aplicación
 * @param urlXMLPatterns Fichero XML con los patrones
 * @param urlXSDPatterns Fichero XSD de los patrones
 * @param urlXMLObjects Fichero XML con los objetos
 * @param urlXSDObjects Fichero XSD de los objetos
 * @throws LoadXMLException Si hay algún error en la carga
 * @throws XPathExpressionException Si se produce un error en una
 *     expresión XPath
 * @throws NoSuchExtensionObject3DException Si el objeto no tiene
 *     una extensión 3D válida
 * @throws IOException En caso de que no se pueda acceder a la URL
 *     lanzaría una IOException
 */
public static void compile(Resources res, String urlXMLPatterns,
    String urlXSDPatterns,
    String urlXMLObjects, String urlXSDObjects ) throws
    LoadXMLException, XPathExpressionException,
    NoSuchExtensionObject3DException, IOException
{
    AugmentedRealityXMLCompiler compilerPatt = new
    AugmentedRealityXMLCompiler();
    compilerPatt.load(res, urlXMLPatterns, urlXSDPatterns,
    urlXMLObjects, urlXSDObjects);
}
}
```

Paquete data C.16: CompilerGenerator.java

### KnowledgeBase.java

```
package com.ivanlorenzo.data;

import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

import com.ivanlorenzo.data.objects.Object3D;

import min3d.animation.AnimationObject3d;
import min3d.core.Object3d;
import min3d.core.Object3dContainer;

import com.ivanlorenzo.parser.IParser;
import com.ivanlorenzo.parser.Parser;
import com.ivanlorenzo.parser.Parser.Type;
```

```

import android.content.res.Resources;

import com.ivanlorenzo.gui.R;

/**
 * Clase <em>Singleton</em> que contiene la base de conocimiento de la
 * aplicación. Contendr tanto la lista de patrones (markers) como la lista
 * de localizaciones y de objetos.
 * @author Iván Lorenzo Rubio
 *
 */
public class KnowledgeBase
{
    /**
     * Única instancia de la base de conocimiento
     */
    private static KnowledgeBase instance = new KnowledgeBase();

    /**
     * Lista de patrones (markers) de la aplicación
     */
    private List<Pattern> patterns;

    /**
     * Lista de geolocalizaciones de la aplicación
     */
    private List<GeoLocation> locations;

    /**
     * Lista de objetos cargados en la aplicación
     */
    private List<Object3D> objects;

    /**
     * Constructor sin parámetros que simplemente crea las estructuras
     * de datos para almacenar la información
     */
    private KnowledgeBase()
    {
        patterns = new ArrayList<Pattern>();
        objects = new ArrayList<Object3D>();
        locations = new ArrayList<GeoLocation>();
    }

    /**
     * Método estático que devuelve la única instancia de la clase
     * @return La instancia de la clase
     */
    public static KnowledgeBase getInstance()
    {
        return instance;
    }

    /**

```

```
* Guarda un objeto 3d
* @param objData Objeto a guardar
*/
public void saveObject(Object3D objData)
{
    objects.add(objData);
}

/**
* Guarda un patrón
* @param patData Patrón a guardar
*/
public void savePattern(Pattern patData)
{
    patterns.add(patData);
}

/**
* Guarda una localización GPS
* @param location Localización a guardar
*/
public void saveLocation(GeoLocation location)
{
    locations.add(location);
}

/**
* Método que borra las listas de objetos y patrones
*/
public void clear()
{
    patterns.clear();
    objects.clear();
    locations.clear();
}

/**
* Obtiene la lista de patrones de la base de conocimiento
* @return Lista de patrones
*/
public List<Pattern> getPatterns()
{
    return patterns;
}

/**
* Devuelve una lista de flujo de bytes correspondiente a los
    patrones
* @param res Recursos de la aplicación
* @return Lista de flujo de bytes correspondientes a los patrones
*/
public ArrayList<InputStream> getPatterns(Resources res)
{
    ArrayList<InputStream> patt = new ArrayList<InputStream>();
    for(Pattern pattern : getPatterns())
    {
```

```

        patt.add(
            res.openRawResource (
                res.getIdentifier(
                    pattern.
                        getName
                        () ,"raw"
                        , R.
                        class.
                        getPackage
                        ().
                        getName
                        ())
                ) );
    }
    return patt;
}

/**
 * Devuelve el patrón que se corresponde con un identificador
 * @param id Identificador del patrón
 * @return Patrón buscado
 */
public Pattern getPattern(int id)
{
    Pattern searchpattern = null;
    for(Pattern pattern : patterns)
    {
        if (pattern.getId() == id )
            searchpattern = pattern;
    }
    return searchpattern;
}

/**
 * Listado de localizaciones
 * @return Lista de geolocalizaciones de la base de conocimiento
 */
public List<GeoLocation> getLocations()
{
    return locations;
}

/**
 * Devuelve la lista de objetos de la base de conocimiento
 * @return Lista de objetos
 */
public List<Object3D> getObjects()
{
    return objects;
}

```

```
/**
 * Devuelve un listado de objetos que se deben visualizar en la
 * aplicación
 * @param res Recursos de la aplicación
 * @return Listado de objetos para ser visualizados
 */
public List<Object3d> getObjectsToScene(Resources res)
{

    List<Object3d> objectsToScene = new ArrayList<Object3d>();
    IParser parser = null;

    for(Object3D object : objects )
    {

        if(object.getFile().startsWith("http://"))
            parser = Parser.createParser(object.getType()
            (),
            object.getFile(), object.
            getTexture());

        else
            parser = Parser.createParser(object.getType()
            (),
            res, R.class.getPackage().
            getName()+" :raw/"+
            object.
            getFileWithoutExtension
            ()
            , false);

        if(object.getType() == Type.MD2)
        {
            AnimationObject3d animationObject3d = null;
            parser.parse();

            animationObject3d = parser.
            getParsedAnimationObject();
            animationObject3d.rotation().z = -90.0f;
            animationObject3d.scale().x =
            animationObject3d.scale().y =
            animationObject3d.scale().z = 1.0f;
            animationObject3d.setFps(30);
            objectsToScene.add(animationObject3d);
        }

        else if( object.getType() == Type.MAX3DS ||
            object.getType() == Type.OBJ )
        {
            Object3dContainer container;
            parser.parse();

            container = parser.getParsedObject();
        }
    }
}
```

```

        for(Object3d subObject : container.children
            ())
        {
            subObject.rotation().x = 90.0f;
            subObject.scale().x = subObject.
                scale().y = subObject.scale().z
                = 1f;
            objectsToScene.add(subObject);
        }
    }
    return objectsToScene;
}

/**
 * Devuelve un objeto buscado por su identificador
 * @param id Identificador del objeto a buscar
 * @return Objeto buscado
 */
public Object3D getObject(int id)
{
    Object3D searchobject = null;
    for(Object3D object : objects)
    {
        if (object.getId() == id )
            searchobject = object;
    }
    return searchobject;
}
}

```

Paquete data C.17: KnowledgeBase.java

### Pattern.java

```

package com.ivanlorenzo.data;

/**
 * Clase que contiene las propiedades del patrón al que le irá asignado un
 * objeto
 * @author Iván Lorenzo Rubio
 *
 */
public class Pattern

```

```
{

    /**
     * Identificador del patrón
     */
    private int id;

    /**
     * Nombre del patrón
     */
    private String name;

    /**
     * Identificador del objeto que tiene asignado
     */
    private int idObject;

    /**
     * Constructor sin parámetros
     */
    public Pattern()
    {
    }

    /**
     * Constructor
     * @param id Identificador del patrón
     * @param name Nombre del patrón
     * @param idObject Identificador del objeto que tiene asignado
     */
    public Pattern( int id, String name, int idObject)
    {
        this.id = id;
        this.name = name;
        this.idObject = idObject;
    }

    /**
     * Devuelve el identificador del patrón
     * @return Identificador del patrón
     */
    public int getId() {
        return id;
    }

    /**
     * Establece el identificador del patrón
     * @param id Identificador del patrón
     */
    public void setId(int id) {
        this.id = id;
    }
}
```

```

    /**
     * Devuelve el identificador del objeto asignado al patrón
     * @return Identificador del objeto asignado al patrón
     */
    public int getIdObject() {
        return idObject;
    }

    /**
     * Establece el identificador del objeto asignado al patrón
     * @param idObject Identificador del objeto asignado al patrón
     */
    public void setIdObject(int idObject) {
        this.idObject = idObject;
    }

    /**
     * Devuelve el nombre del patrón
     * @return El nombre del patrón
     */
    public String getName() {
        return name;
    }

    /**
     * Establece el nombre del patrón
     * @param name El nombre del patrón
     */
    public void setName(String name) {
        this.name = name;
    }
}

```

Paquete data C.18: Pattern.java

### GeoLocation.java

```

package com.ivanlorenzo.data;

import android.location.Location;

/**
 * Clase que se corresponde con una localización de los ficheros de
 * configuración de la aplicación. Tendrá un identificador, un nombre, una
 * localización GPS
 * y un objeto asociado
 * @author Iván Lorenzo Rubio
 */
public class GeoLocation
{

```

```
/**
 * Identificador de la localización
 */
private int id;

/**
 * Nombre del lugar
 */
private String name;

/**
 * Localización GPS
 */
private Location location;

/**
 * Identificador del objeto asociado
 */
private int idObject;

/**
 * Constructor en el que se establecen los atributos
 * @param id Identificador de la localización
 * @param name Nombre del lugar
 * @param latitude Latitud geográfica
 * @param longitude Longitud geográfica
 * @param idObject Identificador del objeto asociado
 */
public GeoLocation(int id, String name, double latitude, double
    longitude, int idObject)
{
    location = new Location("ObjectLocation");
    location.setLatitude(latitude);
    location.setLongitude(longitude);
    this.id = id;
    this.name = name;
    this.idObject = idObject;
}

/**
 * Devuelve la localización
 * @return Localización
 */
public Location getLocation()
{
    return location;
}

/**
 * Devuelve el identificador del objeto asociado
```

```

        * @return Identificador del objeto asociado
        */
    public int getIdObject()
    {
        return idObject;
    }

    /**
     * Devuelve el identificador de la localización
     * @return Identificador de la localización
     */
    public int getId() {
        return id;
    }

    /**
     * Devuelve el nombre del lugar
     * @return Nombre del lugar
     */
    public String getName() {
        return name;
    }
}

```

Paquete data C.19: GeoLocation.java

## C.4.2. Paquete com.ivanlorenzo.data.objects

### objects/Object3DCreator.java

```

package com.ivanlorenzo.data.objects;

import com.ivanlorenzo.gui.R;

import android.content.res.Resources;

import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;
import com.ivanlorenzo.data.objects.audio.Audio;
import com.ivanlorenzo.data.objects.audio.LocalAudio;
import com.ivanlorenzo.data.objects.audio.RemoteAudio;

/**
 * Clase factoría que será la encargada de crear un objeto determinado en
 * función de su tipo. Para esto
 * utiliza alguno de sus métodos estáticos
 * @author Iván Lorenzo Rubio
 *
 */

```

```
public final class Object3DCreator
{
    /**
     * Constructor privado para que no se puedan crear instancias de
     * esta clase
     */
    private Object3DCreator() {}

    /**
     * Método encargado de crear un objeto determinado en función del
     * tipo del fichero leído
     * @param res Recursos de la aplicación
     * @param id Identificador del objeto
     * @param name Nombre del objeto
     * @param description Descripción del objeto
     * @param file Archivo leído
     * @param type Tipo del fichero
     * @param texture Imagen de textura
     * @param mediaPlayer Audio asignado al objeto
     * @param order Orden en que se almacenará
     * @return El objeto creado
     * @throws NoSuchExtensionObject3DException Se lanzará una
     * excepción si no coincide con ninguno de los
     * tipos de objetos soportados en la aplicación
     */
    public static Object3D createObject(Resources res, int id, String
        name,
        String description, String file, String type,
        String texture,
        String mediaPlayer, int order) throws
        NoSuchExtensionObject3DException
    {
        Audio audio = generateAudio(res, mediaPlayer, id);

        if(type.equalsIgnoreCase("3DS"))
            return new Max3DSObject(id, name, description, file
                , texture, audio, order);
        else if(type.equalsIgnoreCase("OBJ"))
            return new OBJObject(id, name, description, file,
                texture, audio, order);
        else if(type.equalsIgnoreCase("MD2"))
            return new MD2Object(id, name, description, file,
                texture, audio, order);
        else if(type.equalsIgnoreCase(""))
            return new Max3DSObject(id, name, description, res
                .getString(R.string.NO_OBJECT_3D), texture,
                audio, order);
        throw new NoSuchExtensionObject3DException();
    }

    /**
     * Método que genera un Audio concreto
     * @param res Recursos de la aplicación
     * @param audio URL del fichero de sonido
     */
}
```

```

    * @param id Identificador del objeto al que se asociará el audio
    * @return Una instancia concreta de Audio
    */
    private static Audio generateAudio(Resources res, String audio, int
        id)
    {
        if (audio == "")
            return null;
        if (audio.startsWith("http://"))
        {
            return new RemoteAudio(audio, id);
        }

        int dotposition= audio.lastIndexOf(".");
        return new LocalAudio(
            res.getIdentifier(
                audio.substring(0,
                    dotposition), "raw",
                R.class.getPackage().
                    getName()));
    }
}

```

Paquete data C.20: objects/Object3DCreator.java

### objects/Object3D.java

```

package com.ivanlorenzo.data.objects;

import com.ivanlorenzo.data.objects.audio.Audio;

import com.ivanlorenzo.parser.Parser.Type;

/**
 * Clase abstracta que representa un objeto 3d proveniente de los ficheros
 * de configuración
 * @author Iván Lorenzo Rubio
 *
 */
public abstract class Object3D
{
    /**
     * Identificador del objeto
     */
    int id;

    /**
     * Nombre del objeto
     */
    String name;

    /**
     * Fichero 3D
     */
}

```

```
    */
    File3D file3D;

    /**
     * Textura del objeto
     */
    String texture;

    /**
     * Descripción sobre el objeto
     */
    String description;

    /**
     * Audio
     */
    Audio audio;

    /**
     * Orden en que el objeto se almacena
     */
    int order;

    /**
     * Constructor de la clase
     * @param id Identificador del objeto
     * @param name Nombre del objeto
     * @param description Descripción del objeto
     * @param file Nombre del fichero
     * @param texture Textura
     * @param audio Audio del objeto
     * @param order Orden en el que se almacena
     */
    public Object3D(int id, String name, String description, String
        file, String texture,
        Audio audio, int order)
    {
        this.id = id;
        this.name = name;
        this.description = description;
        this.audio = audio;
        this.order = order;
        this.texture = texture;
    }

    /**
     * Devuelve el identificador
     * @return Identificador del objeto
     */
    public int getId() {
        return id;
    }

    /**
     * Devuelve el nombre
```

```

    * @return Nombre del objeto
    */
    public String getName() {
        return name;
    }

    /**
     * Comprueba si el objeto tiene un fichero 3d asociado
     * @return Verdadero si dispone de un fichero 3d, falso en caso
     * contrario
     */
    public boolean has3DFile()
    {
        return file3D!=null ? true : false;
    }

    /**
     * Devuelve el nombre del fichero
     * @return Nombre del fichero 3D
     */
    public String getFile()
    {
        return file3D.getName();
    }

    /**
     * Devuelve el nombre del fichero sin extensión
     * @return Nombre del fichero suprimiendo la extensión
     */
    public String getFileWithoutExtension()
    {
        int dotposition= file3D.getName().lastIndexOf(".");
        return file3D.getName().substring(0,dotposition);
    }

    /**
     * Tipo del fichero
     * @return El tipo del fichero
     */
    public Type getType()
    {
        return file3D.getType();
    }

    /**
     * Devuelve la descripción
     * @return La descripción del objeto
     */
    public String getDescription() {
        return description;
    }

    /**
     * Devuelve la textura

```

```
    * @return La textura del objeto
    */
    public String getTexture()
    {
        return texture;
    }

    /**
     * Devuelve el audio
     * @return Audio del objeto
     */
    public Audio getAudio()
    {
        return audio;
    }

    /**
     * Devuelve el identificador del audio
     * @return Identificador del audio
     */
    public int getAudioID() {
        return audio.getId();
    }

    /**
     * Devuelve el orden del objeto
     * @return Orden del objeto
     */
    public int getOrder()
    {
        return order;
    }

    /**
     * Clase que representa un fichero 3D
     * @author Iván Lorenzo Rubio
     */
    class File3D
    {
        /**
         * Nombre del fichero 3D
         */
        private String name;

        public String getName() {
            return name;
        }

        /**
         * Tipo del fichero 3D
         */
        private Type type;

        public Type getType() {
```

```

        return type;
    }

    /**
     * Constructo
     * @param name Nombre del fichero
     * @param type Extensión del fichero
     */
    public File3D(String name, Type type)
    {
        this.name = name;
        this.type = type;
    }
}
}

```

Paquete data C.21: objects/Object3D.java

#### objects/Max3DSObject.java

```

package com.ivanlorenzo.data.objects;

import com.ivanlorenzo.data.objects.audio.Audio;

import com.ivanlorenzo.parser.Parser;

/**
 * Clase que representa un objeto con formato 3ds. Esta clase hereda de
 * Object3D
 * @author Iván Lorenzo Rubio
 */
public class Max3DSObject extends Object3D
{
    /**
     * Constructor
     * @param id Identificador del objeto
     * @param name Nombre del objeto
     * @param description Descripción del objeto
     * @param file Fichero 3d
     * @param texture Imagen de textura
     * @param audio Audio del objeto
     * @param order Número de orden en el que almacenará
     */
    public Max3DSObject(int id, String name, String description, String
        file, String texture,
        Audio audio, int order) {
        super(id, name, description, file, texture, audio, order);
        this.file3D = new File3D(file, Parser.Type.MAX_3DS);
    }
}

```

```
}
```

Paquete data C.22: objects/Max3DSObject.java

### objects/MD2Object.java

```
package com.ivanlorenzo.data.objects;

import com.ivanlorenzo.data.objects.audio.Audio;

import com.ivanlorenzo.parser.Parser;

/**
 * Clase que representa un objeto con formato md2. Esta clase hereda de
 *   Object3D
 * @author Iván Lorenzo Rubio
 *
 */
public class MD2Object extends Object3D
{

    /**
     * Constructor
     * @param id Identificador del objeto
     * @param name Nombre del objeto
     * @param description Descripción del objeto
     * @param file Fichero 3d
     * @param texture Imagen de textura
     * @param audio Audio del objeto
     * @param order Número de orden en el que almacenará
     */
    public MD2Object(int id, String name, String description, String
        file, String texture,
        Audio audio, int order) {
        super(id, name, description, file, texture, audio, order);
        this.file3D = new File3D(file, Parser.Type.MD2);
    }
}
```

Paquete data C.23: objects/MD2Object.java

### objects/OBJObject.java

```
package com.ivanlorenzo.data.objects;

import com.ivanlorenzo.data.objects.audio.Audio;

import com.ivanlorenzo.parser.Parser;

/**
 * Clase que representa un objeto con formato obj. Esta clase hereda de
 *   Object3D

```

```

* @author Iván Lorenzo Rubio
*
*/
public class OBJObject extends Object3D
{
    /**
     * Constructor
     * @param id Identificador del objeto
     * @param name Nombre del objeto
     * @param description Descripción del objeto
     * @param file Fichero 3d
     * @param texture Imagen de textura
     * @param audio Audio del objeto
     * @param order Número de orden en el que almacenará
     */
    public OBJObject(int id, String name, String description, String
        file, String texture,
        Audio audio, int order) {
        super(id, name, description, file, texture, audio, order);
        this.file3D = new File3D(file, Parser.Type.OBJ);
    }
}

```

Paquete data C.24: objects/OBJObject.java

### C.4.3. Paquete com.ivanlorenzo.data.objects.audio

objects/audio/MediaPlayerAdapter.java

```

package com.ivanlorenzo.data.objects.audio;

import java.io.IOException;

import com.ivanlorenzo.gui.ARStatus;
import com.ivanlorenzo.location.LocationData;
import com.ivanlorenzo.location.observer.ARObservable;
import com.ivanlorenzo.location.observer.IARObserver;

import android.content.Context;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;

/**
 * Clase adaptadora del patrón <em>Adapter</em> que adapta la clase
 * MediaPlayer y además es un sujeto observable
 * ya que cuando finalice el audio deberá avisar a sus observadores para
 * que realicen las acciones oportunas
 * @author Iván Lorenzo Rubio
 *
 */
public class MediaPlayerAdapter extends ARObservable<LocationData>
{

```

```
/**
 * MediaPlayer que vamos a adaptar
 */
private MediaPlayer mediaPlayer;

/**
 * Identificador del player
 */
private int resID;

/**
 * Variable que nos indicará si está sonando o no
 */
private boolean isPlaying = false;

/**
 * Variable que nos indica si está marcado como repetición
 */
private boolean loop = false;

/**
 * Constructor de la clase en el que se crea un objeto MediaPlayer
 * y gestiona el evento onCompletion,
 * que informará cuando finalice el audio
 * @param ctx Contexto de la aplicación
 * @param resID Identificador del audio
 */
public MediaPlayerAdapter(Context ctx, int resID)
{
    this.resID = resID;
    mediaPlayer = MediaPlayer.create(ctx, resID);
    mediaPlayer.setOnCompletionListener(new MediaPlayer.
        OnCompletionListener()
        {
            @Override
            public void onCompletion(MediaPlayer mp)
            {
                isPlaying = false;
                if (loop)
                {
                    mp.start();
                }
                notifyObservers();
            }
        });
    mediaPlayer.start();
}

/**
 * Constructor de la clase usado para reproducir un audio remoto,
 * en el que se crea un objeto
 * de la clase MediaPlayer y se le establece un datasource que será
 * la url con el fichero a reproducir.
```

```

    * También gestiona el evento onCompletion que informará cuando
      finalice el audio.
    * @param ctx Contexto de la aplicación
    * @param url Dirección del fichero a reproducir
    * @param id Identificador del audio
    */
public MediaPlayerAdapter(Context ctx, String url, int id)
{
    this.resID = id;
    try
    {
        //this.resID = resID;
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setDataSource(ctx, Uri.parse(url));
        mediaPlayer.setAudioStreamType(AudioManager.
            STREAMMUSIC);
        mediaPlayer.prepare();

        mediaPlayer.setOnCompletionListener(new MediaPlayer
            .OnCompletionListener()
        {
            @Override
            public void onCompletion(MediaPlayer mp)
            {
                isPlaying = false;
                if (loop)
                {
                    mp.start();
                }
                notifyObservers();
            }
        });

        isPlaying = true;
        mediaPlayer.start();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}

/**
 * Método que inicia la reproducción del audio
 */
public synchronized void play()
{
    if (isPlaying)
        return;
    if (mediaPlayer != null) {
        isPlaying = true;
        mediaPlayer.start();
    }
}

/**

```

```
    * Método que detiene la reproducción del audio
    */
    public synchronized void stop()
    {
        try
        {
            loop = false;
            if (isPlaying)
            {
                isPlaying = false;
                mediaPlayer.pause();
            }
        } catch (Exception e)
        {
            System.err.println("Error");
        }
    }

    /**
     * Método que pone el reproductor en modo repetición
     */
    public synchronized void loop()
    {
        loop = true;
        isPlaying = true;
        mediaPlayer.start();
    }

    /**
     * Libera el reproductor
     */
    public void release()
    {
        if (mediaPlayer != null)
        {
            mediaPlayer.release();
            mediaPlayer = null;
        }
    }

    /**
     * Devuelve el identificador del audio
     * @return El identificador
     */
    public int getResID()
    {
        return resID;
    }

    @Override
    public void notifyObservers()
    {

```

```

        LocationData data = LocationData.getInstance();
        data.setObject(null);
        data.setStatus(ARStatus.AR_Marker);
        for(IARObserver<LocationData> observer : observers)
            observer.update(data);
    }
}

```

Paquete data C.25: objects/audio/MediaPlayerAdapter.java

### objects/audio/Audio.java

```

package com.ivanlorenzo.data.objects.audio;

import android.content.Context;

/**
 * Clase abstracta Audio que tendrá un método abstracto que devolverá una
 * objeto de la clase
 * MediaPlayerAdapter
 * @author Iván Lorenzo Rubio
 *
 */
public abstract class Audio
{
    /**
     * Identificador del objeto al que pertenece el audio
     */
    int id;

    /**
     * Devuelve el identificador del objeto
     * @return El identificador del objeto al que pertenece el audio
     */
    public int getId()
    {
        return id;
    }

    /**
     * Método que devuelve la instancia de la clase MediaPlayerAdapter
     * @param ctx Contexto de la aplicación
     * @return Un objeto MediaPlayerAdapter
     */
    public abstract MediaPlayerAdapter getMediaPlayer(Context ctx);
}

```

Paquete data C.26: objects/audio/Audio.java

### objects/audio/LocalAudio.java

```

package com.ivanlorenzo.data.objects.audio;

```

```
import android.content.Context;

/**
 * Clase que hereda de Audio. Se utilizará para representar audios que
 * estén en el dispositivo
 * @author Iván Lorenzo Rubio
 *
 */
public class LocalAudio extends Audio
{
    /**
     * Constructor
     * @param id Identificador del objeto
     */
    public LocalAudio(int id)
    {
        this.id = id;
    }

    @Override
    public MediaPlayerAdapter getMediaPlayer(Context ctx)
    {
        return new MediaPlayerAdapter(ctx, id);
    }
}
```

Paquete data C.27: objects/audio/LocalAudio.java

### objects/audio/RemoteAudio.java

```
package com.ivanlorenzo.data.objects.audio;

import android.content.Context;

/**
 * Representa un audio localizado en una ubicación remota. Hereda de la
 * clase Audio
 * @author Iván Lorenzo Rubio
 *
 */
public class RemoteAudio extends Audio
{
    /**
     * URL donde se encuentra el fichero remotamente
     */
    private String path;

    /**
     * Constructor
     * @param path URL remota
     * @param id Identificador del objeto que tiene asociado el audio
     */
    public RemoteAudio (String path, int id)
```

```

    {
        this.path = path;
        this.id = id;
    }

    @Override
    public MediaPlayerAdapter getMediaPlayer(Context ctx)
    {
        return new MediaPlayerAdapter(ctx, path, id);
    }
}

```

Paquete data C.28: objects/audio/RemoteAudio.java

#### C.4.4. Paquete com.ivanlorenzo.data.exceptions

##### exceptions/LoadXMLException.java

```

package com.ivanlorenzo.data.exceptions;

/**
 * Excepción en el caso que se produzca algún error en la carga de datos
 * @author Iván Lorenzo Rubio
 *
 */
public class LoadXMLException extends Exception
{

    /**
     * UID
     */
    private static final long serialVersionUID = 4949399250566900013L;

    /**
     * Crea una nueva excepción
     */
    public LoadXMLException()
    {
    }

    /**
     * Crea una nueva excepción
     * @param mensaje Mensaje a mostrar
     */
    public LoadXMLException(String mensaje)
    {
        super(mensaje);
    }
}

```

Paquete data C.29: exceptions/LoadXMLException.java

### exceptions/NoSuchExtensionObject3DException.java

```
package com.ivanlorenzo.data.exceptions;

/**
 * Excepción que se lanzará cuando intentemos cargar un fichero 3D cuya
 * extensión no sea .3DS, .OBJ o .MAX
 * @author Iván Lorenzo Rubio
 *
 */
public class NoSuchExtensionObject3DException extends Exception
{
    /**
     * UID
     */
    private static final long serialVersionUID = 1L;
}
```

Paquete data C.30: exceptions/NoSuchExtensionObject3DException.java

### C.4.5. Paquete com.ivanlorenzo.location

#### ARLocationListener.java

```
package com.ivanlorenzo.location;

import java.util.List;

import com.ivanlorenzo.data.KnowledgeBase;
import com.ivanlorenzo.data.GeoLocation;
import com.ivanlorenzo.gui.ARStatus;
import com.ivanlorenzo.location.observer.ARObservable;
import com.ivanlorenzo.location.observer.IARObserver;

import android.location.Location;
import android.location.LocationListener;
import android.os.Bundle;

/**
 * Clase "sujeto observable" del patrón observer que, a su vez, implemente
 * LocationListener y recibe cuándo
 * se produce un cambio en la posición GPS y en caso de que esta posición
 * contenga cercano una localización
 * de los ficheros de configuración, notificará a todos los observadores
 * que debe mostrarse la información
 * asociada a dicha posición.
 * @author Iván Lorenzo Rubio
 *
 */
```

```

public class ARLocationListener extends ARObservable<LocationData>
implements LocationListener
{

    /**
     * Distancia máxima a la que mostraremos la información de las
     * posiciones
     */
    private float distance;

    /**
     * Constructor de la clase que recibe un observador y la distancia
     * máxima de la nuestra a la que mostrar
     * información
     * @param observer Observador
     * @param distance Distancia máxima de la nuestra a la que mostrar
     * información
     */
    public ARLocationListener(IARObserver<LocationData> observer , float
        distance)
    {
        this.addObserver(observer);
        this.distance = distance;
    }

    @Override
    public void onLocationChanged(Location loc)
    {
        loc.getLatitude();
        loc.getLongitude();
        this.data = getLocationData(loc);
        notifyObservers();
    }

    /**
     * Devuelve el estado de la localización. En caso que exista alguna
     * localización cercana, devolverá el objeto
     * asociado junto con el estado.
     * @param currentLocation Nuestra posición actual
     * @return El estado de la localización
     */
    private LocationData getLocationData(Location currentLocation)
    {
        LocationData data = LocationData.getInstance();
        List<GeoLocation> locations = KnowledgeBase.getInstance().
            getLocations();
        for(GeoLocation location : locations)
        {
            if( currentLocation.distanceTo(location.getLocation
                ()) < distance )
            {
                data.setObject( KnowledgeBase.getInstance
                    ().getObject(location.getIdObject()) );
            }
        }
    }
}

```

```
                data.setStatus(ARStatus.AR_GPS);
                return data;
            }

            data.setObject(null);
            data.setStatus(ARStatus.AR_Marker);
            return data;
        }

        @Override
        public void onProviderDisabled(String provider)
        {
        }

        @Override
        public void onProviderEnabled(String provider)
        {
        }

        @Override
        public void onStatusChanged(String provider, int status, Bundle
            extras)
        {
        }
    }
}
```

Paquete location C.31: ARLocationListener.java

### LocationData.java

```
package com.ivanlorenzo.location;

import com.ivanlorenzo.data.objects.Object3D;
import com.ivanlorenzo.gui.ARStatus;

/**
 * Clase que representa el estado de la localización. Se implementa como un
 * Singleton ya que sólo queremos tener
 * una única instancia
 * @author Iván Lorenzo Rubio
 */
public class LocationData
{
    /**
     * Instancia de la clase
     */
}
```

```
private static LocationData instance = new LocationData();

/**
 * Devuelve la única instancia de la clase
 * @return Instancia
 */
public static LocationData getInstance()
{
    return instance;
}

/**
 * Estado del sistema de realidad aumentada
 */
private ARStatus status;

/**
 * Objeto 3D que se está mostrando
 */
private Object3D object;

/**
 * Establece el estado del sistema
 * @param status Estado establecido
 */
public void setStatus(ARStatus status) {
    this.status = status;
}

/**
 * Establece el objeto 3D que se muestra
 * @param object Objeto 3D a mostrar
 */
public void setObject(Object3D object) {
    this.object = object;
}

/**
 * Devuelve el estado del sistema
 * @return Estado
 */
public ARStatus getStatus() {
    return status;
}

/**
 * Devuelve el objeto 3d que se mostrará
 * @return El objeto a mostrar
 */
public Object3D getObject() {
    return object;
}
```

```
}
```

Paquete location C.32: LocationData.java

### C.4.6. Paquete com.ivanlorenzo.location.observer

observer/IARObserver.java

```
package com.ivanlorenzo.location.observer;

/**
 * Interface genérica Observadora del patrón Observer. Este patrón define
 * una dependencia del tipo uno-a-muchos
 * entre objetos, de manera que cuando uno de los objetos cambia su estado,
 * el observador se encarga de notificar
 * este cambio a todos los otros dependientes.
 * @author Iván Lorenzo Rubio
 *
 * @param <T> El sujeto observado
 */
public interface IARObserver<T>
{
    /**
     * Realiza los cambios oportunos en el observador pasándole un
     * parámetro genérico
     * @param subject Se podrá corresponder con el estado del sujeto
     */
    public void update(T subject);
}
```

Paquete location C.33: observer/IARObserver.java

observer/IARObservable.java

```
package com.ivanlorenzo.location.observer;

/**
 * Interface genérica Observable del patrón Observer que tendrá un conjunto
 * de observadores y notificará a
 * estos cuando se produzca algún cambio de estado
 * @author Iván Lorenzo Rubio
 *
 * @param <T> El observador
 */
public interface IARObservable<T>
{
    /**
     * Método que añade un observador a la lista
     * @param observer Observador a añadir
     */
}
```

```

    public void addObserver(IARObserver<T> observer);

    /**
     * Borra un observador de la lista
     * @param observer Observador a borrar
     */
    public void removeObserver(IARObserver<T> observer);

    /**
     * Cuando se produzca un cambio en él, notificará a todos los
     * observadores a través de este método
     */
    public void notifyObservers();
}

```

Paquete location C.34: observer/IARObservable.java

#### observer/ARObservable.java

```

package com.ivanlorenzo.location.observer;

import java.util.ArrayList;
import java.util.List;

/**
 * Clase abstracta que implementa la interface IARObservable<T>. Contiene
 * un elemento genérico y una lista de
 * observadores genéricos
 * @author Iván Lorenzo Rubio
 *
 * @param <T> El observador
 */
public abstract class ARObservable<T> implements IARObservable<T>
{
    /**
     * Elemento genérico
     */
    protected T data;

    /**
     * Lista de observadores genéricos
     */
    protected List<IARObserver<T>> observers = new ArrayList<
        IARObserver<T>>();

    @Override
    public void addObserver(IARObserver<T> observer)
    {
        observers.add(observer);
    }

    @Override
    public void removeObserver(IARObserver<T> observer)

```

```
{
    observers.remove(observer);
}

@Override
public void notifyObservers()
{
    for (IAObserver<T> observer : observers)
        observer.update(data);
}
}
```

Paquete location C.35: observer/ARObservable.java

### C.4.7. Paquete com.ivanlorenzo.parser

#### IParser.java

```
package com.ivanlorenzo.parser;

import min3d.animation.AnimationObject3d;
import min3d.core.Object3dContainer;

/**
 * Interface para los parsers de objetos 3D basada en min3d
 * @author Iván Lorenzo Rubio
 */
public interface IParser {
    /**
     * Empieza a parsear el objeto 3d
     */
    public void parse();

    /**
     * Devuelve el objeto parseado
     * @return El objeto parseado
     */
    public Object3dContainer getParsedObject();

    /**
     * Devuelve el objeto animado parseado
     * @return El objeto animado
     */
    public AnimationObject3d getParsedAnimationObject();
}
```

Paquete parser C.36: IParser.java

#### AParser.java

```
package com.ivanlorenzo.parser;
```

```

import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;

import min3d.Min3d;
import min3d.Shared;
import min3d.Utls;
import min3d.animation.AnimationObject3d;
import min3d.core.Object3dContainer;
import com.ivanlorenzo.parser.ParseObjectData;
import min3d.vos.Color4;
import min3d.vos.Number3d;
import min3d.vos.Uv;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.Bitmap.Config;
import android.util.Log;

/**
 * Clase abstracta Parser con la funcionalidad básica del proceso de parseo
 *
 *
 * @author Iván Lorenzo Rubio
 *
 */
public abstract class AParser implements IParser {

    protected Resources resources;
    protected String resourceID;
    protected String packageID;
    protected String currentMaterialKey;
    protected ArrayList<ParseObjectData> parseObjects;
    protected ParseObjectData co;
    protected boolean firstObject;
    protected TextureAtlas textureAtlas;
    protected ArrayList<Number3d> vertices;
    protected ArrayList<Uv> texCoords;
    protected ArrayList<Number3d> normals;
    protected boolean generateMipMap;
    protected HashMap<String, Material> materialMap;
    protected InputStream fileIn;

    protected boolean remoteLocation;
    protected String url;

    /**
     * Constructor básico
     */
    public AParser()
    {
        vertices = new ArrayList<Number3d>();

```

```
        texCoords = new ArrayList<Uv>();
        normals = new ArrayList<Number3d>();
        parseObjects = new ArrayList<ParseObjectData>();
        textureAtlas = new TextureAtlas();
        firstObject = true;
        materialMap = new HashMap<String, Material>();
    }

    /**
     * Constructor en el que se le pasa la dirección url del objeto a
     * parsear
     * @param url Dirección del objeto a parsear
     */
    public AParser(String url)
    {
        this();
        this.remoteLocation = true;
        this.packageID = "";
        this.url = url;
        try {
            fileIn = new URL(url).openConnection().
                getInputStream();
        } catch (IOException e)
        {
            e.printStackTrace();
        }
    }

    /**
     * Constructor del parser que recibe el recurso específico
     * @param resources Los recursos de la aplicación
     * @param resourceID El identificador del recurso a cargar
     * @param generateMipMap
     */
    public AParser(Resources resources, String resourceID, Boolean
        generateMipMap)
    {
        this();
        this.resources = resources;
        this.resourceID = resourceID;
        if (resourceID.indexOf(":") > -1)
            this.packageID = resourceID.split(":")[0];
        this.generateMipMap = generateMipMap;
        fileIn = resources.openRawResource(resources.getIdentifier(
            resourceID, null, null));
    }

    /**
     * Borra todas las estructuras de datos
     */
    protected void cleanup()
    {
        parseObjects.clear();
        textureAtlas.cleanup();
        vertices.clear();
        texCoords.clear();
    }
}
```

```

        normals.clear();
    }

    /**
     * Devuelve el objeto parseado. Será sobrescrito en el parseador
     * concreto.
     */
    public Object3dContainer getParsedObject() {
        return null;
    }

    /**
     * Devuelve el objeto animado parseado. Será sobrescrito en el
     * parseador concreto.
     */
    public AnimationObject3d getParsedAnimationObject() {
        return null;
    }

    /**
     * Lee una cadena de caracteres de un InputStream
     * @param stream Flujo de bytes a leer
     * @return El string leído
     * @throws IOException En caso de que se produzca alguna excepción
     */
    protected String readString(InputStream stream) throws IOException
    {
        String result = new String();
        byte inByte;
        while ((inByte = (byte) stream.read()) != 0)
            result += (char) inByte;
        return result;
    }

    /**
     * Lee un entero de un InputStream
     * @param stream Flujo de bytes a leer
     * @return El entero leído
     * @throws IOException En caso de que se produzca alguna excepción
     */
    protected int readInt(InputStream stream) throws IOException {
        return stream.read() | (stream.read() << 8) | (stream.read() << 16)
            | (stream.read() << 24);
    }

    /**
     * Lee un short de caracteres de un InputStream
     * @param stream Flujo de bytes a leer
     * @return El short leído
     * @throws IOException En caso de que se produzca alguna excepción
     */
    protected int readShort(InputStream stream) throws IOException {
        return (stream.read() | (stream.read() << 8));
    }

```

```
/**
 * Lee un número flotante de caracteres de un InputStream
 * @param stream Flujo de bytes a leer
 * @return El float leído
 * @throws IOException En caso de que se produzca alguna excepción
 */
protected float readFloat(InputStream stream) throws IOException {
    return Float.intBitsToFloat(readInt(stream));
}

/**
 * Realiza el parseo del objeto. Se sobrescribe en el parseador
 * concreto
 */
public void parse() {

/**
 * Contiene información sobre la textura.
 *
 * @author Iván Lorenzo Rubio
 */
protected class BitmapAsset
{
    /**
     * El bitmap de la textura
     */
    public Bitmap bitmap;
    /**
     * El identificador de la textura
     */
    public String key;
    /**
     * ID del recurso
     */
    public String resourceID;
    /**
     * U-coordinate offset
     */
    public float uOffset;
    /**
     * V-coordinate offset
     */
    public float vOffset;
    /**
     * U-coordinate scaling value
     */
    public float uScale;
    /**
     * V-coordinate scaling value
     */
}
```

```

        */
        public float vScale;
        public boolean useForAtlasDimensions;

        /**
         * Crear un objeto BitmapAsset
         * @param key Identificador de la textura
         * @param resourceID Identificador del recurso
         */
        public BitmapAsset(String key, String resourceID)
        {
            this.key = key;
            this.resourceID = resourceID;
            useForAtlasDimensions = false;
        }
    }

    /**
     * Cuando un modelo contiene una textura por cara, se crea un
     * TextureAtlas. Así
     * combinamos varias texturas
     *
     * @author Iván Lorenzo Rubio
     */
    protected class TextureAtlas {
        /**
         * El conjunto de textura combinadas en una
         */
        private ArrayList<BitmapAsset> bitmaps;
        /**
         * El bitmap de la textura
         */
        private Bitmap atlas;

        /**
         * Crear una instancia de una TextureAtlas
         */
        public TextureAtlas() {
            bitmaps = new ArrayList<BitmapAsset>();
        }

        /**
         * Identificador del atlas
         */
        private String atlasId;

        /**
         * Añade un BitmapAsset

```

```
    * @param ba
    */
    public void addBitmapAssetRemote(BitmapAsset ba)
    {
        bitmaps.add(ba);
    }

    /**
     * Añade un bitmap al atlas
     *
     * @param bitmap Bitmap añadido
     */
    public void addBitmapAsset(BitmapAsset ba) {
        BitmapAsset existingBA = getBitmapAssetByResourceID(
            ba.resourceID);

        if(existingBA == null)
        {
            int bmResourceID = resources.getIdentifier(
                ba.resourceID, null, null);
            if(bmResourceID == 0)
            {
                Log.d(Min3d.TAG, "Texture not found
                    : " + ba.resourceID);
                return;
            }

            Log.d(Min3d.TAG, "Adding texture " + ba.
                resourceID);

            Bitmap b = Utils.makeBitmapFromResourceId(
                bmResourceID);
            ba.useForAtlasDimensions = true;
            ba.bitmap = b;
        }
        else
        {
            ba.bitmap = existingBA.bitmap;
        }

        bitmaps.add(ba);
    }

    public BitmapAsset getBitmapAssetByResourceID(String
        resourceID)
    {
        int numBitmaps = bitmaps.size();

        for(int i=0; i<numBitmaps; i++)
        {
            if(bitmaps.get(i).resourceID.equals(
                resourceID))
                return bitmaps.get(i);
        }

        return null;
    }
}
```

```

    }

    /**
     * Genera una nueva textureAtlas
     */
    public void generate() {
        Collections.sort(bitmaps, new BitmapHeightComparer());

        if(bitmaps.size() == 0) return;

        BitmapAsset largestBitmap = bitmaps.get(0);
        int totalWidth = 0;
        int numBitmaps = bitmaps.size();
        int uOffset = 0;
        int vOffset = 0;

        for (int i = 0; i < numBitmaps; i++) {
            if(bitmaps.get(i).useForAtlasDimensions)
                totalWidth += bitmaps.get(i).bitmap
                    .getWidth();
        }

        atlas = Bitmap.createBitmap(totalWidth,
            largestBitmap.bitmap
                .getHeight(), Config.ARGB_8888);

        for (int i = 0; i < numBitmaps; i++) {
            BitmapAsset ba = bitmaps.get(i);
            BitmapAsset existingBA =
                getBitmapAssetByResourceID(ba.resourceID);

            if(ba.useForAtlasDimensions)
            {
                Bitmap b = ba.bitmap;
                int w = b.getWidth();
                int h = b.getHeight();
                int[] pixels = new int[w * h];

                b.getPixels(pixels, 0, w, 0, 0, w,
                    h);
                atlas.setPixels(pixels, 0, w,
                    uOffset, vOffset, w, h);

                ba.uOffset = (float) uOffset /
                    totalWidth;
                ba.vOffset = 0;
                ba.uScale = (float) w / (float)
                    totalWidth;
                ba.vScale = (float) h / (float)
                    largestBitmap.bitmap.getHeight();

                uOffset += w;
                b.recycle();
            }
        }
    }
}

```

```
        }
        else
        {
            ba.uOffset = existingBA.uOffset;
            ba.vOffset = existingBA.vOffset;
            ba.uScale = existingBA.uScale;
            ba.vScale = existingBA.vScale;
        }
    }
    /*
    FileOutputStream fos;
    try {
        fos = new FileOutputStream("/data/
            screenshot.png");
        atlas.compress(Bitmap.CompressFormat.PNG,
            100, fos);
        fos.flush();
        fos.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    */
    setId(Shared.textureManager().getNewAtlasId());
}

/**
 * Devuelve la textura generada
 *
 * @return El atlas generado
 */
public Bitmap getBitmap() {
    return atlas;
}

/**
 * Nos indica si el textureAtlas tiene algún bitmap
 *
 * @return Verdadero si tiene algún bitmap, falso en caso
        contrario
 */
public boolean hasBitmaps() {
    return bitmaps.size() > 0;
}

/**
 * Devuelve un bitmap con una textura determinada
 *
 * @param materialKey El nombre de la textura
 * @return El bitmap devuelto
 */
public BitmapAsset getBitmapAssetByName(String materialKey)
{

```

```

        int numBitmaps = bitmaps.size();

        for (int i = 0; i < numBitmaps; i++) {
            if (bitmaps.get(i).key.equals(materialKey))
                return bitmaps.get(i);
        }

        return null;
    }

    public void cleanup()
    {
        int numBitmaps = bitmaps.size();

        for (int i = 0; i < numBitmaps; i++) {
            bitmaps.get(i).bitmap.recycle();
        }

        if(atlas != null) atlas.recycle();
        bitmaps.clear();
        vertices.clear();
        texCoords.clear();
        normals.clear();
    }

    public void setId(String newAtlasId) {
        atlasId = newAtlasId;
    }

    public String getId() {
        return atlasId;
    }

    /**
     * Clase que compara el peso de dos bitmaps
     *
     * @author Iván Lorenzo Rubio
     */
    private class BitmapHeightComparer implements Comparator<
        BitmapAsset>
    {
        public int compare(BitmapAsset b1, BitmapAsset b2)
        {
            int height1 = b1.bitmap.getHeight();
            int height2 = b2.bitmap.getHeight();

            if (height1 < height2) {
                return 1;
            } else if (height1 == height2) {
                return 0;
            } else {
                return -1;
            }
        }
    }

```

```
        }

    }

    /**
     * Clase que representa un material para una textura
     * @author Iván Lorenzo Rubio
     *
     */
    protected class Material
    {
        public String name;
        public String diffuseTextureMap;
        public Color4 diffuseColor;

        public Material(String name) {
            this.name = name;
        }
    }
}
```

Paquete parser C.37: AParser.java

### Parser.java

```
package com.ivanlorenzo.parser;

import android.content.res.Resources;

/**
 * Clase factoría de Parsers. Especifica el tipo de parser y devuelve la
 * clase especializada de parser.
 * @author Iván Lorenzo Rubio
 *
 */
public class Parser {

    /**
     * Enumeración de los tipos de parsers de la aplicación
     * @author Iván Lorenzo Rubio
     */
    public static enum Type
    {
        OBJ,
        MAX_3DS,
        MD2
    };

    /**
     * Crea un parser de un tipo determinado
     * @param type Tipo de parser
     * @param resources Recursos de la aplicación
     * @param resourceID Identificador del recurso
     * @param generateMipMap
     */
}
```

```

    * @return Parser del tipo correspondiente
    */
    public static IParser createParser(Type type, Resources resources,
        String resourceID, boolean generateMipMap)
    {
        switch(type)
        {
            case OBJ:
                return new ObjParser(resources, resourceID,
                    generateMipMap);
            case MAX_3DS:
                return new Max3DSParser(resources,
                    resourceID, generateMipMap);
            case MD2:
                return new MD2Parser(resources, resourceID,
                    generateMipMap);
        }

        return null;
    }

    /**
    * Crea un parser de un tipo determinado
    * @param type Tipo de parser
    * @param url La dirección url del objeto
    * @param texture La dirección de la textura
    * @return Parser del tipo correspondiente
    */
    public static IParser createParser(Type type, String url, String
        texture)
    {
        switch(type)
        {
            case OBJ:
                return new ObjParser(url);
            case MAX_3DS:
                return new Max3DSParser(url, texture);
            case MD2:
                return new MD2Parser(url);
        }

        return null;
    }
}

```

Paquete parser C.38: Parser.java

### ParseObjectData.java

```

package com.ivanlorenzo.parser;

import java.util.ArrayList;
import java.util.HashMap;

```

```
import min3d.animation.AnimationObject3d;
import min3d.animation.KeyFrame;
import min3d.core.Object3d;
import com.ivanolorenzo.parser.AParser.BitmapAsset;
import com.ivanolorenzo.parser.AParser.Material;
import com.ivanolorenzo.parser.AParser.TextureAtlas;

import min3d.parser.ParseObjectFace;
import min3d.vos.Color4;
import min3d.vos.Face;
import min3d.vos.Number3d;
import min3d.vos.Uv;

/**
 * Clase que contiene la información de un objeto parseado
 * @author Iván Lorenzo Rubio
 *
 */
public class ParseObjectData {

    /**
     * Caras del objeto
     */
    protected ArrayList<ParseObjectFace> faces;

    /**
     * Número de caras
     */
    protected int numFaces = 0;

    /**
     * Vértices del objeto
     */
    protected ArrayList<Number3d> vertices;

    /**
     * Coordenadas de las texturas del objeto
     */
    protected ArrayList<Uv> texCoords;

    /**
     * Las normales del objeto
     */
    protected ArrayList<Number3d> normals;

    /**
     * Nombre del objeto
     */
    public String name;

    /**
     * Constructor sin parámetros
     */
    public ParseObjectData()
```

```

{
    this.vertices = new ArrayList<Number3d>();
    this.texCoords = new ArrayList<Uv>();
    this.normals = new ArrayList<Number3d>();
    this.name = "";
    faces = new ArrayList<ParseObjectFace>();
}

/**
 * Constructor que inicializa un objeto parseado
 * @param vertices ArrayList con los vértices
 * @param texCoords ArrayList con las coordenadas de las texturas
 * @param normals ArrayList con las normales del objeto
 */
public ParseObjectData(ArrayList<Number3d> vertices , ArrayList<Uv>
    texCoords , ArrayList<Number3d> normals)
{
    this.vertices = vertices;
    this.texCoords = texCoords;
    this.normals = normals;
    this.name = "";
    faces = new ArrayList<ParseObjectFace>();
}

/**
 * Devuelve un objeto AnimationObject3d parseado
 * @param textureAtlas
 * @param materialMap
 * @param frames
 * @return Objeto AnimationObject3d parseado
 */
public AnimationObject3d getParsedObject(TextureAtlas textureAtlas ,
    HashMap<String , Material> materialMap , KeyFrame[] frames)
{
    AnimationObject3d obj = new AnimationObject3d(numFaces * 3,
        numFaces, frames.length);
    obj.name(name);
    obj.setFrames(frames);

    parseObject(obj , materialMap , textureAtlas);

    return obj;
}

/**
 * Devuelve un Object3d parseado
 * @param materialMap
 * @param textureAtlas
 * @return Objeto Object3d parseado
 */
public Object3d getParsedObject(HashMap<String , Material>
    materialMap , TextureAtlas textureAtlas) {
    Object3d obj = new Object3d(numFaces * 3, numFaces);
    obj.name(name);

    parseObject(obj , materialMap , textureAtlas);
}

```

```
        return obj;
    }

    /**
     * Parsea un objeto
     * @param obj
     * @param materialMap
     * @param textureAtlas
     */
    private void parseObject(Object3d obj, HashMap<String, Material>
        materialMap, TextureAtlas textureAtlas)
    {
        int numFaces = faces.size();
        int faceIndex = 0;
        boolean hasBitmaps = textureAtlas.hasBitmaps();

        for (int i = 0; i < numFaces; i++) {
            ParseObjectFace face = faces.get(i);
            BitmapAsset ba = textureAtlas
                .getBitmapAssetByName(face.
                    materialKey);

            for (int j = 0; j < face.faceLength; j++) {
                Number3d newVertex = vertices.get(face.v[j]
                    );

                Uv newUv = face.hasuv ? texCoords.get(face.
                    uv[j]).clone()
                    : new Uv();
                Number3d newNormal = face.hasn ? normals.
                    get(face.n[j])
                    : new Number3d();
                Material material = materialMap.get(face.
                    materialKey);

                Color4 newColor = new Color4(255, 255, 0,
                    255);
                if(material != null && material.
                    diffuseColor != null)
                {
                    newColor.r = material.diffuseColor.
                        r;
                    newColor.g = material.diffuseColor.
                        g;
                    newColor.b = material.diffuseColor.
                        b;
                    newColor.a = material.diffuseColor.
                        a;
                }

                if(hasBitmaps && (ba != null))
                {
                    newUv.u = ba.uOffset + newUv.u * ba
                        .uScale;
                }
            }
        }
    }
}
```

```

        newUv.v = ba.vOffset + ((newUv.v +
            1) * ba.vScale) - 1;
    }
    obj.vertices().addVertex(newVertex, newUv,
        newNormal, newColor);
}

if (face.faceLength == 3) {
    obj.faces().add(
        new Face(faceIndex,
            faceIndex + 1, faceIndex
            + 2));
} else if (face.faceLength == 4) {
    obj.faces().add(
        new Face(faceIndex,
            faceIndex + 1, faceIndex
            + 3));

    obj.faces().add(
        new Face(faceIndex + 1,
            faceIndex + 2, faceIndex
            + 3));
}

faceIndex += face.faceLength;
}

if (hasBitmaps) {
    obj.textures().addById(textureAtlas.getId());
}

cleanup();
}

/**
 * Calcula la normal de un objeto
 * @param face
 */
public void calculateFaceNormal(ParseObjectFace face)
{
    Number3d v1 = vertices.get(face.v[0]);
    Number3d v2 = vertices.get(face.v[1]);
    Number3d v3 = vertices.get(face.v[2]);

    Number3d vector1 = Number3d.subtract(v2, v1);
    Number3d vector2 = Number3d.subtract(v3, v1);

    Number3d normal = new Number3d();
    normal.x = (vector1.y * vector2.z) - (vector1.z * vector2.y
    );
    normal.y = -((vector2.z * vector1.x) - (vector2.x * vector1
    .z));
    normal.z = (vector1.x * vector2.y) - (vector1.y * vector2.x
    );

    double normFactor = Math.sqrt((normal.x * normal.x) + (
        normal.y * normal.y) + (normal.z * normal.z));

```

```
        normal.x /= normFactor;
        normal.y /= normFactor;
        normal.z /= normFactor;

        normals.add(normal);

        int index = normals.size() - 1;
        face.n = new int [3];
        face.n[0] = index;
        face.n[1] = index;
        face.n[2] = index;
        face.hasn = true;
    }

    /**
     * Borra las caras
     */
    protected void cleanup() {
        faces.clear();
    }
}
```

Paquete parser C.39: ParseObjectData.java

### Max3DSParser.java

```
package com.ivanlorenzo.parser;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;

import min3d.Min3d;
import min3d.Shared;
import min3d.core.Object3dContainer;
import min3d.parser.ParseObjectFace;
import min3d.vos.Number3d;
import min3d.vos.Uv;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Log;

/**
 * Parser específico para los objetos Max3DS
 * @author Iván Lorenzo Rubio
 */
public class Max3DSParser extends AParser implements IParser {
    private final int IDENTIFIER_3DS = 0x4D4D;
    private final int MESHBLOCK = 0x3D3D;
    private final int OBJECTBLOCK = 0x4000;
```

```

private final int TRIMESH = 0x4100;
private final int TRLMATERIAL = 0x4130;
private final int VERTICES = 0x4110;
private final int FACES = 0x4120;
private final int TEXCOORD = 0x4140;
private final int TEX_MAP = 0xA200;
private final int TEX_NAME = 0xA000;
private final int TEX_FILENAME = 0xA300;
private final int MATERIAL = 0xAFFF;

private int chunkID;
private int chunkEndOffset;
private boolean endReached;
private String currentObjName;

private String texture;

/**
 * Constructor del parser
 * @param resources Recursos de la aplicación
 * @param resourceID Identificador del recurso
 * @param generateMipMap
 */
public Max3DParser(Resources resources, String resourceID, boolean
generateMipMap) {
    super(resources, resourceID, generateMipMap);
}

/**
 * Constructor de la clase en el que se le pasa la dirección del
 * objeto y la textura
 * @param url
 * @param texture
 */
public Max3DParser(String url, String texture)
{
    super(url);
    this.texture = texture;
}

@Override
public void parse()
{
    BufferedInputStream stream = new BufferedInputStream(fileIn
);

    Log.d(Min3d.TAG, "Start parsing object");

    co = new ParseObjectData();
    parseObjects.add(co);

    try {

```

```

        readHeader(stream);
        if (chunkID != IDENTIFIER_3DS)
        {
            Log.d(Min3d.TAG, "Not a valid .3DS file!");
            return;
        }
        else
        {
            Log.d(Min3d.TAG, "Found a valid .3DS file")
            ;
        }
        while (!endReached)
        {
            readChunk(stream);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    Log.d(Min3d.TAG, "End parsing object");
}

/**
 * Lee la cabecera del fichero 3ds
 * @param stream Stream con los datos a leer
 * @throws IOException
 */
private void readHeader(InputStream stream) throws IOException {
    chunkID = readShort(stream);
    chunkEndOffset = readInt(stream);
    endReached = chunkID < 0;
}

/**
 * Lee un pedazo del fichero
 * @param stream Stream con los datos a leer
 * @throws IOException
 */
private void readChunk(InputStream stream) throws IOException {
    readHeader(stream);

    switch (chunkID) {
        case MESHBLOCK:
            break;
        case OBJECTBLOCK:
            currentObjName = readString(stream);
            Log.d(Min3d.TAG, "Parsing object " + currentObjName
            );
            break;
        case TRIMESH:
            if (firstObject)
            {
                co.name = currentObjName;
                firstObject = false;
            }
    }
}

```

```

        else
        {
            co = new ParseObjectData();
            co.name = currentObjName;
            parseObjects.add(co);
        }
        break;
    case VERTICES:
        readVertices(stream);
        break;
    case FACES:
        readFaces(stream);
        break;
    case TEXCOORD:
        readTexCoords(stream);
        break;
    case TEX_NAME:
        currentMaterialKey = readString(stream);
        break;
    case TEX_FILENAME:
        readTexture(stream);

        break;
    case TRI_MATERIAL:
        String materialName = readString(stream);
        int numFaces = readShort(stream);

        for(int i=0; i<numFaces; i++)
        {
            int faceIndex = readShort(stream);
            co.faces.get(faceIndex).materialKey =
                materialName;
        }
        break;
    case MATERIAL:
        break;
    case TEX_MAP:
        break;
    default:
        skipRead(stream);
    }
}

/**
 * Finaliza la lectura del fichero
 * @param stream Stream con los datos a leer
 * @throws IOException
 */
private void skipRead(InputStream stream) throws IOException
{
    for(int i=0; (i<chunkEndOffset - 6) && !endReached; i++)
    {
        endReached = stream.read() < 0;
    }
}

```

```
/**
 * Lee los vértices del fichero 3ds
 * @param buffer Stream con los datos a leer
 * @throws IOException
 */
private void readVertices(InputStream buffer) throws IOException {
    float x, y, z, tmpy;
    int numVertices = readShort(buffer);

    for (int i = 0; i < numVertices; i++) {
        x = readFloat(buffer);
        y = readFloat(buffer);
        z = readFloat(buffer);
        tmpy = y;
        y = z;
        z = -tmpy;

        co.vertices.add(new Number3d(x, y, z));
    }
}

/**
 * Lee las caras del fichero 3ds
 * @param buffer Stream con los datos a leer
 * @throws IOException
 */
private void readFaces(InputStream buffer) throws IOException {
    int triangles = readShort(buffer);
    for (int i = 0; i < triangles; i++) {
        int[] vertexIDs = new int[3];
        vertexIDs[0] = readShort(buffer);
        vertexIDs[1] = readShort(buffer);
        vertexIDs[2] = readShort(buffer);
        readShort(buffer);
        ParseObjectFace face = new ParseObjectFace();
        face.v = vertexIDs;
        face.uv = vertexIDs;
        face.faceLength = 3;
        face.hasuv = true;
        co.numFaces++;
        co.faces.add(face);

        co.calculateFaceNormal(face);
    }
}

/**
 * Lee las texturas del fichero 3ds
 * @param buffer Stream con los datos a leer
 * @throws IOException
 */
private void readTexCoords(InputStream buffer) throws IOException {
    int numVertices = readShort(buffer);
```

```

    for (int i = 0; i < numVertices; i++) {
        Uv uv = new Uv();
        uv.u = readFloat(buffer);
        uv.v = readFloat(buffer) * -1f;
        co.texCoords.add(uv);
    }
}

@Override
public Object3dContainer getParsedObject()
{
    Log.d(Min3d.TAG, "Start object creation");
    Object3dContainer obj = new Object3dContainer(0, 0);
    int numObjects = parseObjects.size();
    Bitmap texture = null;

    if(textureAtlas.hasBitmaps())
    {
        textureAtlas.generate();
        texture = textureAtlas.getBitmap();
        Shared.textureManager().addTextureId(texture,
            textureAtlas.getId(), generateMipMap);
    }

    for (int i = 0; i < numObjects; i++) {
        ParseObjectData o = parseObjects.get(i);
        Log.d(Min3d.TAG, "Creating object " + o.name);
        obj.addChild(o.getParsedObject(materialMap,
            textureAtlas));
    }

    if(textureAtlas.hasBitmaps())
    {
        if(texture != null) texture.recycle();
    }
    Log.d(Min3d.TAG, "Object creation finished");

    super.cleanup();

    return obj;
}

/**
 * Lee una textura del fichero 3ds
 * @param stream Stream con los datos a leer
 * @throws IOException
 */
private void readTexture(InputStream stream) throws IOException
{
    String fileName = readString(stream);
    StringBuffer texture = new StringBuffer(packageID);
    texture.append(":drawable/");
}

```

```
StringBuffer textureName = new StringBuffer(fileName.
    toLowerCase());
int dotIndex = textureName.lastIndexOf(".");
if (dotIndex > -1)
    texture.append(textureName.substring(0, dotIndex));
else
    texture.append(textureName);

if (!remoteLocation)
    textureAtlas.addBitmapAsset(new BitmapAsset(
        currentMaterialKey, texture.toString()));

else
{
    String remoteTexture = getRemoteTexture(textureName
        .toString());

    BitmapAsset baAsset = new BitmapAsset(
        currentMaterialKey, texture.toString());
    baAsset.bitmap = BitmapFactory.decodeStream(
        new URL(
            remoteTexture
        ),
        null,
        openConnection
        ().
        getInputStream
        ());

    baAsset.useForAtlasDimensions = true;
    textureAtlas.addBitmapAssetRemote(baAsset);
}
}

/**
 * Método que devuelve la textura. Si en la configuración ya
 * dispone explícitamente de una, se devolverá esta,
 * en caso contrario, se devolverá la que tiene asociada el objeto
 * 3d
 * @param objectTexture El nombre de la textura contenida en el
 * fichero 3d
 * @return La textura remota
 */
private String getRemoteTexture(String objectTexture)
{
    if (!texture.equalsIgnoreCase(""))
        return texture;

    String remoteTexture = "";
    int slashIndex = url.lastIndexOf("/");
    if (slashIndex > -1)
        remoteTexture = url.substring(0, slashIndex+1);

    remoteTexture += objectTexture;
    return remoteTexture;
}
```

```

    }
}

```

Paquete parser C.40: Max3DSParser.java

### MD2Parser.java

```

package com.ivanlorenzo.parser;

import java.io.BufferedInputStream;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
//import java.util.ArrayList;

import min3d.Min3d;
import min3d.Shared;
import min3d.animation.AnimationObject3d;
import min3d.animation.KeyFrame;
import min3d.parser.LittleEndianDataInputStream;
import min3d.parser.ParseObjectFace;
import min3d.vos.Number3d;
import min3d.vos.Uv;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Log;

/**
 * Parser específico de los objetos MD2
 * @author Iván Lorenzo Rubio
 *
 */
public class MD2Parser extends AParser implements IParser {

    private MD2Header header;
    private String currentTextureName;
    private KeyFrame[] frames;

    /**
     * Constructor del parser MD2
     * @param resources
     * @param resourceID
     * @param generateMipMap
     */
    public MD2Parser(Resources resources, String resourceID, boolean
        generateMipMap) {
        super(resources, resourceID, generateMipMap);
    }

    /**

```

```
* Constructor del parser en el que se le pasa la dirección donde  
se encontrará el objeto a parsear  
* @param url  
*/  
public MD2Parser(String url) {  
    super(url);  
}  
  
@Override  
public AnimationObject3d getParsedAnimationObject() {  
    Log.d(Min3d.TAG, "Start object creation");  
    Bitmap texture = null;  
    AnimationObject3d animObj;  
  
    if (textureAtlas.hasBitmaps()) {  
        textureAtlas.generate();  
        texture = textureAtlas.getBitmap();  
        Shared.textureManager().addTextureId(texture,  
            textureAtlas.getId(), generateMipMap);  
    }  
  
    Log.d(Min3d.TAG, "Creating object " + co.name);  
    animObj = co.getParsedObject(textureAtlas, materialMap,  
        frames);  
  
    if (textureAtlas.hasBitmaps()) {  
        if (texture != null)  
            texture.recycle();  
    }  
    Log.d(Min3d.TAG, "Object creation finished");  
  
    super.cleanup();  
  
    return animObj;  
}  
  
@Override  
public void parse() {  
  
    BufferedInputStream stream = new BufferedInputStream(fileIn  
        );  
  
    co = new ParseObjectData();  
    header = new MD2Header();  
  
    Log.d(Min3d.TAG, "Start parsing MD2 file");  
    try {  
        header.parse(stream);  
        frames = new KeyFrame[header.numFrames];  
        byte[] bytes = new byte[header.offsetEnd - 68];  
        stream.read(bytes);  
        getMaterials(stream, bytes);  
        getTexCoords(stream, bytes);  
        getFrames(stream, bytes);  
        getTriangles(stream, bytes);  
    } catch (Exception e) {
```

```

        e.printStackTrace();
    }
}

private void getMaterials(BufferedInputStream stream, byte[] bytes)
    throws IOException {
    ByteArrayInputStream ba = new ByteArrayInputStream(bytes,
        header.offsetSkins - 68, bytes.length -
        header.offsetSkins);
    LittleEndianDataInputStream is = new
        LittleEndianDataInputStream(ba);

    for (int i = 0; i < header.numSkins; i++) {
        String skinPath = is.readString(64);
        StringBuffer texture = new StringBuffer(packageID);
        texture.append(":drawable/");

        skinPath = skinPath.substring(skinPath.lastIndexOf(
            "/" ) + 1,
            skinPath.length());
        StringBuffer textureName = new StringBuffer(
            skinPath.toLowerCase());
        int dotIndex = textureName.lastIndexOf(".");
        if (dotIndex > -1)
            texture.append(textureName.substring(0,
                dotIndex));
        else
            texture.append(textureName);

        currentTextureName = texture.toString();

        if (!remoteLocation)
            textureAtlas.addBitmapAsset(new BitmapAsset(
                currentTextureName,
                currentTextureName));

        else
        {
            String remoteTexture = "";
            int slashIndex = url.lastIndexOf("/");
            if (slashIndex > -1)
                remoteTexture = url.substring(0,
                    slashIndex+1);

            remoteTexture+= textureName;

            BitmapAsset baAsset = new BitmapAsset(
                currentTextureName, currentTextureName);
            baAsset.bitmap = BitmapFactory.decodeStream(
                (

```

new

URL

(

remoteTexture

```
        )
        .openConnection()
        .getInputStream()
        );

        baAsset.useForAtlasDimensions = true;
        textureAtlas.addBitmapAssetRemote(baAsset);
    }
}

private void getTexCoords(BufferedInputStream stream, byte[] bytes)
    throws IOException {
    ByteArrayInputStream ba = new ByteArrayInputStream(bytes,
        header.offsetTexCoord - 68, bytes.length
        - header.offsetTexCoord);
    LittleEndianDataInputStream is = new
        LittleEndianDataInputStream(ba);

    for (int i = 0; i < header.numTexCoord; i++) {
        co.texCoords.add(new Uv((float)is.readShort() / (
            float)header.skinWidth, (float)is.readShort() /
            (float)header.skinHeight));
    }
}

private void getFrames(BufferedInputStream stream, byte[] bytes)
    throws IOException {
    ByteArrayInputStream ba = new ByteArrayInputStream(bytes,
        header.offsetFrames - 68, bytes.length -
        header.offsetFrames);
    LittleEndianDataInputStream is = new
        LittleEndianDataInputStream(ba);
    //ArrayList<Number3d> firstFrameVerts = new ArrayList<
        Number3d>();

    for (int i = 0; i < header.numFrames; i++) {
        float scaleX = is.readFloat();
        float scaleY = is.readFloat();
        float scaleZ = is.readFloat();
        float translateX = is.readFloat();
        float translateY = is.readFloat();
        float translateZ = is.readFloat();
        String name = is.readString(16);

        if (name.indexOf("_") > 0)
            name = name.subSequence(0, name.lastIndexOf(
                "_")).toString();
        else
```

```

        name = name.substring(0, 6).replaceAll("[0-9]{1,2}$", "");

        Log.d(Min3d.TAG, "frame name: " + name);
        float vertices[] = new float[header.numVerts * 3];
        int index = 0;

        for (int j = 0; j < header.numVerts; j++) {
            vertices[index++] = scaleX * is.
                readUnsignedByte() + translateX;
            vertices[index++] = scaleY * is.
                readUnsignedByte() + translateY;
            vertices[index++] = scaleZ * is.
                readUnsignedByte() + translateZ;

            //int normalIndex = is.readUnsignedByte();
            if (i == 0)
                co.vertices.add(new Number3d(
                    vertices[index - 3],
                    vertices[index - 2], vertices[
                        index - 1]));
        }

        frames[i] = new KeyFrame(name, vertices);
    }
}

private void getTriangles(BufferedInputStream stream, byte[] bytes)
    throws IOException {
    ByteArrayInputStream ba = new ByteArrayInputStream(bytes,
        header.offsetTriangles - 68, bytes.length
            - header.offsetTriangles);
    LittleEndianDataInputStream is = new
        LittleEndianDataInputStream(ba);
    int[] indices = new int[header.numTriangles*3];
    int index = 0;

    for (int i = 0; i < header.numTriangles; i++) {
        int[] vertexIDs = new int[3];
        int[] uvIDS = new int[3];

        indices[index+2] = vertexIDs[2] = is.
            readUnsignedShort();
        indices[index+1] = vertexIDs[1] = is.
            readUnsignedShort();
        indices[index] = vertexIDs[0] = is.
            readUnsignedShort();
        index += 3;
        uvIDS[2] = is.readUnsignedShort();
        uvIDS[1] = is.readUnsignedShort();
        uvIDS[0] = is.readUnsignedShort();

        ParseObjectFace f = new ParseObjectFace();
        f.v = vertexIDs;
        f.uv = uvIDS;
    }
}

```

```
        f.hasn = f.hasuv = true;
        f.faceLength = 3;
        f.materialKey = currentTextureName;
        co.numFaces++;
        co.faces.add(f);
        co.calculateFaceNormal(f);
    }

    for(int j=0; j<header.numFrames; j++)
    {
        frames[j].setIndices(indices);
    }
}

/**
 * Clase que representa la cabecera de los ficheros MD2
 * @author Iván Lorenzo Rubio
 */
private class MD2Header {
    public int id;
    public int version;
    public int skinWidth;
    public int skinHeight;
    @SuppressWarnings("unused")
    public int frameSize;
    public int numSkins;
    public int numVerts;
    public int numTexCoord;
    public int numTriangles;
    @SuppressWarnings("unused")
    public int numGLCommands;
    public int numFrames;
    public int offsetSkins;
    public int offsetTexCoord;
    public int offsetTriangles;
    public int offsetFrames;
    @SuppressWarnings("unused")
    public int offsetGLCommands;
    public int offsetEnd;

    public void parse(InputStream stream) throws Exception {
        id = readInt(stream);
        version = readInt(stream);

        if (id != 844121161 || version != 8)
            throw new Exception("This is not a valid
                                MD2 file.");

        skinWidth = readInt(stream);
        skinHeight = readInt(stream);
        frameSize = readInt(stream);

        numSkins = readInt(stream);
        numVerts = readInt(stream);
        numTexCoord = readInt(stream);
    }
}
```

```

        numTriangles = readInt(stream);
        numGLCommands = readInt(stream);
        numFrames = readInt(stream);

        offsetSkins = readInt(stream);
        offsetTexCoord = readInt(stream);
        offsetTriangles = readInt(stream);
        offsetFrames = readInt(stream);
        offsetGLCommands = readInt(stream);
        offsetEnd = readInt(stream);
    }
}

```

Paquete parser C.41: MD2Parser.java

### ObjParser.java

```

package com.ivanlorenzo.parser;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Calendar;
import java.util.StringTokenizer;

import min3d.Min3d;
import min3d.Shared;
//import min3d.Utills;
import min3d.core.Object3dContainer;
import min3d.parser.ParseObjectFace;
import min3d.vos.Color4;
import min3d.vos.Number3d;
import min3d.vos.Uv;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.util.Log;

/**
 * Parser específico de los objetos obj
 * @author Iván Lorenzo Rubio
 */
public class ObjParser extends AParser implements IParser {

    private final String VERTEX = "v";
    private final String FACE = "f";
    private final String TEXCOORD = "vt";
    private final String NORMAL = "vn";
    private final String OBJECT = "o";
    private final String MATERIAL_LIB = "mtllib";
    private final String USE_MATERIAL = "usemtl";
    private final String NEW_MATERIAL = "newmtl";
    private final String DIFFUSE_COLOR = "Kd";

```

```
private final String DIFFUSE_TEX_MAP = "map_Kd";

/**
 * Crea una nueva instancia de un parser OBJ
 *
 * @param resources Recursos de la aplicación
 * @param resourceID Identificador del recurso
 * @param generateMipMap
 */
public ObjParser(Resources resources, String resourceID, boolean
    generateMipMap) {
    super(resources, resourceID, generateMipMap);
}

/**
 * Crea una nueva instancia de un parser OBJ
 * @param url Dirección en la que se encuentra el objeto a parsear
 */
public ObjParser(String url)
{
    super(url);
}

@Override
public void parse() {
    long startTime = Calendar.getInstance().getTimeInMillis();

    BufferedReader buffer = new BufferedReader(
        new InputStreamReader(fileIn));

    String line;
    co = new ParseObjectData(vertices, texCoords, normals);
    parseObjects.add(co);

    Log.d(Min3d.TAG, "Start parsing object " + resourceID);
    Log.d(Min3d.TAG, "Start time " + startTime);

    try {
        while ((line = buffer.readLine()) != null) {
            // remove duplicate whitespace
            // line = line.replaceAll("\\s+", " ");
            // String[] parts = line.split(" ");
            StringTokenizer parts = new StringTokenizer
                (line, " ");
            int numTokens = parts.countTokens();
            if (numTokens == 0)
                continue;
            String type = parts.nextToken();

            if (type.equals(VERTEX)) {
                Number3d vertex = new Number3d();
                vertex.x = Float.parseFloat(parts.
                    nextToken());
                vertex.y = Float.parseFloat(parts.
                    nextToken());
            }
        }
    }
}
```

```

        vertex.z = Float.parseFloat(parts.
            nextToken());
        vertices.add(vertex);
    } else if (type.equals(FACE)) {
        if (numTokens == 4) {
            co.numFaces++;
            co.faces.add(new ObjFace(
                line, currentMaterialKey
                , 3));
        } else if (numTokens == 5) {
            co.numFaces += 2;
            co.faces.add(new ObjFace(
                line, currentMaterialKey
                , 4));
        }
    } else if (type.equals(TEXCOORD)) {
        Uv texCoord = new Uv();
        texCoord.u = Float.parseFloat(parts
            .nextToken());
        texCoord.v = Float.parseFloat(parts
            .nextToken()) * -1f;
        texCoords.add(texCoord);
    } else if (type.equals(NORMAL)) {
        Number3d normal = new Number3d();
        normal.x = Float.parseFloat(parts.
            nextToken());
        normal.y = Float.parseFloat(parts.
            nextToken());
        normal.z = Float.parseFloat(parts.
            nextToken());
        normals.add(normal);
    } else if (type.equals(MATERIAL_LIB)) {
        readMaterialLib(parts.nextToken());
    } else if (type.equals(USE_MATERIAL)) {
        currentMaterialKey = parts.
            nextToken();
    } else if (type.equals(OBJECT)) {
        String objName = parts.
            hasMoreTokens() ? parts.
            nextToken() : "";
        if (firstObject)
        {
            Log.d(Min3d.TAG, "Create
                object " + objName);
            co.name = objName;
            firstObject = false;
        }
        else
        {
            Log.d(Min3d.TAG, "Create
                object " + objName);
            co = new ParseObjectData(
                vertices, texCoords,
                normals);
            co.name = objName;
            parseObjects.add(co);
        }
    }

```

```
        }
    }
} catch (IOException e) {
    e.printStackTrace();
}

long endTime = Calendar.getInstance().getTimeInMillis();
Log.d(Min3d.TAG, "End time " + (endTime - startTime));
}

@Override
public Object3dContainer getParsedObject() {
    Log.d(Min3d.TAG, "Start object creation");
    Object3dContainer obj = new Object3dContainer(0, 0);
    int numObjects = parseObjects.size();
    Bitmap texture = null;

    if(textureAtlas.hasBitmaps())
    {
        textureAtlas.generate();
        texture = textureAtlas.getBitmap();
        Shared.textureManager().addTextureId(texture,
            textureAtlas.getId(), generateMipMap);
    }

    for (int i = 0; i < numObjects; i++) {
        ParseObjectData o = parseObjects.get(i);
        Log.d(Min3d.TAG, "Creating object " + o.name);
        obj.addChild(o.getParsedObject(materialMap,
            textureAtlas));
    }

    if(textureAtlas.hasBitmaps())
    {
        if(texture != null) texture.recycle();
    }
    Log.d(Min3d.TAG, "Object creation finished");

    cleanup();

    return obj;
}

/**
 * Lee el material
 * @param libID Identificador del material
 */
private void readMaterialLib(String libID) {
    StringBuffer resourceID = new StringBuffer(packageID);
    StringBuffer libIDSbuf = new StringBuffer(libID);
    int dotIndex = libIDSbuf.lastIndexOf(".");
    if (dotIndex > -1)
        libIDSbuf = libIDSbuf.replace(dotIndex, dotIndex +
            1, "_");
}
```

```

resourceID.append(":raw/");
resourceID.append(libIDSbuf.toString());

InputStream fileIn = resources.openRawResource(resources.
    getIdentifier(
        resourceID.toString(), null, null));
BufferedReader buffer = new BufferedReader(
    new InputStreamReader(fileIn));

String line;
String currentMaterial = "";

try {
    while ((line = buffer.readLine()) != null) {
        String[] parts = line.split(" ");
        if (parts.length == 0)
            continue;
        String type = parts[0];

        if (type.equals(NEW_MATERIAL)) {
            if (parts.length > 1) {
                currentMaterial = parts[1];
                materialMap.put(
                    currentMaterial, new
                        Material(
                            currentMaterial
                                ));
            }
        } else if (type.equals(DIFFUSE_COLOR) && !
            type.equals(DIFFUSE_TEX_MAP)) {
            Color4 diffuseColor = new Color4(
                Float.parseFloat(parts[1]) *
                255.0f, Float.parseFloat(parts
                    [2]) * 255.0f, Float.parseFloat(
                    parts[3]) * 255.0f, 255.0f);
            materialMap.get(currentMaterial).
                diffuseColor = diffuseColor;
        } else if (type.equals(DIFFUSE_TEX_MAP)) {
            if (parts.length > 1) {
                materialMap.get(
                    currentMaterial).
                    diffuseTextureMap =
                        parts[1];
                StringBuffer texture = new
                    StringBuffer(packageID);
                texture.append(":drawable/"
                    );

                StringBuffer textureName =
                    new StringBuffer(parts
                        [1]);
                dotIndex = textureName.
                    lastIndexOf(".");
                if (dotIndex > -1)
                    texture.append(
                        textureName.

```

```
                substring(0,
                        dotIndex));
        else
            texture.append(
                textureName);

        //int bmResourceID =
            resources.getIdentifier(
                texture
                .toString()
                , null, null);
        //Bitmap b = Utils.
            makeBitmapFromResourceId
                (bmResourceID);
        textureAtlas.addBitmapAsset
            (new BitmapAsset(
                currentMaterial, texture
                .toString()));
    }
}

} catch (IOException e) {
    e.printStackTrace();
}

}

@Override
protected void cleanup() {
    super.cleanup();
    materialMap.clear();
}

/**
 * Clase privada que representa una cara del objeto
 * @author Iván Lorenzo Rubio
 */
private class ObjFace extends ParseObjectFace
{
    public ObjFace(String line, String materialKey, int
        faceLength) {
        super();
        this.materialKey = materialKey;
        this.faceLength = faceLength;
        boolean emptyVt = line.indexOf("//") > -1;
        if(emptyVt) line = line.replace("//", "/");
        StringTokenizer parts = new StringTokenizer(line);
        parts.nextToken();
        StringTokenizer subParts = new StringTokenizer(
            parts.nextToken(), "/");
        int partLength = subParts.countTokens();
        hasuv = partLength >= 2 && !emptyVt;
        hasn = partLength == 3 || (partLength == 2 &&
            emptyVt);
    }
}
```

```

        v = new int[faceLength];
        if (hasuv)
            uv = new int[faceLength];
        if (hasn)
            n = new int[faceLength];

        for (int i = 1; i < faceLength + 1; i++) {
            if (i > 1)
                subParts = new StringTokenizer(
                    parts.nextToken(), "/");

            int index = i - 1;
            v[index] = (short) (Short.parseShort(
                subParts.nextToken()) - 1);
            if (hasuv)
                uv[index] = (short) (Short.
                    parseShort(subParts.nextToken())
                    - 1);
            if (hasn)
                n[index] = (short) (Short.
                    parseShort(subParts.nextToken())
                    - 1);
        }
    }
}

```

Paquete parser C.42: ObjParser.java

### C.4.8. Paquete com.ivanlorenzo.connection

#### ConnectionStatus.java

```

package com.ivanlorenzo.connection;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;

/**
 * Clase que utilizaremos para comprobar el estado de la conexión a
 * Internet
 * @author Iván Lorenzo Rubio
 *
 */
public class ConnectionStatus
{
    /**
     * Comprueba si tenemos o no conexión a Internet
     * @param context El contexto de la aplicación
     * @return Devuelve verdadero en caso de tener conexión y falso en
     * caso contrario
     */
}

```

```
    */
    public static boolean checkInternetConnection(Context context)
    {
        ConnectivityManager conMgr = (ConnectivityManager)
            context.getSystemService(Context.
                CONNECTIVITY_SERVICE);

        if ( conMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE)
            .getState() == NetworkInfo.State.DISCONNECTED
            && conMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI) .
            getState() == NetworkInfo.State.DISCONNECTED )
            return false;

        return true;
    }
}
```

Paquete connection C.43: ConnectionStatus.java

### C.4.9. Paquete com.ivanlorenzo.gui

#### TFGARActivity.java

```
package com.ivanlorenzo.gui;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;

import javax.xml.xpath.XPathExpressionException;

import com.ivanlorenzo.gui.R;

import com.android.camera.CameraHardwareException;
import com.android.camera.CameraHolder;
import com.ivanlorenzo.connection.ConnectionStatus;
import com.ivanlorenzo.data.CompilerGenerator;
import com.ivanlorenzo.data.KnowledgeBase;
import com.ivanlorenzo.data.exceptions.LoadXMLException;
import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;
import com.ivanlorenzo.gui.dialogs.DialogType;
import com.ivanlorenzo.gui.preferences.TFGPreferencesActivity;
import com.ivanlorenzo.location.ARLocationListener;
import com.ivanlorenzo.location.LocationData;
import com.ivanlorenzo.location.observer.IARObserver;

import min3d.Shared;
import min3d.core.Object3d;
import min3d.core.Renderer;
import min3d.core.Scene;
import min3d.vos.Light;
```

```

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
import android.content.res.Resources;
import android.content.res.Resources.NotFoundException;
import android.graphics.PixelFormat;
import android.hardware.Camera;
import android.location.Criteria;
import android.location.LocationListener;
import android.location.LocationManager;
import android.media.MediaPlayer;
import android.opengl.GLSurfaceView;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.OrientationEventListener;
import android.view.ScaleGestureDetector;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
//import android.view.ViewGroup.LayoutParams;
import android.widget.FrameLayout;

/**
 * Clase Activity principal de la aplicación. Será la encargada de manejar
 * la recepción de datos de la cámara.
 * Además, recibirá los eventos táctiles producidos en la pantalla y
 * gestionará la escena en la que se encontrarán
 * todos los objetos.
 * @author Iván Lorenzo Rubio
 *
 */
public class TFGARActivity extends Activity implements View.OnClickListener
    , SurfaceHolder.Callback,
    min3d.interfaces.ISceneController , IARObserver<
    LocationData>
{
    /**
     * Array con los mensajes de la aplicación
     */
    private static MessageWhat[] messageWhat = MessageWhat.values();

```

```
    /**
     * Tiempo de delay de la pantalla
     */
    private static final int SCREEN_DELAY = 2 * 60 * 1000;

    /**
     * Parámetros de la cámara
     */
    private android.hardware.Camera.Parameters mParameters;

    /**
     * Manejador de eventos de la orientación
     */
    private OrientationEventListener mOrientationListener;
    /**
     * Última orientación del dispositivo
     */
    private int mLastOrientation = 0;

    /**
     * Preferencias compartidas de la aplicación
     */
    private SharedPreferences mPreferences;

    /**
     * Dispositivo de cámara
     */
    private android.hardware.Camera mCameraDevice;

    private SurfaceView mSurfaceView;
    private SurfaceHolder mSurfaceHolder = null;
    private boolean mStartPreviewFail = false;
    private GLSurfaceView mGLSurfaceView = null;

    /**
     * Renderizador de min3d
     */
    private Renderer mRenderer;
    /**
     * Nos indica si estamos capturando imágenes previas
     */
    private boolean mPreviewing;
    /**
     * Nos indica si está pausado
     */
    private boolean mPausing;
    /**
     * Nos indica si ya está inicializada la aplicación
     */
    private boolean mFirstTimeInitialized;
    /**
     * Manejador principal
     */
    private Handler mHandler = new MainHandler();
    /**
     * Llamada a la vista previa
```

```

    */
    private PreviewCallback mPreviewCallback = new PreviewCallback();
    /**
     * Instancia encargada de dibujar los objetos
     */
    private ARToolkitDrawer arToolkitDrawer = null;
    /**
     * MediaPlayer
     */
    private MediaPlayer mMediaPlayer = null;
    /**
     * Estado de la aplicación
     */
    private ARStatus arStatus = ARStatus.AR_Marker;

    /**
     * Se llama cuando se crea la actividad
     */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        mPreferences = Util.getSharedPreferences(this);

        // Renderer of min3d
        _initSceneHandler = new Handler();
        _updateSceneHandler = new Handler();

        Shared.context(this);
        scene = new Scene(this);
        scene.backgroundTransparent(true);
        mRenderer = new Renderer(scene);
        Shared.renderer(mRenderer);

        requestWindowFeature(Window.FEATURE_PROGRESS);

        Window win = getWindow();
        win.addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
        win.addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.main);
        mSurfaceView = (SurfaceView) findViewById(R.id.camera_preview);

        mSurfaceView.setKeepScreenOn(true);

        SurfaceHolder holder = mSurfaceView.getHolder();
        holder.addCallback(this);
        holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);

        //Comprobar la conexión
        if(!ConnectionStatus.checkInternetConnection(this))

```

```
        showDialog(DialogType.DIALOG_ERROR_CONNECTION.ordinal());

        //Iniciamos la recepción de nuestra posición
        this.startGPSLocation();

        mScaleDetector = new ScaleGestureDetector(this, new ScaleListener()
        );

    }

    @Override
    public void onStart()
    {
        super.onStart();
    }

    @Override
    public void onResume()
    {
        super.onResume();
        mPausing = false;

        // Start the preview if it is not started.
        if (!mPreviewing && !mStartPreviewFail && (mSurfaceHolder != null))
        {
            try {
                startPreview();
            } catch (Exception e) {
                showCameraErrorAndFinish();
                return;
            }
        }

        if (mSurfaceHolder != null) {
            // If first time initialization is not finished, put it in the
            // message queue.
            if (!mFirstTimeInitialized)
            {
                mHandler.sendMessage(MessageWhat.FIRST_TIME_INIT.
                    ordinal());
            } else {
                initializeSecondTime();
            }
        }
        keepScreenOnAwhile();

        createCompilerXML();
    }

    @Override
    protected void onPause()
```

```

{
    mPausing = true;
    stopPreview();
    // Close the camera now because other activities may need to use it
    .
    closeCamera();
    resetScreenOn();
    changeGLSurfaceViewState();

    if (mFirstTimeInitialized) {
        mOrientationListener.disable();
    }

    // Remove the messages in the event queue.
    mHandler.removeMessages(MessageWhat.RESTART_PREVIEW.ordinal());
    mHandler.removeMessages(MessageWhat.FIRST_TIME_INIT.ordinal());

    super.onPause();
}

@Override
public void onStop()
{
    super.onStop();
}

@Override
protected void onDestroy()
{
    super.onDestroy();
    finish();
    System.exit(0);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
    Intent data)
{
    switch (messageWhat[requestCode])
    {
        case CROP_MSG:
        {
            Intent intent = new Intent();
            if (data != null) {
                Bundle extras = data.getExtras();
                if (extras != null) {
                    intent.putExtras(extras);
                }
            }
            setResult(resultCode, intent);
            finish();
            break;
        }
    }
}

```

```
    }  
}  
  
@Override  
public boolean onCreateOptionsMenu(Menu menu)  
{  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu, menu);  
    return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item)  
{  
    switch (item.getItemId()) {  
  
        case R.id.configuration:  
  
            Intent intent = new Intent(this,  
                TFGPreferencesActivity.class);  
            startActivity(intent);  
            break;  
  
        case R.id.exit:  
            showDialog(DialogType.DIALOG_EXIT.ordinal());  
            break;  
    }  
    return true;  
}  
  
@Override  
protected Dialog onCreateDialog(int id)  
{  
    final DialogType[] dialogTypes = DialogType.values();  
    switch (dialogTypes[id]) {  
        case DIALOG_LOADING: {  
            ProgressDialog dialog = new ProgressDialog(this);  
            dialog.setMessage(getString(R.string.loading));  
            dialog.setCancelable(false);  
            dialog.getWindow().setFlags  
                (WindowManager.LayoutParams.  
                    FLAG_BLUR_BEHIND,  
                    WindowManager.LayoutParams.  
                        FLAG_BLUR_BEHIND);  
            return dialog;  
        }  
        case DIALOG_ERROR_CONNECTION: {  
            setRequestedOrientation(ActivityInfo.  
                SCREEN_ORIENTATION_PORTRAIT);  
            return new AlertDialog.Builder(this)  
                .setMessage(getString(R.string.  
                    internet_connection))
```

```

        .setTitle(getString(R.string.
            connection_error))
        .setCancelable(false)
        .setNeutralButton(android.R.string.ok,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface
                    dialog, int whichButton)
                {
                    onDestroy();
                }
            })
        .show();
    }
    case DIALOG_EXIT:{
        return new AlertDialog.Builder(this)
            .setMessage(getString(R.string.sure_exit))
            .setCancelable(false)
            .setPositiveButton(getString(R.string.yes),
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
                        id) {
                        onDestroy();
                    }
                })
            .setNegativeButton(getString(R.string.no),
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
                        dialog, int id) {
                        dialog.cancel();
                    }
                })
            .show();
    }

    // case ALERT_DIALOG_LOADED_DATA:{
    // return new AlertDialog.Builder(this)
    // .setMessage("Cargados los datos")
    // .setCancelable(false)
    // .setNegativeButton("OK", new
    // DialogInterface.OnClickListener() {
    // public void onClick(DialogInterface dialog, int
    // id) {
    // dialog.cancel();
    // }
    // })
    // .show();
    // }
    // //

    default :
        return super.onCreateDialog(id);
    }
}

```

```
@Override
public void onBackPressed()
{
    showDialog(DialogType.DIALOG_EXIT.ordinal());
}

/**
 * Empezaremos a tomar capturas de la cámara. Debería ser llamado una
 * sola vez.
 * Esto podría haberse hecho en el método onCreate(), pero queremos
 * así logramos que
 * aparezca la pantalla de la cámara tan pronto como sea posible.
 */
private void initializeFirstTime()
{
    if (mFirstTimeInitialized) return;

    //Creamos el listener de orientación. Esto se debe hacer en primer
    //lugar porque
    //se necesita cierto tiempo para conseguir la primera orientación.
    mOrientationListener =
        new OrientationEventListener(this) {
            @Override
            public void onOrientationChanged(int orientation) {
                //Mantenemos la última orientación conocida
                if (orientation != ORIENTATION_UNKNOWN) {
                    orientation += 90;
                }
                orientation = Util.roundOrientation(orientation);
                if (orientation != mLastOrientation) {
                    mLastOrientation = orientation;
                }
            }
        };
    mOrientationListener.enable();
    mFirstTimeInitialized = true;
    changeGLSurfaceViewState();
}

/**
 * Si la actividad es pausada y luego se continúa, se llamará a este
 * método en onResume
 */
private void initializeSecondTime()
{
    //Iniciamos el orientationListener si es posible
    mOrientationListener.enable();
}
```

```

        changeGLSurfaceViewState();
    }

    /**
     * Clase manejadora que se utiliza para enviar mensajes al hilo
     * principal de la aplicación
     * @author Iván Lorenzo Rubio
     *
     */
    private class MainHandler extends Handler
    {
        @Override
        public void handleMessage(Message msg)
        {
            switch (messageWhat[msg.what])
            {
                case RESTART_PREVIEW:
                    {
                        restartPreview();
                        break;
                    }
                case CLEAR_SCREEN_DELAY:
                    {
                        getWindow().clearFlags(WindowManager.LayoutParams.
                            LayoutParams.

                case FIRST_TIME_INIT:
                    {
                        initializeFirstTime
                        ();
                        break;
                    }
                case SHOW_LOADING:
                    {
                        showDialog(
                            DialogType.DIALOG_LOADING.ordinal());
                        break;
                    }
                case HIDE_LOADING:
                    {
                        try {
                            dismissDialog(DialogType.
                                DIALOG_LOADING.ordinal()
                            );
                            removeDialog(DialogType.
                                DIALOG_LOADING.ordinal()
                            );
                        } catch (IllegalArgumentException e
                            ) {}

                        break;
                    }
            }
        }
    }

```

```

    }
}

/**
 * Inicia el servicio de localización utilizando el mejor proveedor
 * , bien sea GPS o a través de la conexión a Internet.
 * La búsqueda se realiza cada 120.000 ms = 2 minutos de tiempo.
 */
private void startGPSLocation()
{
    String bestProvider;
    Criteria criteria = new Criteria();

    /* Usamos la clase LocationManager para obtener localizaciones GPS
    */
    LocationManager mlocManager = (LocationManager) getSystemService(
        Context.LOCATION_SERVICE);
    bestProvider = mlocManager.getBestProvider(criteria, false);

    LocationListener mlocListener = new ARLocationListener(this, 10);
    mlocManager.requestLocationUpdates( bestProvider, 120000, 0,
        mlocListener);
}

/**
 * Crea el compilador XML pasándole las URLs de los archivos de
 * configuración y se muestra
 * un diálogo informado de que ha terminado la carga
 */
private void createCompilerXML()
{
    try
    {
        CompilerGenerator.BASE.clear();
        CompilerGenerator.compile(
            getResources(),
            mPreferences.getString("
                URL_PATTERNS", "http://tfg.
                ivanlorenzo.es/data/patterns.dtd
            "), "",
            mPreferences.getString("URL_OBJECTS
                ", "http://tfg.ivanlorenzo.es/
                data/objects.dtd"), "");
    }
    catch (LoadXMLException lxMLE)
    {
        lxMLE.printStackTrace();
    }
}

```

```

        catch (XPathExpressionException xpee)          { xpee.
            printStackTrace();}
        catch (NotFoundException e)
            { e.printStackTrace();}
        catch (NoSuchExtensionObject3DException e)      { e.
            printStackTrace();}
        catch (IOException e)
            { showDialog(DialogType.
                DIALOG_ERROR_CONNECTION.ordinal()); }
    }

    /**
     * Clase utilizada para enviar y recibir copias de los frames
     * mostrados
     * @author Iván Lorenzo Rubio
     */
    private final class PreviewCallback implements android.hardware.Camera.
        PreviewCallback
    {
        @Override
        public void onPreviewFrame(byte [] data, Camera camera)
        {
            if (mPausing)
                return;

            if (data != null)
            {
                if (arToolkitDrawer != null && arStatus ==
                    ARStatus.AR_Marker)
                    arToolkitDrawer.draw(data);

                } else {
                    try {
                        // The measure against over load.
                        Thread.sleep(500);
                    } catch (InterruptedException e) {}
                }
            restartPreview();
        }
    }

    /**
     * Cambia la vista
     */
    private void changeGLSurfaceViewState()
    {
        Configuration config = getResources().getConfiguration();
        if (config.orientation == Configuration.ORIENTATION_LANDSCAPE
            && !mPausing && mFirstTimeInitialized) {

```

```

        if (mGLSurfaceView == null) initializeGLSurfaceView();
    } else if (mGLSurfaceView != null) {
        finalizeGLSurfaceView();
    }
}

/**
 * Inicializa la vista
 */
private void initializeGLSurfaceView()
{
    // init ARToolkit.
    if (arToolkitDrawer == null) {
        InputStream camePara = getResources().
            openRawResource(R.raw.camera_para);

        //Obtenemos los patrones
        ArrayList<InputStream> patt = KnowledgeBase.
            getInstance().getPatterns(getResources());

        int[] width = new int[patt.size()];
        for (int i = 0; i < patt.size(); i++)
        {
            width[i] = 80;
        }

        arToolkitDrawer = new ARToolkitDrawer(camePara,
            width, patt, mRenderer);

        //Para poder crear audios
        arToolkitDrawer.setContext(this);
    }

    FrameLayout frame = (FrameLayout) findViewById(R.id.frame);
    mGLSurfaceView = new GLSurfaceView(this);
    mGLSurfaceView.setEGLConfigChooser(8, 8, 8, 8, 16, 0);
    mGLSurfaceView.getHolder().setFormat(PixelFormat.
        TRANSLUCENT);
    mGLSurfaceView.setZOrderOnTop(true);
    mGLSurfaceView.setRenderer(mRenderer);
    frame.addView(mGLSurfaceView);
}

/**
 * Finaliza la vista
 */
private void finalizeGLSurfaceView()
{
    FrameLayout frame = (FrameLayout) findViewById(R.id.frame);
    frame.removeView(mGLSurfaceView);
    mGLSurfaceView = null;

    if (mMediaPlayer != null)

```

```

        mMediaPlayer.release();
        mMediaPlayer = null;

        arToolkitDrawer = null;
    }

    @Override
    public void surfaceChanged(SurfaceHolder holder, int format, int w,
        int h)
    {
        if (holder.getSurface() == null)
            return;

        //Guardamos el holder para un recuperación posterior
        mSurfaceHolder = holder;

        if (mCameraDevice == null)
        {
            /*
             * Para reducir el tiempo, iniciamos la vista en otro hilo
             */
            Thread startPreviewThread = new Thread(new Runnable()
            {
                public void run() {
                    try {
                        mStartPreviewFail = false;
                        startPreview();
                    } catch (Exception e) {
                        if ("eng".equals(Build.TYPE)) {
                            throw new RuntimeException(
                                e);
                        }
                        mStartPreviewFail = true;
                    }
                }
            });
            startPreviewThread.start();

            // Nos aseguramos que está iniciado
            try {
                startPreviewThread.join();
                if (mStartPreviewFail) {
                    showCameraErrorAndFinish();
                    return;
                }
            } catch (InterruptedException ex) {}
        }

        if (mPausing || isFinishing()) return;

        //Si la primera inicialización no terminó, enviar un
        //mensaje para hacerlo después
        if (!mFirstTimeInitialized) {

```

```
        mHandler.sendMessage(MessageWhat.FIRST_TIME_INIT.ordinal());
    } else {
        initializeSecondTime();
    }
}

@Override
public void surfaceCreated(SurfaceHolder holder) {

    @Override
    public void surfaceDestroyed(SurfaceHolder holder) {
        stopPreview();
        mSurfaceHolder = null;
    }

    /**
     * Apagamos la cámara
     */
    private void closeCamera() {
        if (mCameraDevice != null) {
            CameraHolder.instance().release();
            mCameraDevice = null;
            mPreviewing = false;
        }
    }

    /**
     * Nos aseguramos de que la cámara está disponible
     * @throws CameraHardwareException En caso de que el hardware de la
     * cámara nos dé algún error
     */
    private void ensureCameraDevice() throws CameraHardwareException {
        if (mCameraDevice == null) {
            mCameraDevice = CameraHolder.instance().open();
        }
    }

    /**
     * Mostrar el error de la cámara y finalizar
     */
    private void showCameraErrorAndFinish()
    {
        Resources ress = getResources();
        com.android.camera.Util.showFatalErrorAndFinish(TFGARActivity.this,
            ress.getString(R.string.camera_error_title),
            ress.getString(R.string.cannot_connect_camera));
    }

    /**
     * Se restaura la preview
     */
    public void restartPreview()
    {
```

```

    try {
        startPreview();
    } catch (CameraHardwareException e) {
        showCameraErrorAndFinish();
        return;
    }
}

/**
 * Establecemos la vista
 * @param holder Holder a establecer
 */
private void setPreviewDisplay(SurfaceHolder holder) {
    try {
        mCameraDevice.setPreviewDisplay(holder);
    } catch (Throwable ex) {
        closeCamera();
        throw new RuntimeException("setPreviewDisplay
            failed", ex);
    }
}

/**
 * Iniciamos la vista
 * @throws CameraHardwareException En caso de encontrarse algún
 *     problema con la cámara del dispositivo
 */
private void startPreview() throws CameraHardwareException
{
    if (mPausing || isFinishing()) return;

    ensureCameraDevice();
    setPreviewDisplay(mSurfaceHolder);
    if (!mPreviewing)
        setCameraParameters();

    mCameraDevice.setOneShotPreviewCallback(mPreviewCallback);

    try {
        mCameraDevice.startPreview();
    } catch (Throwable ex) {
        closeCamera();
        throw new RuntimeException("startPreview failed",
            ex);
    }
    mPreviewing = true;
}

/**
 * Paramos la vista
 */
private void stopPreview()
{
    if (mCameraDevice != null && mPreviewing)
    {
        mCameraDevice.setOneShotPreviewCallback(null);
    }
}

```

```
        mCameraDevice.stopPreview();
    }
    mPreviewing = false;
}

/**
 * Se establecen los parámetros de la cámara
 */
private void setCameraParameters()
{
    mParameters = mCameraDevice.getParameters();
    mParameters.setPreviewSize(320, 240);

    mCameraDevice.setParameters(mParameters);
}

/**
 * Reseteamos la pantalla
 */
private void resetScreenOn()
{
    mHandler.removeMessages(MessageWhat.CLEAR_SCREEN_DELAY.ordinal());
    getWindow().clearFlags(WindowManager.LayoutParams.
        FLAG_KEEP_SCREEN_ON);
}

/**
 * Mantenemos la pantalla en espera
 */
private void keepScreenOnAwhile()
{
    mHandler.removeMessages(MessageWhat.CLEAR_SCREEN_DELAY.ordinal());
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    mHandler.sendMessageDelayed(MessageWhat.CLEAR_SCREEN_DELAY.
        ordinal(), SCREEN_DELAY);
}

@Override
public void onClick(View v) {    ;;    }

//----- Movimientos -----

private static final int INVALID_POINTER_ID = -1;
```

```

private float mLastTouchX;
private float mLastTouchY;

private int mActivePointerId = INVALID_POINTER_ID;

private ScaleGestureDetector mScaleDetector;
private float mScaleFactor = 1.f;

@Override
public boolean onTouchEvent(MotionEvent event)
{
    // Let the ScaleGestureDetector inspect all events.
    mScaleDetector.onTouchEvent(event);

    switch (event.getAction())
    {
        case MotionEvent.ACTION_DOWN:
        {
            final float x = event.getX();
            final float y = event.getY();

            mLastTouchX = x;
            mLastTouchY = y;
            mActivePointerId = event.getPointerId(0);
            break;
        }

        case MotionEvent.ACTION_MOVE:
        {
            final int pointerIndex = event.
                findPointerIndex(mActivePointerId);
            final float x = event.getX(pointerIndex);
            final float y = event.getY(pointerIndex);

            // Only move if the ScaleGestureDetector isn't
            // processing a gesture.
            if (!mScaleDetector.isInProgress())
            {
                final float dx = x - mLastTouchX;
                final float dy = y - mLastTouchY;
                Util.updatePositionObjectsScene(scene, dx, dy,
                    mLastOrientation);
            }

            mLastTouchX = x;
            mLastTouchY = y;
            break;
        }

        case MotionEvent.ACTION_POINTER_UP: {
            final int pointerIndex = (event.getAction() &
                MotionEvent.ACTION_POINTER_INDEX_MASK)
                >> MotionEvent.ACTION_POINTER_INDEX_SHIFT;

```

```
        final int pointerId = event.getPointerId(pointerIndex);
        if (pointerId == mActivePointerId) {
            // This was our active pointer going up. Choose a
            // new
            // active pointer and adjust accordingly.
            final int newPointerIndex = pointerIndex == 0 ? 1 :
                0;
            mLastTouchX = event.getX(newPointerIndex);
            mLastTouchY = event.getY(newPointerIndex);
            mActivePointerId = event.getPointerId(
                newPointerIndex);
        }
        break;
    }

    case MotionEvent.ACTION_UP:
        {
            mActivePointerId =
                INVALID_POINTER_ID; break; }
    case MotionEvent.ACTION_CANCEL:
        {
            mActivePointerId = INVALID_POINTER_ID;
            break; }
    case MotionEvent.ACTION_POINTER_DOWN:
        {
            break; }

    }

    return true;
}

/**
 * Clase que gestiona el gesto de escalar un objeto
 * @author Iván Lorenzo Rubio
 */
private class ScaleListener extends ScaleGestureDetector.
    SimpleOnScaleGestureListener
{
    @Override
    public boolean onScale(ScaleGestureDetector detector)
    {
        mScaleFactor *= detector.getScaleFactor();

        // Evitamos que el objeto sea muy grande o muy pequeño
        mScaleFactor = Math.max(0.1f, Math.min(mScaleFactor, 5.0f));
        return true;
    }
}

//----- FIN Movimientos -----
```

```
//----- Escena min3d -----

/**
 * Escena que mantendrá todos los objetos
 */
public Scene scene;

/**
 * Menajador de inicio de la escena
 */
protected Handler _initSceneHandler;

/**
 * Manejador de actualización de la escena
 */
protected Handler _updateSceneHandler;

final Runnable _initSceneRunnable = new Runnable()
{
    public void run() {
        onInitScene();
    }
};

final Runnable _updateSceneRunnable = new Runnable()
{
    public void run() {
        onUpdateScene();
    }
};

/**
 * Inicializamos los objetos 3d, establecemos sus propiedades y
 * añadimos los objetos
 * a la escena
 */
@Override
public void initScene()
{
    scene.lights().add(new Light());
    scene.camera().frustum.zFar(10000.0f);
    //scene.camera().frustum.shortSideLength(0.77f);

    //Añadimos los objetos a la escena
    for (Object3d object : KnowledgeBase.getInstance().
        getObjectsToScene(getResources())) {
        scene.addChild(object);
    }
}
```

```
}

/**
 * Cualquier manipulación de la escena y de los objetos 3d irá aquí
 *
 * En nuestro caso tan solo modificaremos la escala por aquella que
 * tengan en cada momento.
 */
@Override
public void updateScene()
{
    Util.updateZoomObjectsScene(scene, mScaleFactor);
}

/**
 * Llamado justo después de iniciarse la escena
 */
public void onInitScene()
{
}

/**
 * Llamado justo después de actualizarse la escena
 */
public void onUpdateScene()
{
}

/**
 * @see min3d.interfaces.ISceneController#getInitSceneHandler()
 */
@Override
public Handler getInitSceneHandler()
{
    return _initSceneHandler;
}

/**
 * @see min3d.interfaces.ISceneController#getUpdateSceneHandler()
 */
@Override
public Handler getUpdateSceneHandler()
{
    return _updateSceneHandler;
}

/**
 * @see min3d.interfaces.ISceneController#getInitSceneRunnable()
 */
@Override
public Runnable getInitSceneRunnable()
{
    return _initSceneRunnable;
}
```

```

    }

    /**
     * @see min3d.interfaces.ISceneController#getUpdateSceneRunnable()
     */
    @Override
    public Runnable getUpdateSceneRunnable()
    {
        return _updateSceneRunnable;
    }
}

//----- FIN Escena min3d -----

//----- LOCATION -----

/**
 * Establecemos el estado de la aplicación: Marker o GPS y en caso
 * de ser localización, lo mandamos dibujar
 */
@Override
public void update(LocationData data)
{
    arStatus = data.getStatus();

    if (data.getStatus() == ARStatus.AR_GPS)
    {
        if (arToolkitDrawer != null)
        {
            arToolkitDrawer.draw(this, new int[] { data.
                getObject().getOrder() });
        }
    }
}
}

```

Paquete gui C.44: TFGARActivity.java

### ARToolkitDrawer.java

```

package com.ivanlorenzo.gui;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.os.SystemClock;
import android.util.Log;
import android.view.Gravity;
import android.view.WindowManager;
import android.widget.TextView;

```

```
import java.io.InputStream;
import java.util.ArrayList;

import com.ivanlorenzo.data.KnowledgeBase;
import com.ivanlorenzo.data.objects.audio.Audio;
import com.ivanlorenzo.data.objects.audio.MediaPlayerAdapter;
import com.ivanlorenzo.location.LocationData;
import com.ivanlorenzo.location.observer.IARObserver;

import min3d.core.Renderer;

import jp.nyatla.nyartoolkit.NyAREException;
import jp.nyatla.nyartoolkit.core.NyARCode;
import jp.nyatla.nyartoolkit.core.param.NyARParam;
import jp.nyatla.nyartoolkit.core.raster.rgb.NyARRgbRaster_RGB;
import jp.nyatla.nyartoolkit.core.transmat.NyARTransMatResult;
import jp.nyatla.nyartoolkit.core.types.NyARBufferType;
import jp.nyatla.nyartoolkit.core.types.matrix.NyARDoubleMatrix44;
import jp.nyatla.nyartoolkit.detector.NyARDetectMarker;
import jp.nyatla.nyartoolkit.jogl.utils.NyARGLUtil;

/**
 * Clase responsable de dibujar los objetos. Está basada en min3d
 * @author Iván Lorenzo Rubio
 *
 */
public class ARToolkitDrawer
{

    private int mNumPatt;
    private static final int MARKERMAX = 8;
    private NyARDetectMarker nya = null;
    private NyARRgbRaster_RGB raster = null;
    private NyARParam ar_param = null;
    private NyARCode[] ar_code;
    private double marker_width[];
    private NyARTransMatResult ar_transmat_result = new
        NyARTransMatResult();

    /**
     * Render de min3d
     */
    private Renderer mRenderer;

    //private MediaPlayer mMediaPlayer = null;
    private MediaPlayerAdapter mediaPlayer = null;
    private Context context;

    private Dialog dialog = null;

    static {
        System.loadLibrary("yuv420sp2rgb");
    }
}
```

```

public static native void decodeYUV420SP(int [] rgb, byte [] yuv420sp
    , int width, int height, int type);
public static native void decodeYUV420SP(byte [] rgb, byte []
    yuv420sp, int width, int height, int type);

/**
 * Constructor
 * @param camePara Parámetros de la cámara
 * @param width Ancho de los patrones
 * @param patt Patrones de la aplicación
 * @param mRenderer El render
 */
public ARToolkitDrawer(InputStream camePara, int [] width, ArrayList
    <InputStream> patt, Renderer mRenderer)
{
    this.mRenderer = mRenderer;
    this.initialization(camePara, width, patt);
}

/**
 * Inicializa los elementos de la clase
 * @param camePara Parámetros de la cámara
 * @param width Ancho de los patrones
 * @param patt Lista con los patrones
 */
private void initialization(InputStream camePara, int [] width,
    ArrayList<InputStream> patt)
{
    mNumPatt = patt.size();
    marker_width = new double[mNumPatt];
    ar_code = new NyARCode[mNumPatt];
    try {
        for (int i = 0; i < mNumPatt; i++)
        {
            marker_width[i] = width[i];

            ar_code[i] = new NyARCode(16, 16);
            ar_code[i].loadARPatt(patt.get(i));
        }

        ar_param = new NyARParam();
        ar_param.loadARParam(camePara);
    } catch (Exception e) {
        Log.e("nyar", "resource loading failed", e);
    }
}

/**
 * Se crean los recursos
 * @param w Ancho de la pantalla
 * @param h Altura de la pantalla
 */
private void createNyARTool(int w, int h)
{

```

```
        try {
            if (nya == null && mNumPatt>0)
            {
                ar_param.changeScreenSize(w, h);

                nya = new NyARDetectMarker(ar_param,
                    ar_code, marker_width, mNumPatt,
                    NyARBufferType.BYTE1D_B8G8R8_24);
                nya.setContinueMode(true);
            }
            Log.d("nyar", "resources have been loaded");
        } catch (Exception e) {
            Log.e("nyar", "resource loading failed", e);
        }
    }

    /**
     *
     * @param i_arparam
     * @param o_gl_projection
     */
    public static void toCameraFrustumRHf(NyARParam i_arparam, float []
        o_gl_projection)
    {
        double [] mf = new double [16];
        NyARGLUtil.toCameraFrustumRH(i_arparam, NyARGLUtil.
            SCALE_FACTOR.toCameraFrustumRH_NYAR2, 10, 10000, mf);

        for (int i = 0; i < mf.length; i++) {
            o_gl_projection[i] = (float) mf[i];
        }
    }

    /**
     *
     * @param i_ny_result
     * @param o_gl_result
     */
    public static void toCameraViewRHf(NyARDoubleMatrix44 i_ny_result,
        float [] o_gl_result)
    {
        double [] mf = new double [16];
        NyARGLUtil.toCameraViewRH(i_ny_result, NyARGLUtil.
            SCALE_FACTOR.toCameraViewRH_NYAR2, mf);

        for (int i = 0; i < mf.length; i++) {
            o_gl_result[i] = (float) mf[i];
        }
    }

    /**
     * Método que detecta los patrones y si los encuentra pinta los
     * objetos
     * @param data Datos enviados para pintar
     */
}
```

```

    */
    public void draw(byte[] data) {

        if(data == null) {
            Log.d("AR draw", "data= null");
            return;
        }
        Log.d("AR draw", "data.length= " + data.length);
        int width = 320;
        int height = 240;

        // start coordinates calculation.
        byte[] buf = new byte[width * height * 3];

        // assume YUV420SP
        long time1 = SystemClock.uptimeMillis();
        // convert YUV420SP to RGB24
        decodeYUV420SP(buf, data, width, height, 1);
        long time2 = SystemClock.uptimeMillis();
        Log.d("ARToolkitDrawer", "RGB decode time: " + (time2 -
            time1) + "ms");

        float [][] resultf = new float [MARKERMAX][16];

        int found_markers = 0;
        int ar_code_index [] = new int [MARKERMAX];

        createNyARTool(width, height);
        // Marker detection
        try {
            Log.d("AR draw", "Marker detection.");
            raster = new NyARRgbRaster_RGB(width, height);
            raster.wrapBuffer(buf);
            if(nya!=null)
                found_markers = nya.detectMarkerLite(raster
                    , 100);
        } catch (NyARException e) {
            Log.e("AR draw", "marker detection failed", e);
            return;
        }

        boolean isDetect = false;
        int objectSound = 0;

        // Se pintará un objeto si se encuentra el patrón
        if (found_markers > 0)
        {
            // Projection transformation.
            float [] cameraRHf = new float [16];
            toCameraFrustumRHf(ar_param, cameraRHf);

            if (found_markers > MARKERMAX)
                found_markers = MARKERMAX;
            for (int i = 0; i < found_markers; i++) {

```

```
        if (nya.getConfidence(i) < 0.60f )
            continue;
        try {
            ar_code_index[i] = nya.
                getARCodeIndex(i);
            objectSound = ar_code_index[i];
            NyARTransMatResult transmat_result
                = ar_transmat_result;
            nya.getTransmationMatrix(i,
                transmat_result);
            toCameraViewRHf(transmat_result,
                resultf[i]);
            isDetect = true;
        } catch (NyARException e) {
            Log.e("AR draw", "getCameraViewRH
                failed", e);
            return;
        }
    }
    /*
    Log.d("AR draw", "cameraRHf: " + cameraRHf[0] + ",
    "
    + cameraRHf[1] + ", " + cameraRHf[2] + ", "
    + cameraRHf[3] + ", " + cameraRHf[4] + ", "
    + cameraRHf[5] + ", " + cameraRHf[6] + ", "
    + cameraRHf[7] + ", " + cameraRHf[8] + ", "
    + cameraRHf[9] + ", " + cameraRHf[10] + ", "
    + cameraRHf[11] + ", " + cameraRHf[12] + ", "
    + cameraRHf[13] + ", " + cameraRHf[14] + ", "
    + cameraRHf[15]);
    Log.d("AR draw", "resultf: " + resultf[0][0] + ", "
    + resultf[0][1] + ", " + resultf[0][2] + ", "
    + resultf[0][3] + ", " + resultf[0][4] + ", "
    + resultf[0][5] + ", " + resultf[0][6] + ", "
    + resultf[0][7] + ", " + resultf[0][8] + ", "
    + resultf[0][9] + ", " + resultf[0][10] + ", "
    + resultf[0][11] + ", " + resultf[0][12] + ", "
    + resultf[0][13] + ", " + resultf[0][14] + ", "
    + resultf[0][15]);
    */

    mRenderer.objectPointChanged(found_markers,
        ar_code_index, resultf, cameraRHf);
} else {
    Log.d("AR draw", "not exist marker.");

    mRenderer.objectClear();
}

if (isDetect)
{
    Audio audio = null;
    if (KnowledgeBase.getInstance().getObjects().size()
        > objectSound)
```

```

        {
            audio = KnowledgeBase.getInstance()
                .getObjects().get(objectSound).
                getAudio();

            if(dialog==null || !dialog.
                isShowing())
                dialog = getDescriptionDialog(
                    KnowledgeBase.
                        getInstance().
                        getObjects().get(
                            objectSound).
                        getDescription()
                    );
        }

        if(audio == null)
        {
            if(mediaPlayer != null)
                mediaPlayer.stop();
        }
        else
        {
            //Crearlo en caso de que el patrón tenga
            //sonido asociado
            if(mediaPlayer == null)
                mediaPlayer = audio.getMediaPlayer(
                    context);

            else if (mediaPlayer != null && mediaPlayer
                .getResID() == audio.getId())
                mediaPlayer.play();

            else if (mediaPlayer != null && mediaPlayer
                .getResID() != audio.getId())
            {
                mediaPlayer.stop();
                mediaPlayer = audio.getMediaPlayer(
                    context);
            }
        }
    }
    else
    {
        if(dialog!=null)
            dialog.cancel();

        if (mediaPlayer != null)
            mediaPlayer.stop();
    }
}

```

```
//FIXME
private Dialog getDescriptionDialog(String text)
{
    AlertDialog.Builder builder = new AlertDialog.Builder(
        context);
    builder.setMessage(text);
    dialog = builder.create();

    WindowManager.LayoutParams WMLP = dialog.getWindow().
        getAttributes();
    WMLP.gravity = (Gravity.BOTTOM | Gravity.LEFT);
    WMLP.x = 0;
    WMLP.y = 0;
    dialog.getWindow().setAttributes(WMLP);

    dialog.getWindow().setFlags(WindowManager.LayoutParams.
        FLAG_NOT_TOUCH_MODAL,
        WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL);
    dialog.getWindow().clearFlags(WindowManager.LayoutParams.
        FLAG_DIM_BEHIND);

    dialog.show();

//
//      dialog = new Dialog(context);
//      WindowManager.LayoutParams WMLP = dialog.getWindow().
//      getAttributes();
//      WMLP.x = 500;    //x position
//      WMLP.y = 500;    //y position
//
//      dialog.getWindow().setFlags(WindowManager.LayoutParams.
//      FLAG_NOT_TOUCH_MODAL,
//      WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL);
//
//      dialog.getWindow().setAttributes(WMLP);
//      dialog.show();

    TextView textView = (TextView) dialog.findViewById(android.
        R.id.message);

    //TextView textView = new TextView(context);

    textView.setMaxLines(4);
    textView.setTextSize(11);

    //textView.setText(text);
    //dialog.addView(textView, new LayoutParams(LayoutParams
    .WRAP_CONTENT, LayoutParams.WRAP_CONTENT));
```

```

        return dialog;
    }

    /**
     * Método para pintar un objeto en el centro de la pantalla sin
     * falta de detectar ningún patrón
     * @param observer Datos acerca de la localización actual
     * @param objects Objetos a pintar
     */
    public void draw(IARObserver<LocationData> observer, int[] objects)
    {
        mRenderer.objectClear();
        float[][] resultf = new float[][]
        {
            {0.99811774f, 0.0026676375f, -0.06126828f,
             0.0f, -0.05234584f, 0.55755705f,
             -0.8284866f, 0.0f,
             0.031950463f, 0.83013433f,
             0.55664724f, 0.0f, -0.02831533f,
             -0.26701164f, -5.6218963f, 1.0f},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
        };

        int ar_code_index[] = new int[MARKERMAX];
        boolean isDetect = false;
        int objectSound = 0;

        float[] cameraRHf = new float[] {2.19734f, 0.0f, 0.0f, 0.0f,
            0.0f, 3.038051f, 0.0f, 0.0f, 0.007836991f, 0.010460251f,
            -1.002002f, -1.0f, 0.0f, 0.0f, -20.02002f, 0.012f};

        for (int i = 0; i < objects.length; i++)
        {
            ar_code_index[i] = objects[i];
            Log.i("ARcode", ar_code_index[i] + " : " + i);
            objectSound = ar_code_index[i];
            isDetect = true;
        }
        mRenderer.objectPointChanged(objects.length, ar_code_index,
            resultf, cameraRHf);

        if (isDetect)
        {
            Audio audio = null;

```

```
        if (KnowledgeBase.getInstance().getObjects().size()
            > objectSound)
        {
            audio = KnowledgeBase.getInstance().
                getObjects().get(objectSound).getAudio();
        }

        if (audio == null)
        {
            try { Thread.sleep(10000); }
            catch (InterruptedException e) {}

            LocationData data = LocationData.
                getInstance();
            data.setStatus(ARStatus.AR_Marker);
            observer.update(data);

            if (mediaPlayer != null)
                mediaPlayer.stop();
        }
        else
        {
            //Crearlo en caso de que el patr?n tenga
            sonido asociado
            if (mediaPlayer == null)
            {
                mediaPlayer = audio.getMediaPlayer(
                    context);
            }

            else if (mediaPlayer != null && mediaPlayer
                .getResID() == audio.getId())
                mediaPlayer.play();

            else if (mediaPlayer != null && mediaPlayer
                .getResID() != audio.getId())
            {
                mediaPlayer.stop();
                mediaPlayer = audio.getMediaPlayer(
                    context);
            }
            mediaPlayer.addObserver(observer);
        }
    }
    else
    {
        if (mediaPlayer != null)
            mediaPlayer.stop();
    }
}
```

```

    /**
     * Establece el contexto
     * @param context Contexto de la aplicación
     */
    public void setContext(Context context)
    {
        this.context = context;
    }
}

```

Paquete gui C.45: ARToolkitDrawer.java

### Util.java

```

package com.ivanlorenzo.gui;

import java.io.IOException;
import java.util.Map;

import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.content.Context;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import min3d.core.Scene;

/**
 * Clase de utilidades que contendrá un conjunto de métodos estáticos de
 * ayuda a otras clases
 * @author Iván Lorenzo Rubio
 *
 */
public class Util
{
    /**
     * Método que cambia la orientación de la pantalla redondeándola
     * @param orientationInput Orientación de la pantalla
     * @return La orientación redondeada
     */
    public static int roundOrientation(int orientationInput)
    {
        int orientation = orientationInput;
        if (orientation == -1)
            orientation = 0;

        orientation = orientation % 360;
        int retVal;
    }
}

```

```
        if (orientation < (0*90) + 45) {
            retVal = 0;
        } else if (orientation < (1*90) + 45) {
            retVal = 90;
        } else if (orientation < (2*90) + 45) {
            retVal = 180;
        } else if (orientation < (3*90) + 45) {
            retVal = 270;
        } else {
            retVal = 0;
        }

        return retVal;
    }

    /**
     * Actualiza las posiciones de todos los objetos de la escena
     * @param scene Escena que contendrá los objetos
     * @param mPosX El desplazamiento en el eje de las X
     * @param mPosY El desplazamiento en el eje de las Y
     * @param mLastOrientation La orientación de la pantalla
     */
    public static void updatePositionObjectsScene(Scene scene, float
        mPosX, float mPosY, int mLastOrientation )
    {
        for(int i=0; i<scene.numChildren(); i++)
        {
            if(mLastOrientation == 0)
            {
                scene.getChildAt(i).position().x += mPosX
                    /100;
                scene.getChildAt(i).position().y -= mPosY
                    /100;
            }
            else if (mLastOrientation == 180)
            {
                scene.getChildAt(i).position().x -= mPosX
                    /100;
                scene.getChildAt(i).position().y += mPosY
                    /100;
            }
            else if (mLastOrientation == 90)
            {
                scene.getChildAt(i).position().x -= mPosY
                    /100;
                scene.getChildAt(i).position().y -= mPosX
                    /100;
            }
            else if (mLastOrientation == 270)
            {
                scene.getChildAt(i).position().x += mPosY
                    /100;
                scene.getChildAt(i).position().y += mPosX
                    /100;
            }
        }
    }
}
```

```

    }
}

/**
 * Actualiza la escala de todos los objetos de la escena
 * @param scene Escena
 * @param mScaleFactor El factor de escala a aplicar
 */
public static void updateZoomObjectsScene(Scene scene, float
    mScaleFactor)
{
    for(int i=0; i<scene.numChildren(); i++)
    {
        scene.getChildAt(i).scale().x = scene.getChildAt(i).scale()
            .y =
                scene.getChildAt(i).scale().z =
                    mScaleFactor;
    }
}

/**
 * Obtiene las preferencias. O bien las definidas y guardadas por
 * el usuario o en caso de no estar disponibles, carga unas por
 * defecto
 * @param ctx El contexto de la aplicación
 * @return Las preferencias del usuario
 */
public static SharedPreferences getSharedPreferences (Context ctx)
{
    SharedPreferences preferences = ctx.getSharedPreferences(
        ctx.getResources().getString(R.string.
            FILE_PREFERENCES),
        Context.MODE_PRIVATE);

    if( !checkURL(preferences.getString("URL_PATTERNS", ""))
        ||
            !checkURL(preferences.getString("
                URL_OBJECTS", "")) )
    {

        SharedPreferences.Editor editor = preferences.edit
            ();
        editor.clear();

        PreferenceManager.setDefaultValues(ctx, R.xml.
            preferences, true);
        SharedPreferences defaultPreferences =
            PreferenceManager.getDefaultSharedPreferences(
                ctx);

        for (Map.Entry<String, ?> entry :
            defaultPreferences.getAll().entrySet())
        {

```

```
        editor.putString(entry.getKey(), (String)
            entry.getValue());
    }
    editor.commit();
}
return preferences;

}

/**
 * Método que comprueba si una determinada URL está disponible
 * @param url Dirección a comprobar si está activa
 * @return Verdadero si está disponible, falso en caso contrario
 */
private static boolean checkURL(String url)
{
    if (url.equals("")) return false;

    DefaultHttpClient client = new DefaultHttpClient();
    HttpGet getRequest = new HttpGet(url);
    HttpResponse getResponse;
    try {
        getResponse = client.execute(getRequest);
        final int statusCode = getResponse.getStatusLine().
            getStatusCode();
        if (statusCode == HttpStatus.SC_OK)
            return true;
    } catch (IOException e) {
        return false;
    }
    return false;
}

}
```

Paquete gui C.46: Util.java

### ARStatus.java

```
package com.ivanlorenzo.gui;

/**
 * Enumeración de los estados en los que puede estar funcinando la
 * aplicación
 * @author Iván Lorenzo Rubio
 *
 */
public enum ARStatus
{
    /**
```

```

        * Estado en el que se están buscando markers en las imágenes
          obtenidas por la cámara
    */
    AR_Marker,
    /**
        * Estado en el que se está mostrando información proveniente de
          una localización
    */
    AR_GPS
}

```

Paquete gui C.47: ARStatus.java

### MessageWhat.java

```

package com.ivanlorenzo.gui;

/**
 * Enumeración con los mensajes mostrados por la aplicación
 * @author Iván Lorenzo Rubio
 */
public enum MessageWhat
{
    NONE, //0
    CROP_MSG, //1
    FIRST_TIME_INIT, //2
    RESTART_PREVIEW, //3
    CLEAR_SCREEN_DELAY, //4
    SET_CAMERA_PARAMETERS_WHEN_IDLE, //5
    SHOW_LOADING, //6
    HIDE_LOADING //7
}

```

Paquete gui C.48: MessageWhat.java

## C.4.10. Paquete com.ivanlorenzo.gui.preferences

### preferences/TFGPPreferencesActivity.java

```

package com.ivanlorenzo.gui.preferences;

import com.ivanlorenzo.gui.R;

import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;

/**
 * Activity que gestiona las preferencias del usuario. Carga las
   preferencias del recurso

```

```
* preferences.xml
* @author Iván Lorenzo Rubio
*
*/
public class TFGPreferencesActivity extends PreferenceActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        PreferenceManager prefMgr = getPreferenceManager();
        prefMgr.setSharedPreferencesName(getResources().getString(R.string.
            FILE_PREFERENCES));

        addPreferencesFromResource(R.xml.preferences);
    }
}
```

Paquete gui C.49: preferences/TFGPreferencesActivity.java

### C.4.11. Paquete com.ivanlorenzo.gui.dialogs

dialogs/DialogType.java

```
package com.ivanlorenzo.gui.dialogs;

/**
 * Enumeración de los tipos de diálogos de la aplicación
 * @author Iván Lorenzo Rubio
 *
 */
public enum DialogType
{
    DIALOG_LOADING,                                //0
    DIALOG_ERROR_CONNECTION,                        //1
    DIALOG_EXIT,                                    //2
    //ALERT_DIALOG_LOADED_DATA                      //4
}
```

Paquete gui C.50: dialogs/DialogType.java

## C.5. Pruebas JUnit

DataTest.java

```
package com.ivanlorenzo.data.test;

import java.io.IOException;
```

```

import javax.xml.xpath.XPathExpressionException;

import android.test.ActivityInstrumentationTestCase2;

import com.ivanlorenzo.data.CompilerGenerator;
import com.ivanlorenzo.data.exceptions.LoadXMLException;
import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;
import com.ivanlorenzo.gui.TFGARActivity;

public class DataTest extends ActivityInstrumentationTestCase2<
    TFGARActivity>
{

    public DataTest()
    {
        super("com.ivanlorenzo.gui", TFGARActivity.class);
    }

    @Override
    protected void setUp()
    {
        try {
            CompilerGenerator.compile(
                null,
                "http://tfg.ivanlorenzo.es/data/
                testPatterns.xml", "http://tfg.
                ivanlorenzo.es/data/patterns.dtd
                ",
                "http://tfg.ivanlorenzo.es/data/
                testObjects.xml", "http://tfg.
                ivanlorenzo.es/data/objects.dtd"
            );
        } catch (IOException e) { e.printStackTrace(); }
        catch (XPathExpressionException e) { e.
            printStackTrace(); }
        catch (LoadXMLException e) { e.printStackTrace(); }
        catch (NoSuchExtensionObject3DException e) { e.
            printStackTrace(); }
    }

    /**
     * - Comprueba que el patrón 1 tenga asociado el objeto cuyo id sea
     *   1
     * - Comprueba que se hayan cargado 2 patrones y 1 localización
     */
    public void testCheckPatterns()
    {
        assertEquals(CompilerGenerator.BASE.getPattern(1).
            getIdObject(), 1);

        assertEquals(CompilerGenerator.BASE.getPatterns().size(),
            2);
    }
}

```

## C. Código Fuente

---

```
        assertEquals( CompilerGenerator.BASE.getLocations().size(),
                        1);
    }

    /**
     * Comprobamos que haya 3 objetos cargados
     */
    public void testCheckObjects()
    {
        assertEquals( CompilerGenerator.BASE.getObjects().size(), 3)
        ;
    }

    @Override
    protected void tearDown()
    {
        CompilerGenerator.BASE.clear();
    }
}
```

Test JUnit C.51: DataTest.java

### ParserTest.java

```
package com.ivanlorenzo.parser;

import java.io.IOException;

import javax.xml.xpath.XPathExpressionException;

import android.test.ActivityInstrumentationTestCase2;

import com.ivanlorenzo.data.CompilerGenerator;
import com.ivanlorenzo.data.KnowledgeBase;
import com.ivanlorenzo.data.exceptions.LoadXMLException;
import com.ivanlorenzo.data.exceptions.NoSuchExtensionObject3DException;
import com.ivanlorenzo.gui.TFGARActivity;

public class ParserTest extends ActivityInstrumentationTestCase2<
    TFGARActivity>
{

    public ParserTest()
    {
        super("com.ivanlorenzo.gui", TFGARActivity.class);
    }
}
```

```

@Override
protected void setUp()
{
    try {
        CompilerGenerator.compile(
            null,
            "http://tfg.ivanlorenzo.es/data/
            testPatterns.xml", "http://tfg.
            ivanlorenzo.es/data/patterns.dtd
            ",
            "http://tfg.ivanlorenzo.es/data/
            testObjects.xml", "http://tfg.
            ivanlorenzo.es/data/objects.dtd"
        );
    } catch (IOException e) { e.printStackTrace();}
    catch (XPathExpressionException e) {e.
        printStackTrace();}
    catch (LoadXMLException e) { e.printStackTrace();}
    catch (NoSuchExtensionObject3DException e) { e.
        printStackTrace();}
}

/**
 * Comprobamos si efectivamente hay objetos parseados
 */
public void testParse()
{
    assertTrue(
        KnowledgeBase.getInstance().
        getObjectsToScene(this.getActivity().
        getResources()).size() > 0);
}

@Override
protected void tearDown()
{
    CompilerGenerator.BASE.clear();
}
}

```

Test JUnit C.52: ParserTest.java

