ARLoader. Visor de Realidad Aumentada

Bernat Gómez Garcia del Acebal

Desarrollo de Aplicaciones para Dispositivos Móviles

Carlos Caballero González

23/06/2014

© Bernat Gómez Garcia del Acebal

Reservados todos los derechos. Está prohibida la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	AR Loader. Visor de Realidad Aumentada.
Nombre del autor:	Bernat Gómez Garcia del Acebal
Nombre del consultor:	Carlos Caballero González
Fecha de entrega (mm/aaaa):	23/06/2014
Área del Trabajo Final:	Desarrollo de Aplicaciones para Dispositivos Móviles, Realidad Aumentada
Titulación:	*Posgrado Desarrollo Aplicaciones Dispositivos Móviles*

Resumen del Trabajo (máximo 250 palabras):

Visor de modelos en 3D que combina el uso de marcadores físicos y la realidad aumentada para mostrar representaciones virtuales de diferentes objetos.

Inicialmente, se plantea la compatibilidad del programa con modelos estáticos, definidos en un formato estándar, como el **formato OBJ**. Se podrían incorporar otros formatos en el futuro, aunque ya fuera del ámbito del presente proyecto.

El programa ofrece como principal valor añadido una **experiencia de usuario optimizada**, obtenida gracias a la explotación del contexto que hacen los elementos hardware del dispositivo.

Incluye una **interfaz de usuario intuitiva,** que permite al usuario seleccionar un modelo ya existente o cargar uno nuevo, para posteriormente ver su correspondiente representación.

También se brinda la posibilidad de manipular y modificar el elemento mostrado, utilizando para ello un conjunto de opciones predefinidas por el programa.

El objetivo es desarrollar el núcleo de la aplicación como un **módulo independiente**, a modo de librería, de manera que, una vez implementado, pueda integrarse fácilmente en otros programas (juegos, catálogos virtuales de productos, etc.) que requieran los servicios ofrecidos.

Abstract (in English, 250 words or less):

3D model viewer combining both augmented reality and physical markers in order to display different virtual objects.

To begin with, the program should be able to manage static models using a standard format, such as **OBJ format**. Other formats could be added in the future.

App added value comes from an **optimized user experience**, achieved by getting and exploding data about user's context through sensors available on the mobile device.

The application includes a **user-friendly interface,** allowing the client to select a predefined element or to load a custom model in order to display it. The program also gives users the chance of managing and modifying displayed models by using a set of given options.

Project's main goal consists on independent module development. Once it is finished, its core functionalities could be easily used as a library by others applications (games, virtual catalogues, etc.).

Palabras clave (entre 4 y 8):
Plataforma Android Dispositivos móviles Realidad aumentada con marcadores físicos Visor de modelos 3D Formato Wavefront OBJ Lenguaje programación orientada objetos Java

TABLA DE CONTENIDOS

1. Introducción 15

1.1 Contexto y justificación del trabajo 15

1.2 Objetivos del trabajo 15

1.3 Enfoque y método seguido 16

1.4 Planificación del trabajo 17

1.4.1 Tabla de hitos 17

1.5 Breve sumario de productos obtenidos 18

1.6 Breve descripción del resto de capítulos de la memoria 18

2. Realidad Aumentada 19

2.1 Conceptos básicos 19

2.1.1 Definición 19

2.1.2 Taxonomía 19

2.1.3 Flujo de trabajo para RA basada en marcadores 19

3. Análisis de audiencia 21

3.1 Recolección de datos 21

3.1.1 Metodologías cuantitativas 21

3.1.2 Resultados de las metodologías cuantitativas 21

3.1.3 Metodologías cualitativas 22

3.2 Personas 22

3.2.1 Usuario focal 23

3.2.2 Usuario secundario 24

4. Análisis de mercado 25

4.1 Visión global 25

4.2 Usos actuales en el entorno móvil 25

4.2.1 Navegadores 25

4.2.2 Comerciales 25
4.2.3 Educativas 25
4.2.4 Juegos y entretenimiento 25
4.2.5 Sociales 25
4.2.6 Médicas 25
4.3 Librerías disponibles 26
4.3.1 ARToolkit 26
4.3.2 ARToolworks 26
4.3.3 NYARToolkit 26
4.3.4 NYARToolkit Professional 26
4.3.5 Vuforia 26
4.3.6 BuildAR 27
4.3.7 Designers Augmented Reality Toolkit (DART) 27
4.3.8 AR MonkeyKit 27
4.3.9 Metaio SDK 27
4.3.10 D'Fusion Studio 27
4.3.11 Comparativa 27
4.4 Aplicaciones 28
4.4.1 AndAR Model Viewer 28
4.4.2 AR Model Viewer 28
4.4.3 3D Loader Augmented Reality 28
4.4.4 Augmented 3DS File Viewer 29
5. Análisis del sistema 30
5.1 Necesidades de usuario 30
5.1.1 Listado de necesidades 30
5.2 Requerimientos funcionales 30

5.2.1 Funcionalidad principal 30
5.2.2 Otras funcionalidades 30
5.2.3 Funcioalidades adicionales 31
5.3 Requerimientos no funcionales 31
5.3.1 Listado de restricciones/aspectos a tener en cuenta 31
5.4 Virtudes y beneficios 31
5.5 Amenazas y riesgos 31
5.6 Fundamentos teóricos.....32
5.6.1 Programación orientada a objetos 32
6. Diseño del sistema 33
6.1 Diseño preliminar del sistema.....33
6.2 Diseño de base de datos 33
6.2.1 Diseño conceptual.....34
6.2.2 Diseño lógico.....35
6.2.3 Diseño físico.....36
7. Modelado del sistema 37
7.1 Modelado de comportamiento: casos de uso 37
7.1.1 Diagramas de casos de uso general 37
7.1.2 Administración de categorías y modelos.....38
7.1.3 Visor de modelos 39
7.1.4 Galería de imágenes.....39
7.1.5 Buscador global 40
7.1.6 Ayuda 40
7.2 Modelado de comportamiento: flujos de usuario.....40
7.2.1 Caso de uso “Añadir categoría”.....40
7.2.2 Caso de uso “Añadir modelo” 41

7.2.3 Caso de uso “Listar categorías”.....42
7.2.4 Caso de uso “Listar modelos” 42
7.2.5 Caso de uso “Ver información modelo” 42
7.2.6 Caso de uso “Cargar modelo” 43
7.2.7 Caso de uso “Capturar escena”.....43
7.2.8 Caso de uso “Manipular modelo”.....44
7.2.9 Caso de uso “Publicar modelo” 44
7.2.10 Caso de uso “Añadir comentario” 45
7.2.11 Caso de uso “Mostrar grid de imágenes” 45
7.2.12 Caso de uso “Ver imagen”.....46
7.2.13 Caso de uso “Realizar búsqueda estándar” 46
7.2.14 Caso de uso “Realizar búsqueda con filtros”.....46
7.2.15 Caso de uso “Listar resultados”.....47
7.2.16 Caso de uso “Consultar preguntas frecuentes” 47
7.2.17 Caso de uso “Consultar tema de ayuda”.....48
7.2.18 Caso de uso “Listar temas de ayuda”.....48
7.3 Modelado de comportamiento: diagrama de clases.....48
7.3.1 Paquete “com.example.loader” 49
7.3.2 Paquete “com.example.loader.adapter” 50
7.3.3 Paquete “com.example.loader.ar” 50
7.3.4 Paquete “com.example.loader.constants”.....51
7.3.5 Paquete “com.example.loader.data”.....52
7.3.6 Paquete “com.example.loader.repo”.....53
7.3.7 Paquete “com.example.loader.utils” 53
7.3.8 Paquete “com.example.loader.workers” 54
8. Estructura del sistema.....55

8.1 Módulos de la aplicación.....55

8.2 Pantallas de la aplicación.....55

8.3 Navegación principal 57

8.3.1 Springboard 57

8.4 Navegación secundaria 57

8.4.1 Tablas de datos 57

8.5 Otros recursos utilizados.....58

8.5.1 Ayuda contextual 58

8.5.2 Enlaces de retorno 59

8.5.3 Foco automático 59

8.5.4 Hints 59

8.5.5 Textos alternativos 60

8.6 Patrones de diseño 60

8.6.1 Home 60

8.6.2 Categories 60

8.6.3 New category 61

8.6.4 Models 61

8.6.5 New model.....61

8.6.6 Model detail 61

8.6.7 Model viewer.....61

8.6.8 Gallery 62

8.6.9 Gallery detail.....62

8.6.10 Search.....62

8.6.11 Search results 63

8.6.12 Docs.....63

8.6.13 FAQ.....63

8.6.14 How To 63

8.6.15 Settings 63

9. Conclusiones 65

9.1 Grado de cumplimiento de los objetivos 65

9.2 Extensiones del sistema 65

9.3 Valoración personal.....65

10. Glosario.....67

10.1 Listado de términos 67

11. Bibliografía 68

11.1 Bibliografía básica.....68

11.2 Bibliografía electrónica complementaria 68

Anexos.....71

A1. Listado de recursos gráficos utilizados.....71

A2. Listado de recursos de SW utilizados 71

TABLA DE ILUSTRACIONES

Figura 1. Desarrollo de software según el modelo evolutivo.....16

Figura 2. Relación entre realidad y virtualidad 19

Figura 3. Flujo de trabajo en RA 20

Figura 4. Diagrama de componentes de una aplicación de RA.....33

Figura 5. Diagrama entidad-relación de la aplicación 35

Figura 6. Diagrama relacional de la aplicación 36

Figura 7. Diagrama de casos de uso de la app 37

Figura 8. Detalle del caso de uso "Administración de categorías y modelos" 38

Figura 9. Detalle del caso de uso "Visor de modelos".....39

Figura 10. Detalle del caso de uso "Galería de imágenes" 39

Figura 11. Detalle del caso de uso "Buscador global".....40

Figura 12. Detalle del caso de uso "Ayuda" 40

Figura 13. Detalle del paquete "com.example.loader".....49

Figura 14. Detalle del paquete "com.example.loader.adapter" 50

Figura 15. Detalle del paquete "com.example.loader.ar" 50

Figura 16. Detalle del paquete "com.example.loader.contants".....51

Figura 17. Detalle del paquete "com.example.loader.data".....52

Figura 18. Detalle del paquete "com.example.loader.repo".....53

Figura 19. Detalle del paquete "com.example.loader.utils" 53

Figura 20. Detalle del paquete "com.example.loader.workers".....54

Figura 21. Springboard.....57

Figura 22. Tablas de datos 58

Figura 23. Ayuda contextual.....58

Figura 24. Enlaces de retorno 59

Figura 25. Hints.....60

1. Introducción

1.1 Contexto y justificación del trabajo

La **realidad aumentada (RA)** tiene como objetivo "ofrecer a los usuarios una representación mejorada o aumentada del mundo que les rodea" **[1]**.

Así, mediante la combinación de **elementos virtuales y elementos del mundo real**, se pueden crear programas que presentan un sinfín de aplicaciones para los usuarios finales.

A pesar de su enorme potencial, la realidad aumentada sigue siendo una tecnología todavía por explotar, aunque ya empiezan a aparecer productos que hacen una clara apuesta por esta técnica, como las Google Glass. De hecho, diversos autores auguran que la realidad aumentada será el **futuro de las aplicaciones móviles [e1]**.

El presente posgrado pretende dotar al alumno de una base solida de conocimientos en desarrollo móvil y, por tanto, se centra en la implementación de otro tipo de programas, como, por ejemplo, los servicios de geolocalización. Parece necesario también, por tanto, explorar las posibilidades de la realidad aumentada, aunque solo sea a nivel superficial.

Por este motivo, la realización del proyecto del posgrado de "Desarrollo de Aplicaciones para Dispositivos Móviles" se concibe como una plataforma para estudiar los **fundamentos de la realidad aumentada** y demostrar sus aplicaciones en el desarrollo de programas para dispositivos móviles.

Además, la realización del proyecto también obligará a llevar a cabo **tareas secundarias**, aplicables tanto al desarrollo móvil como a otros sectores dentro de las nuevas tecnologías.

A modo de ejemplo, se puede nombrar el estudio de la **librería OpenGL ES,** versión de OpenGL optimizada para dispositivos móviles que permite representar gráficos avanzados y se utiliza especialmente en el sector de los videojuegos.

1.2 Objetivos del trabajo

1. Desarrollar una **aplicación** que permita al usuario visualizar objetos virtuales mediante técnicas de **RA basada en marcadores**.
2. Integrar en el programa opciones para manipulación y **personalización de los elementos** de la app.
3. Soporte a formato estándar Wavefront OBJ **[e2]**.
4. Clasificar y mantener de manera persistente los **datos del programa**.
5. Priorizar la **usabilidad** del programa, haciéndolo adecuado para personas de diferente perfil tecnológico y/o profesional.

6. Aplicar las capacidades para el **desarrollo móvil** adquiridas durante el transcurso del posgrado y practicar otras competencias, como la programación gráfica.
7. Obtener una base sólida de **conocimientos sobre RA**.

1.3 Enfoque y método seguido

El proyecto se ha desarrollado según el **modelo evolutivo o incremental**.

Este paradigma combina los beneficios que ofrecen metodologías como el ciclo de vida clásico y el desarrollo basado en prototipo, pero sin presentar los inconvenientes de las mismas. Es el punto de partida tomado por otras técnicas de desarrollo muy vigentes actualmente, como las metodologías ágiles.

Así, el modelo evolutivo *"se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado"* **[2]**.

La adopción de esta metodología conlleva definir la denominada **lista de control del proyecto**, que contiene el conjunto de funcionalidades que debe incorporar la versión final del sistema.

El **proceso de trabajo** planteado por este modelo consiste en:

- para cada iteración, se extrae un **elemento de la lista de control** del proyecto.
- la funcionalidad extraída **se diseña, se implementa y se prueba** de manera individual.
- finalmente, se realiza un **análisis del nuevo sistema** obtenido, actualizando la lista de control del proyecto si es necesario.

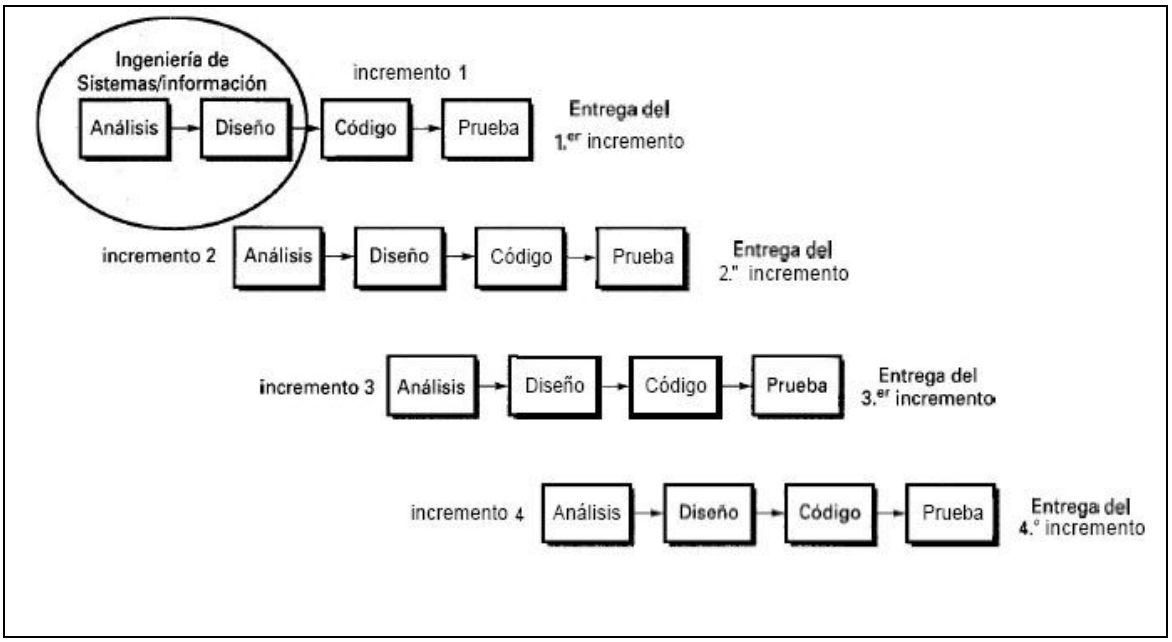

Figura 1. Desarrollo de software según el modelo evolutivo.

El modelo evolutivo presenta diversas **ventajas** respecto a otras metodologías de desarrollo. En primer lugar, proporciona una **prueba del sistema** más exhaustiva, en tanto que realiza pruebas unitarias en cada iteración del proceso. Además, presenta una estructura flexible que permite la **inclusión de nuevos requerimientos** a lo largo del desarrollo, ya que la lista de control del proyecto no es una lista cerrada.

Por otra parte, el principal **inconveniente** que presenta este paradigma radica en el grado de **mantenimiento** que conlleva. La fase de mantenimiento deja de ser una actividad circunstancial y se convierte en una constante que tiene lugar durante todo el desarrollo del sistema.

1.4 Planificación del trabajo

1.4.1 Tabla de hitos

ID	Tarea	Duración	Inicio	Fin	Predecesor	Recursos
1	***INICIACIÓN***	9 días	26-02-14	10-03-14		
2	Elección tema	3 días	26-02-14	28-02-14	-	Proj. Manager
3	Búsqueda bibliográfica	6 días	03-03-14	10-03-14	2	Ana – Prog
4	***ANÁLISIS Y PLANIFICACIÓN***	15 días	11-03-14	31-03-14		
5	Análisis audiencia	4 días	11-03-14	14-03-14	3	Ana – Prog
6	Análisis mercado	7 días	17-03-14	25-03-14	5	Ana – Prog
7	Listado RF	1 días	26-03-14	26-03-14	6	Ana – Prog
8	Listado RNF	1 días	26-03-14	26-03-14	6	Ana – Prog
9	Objetivos	1 días	27-03-14	27-03-14	7; 8	Ana – Prog
10	Alcance	1 días	27-03-14	27-03-14	7; 8	Ana – Prog
11	Plan trabajo	2 días	28-03-14	31-03-14	10	Ana – Prog
12	***EJECUCIÓN***	60 días	01-04-14	23-06-14		
13	DISEÑO	20 días				
14	Casos de uso	1 días	01-04-14	01-04-14	11	Ana – Prog
15	Flujos de trabajo	2 días	02-04-14	03-04-14	14	Ana – Prog
16	Diagrama componentes	1 días	04-04-14	04-04-14	15	Ana – Prog
17	Diagrama clases	1 días	07-04-14	07-04-14	16	Ana – Prog
18	Sketching interfaz usuario	5 días	08-04-14	14-04-14	17	Diseñador
19	Wireframing interfaz usuario	10 días	15-04-14	28-04-14	18	Diseñador
20	Diseño estructuras datos	5 días	08-04-14	14-04-14	17	Ana – Prog
21	Diseño ER de BD	5 días	15-04-14	21-04-14	20	Ana – Prog
22	Diseño relacional de BD	5 días	22-04-14	28-04-14	21	Ana – Prog
23	IMPLEMENTACIÓN	28 días				
24	Interfaz gráfica usuario	10 días	29-04-14	12-05-14	19	Diseñador
25	Base de datos	5 días	29-04-14	05-05-14	22	Ana – Prog
26	Módulo mundo app	5 días	06-05-14	12-05-14	25	Ana – Prog
27	Módulo interacción app	5 días	13-05-14	19-05-14	26	Ana – Prog
28	Módulo presentación app	5 días	20-05-14	26-05-14	27	Ana – Prog
29	Módulo contexto app	5 días	27-05-14	02-06-14	28	Ana – Prog
30	Módulo principal app	15 días	13-05-14	02-06-14	26	Ana – Prog
31	Otras funcionalidades	3 días	03-06-14	05-06-14	29; 30	Ana – Prog
32	TEST	12 días				
33	Test unitario	7 días	06-06-14	16-06-14	31	Ana – Prog
34	Test usabilidad	2 días	06-06-14	09-06-14	31	Tester

ID	Tarea	Duración	Inicio	Fin	Predecesor	Recursos
35	*Test compatibilidad*	*2 días*	*10-06-14*	*11-06-14*	*34*	*Tester*
36	*Test contextual*	*2 días*	*12-06-14*	*13-06-14*	*35*	*Tester*
37	*Test rendimiento*	*1 días*	*16-06-14*	*16-06-14*	*36*	*Tester*
38	*Corrección errores*	*5 días*	*17-06-14*	*23-06-14*	*33; 37*	*Ana – Prog*
39	***CIERRE***	*10 días*	*24-06-14*	*07-07-14*		
40	*Modificaciones memoria*	*5 días*	*24-06-14*	*30-06-14*	*38*	*Ana – Prog*
41	*Presentación*	*5 días*	*24-06-14*	*07-07-14*	*38*	*Ana – Prog*
42	***SEGUIMIENTO Y CONTROL***	*94 días*	*26-02-14*	*07-07-14*		*Proj. Manager*

1.5 Breve sumario de productos obtenidos

El producto principal del desarrollo es una app con las **características descritas** en la siguiente tabla:

Atributo	Valor
Propuesta de título	*AR Loader*
Descripción breve	*Visor de modelos 3D con RA*
Tipo de aplicación	*Nativa*
Plataforma de destino	*Android*
Categoría	*Herramientas / Utilidades*

En resumen, se han obtenido los siguientes **productos**:

- **Aplicación Android** desarrollada**.**
- **Documentación** de la app**.**
- **Memoria** del proyecto**.**
- **Diagramas** sobre el funcionamiento y la estructura de la app.

1.6 Breve descripción del resto de capítulos de la memoria

- **Capítulo 2.** Breve introducción a los conceptos fundamentales sobre la realidad aumentada.
- **Capítulo 3.** Análisis de la audiencia objetiva de la app y descripción de perfiles de usuario.
- **Capítulo 4.** Análisis de las aplicaciones y los frameworks de realidad aumentada presentes en el mercado.
- **Capítulo 5.** Análisis inicial del programa a desarrollar considerando todos los datos obtenidos en los capítulos anteriores.
- **Capítulo 6.** Diseño inicial del sistema y la base de datos.
- **Capítulo 7.** Modelado del sistema mediante diagramas de software.
- **Capítulo 8.** Descripción de la app desarrollada.
- **Capítulo 9.** Conclusiones del proyecto.
- **Capítulo 10.** Recursos bibliográficos consultados para la realización del proyecto.
- **Capítulo 11.** Glosario de términos.

2. Realidad Aumentada

2.1 Conceptos básicos

2.1.1 Definición

La **realidad aumentada** se define como "la visión directa o indirecta de un entorno físico que ha sido mejorado al añadir información generada por computador sobre él" **[3]**.

Esta tecnología combina, por tanto, **elementos reales y contenidos virtuales**. A pesar de todo, no se debe confundir con la realidad virtual (RV), en la que los contenidos digitales suplantan por completo al mundo real.

Se puede plasmar gráficamente la relación entre las diferentes **realidades existentes**, dibujando los respectivos entornos sobre una recta imaginaria **[4]**.

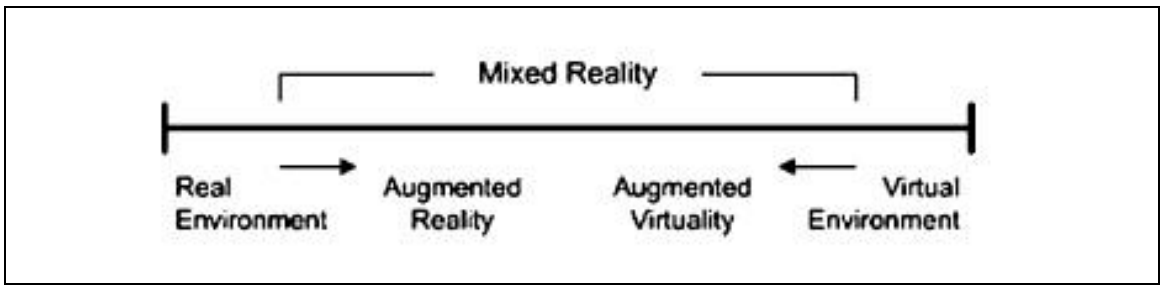

Figura 2. Relación entre realidad y virtualidad

Como se observa en la anterior figura, en los extremos opuestos de la recta se encuentran el mundo real y el entorno virtual respectivamente. La parte central de la línea contiene la **realidad mixta**, que incluye la realidad aumentada (mundo real con elementos virtuales) y la virtualidad aumentada (entorno virtual con elementos reales).

2.1.2 Taxonomía

Se distinguen principalmente **2 tipos** de realidad aumentada **[5]**:

- **Basada en localización,** que utiliza la posición geográfica y la orientación de un dispositivo para mostrar información contextual.

- **Basada en marcadores**, que se apoya en elementos físicos (por ejemplo, una imagen con un patrón determinado) para añadir contenidos virtuales al mundo real.

2.1.3 Flujo de trabajo para RA basada en marcadores

El proceso de **creación de contenidos virtuales** se compone de los pasos descritos a continuación:

1. Obtener una visión del mundo real a través de un dispositivo (cámara)
2. Digitalizar la entrada de datos
3. Detectar el marcador correspondiente en la entrada de datos
4. Asociar el marcador detectado con un contenido virtual concreto
5. Configurar el contenido virtual (posición, escala, orientación)
6. Combinar el contenido real y el contenido virtual en la salida de datos

Diversos autores **[5]** han ilustrado el anterior proceso de manera gráfica, como muestra la siguiente figura.

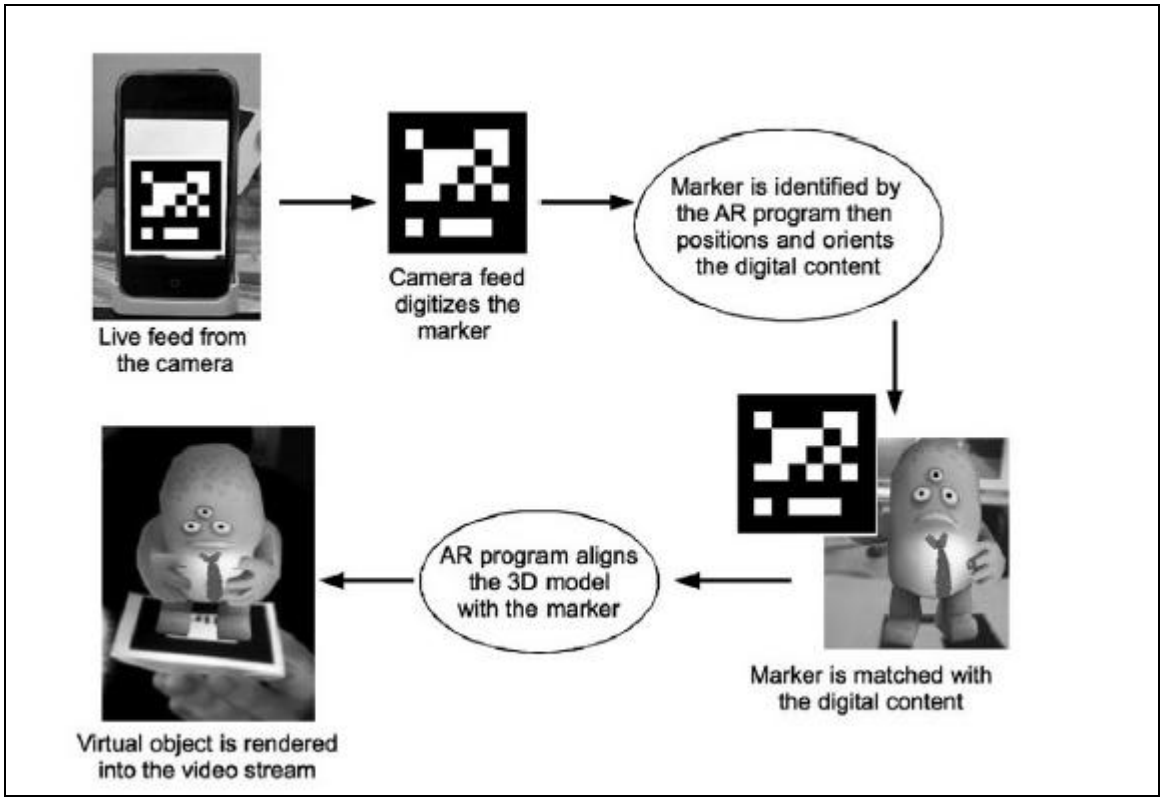

Figura 3. Flujo de trabajo en RA

3. Análisis de audiencia

3.1 Recolección de datos

Como paso previo a la implementación del proyecto, se ha llevado a cabo un **estudio preliminar con usuarios potenciales.**

El objetivo de dicho estudio era determinar las necesidades de usuario y concretar así tanto las **funcionalidades del sistema** como las **características generales de la interfaz**.

La primera fase de dicho estudio, centrada en la recolección de datos, ha combinado **metodologías cuantitativas** y **metodologías cualitativas**.

3.1.1 Metodologías cuantitativas

Para iniciar la recopilación de datos, **9 individuos** han rellenado **2 encuestas anónimas**.

Dichas encuestas, adjuntas con el presente documento, son:

1. **Encuesta general.** Permite obtener una radiografía de los usuarios desde el punto de vista demográfico, académico y profesional.

2. **Encuesta tecnológica**. Indaga en la utilización de las nuevas tecnologías y las preferencias de los usuarios respecto a los dispositivos móviles.

3.1.2 Resultados de las metodologías cuantitativas

Los **datos demográficos** obtenidos se resumen en la siguiente tabla:

Característica	Resultado
Sexo	*2 de cada 3 son mujeres*
Edad	*18 - 55 años*
Estado civil	*1 de cada 3 está casado*
Residencia	*En provincias de Catalunya*

También se observan los siguientes datos sobre el **perfil académico y profesional** de los participantes:

Característica	Resultado
Estudios	*2 de cada 3 tienen estudios superiores*
Situación laboral	*2 de cada 3 con empleo o en búsqueda activa*

Por último, en la siguiente tabla se presentan las conclusiones respecto a las **nuevas tecnologías**:

Característica	Resultado
Plataforma	*2 de cada 3 prefieren Android*

Característica	Resultado
Dispositivos móviles	*Todos tienen Smartphone*
Uso del dispositivo	*Ámbito laboral y personal*
Acceso a Internet	*Fragmentado*
Apps más utilizadas	*Juegos, herramientas, redes sociales*
Mercados de apps	*Uso frecuente*
Usuario de redes sociales	*Sí*
Interfaz de usuario preferida	*Pantallas táctiles*
Preocupaciones	*Seguridad, privacidad*

Ante estos resultados, se puede afirmar que los datos recopilados no permiten dibujar un usuario definido. De hecho, se observa una clara **diversidad de perfiles de usuario**.

Para intentar solventar esta situación, se han utilizado las técnicas cualitativas descritas en el siguiente apartado.

3.1.3 Metodologías cualitativas

La investigación detallada de las características y las necesidades de los usuarios se ha llevado a cabo mediante **entrevistas semiestructuradas**, centradas en aspectos de la **vida profesional y personal** de los entrevistados, así como en su relación con las **nuevas tecnologías**.

La **información extraída** de las entrevistas se ha utilizado para crear a las **personas** descritas en el siguiente apartado.

3.2 Personas

Considerando que la aplicación a desarrollar tiene una **audiencia heterogénea**, se han elaborado **2 arquetipos**, correspondientes al usuario focal y al usuario secundario de la app, para tener una **imagen clara de los usuarios** del programa.

El **usuario focal** descrito presenta un notable dominio de las nuevas tecnologías y utiliza con frecuencia los dispositivos móviles. Además, persigue objetivos profesionales/personales a largo plazo. Por tanto, puede aprovechar al máximo la aplicación desarrollada y, en consecuencia, convertirse en un **usuario habitual** del programa.

El **usuario secundario**, en cambio, tiene unas necesidades puntuales, que pueden resolverse o cambiar a medio plazo. En consecuencia, se prevé una **menor utilización** del programa por parte de los usuarios de este tipo.

A pesar de todo, siempre que sea posible, se deben cubrir las necesidades tanto de los usuarios focales como de los secundarios. Únicamente en caso de conflicto se dará prioridad al primer grupo.

3.2.1 Usuario focal

“Las nuevas tecnologías ya están en todos los ámbitos de nuestro día a día”

DESCRIPCIÓN

Vive en un piso de alquiler con opción de compra en el centro de la ciudad.

Su formación inicial en el campo de las artes plásticas se complementó posteriormente con los estudios correspondientes al grado de “Ingeniería Multimedia”.

Actualmente, compagina su labor como creadora de contenidos digitales en una importante multinacional con trabajos freelance en el sector del diseño gráfico. En constante desarrollo profesional, está cursando también un máster no presencial.

RUTINA DIARIA

Cada mañana, antes de partir hacia el trabajo en coche, comprueba con su smartphone la previsión del tiempo y el estado de las principales vías.

Ya en la oficina, utiliza su teléfono para revisar el correo electrónico o ver las tareas planificadas. Es una persona muy organizada y siempre programa sus actividades en calendarios o agendas electrónicas.

A lo largo de la jornada, también utiliza el tablet para acceder a contenidos de la intranet empresarial, crear presentaciones o comunicarse con compañeros y clientes.

OBJETIVOS PROFESIONALES / PERSONALES

Quiere enfocar su carrera profesional hacia tecnologías emergentes como la realidad aumentada, pues considera que presentan un sinfín de oportunidades.

TECNOLOGÍA

Le gusta suscribirse a boletines de noticias para estar al día.

Es muy crítica con algunos aspectos derivados del uso de las aplicaciones y los dispositivos móviles. No soporta las app de móvil que parecen estar diseñadas para ordenadores de sobremesa.

Afirma que el número de aplicaciones de realidad aumentada disponible es todavía muy bajo y reclama más presencia para los programas de este tipo.

Cuando navega por la Red, no le gusta tener que revisar una infinidad de datos no categorizados hasta encontrar lo que busca.

María Ribas

DEMOGRAFÍA

- *29 años*
- *Soltera con compromiso*
- *Tecnologías de la información*
- *Sabadell*

DISPOSITIVOS

- *Galaxy S4*
- *iPhone 5*
- *iPad*
- *Mac Book Pro*
- *iMac*

INTERESES

- *Tecnologías emergentes*
- *Fitness*
- *Cine & TV*
- *Fotografía*

FRUSTACIONES

- *Desorganización*
- *Atascos viarios*
- *Interface stupidity*

3.2.2 Usuario secundario

“Estoy conectado ininterrumpidamente con mis amigos gracias al móvil y las redes sociales”

DESCRIPCIÓN

Vive con sus padres en las afueras de la ciudad.

Está cursando el cuarto año del grado en “Diseño de Interiores” en una universidad pública.

RUTINA DIARIA

Mientras desayuna, revisa los mensajes en el correo y las redes sociales. Se desplaza hacia la universidad en tren. Durante el viaje, utiliza su teléfono para escuchar música y revisar contenidos académicos de las asignaturas en curso.

Por las tardes, realiza prácticas laborales en una pequeña empresa de decoración. Su labor consiste en visitar a diferentes clientes, para redactar informes de seguimiento sobre los trabajos realizados y ofrecer servicios de consultoría.

Durante la jornada laboral, utiliza frecuentemente su dispositivo para tomar fotografías, llamar a clientes, enviar documentos, buscar información por Internet, etc.

OBJETIVOS PROFESIONALES / PERSONALES

Las prácticas laborales le obligan a representar total o parcialmente espacios virtuales y/o reales. Necesita una herramienta móvil que le permita añadir fácilmente objetos a dichos espacios, para visualizar claramente el resultado final de sus ideas.

TECNOLOGÍA

Se desenvuelve de forma natural en el mundo de las nuevas tecnologías.

Siempre está en permanente contacto con sus familiares y amigos a través de su dispositivo móvil. Todas las aplicaciones que utiliza ofrecen mecanismos de comunicación entre usuarios.

Le gustan los programas simples, sin opciones innecesarias, que presenten la información de manera ordenada.

Se muestra impaciente cuando utiliza una app y tardan en aparecer los contenidos.

Publica frecuentemente comentarios y materiales en Internet.

Andreu Ferrer

DEMOGRAFÍA

- *22 años*
- *Soltero*
- *Estudiante*
- *Barcelona*

DISPOSITIVOS

- *Galaxy S3*
- *Galaxy Tab*
- *Acer Notebook*

INTERESES

- *Videojuegos*
- *Música*
- *Fútbol*

FRUSTACIONES

- *Tiempos de espera*
- *Exámenes*

4. Análisis de mercado

4.1 Visión global

Según varios autores, la RA en el entorno móvil "todavía está en la infancia, dando sus primeros pasos" **[3]**.

A pesar de todo, la **constante evolución de los dispositivos** ya ha permitido lanzar al mercado numerosas aplicaciones que incorporan esta tecnología, como se describe a continuación.

4.2 Usos actuales en el entorno móvil

4.2.1 Navegadores

Aplicaciones como "Wikitude" **[e3]**, que, una vez determinada la posición del usuario, utilizan elementos virtuales para listar puntos de interés o proporcionar indicaciones para llegar a un determinado destino.

4.2.2 Comerciales

Programas como "Beyond Reality" **[e4]**, que emplean la tecnología para promocionar productos o servicios.

4.2.3 Educativas

Aplicativos similares a "Augmented Class!" **[e5]**, que permiten impartir la docencia desde un punto de vista novedoso.

4.2.4 Juegos y entretenimiento

Juegos como "Droid Shooting" **[e6]**, que ofrecen una experiencia inmersiva, pues permiten a los usuarios iniciar una partida en cualquier momento, utilizando el contexto actual como campo de juego.

4.2.5 Sociales

Similares a "TAT Augmented ID" **[e7]**, para obtener de manera selectiva información sobre las personas alrededor del usuario.

4.2.6 Médicas

Se utilizan principalmente para ofrecer tratamientos alternativos a diferentes patologías físicas y/o psíquicas.

Ya se han distribuido aplicaciones concretas, por ejemplo, "Hallux Angles" **[e8]**, que permite complementar las técnicas de radiografía del pie tradicionales.

4.3 Librerías disponibles

4.3.1 ARToolkit

ARToolkit **[e9]** es una biblioteca de código abierto, escrita en C/C++, que permite implementar aplicaciones de **RA basada en marcadores**.

Desarrollada inicialmente en el entorno académico de la Universidad de Washington, posteriormente ha sido utilizada como punto de partida para crear otros frameworks.

4.3.2 ARToolworks

ARToolworks **[e10]** es un conjunto de herramientas propietarias, basado en ARToolkit, que permite desarrollar experiencias de **RA para diferentes entornos** (escritorio, móvil y web).

Se distribuye para cada plataforma móvil con un nombre comercial diferente, como indica la siguiente tabla:

Plataforma	Nombre comercial
Android	*AndAR*
iOS	*ARToolkit for iOS*
Android	*ARToolkit for Android*

4.3.3 NYARToolkit

NYARToolkit **[e11]** es un framework de código abierto para el **desarrollo de aplicaciones RA**, también basado en ARToolkit, que está implementado mediante el lenguaje de programación Java.

Incorpora una funcionalidad que permite usar el dispositivo móvil como cliente de una aplicación RA ejecutada en un servidor remoto.

4.3.4 NYARToolkit Professional

Versión **comercial** de NyARToolkit, con funcionalidades adicionales.

4.3.5 Vuforia

Vuforia **[e12]** es una plataforma de software comercial, desarrollada por Qualcomm, que integra numerosas funcionalidades de **visión por computador** y se puede utilizar para el desarrollo de aplicaciones RA.

En el entorno móvil, está disponible para las 2 plataformas mayoritarias (Android, iOS).

4.3.6 BuildAR

BuildAR **[e13]** es una aplicación propietaria que permite la **carga de modelos en 3D**. También ofrece la posibilidad de definir marcadores personalizados.

4.3.7 Designers Augmented Reality Toolkit (DART)

DART **[e14]** constituye un conjunto de herramientas de software para la **creación de aplicaciones de RA** mediante el entorno multimedia Adobe Director.

4.3.8 AR MonkeyKit

ARMonkeyKit **[e15]** es una librería diseñada especialmente para diseñar **prototipos en RA**.

4.3.9 Metaio SDK

Metaio **[e16]** es un kit de desarrollo comercial, distribuido por la compañía Metaio, que posibilita el **desarrollo de aplicaciones RA multiplataforma** ("develop once, deploy everywhere").

4.3.10 D'Fusion Studio

D'Fusion Studio **[e17]** es una completa suite de herramientas, desarrollada por la compañía Total Inmersion, que facilita el **desarrollo de aplicaciones en RA**. Es compatible con iOS y Android.

4.3.11 Comparativa

La siguiente tabla **[6]** resume las **características principales** de las herramientas descritas anteriormente.

Framework	Licencia	Plataforma	Rendimiento	Contenidos
ARToolkit	*Libre*	*Sobremesa*	*Óptimo*	*3D*
And AR	*Propietaria*	*Android*	*Medio*	*3D*
ARToolkit for iOS	*Propietaria*	*iOS*	*Óptimo*	*3D*
ARToolkit for Android	*Propietaria*	*Android*	*Óptimo*	*3D*
NyARToolkit	*Libre*	*Android*	*Medio*	*3D*
NyARToolkit Professional	*Propietaria*	*Android*	*Óptimo*	*3D*
Vuforia	*Propietaria*	*Android iOS*	*Óptimo*	*3D*
BuildAR	*Propietaria*	*Sobremesa*	*Óptimo*	*3D*
DART	*Propietaria*	*Sobremesa*	*Óptimo*	*3D*
AR MonkeyKit	*Propietaria*	*Sobremesa*	*Óptimo*	*3D*
Metaio SDK	*Propietaria*	*Multiplat.*	*Óptimo*	*3D*
D'Fusion Studio	*Propietaria*	*Android iOS*	*Óptimo*	*Formato propio*

4.4 Aplicaciones

4.4.1 AndAR Model Viewer

Principales características de la app AndAR Model Viewer **[e18]:**

Atributo	Valor
Plataforma	*Android*
Descripción	*Visor de modelos 3D con RA*
Virtudes	- *Compatibilidad con formato .obj* - *Permite manipulación del modelo*
Defectos	- *Requiere la instalación adicional de otros programas para tareas secundarias* - *Inestabilidad en función del dispositivo*

4.4.2 AR Model Viewer

Principales características de la app AR Model Viewer **[e19]:**

Atributo	Valor
Plataforma	*Android*
Descripción	*Visor de objetos 3D con RA*
Virtudes	- *Permite realizar capturas* - *Bluetooth* - *Redes sociales*
Defectos	- *Solo trabaja con objetos predefinidos* - *Tiempos de respuesta*

4.4.3 3D Loader Augmented Reality

Principales características de la app LoaderAugmented Reality **[e20]:**

Atributo	Valor
Plataforma	*Android*
Descripción	*Cargador de modelos 3D con RA*
Virtudes	- *Compatibilidad con formato .obj* - *Soporte para texturas* - *Permite realizar capturas de pantalla*
Defectos	- *Inestabilidad*

4.4.4 Augmented 3DS File Viewer

Principales características de la app Augmented 3DS File Viewer **[e21]:**

Atributo	Valor
Plataforma	*iOS*
Descripción	*Visor de objetos 3D con RA*
Virtudes	- *Compatibilidad con formato .3ds* - *Permite manipulación del objeto* - *Interfaz de usuario intuitiva*
Defectos	- *Formato poco extendido* - *Proceso de carga de objetos*

5. Análisis del sistema

5.1 Necesidades de usuario

Se presentan a continuación las **necesidades detectadas** a partir del estudio con usuarios descrito anteriormente. Se indica, entre paréntesis, el usuario que requiere cada necesidad. Algunos elementos pueden ser compartidos.

5.1.1 Listado de necesidades

1. Comunicación usuario – sistema (focal)
2. Categorización de contenidos (focal)
3. Búsqueda con filtros (focal)
4. Usabilidad (focal)
5. Comunicación usuario – usuario (secundario)
6. Tiempos de respuesta (secundario)
7. Publicación de contenidos en Internet (secundario)
8. Diseño simplificado (secundario)

Como se detalla en los siguientes apartados, algunas de las anteriores necesidades se corresponden con **requerimientos funcionales** (RF) de la app. En cambio, otras se enmarcan dentro de los **requerimientos no funcionales** (RNF) asociados al desarrollo.

5.2 Requerimientos funcionales

5.2.1 Funcionalidad principal

- **Visualización** de modelos (objetos 3D) mediante realidad aumentada.

5.2.2 Otras funcionalidades

- Parser para ficheros en **formato OBJ.**
- Reconocimiento de **marcadores físicos**.
- **Buscador** de modelos.
- Uso de **modelos predefinidos**, ya incluidos por defecto en la app.
- Uso de **modelos personalizados**, indicados por el usuario.
- **Clasificación** de modelos en grupos, utilizando una jerarquía de categorías.
- Posibilidad de **manipulación del modelo mostrado**: traslación, rotación y escalado.
- Inclusión de **comentarios** y anotaciones para cada modelo.
- **Captura de la escena** mediante la cámara**.**
- Mecanismo de **comunicación** (correo electrónico).
- **Publicación** de contenidos en Internet**.**
- **Persistencia** de datos.

5.2.3 Funcioalidades adicionales

Adicionalmente, una vez desarrollado el proyecto, se podrían implementar también las funcionalidades listadas a continuación, aunque éstas quedan **fuera del ámbito del presente proyecto**.

- Definición de **marcadores físicos personalizados.**
- Compatibilidad con **otros formatos** (md2, collada, 3ds, dae).
- Inclusión de la aplicación desarrollada en otro programa, por ejemplo, un juego, a modo de **librería externa**.

5.3 Requerimientos no funcionales

5.3.1 Listado de restricciones/aspectos a tener en cuenta

- **Usabilidad**. Facilitar el uso del programa a usuarios de diferente perfil.
- **Tiempos de respuesta.** Utilizar los recursos de manera eficiente y notificar al usuario las tareas pesadas realizadas en segundo plano**.**
- **Asistencia**. Ubicar al usuario en todo momento y mostrar claramente las opciones ofrecidas por la app.

5.4 Virtudes y beneficios

- **Innovación**. Uso de una tecnología emergente, no extendida todavía en las plataformas móviles.
- **Explotación del contexto**. Utilización del programa en múltiples ámbitos de diferente naturaleza.
- **Portabilidad**. La arquitectura propuesta posibilita el uso de la app como un módulo independiente integrado en otros programas.
- **Flexibilidad**. Capacidad del programa para reconocer y manipular diferentes formatos gráficos.
- **Personalización**. Posibilidad de definir tanto los elementos que se quieren visualizar como los marcadores físicos a utilizar.

5.5 Amenazas y riesgos

- **Fragmentación** de la plataforma Android, que conlleva una mayor carga de trabajo en la fase de desarrollo.
- **Demoras** en la planificación del proyecto**.**
- **Conocimientos técnicos requeridos** sobre RA y programación gráfica, imprescindibles para implementar el proyecto con éxito.
- **Competencia**, otros programas similares ya existentes en mercado.
- Saturación en **tiendas de aplicaciones**, que dificulta la visibilidad de las app.
- Cambios en la **definición estándar de los formatos** soportados, que implicarían la modificación del programa.

5.6 Fundamentos teóricos

5.6.1 Programación orientada a objetos

La **programación orientada a objetos** (POO) es *"una manera de pensar sobre el sistema que se va a desarrollar. En lugar de pensar en los datos y la lógica de programación de forma separada, que es el enfoque de la programación estructurada, trata de dividir los programas en sus componentes y concentrarse en la interacción entre esas partes"* **[7]**.

Actualmente, la orientación a objeto se presenta como una de las metodologías imperantes en el campo de la programación, ya que permite modelar más fácilmente las entidades del mundo real.

En este estilo de programación, los componentes de un sistema se denominan **objetos**. Un objeto es una entidad que agrupa:

- **Estado**. Formado por un conjunto de propiedades o **atributos**, que almacenan la información de interés del objeto.
- **Comportamiento**. Formado por una serie de operaciones o **métodos**, que definen la lógica de programación y operan sobre el estado del objeto.

En el paradigma orientado a objetos, las **clases** son la principal estructura de datos utilizada. La clase, de hecho, es el patrón que se utiliza para definir un tipo de objetos determinado. La definición de una clase especifica con detalle los atributos y los métodos que poseerán todos los objetos de la misma.

Para ejemplificar el planteamiento de la POO, puede suponerse la creación de un hipotético programa informático para almacenar fechas. Dicho programa contendría un objeto de una clase denominada Fecha, que presentaría los atributos día, mes y año para almacenar la información correspondiente. Además, esta clase también definiría diferentes operaciones que permitirían, por ejemplo, mostrar la fecha almacenada por pantalla. El objeto de la clase Fecha definido podría interactuar con objetos de otras clases, que, a su vez, poseerían sus propios atributos y métodos. La **interacción** entre los diferentes objetos del programa proporcionaría, en última instancia, el **comportamiento esperado** del sistema.

La orientación a objeto, por último, posee una amplia **terminología** propia. Los objetos, por ejemplo, también se denominan instancias, para indicar que se han construido en base a una clase determinada. Entre los términos definidos por este paradigma, destacan los conceptos de:

- **Abstracción**. Capacidad para distinguir las características esenciales de un objeto, permitiendo así diferenciarlo del resto de objetos existentes.
- **Encapsulación**. Agrupación de las estructuras de datos y los algoritmos en un objeto. Ocultación de los detalles de la implementación de dicho objeto.
- **Jerarquía**. Establecimiento de una ordenación jerárquica de clases, basándose en la agrupación de atributos y/o métodos comunes.
- **Polimorfismo**. Capacidad de una instancia para hacer referencia a clases diferentes de objetos.

6. Diseño del sistema

6.1 Diseño preliminar del sistema

Las aplicaciones de RA presentan una **arquitectura interna basada en componentes [8]**, como se muestra en la siguiente figura.

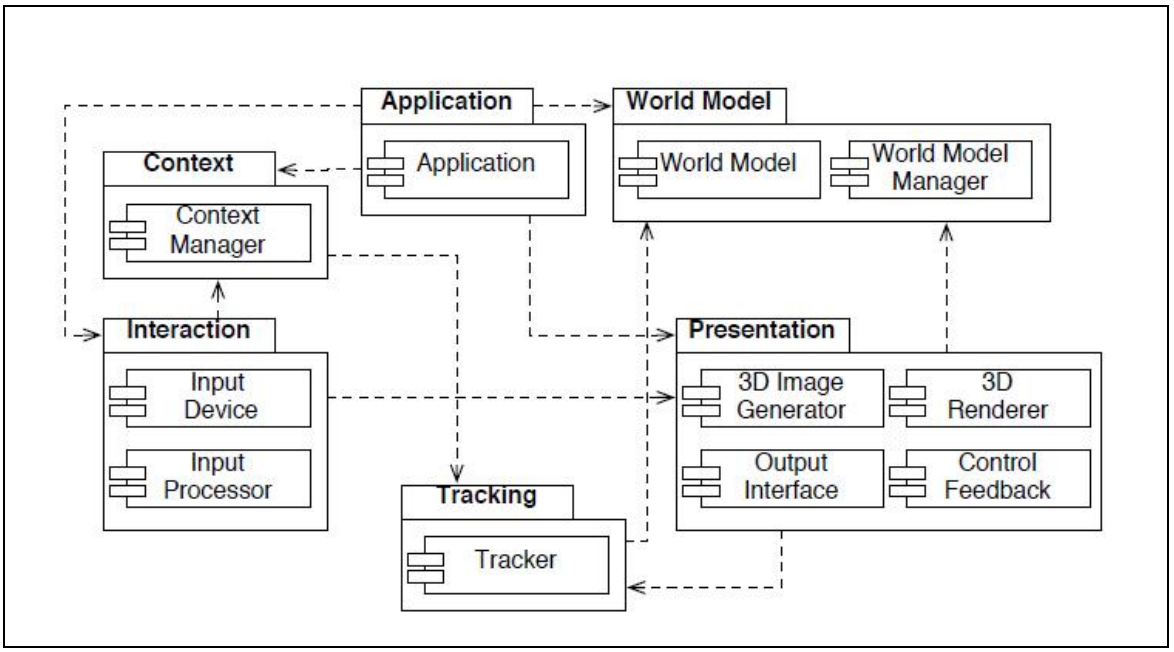

Figura 4. Diagrama de componentes de una aplicación de RA

La anterior estructura contiene los siguientes **módulos**:

- **Aplicación.** Administrador principal del programa, incluye la lógica necesaria para implementar todas las funcionalidades.
- **Interacción.** Encargado de capturar las entradas de datos.
- **Seguimiento.** Gestor de la posición y la orientación actual del dispositivo.
- **Presentación.** Responsable de mostrar en pantalla el resultado final de la escena.
- **Mundo real.** Para almacenar la información del entorno exterior manipulada por el programa.
- **Contexto.** Mecanismo para recolectar, procesar y distribuir datos entre los diferentes componentes de la aplicación.

6.2 Diseño de base de datos

La fase de diseño de la **base de datos** (BD) tiene como principal objetivo definir la estructura del repositorio del sistema. Esta fase se divide en diferentes etapas, que se describen a continuación.

6.2.1 Diseño conceptual

El **diseño conceptual** consiste en determinar la organización de la base de datos, pero obviando la tecnología que ésta utilizará. Para establecer dicha organización, se toman como punto de partida los requerimientos de datos y se utiliza el modelo entidad-relación.

El **modelo entidad-relación** es un modelo de datos semántico, utilizado para capturar el significado de la información a modelar. Principalmente, permite representar:

- **Entidades**. Conjunto de acciones u objetos distinguibles, representadas en el modelo mediante rectángulos.
- **Relaciones**. Conjunto de interrelaciones entre las entidades, representadas por rombos.
- **Atributos**. Propiedades de las entidades o las relaciones, representadas a través de elipses.

En el desarrollo de la BD del sistema, se han definido las siguientes **entidades**:

- **CATEGORIES**. Representa un conjunto de elementos de la app.
- **MODELS**. Caracteriza un modelo 3D que puede ser visualizado mediante realidad aumentada.
- **MODEL_COMMENTS**. Almacena las anotaciones del usuario para cada modelo.
- **FAQS**. Incluye respuestas a preguntas frecuentes sobre la aplicación.
- **HOW_TOS**. Guarda breves tutoriales sobre el funcionamiento del programa.

Del mismo modo, se han establecido las **interrelaciones** que se listan a continuación:

- **MODELS - CATEGORIES**. Describe la relación entre las entidades participantes:
  - Una categoría está formada por diversos modelos.
  - Un modelo solo pertenece a una categoría.
- **MODELS – MODEL_COMMENTS**. Describe la relación entre las entidades participantes:
  - Un modelo puede tener asociadas varias anotaciones/comentarios.
  - Un comentario solo pertenece a un modelo.

El diseño conceptual del repositorio del sistema se ilustra en el correspondiente **diagrama entidad-relación**:

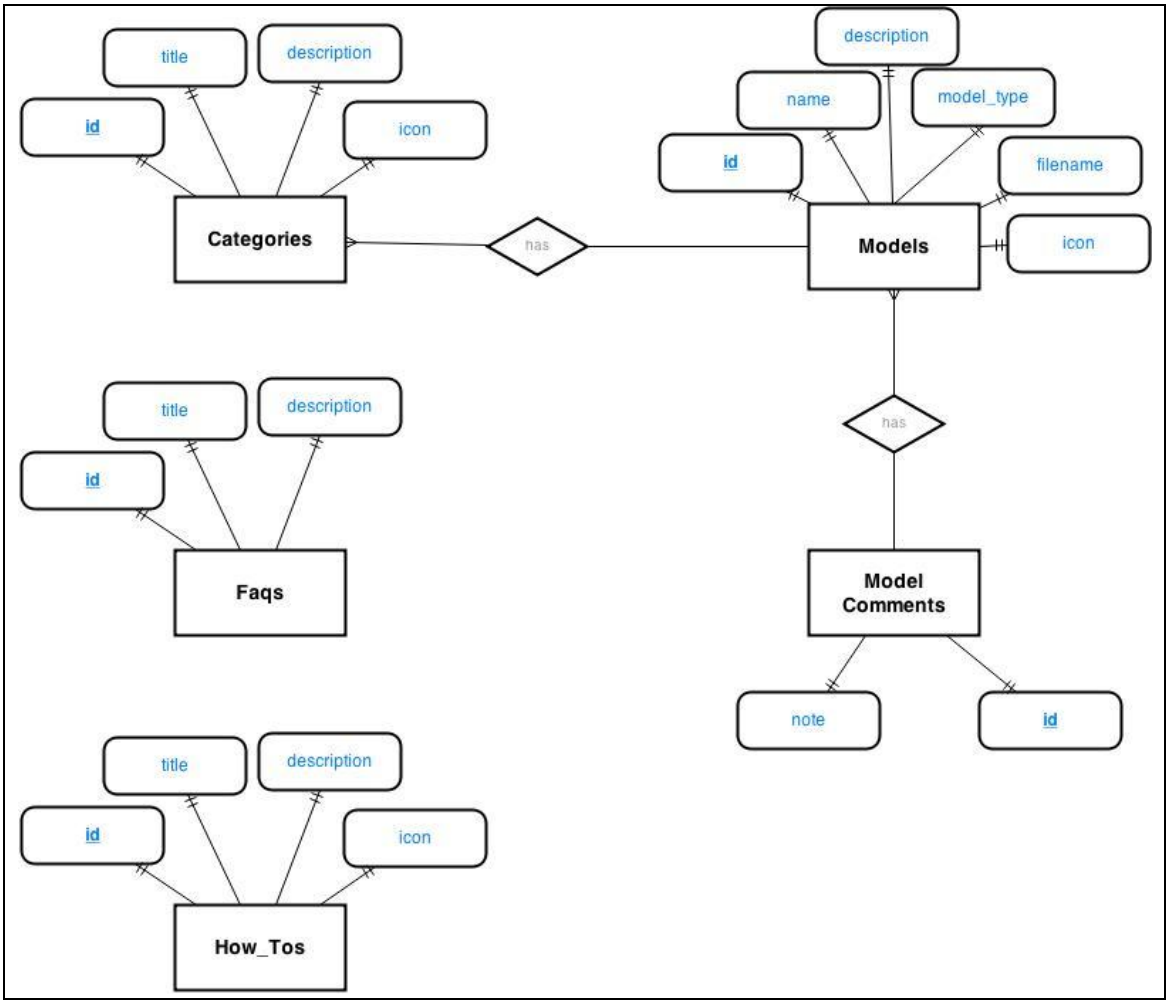

Figura 5. Diagrama entidad-relación de la aplicación

6.2.2 Diseño lógico

El **diseño lógico** es la conversión del esquema entidad-relación, obtenido en la anterior etapa, a un esquema relacional, que se adapta a la tecnología de la base de datos. Para realizar dicha traducción, se utiliza el modelo relacional.

El **modelo relacional** es el modelo de datos funcional implementado por la mayoría de los sistemas gestores de base de datos, también en SQLite. Se basa en el uso de relaciones, es decir, tablas formadas por filas y columnas. Define, por tanto:

- **Relaciones**. Tablas que almacenan la información.
- **Atributos**. Campos (columnas) de información de las relaciones (tablas).

En el desarrollo de la BD del sistema, se han definido las siguientes **relaciones**:

- **CATEGORIES**. Tabla que almacena datos relativos a las categorías de la app.
- **MODELS**. Tabla que guarda los modelos 3D utilizados por el programa.
- **MODEL_COMMENTS**. Tabla que incluye las anotaciones de usuario.
- **FAQS**. Tabla que encapsula información general del programa.

- **HOW_TOS**. Tabla que proporciona datos sobre el funcionamiento de la app.

El diseño lógico del repositorio del sistema se ilustra en el correspondiente **diagrama relacional**:

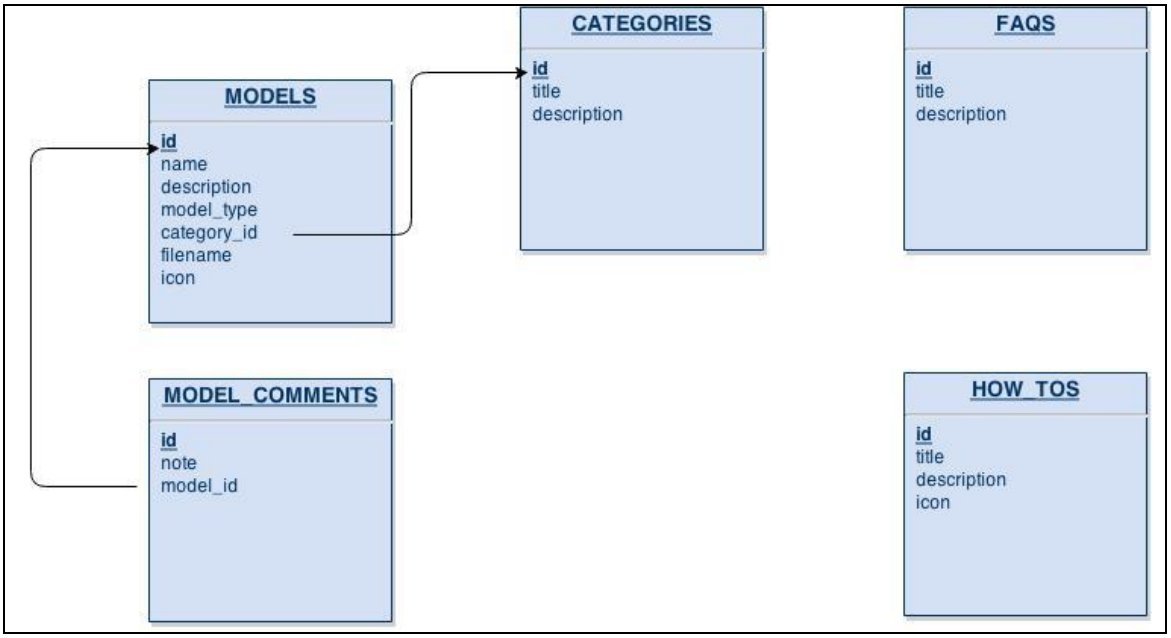

Figura 6. Diagrama relacional de la aplicación

6.2.3 Diseño físico

El **diseño físico**, por último, consiste en definir las características físicas de la base de datos, especificando el conjunto de ficheros con instrucciones que la forman.

Los script de BD necesarios para implementar el repositorio descrito anteriormente están incluidos en el directorio de activos del proyecto.

7. Modelado del sistema

7.1 Modelado de comportamiento: casos de uso

7.1.1 Diagramas de casos de uso general

La siguiente figura muestra el **diagrama de casos de uso de la app**:

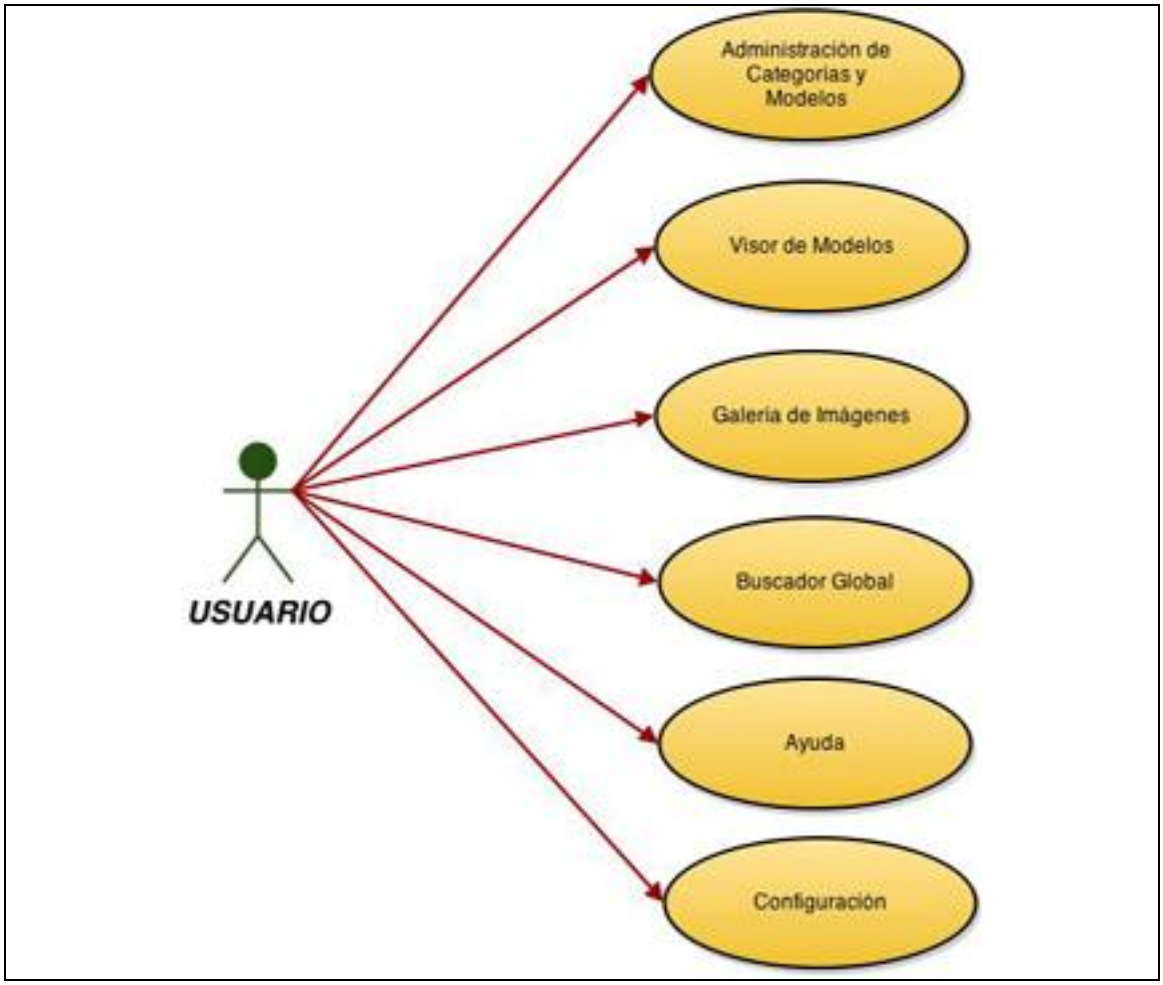

Figura 7. Diagrama de casos de uso de la app

En el anterior diagrama, los casos de uso listados a continuación incluyen **diversas funcionalidades**:

- Administración de categorías y modelos
- Visor de modelos
- Galería de imágenes
- Buscador global
- Ayuda

Estos componentes se desglosan en los siguientes apartados.

7.1.2 Administración de categorías y modelos

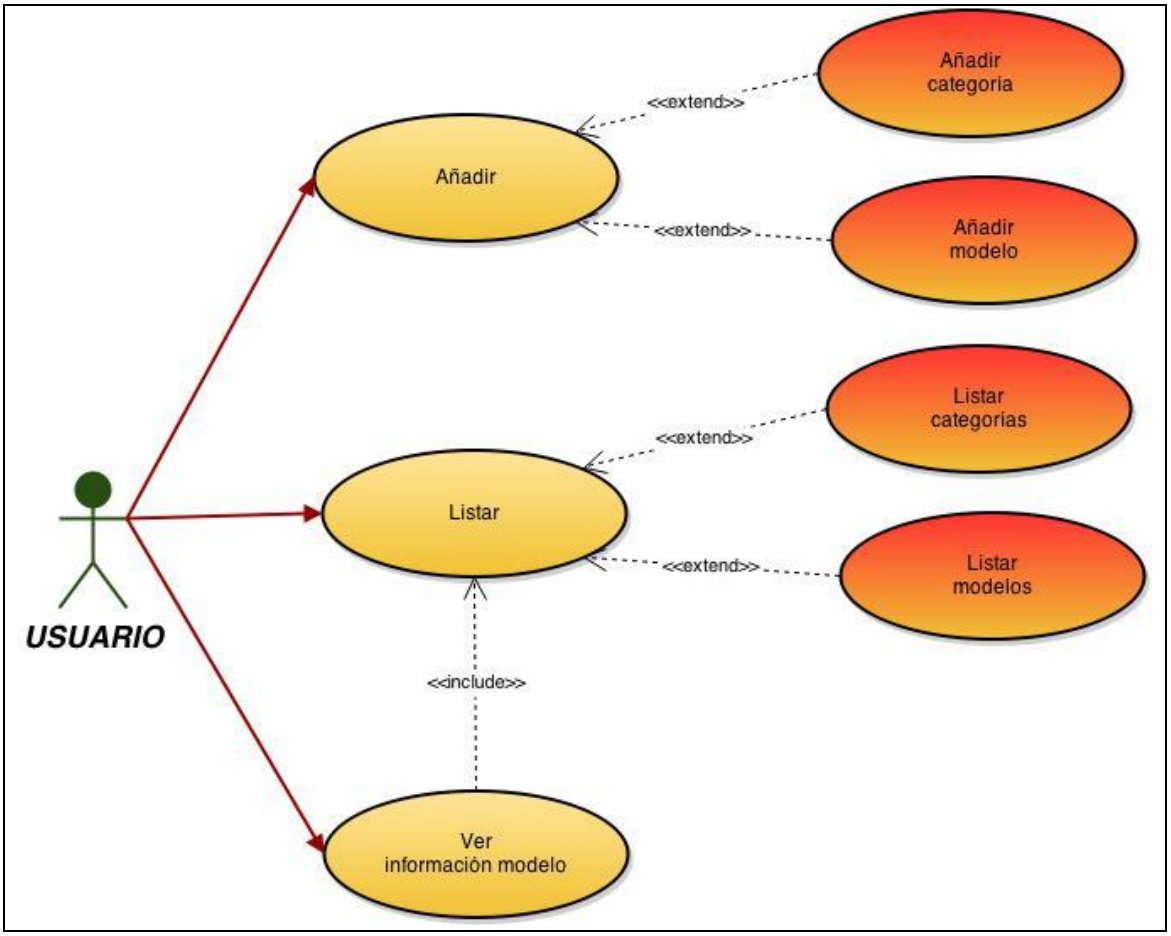

Figura 8. Detalle del caso de uso “Administración de categorías y modelos”

7.1.3 Visor de modelos

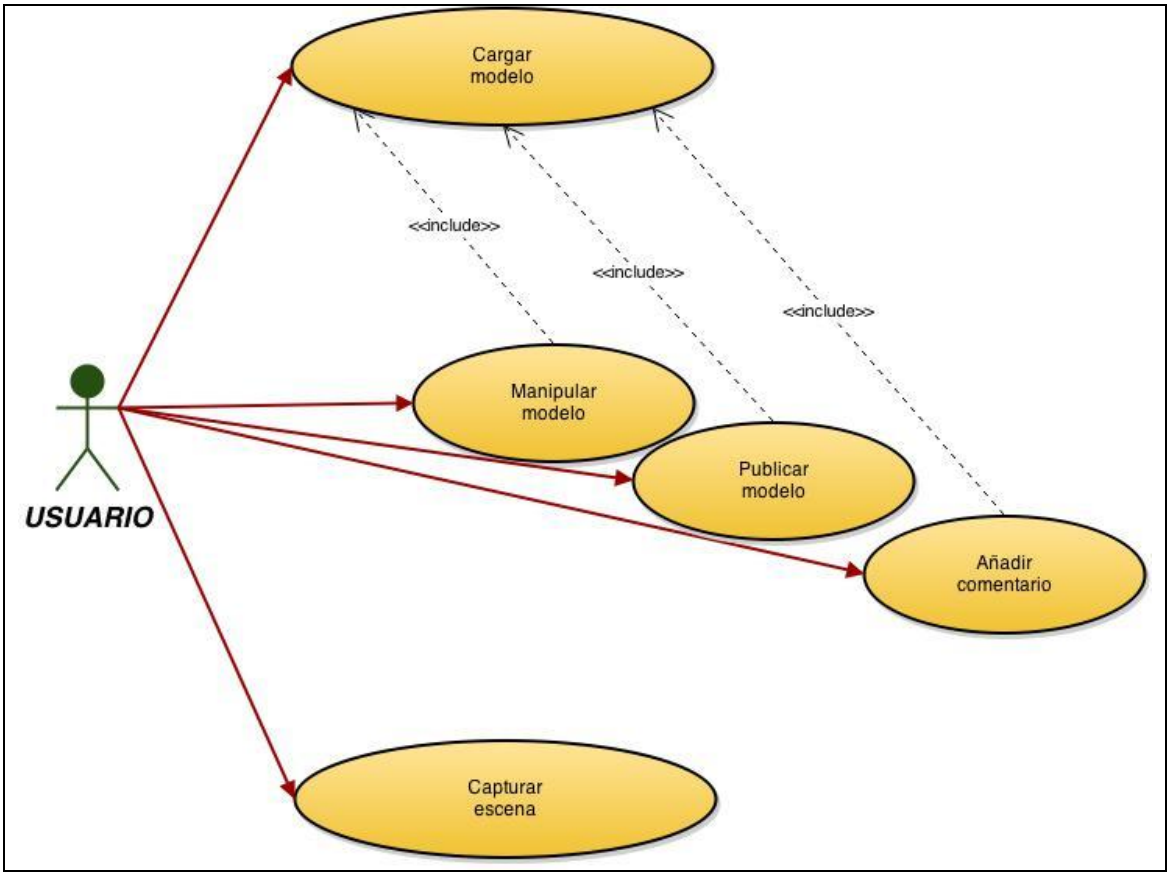

Figura 9. Detalle del caso de uso “Visor de modelos”

7.1.4 Galería de imágenes

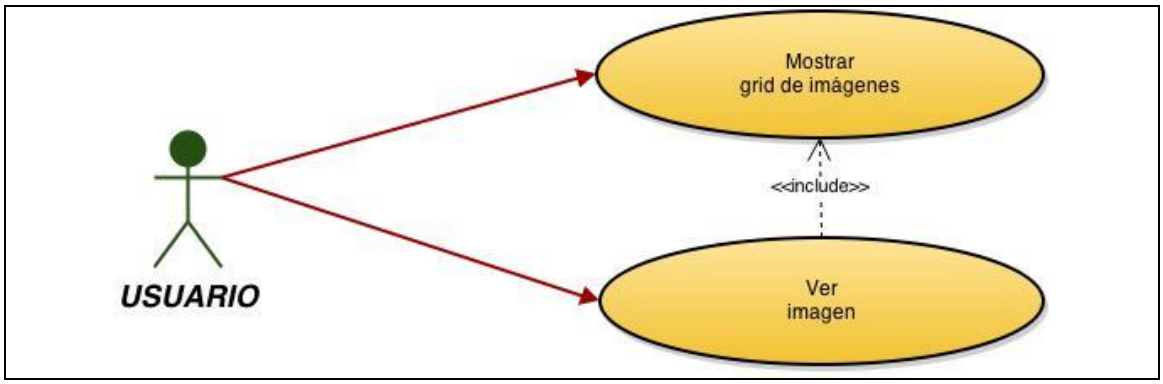

Figura 10. Detalle del caso de uso “Galería de imágenes”

7.1.5 Buscador global

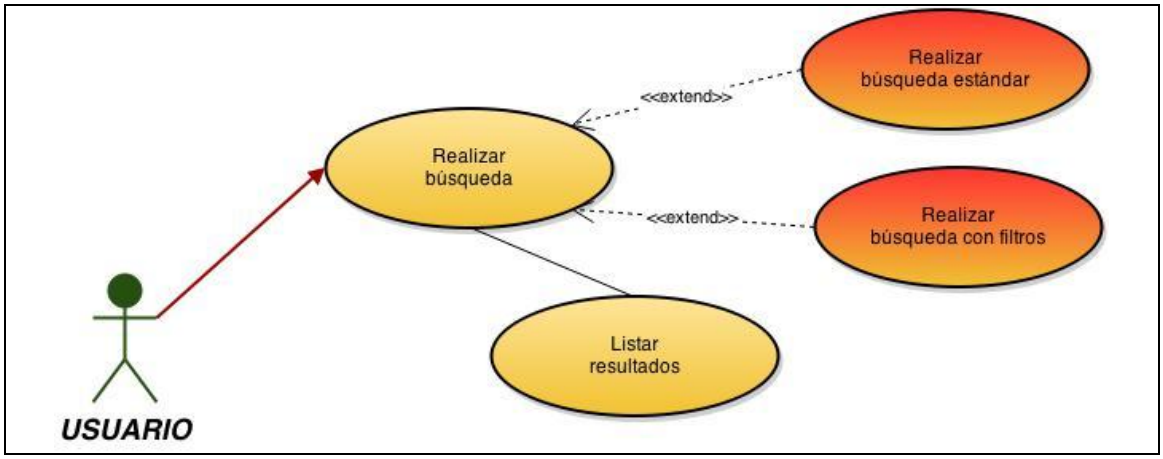

Figura 11. Detalle del caso de uso "Buscador global"

7.1.6 Ayuda

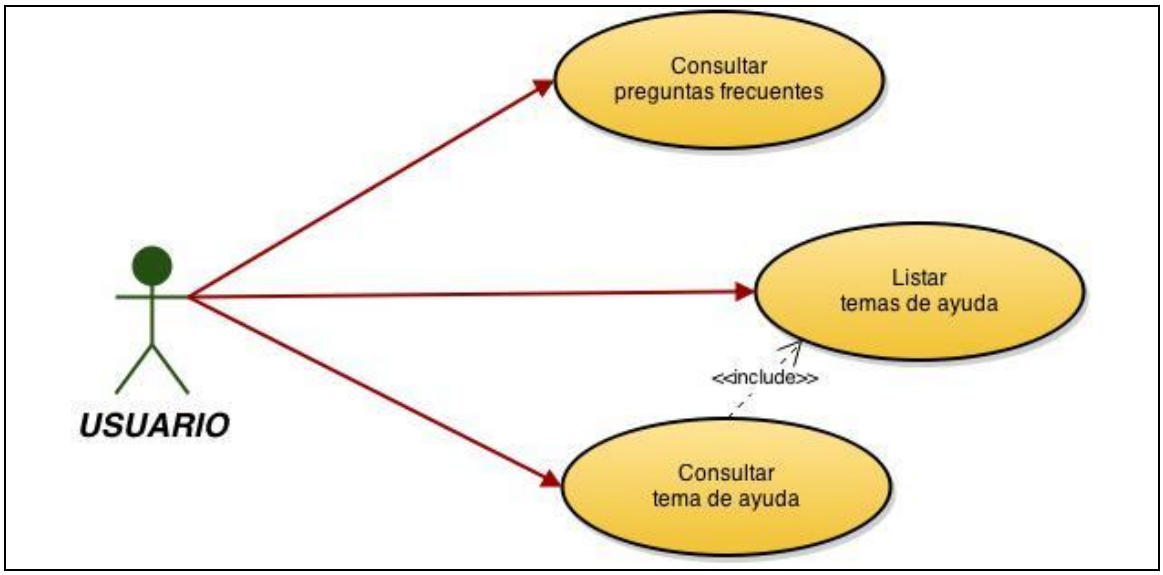

Figura 12. Detalle del caso de uso "Ayuda"

7.2 Modelado de comportamiento: flujos de usuario

7.2.1 Caso de uso "Añadir categoría"

Añadir categoría	
Descripción	*Proceso para crear una nueva categoría.*
Actores	*Usuario*
Precondiciones	

Añadir categoría	
Flujo de eventos principal	*1. El usuario accede al módulo "Elements"* *2. Click en "Add new category"* *3. Introduce nombre de la categoría* *4. Introduce descripción de la categoría* *5. Click en "Save"* *6. Se muestra un mensaje breve de confirmación al usuario*
Flujo de eventos secundario	*Cancelación de la acción* *Si el usuario cancela la operación, finaliza la ejecución del caso de uso.*
Flujo de eventos alternativo	*Error de BD* *Si se produce un error en la capa de datos, se muestra el correspondiente mensaje.*
Poscondiciones	*Categoría insertada en BD*
Requerimientos no funcionales asociados	*Asistencia al usuario* *Integridad de datos*

7.2.2 Caso de uso "Añadir modelo"

Añadir modelo	
Descripción	*Proceso para crear un nuevo modelo.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	*1. El usuario accede al módulo "Elements"* *2. Selecciona una categoría de la lista* *3. Click en "Add new model"* *4. Introduce nombre del modelo* *5. Introduce descripción del modelo* *6. Click en "Save/Continue"* *7. Selecciona el archivo *.obj asociado* *8. Click en "Save"* *9. Se muestra un mensaje breve de confirmación al usuario*
Flujo de eventos secundario	*Cancelación de la acción* *Si el usuario cancela la operación, finaliza la ejecución del caso de uso.* *Archivo no válido* *Si el usuario selecciona un archivo con formato no válido, el sistema muestra un mensaje informativo.*
Flujo de eventos alternativo	*Error de BD* *Si se produce un error en la capa de datos, se muestra el correspondiente mensaje.*
Poscondiciones	*Modelo insertado en BD*

Añadir modelo	
Requerimientos no funcionales asociados	*Asistencia al usuario* *Integridad de datos* *Prevención de errores*

7.2.3 Caso de uso "Listar categorías"

Listar categorías	
Descripción	*Proceso para ver la lista de categorías que agrupan los diferentes modelos.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	1. *El usuario accede al módulo "Elements"* 2. *Consulta la lista de categorías mostrada en pantalla*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.2.4 Caso de uso "Listar modelos"

Listar modelos	
Descripción	*Proceso para ver la lista de modelos asociados a una categoría concreta.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	1. *El usuario accede al módulo "Elements"* 2. *Consulta la lista de categorías mostrada en pantalla* 3. *Selecciona una categoría* 4. *Consulta la lista de modelos contenidos en la categoría seleccionada*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.2.5 Caso de uso "Ver información modelo"

Ver información modelo	
Descripción	*Proceso para ver los datos de un modelo 3D concreto.*
Actores	*Usuario*
Precondiciones	

Ver información modelo	
Flujo de eventos principal	*1. El usuario accede al módulo "Elements"* *2. Consulta la lista de categorías mostrada en pantalla* *3. Selecciona una categoría* *4. Consulta la lista de modelos de la categoría seleccionada* *5. Selecciona un modelo* *6. Visualiza la información mostrada sobre el modelo*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.2.6 Caso de uso "Cargar modelo"

Cargar modelo	
Descripción	*Proceso para mostrar un modelo 3D sobre un marcador físico.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	*1. El usuario accede al módulo "Viewer"* *2. Selecciona una categoría* *3. Selecciona un modelo relativo a la anterior categoría* *4. Click en "Load"* *5. Se muestra un mensaje de carga* *6. Aparece un visor que captura la entrada de video del dispositivo* *7. Cuando se detecta el marcador correspondiente, se renderiza el modelo 3D correspondiente*
Flujo de eventos secundario	
Flujo de eventos alternativo	*Error de carga* *Si se produce un error durante la carga del modelo, se muestra el correspondiente mensaje.*
Poscondiciones	*Modelo cargado*
Requerimientos no funcionales asociados	*Tiempos de espera* *Asistencia al usuario* *Prevención de errores*

7.2.7 Caso de uso "Capturar escena"

Capturar escena	
Descripción	*Proceso para guardar la escena visualizada a través del dispositivo como imagen.*
Actores	*Usuario*

Capturar escena	
Precondiciones	
Flujo de eventos principal	*1. Click en "Menu" para mostrar menú contextual* *2. Selecciona la opción "Capture"* *3. El sistema almacena la imagen en el sistema de archivos* *4. Se muestra un mensaje breve de confirmación al usuario*
Flujo de eventos secundario	
Flujo de eventos alternativo	*Error de sistema de ficheros* *Si se produce un error durante el almacenamiento, se muestra el correspondiente mensaje.*
Poscondiciones	
Requerimientos no funcionales asociados	*Asistencia al usuario*

7.2.8 Caso de uso "Manipular modelo"

Manipular modelo	
Descripción	*Proceso para trasladar, escalar y rotar el modelo visualizado.*
Actores	*Usuario*
Precondiciones	*Modelo cargado*
Flujo de eventos principal	*1. Click en "Menu" para mostrar menú contextual* *2. Selecciona la opción "Scale/Rotate/Translate"* *3. El usuario realiza un touch/gesture sobre la pantalla táctil* *4. El sistema aplica la transformación correspondiente en tiempo real* *5. Observa el resultado de la transformación*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Tiempos de respuesta* *Usabilidad*

7.2.9 Caso de uso "Publicar modelo"

Publicar modelo	
Descripción	*Proceso para publicar en Internet el modelo visualizado.*
Actores	*Usuario*
Precondiciones	*Modelo cargado*

Publicar modelo	
Flujo de eventos principal	*1. Click en "Menu" para mostrar menú contextual* *2. Selecciona la opción "Publish"* *3. Consulta listado de aplicaciones/redes sociales* *4. Escoge el programa a utilizar*
Flujo de eventos secundario	
Flujo de eventos alternativo	*Error de conexión* *Si se produce un error de acceso a redes, se muestra el correspondiente mensaje.*
Poscondiciones	
Requerimientos no funcionales asociados	*Asistencia al usuario* *Prevención de errores*

7.2.10 Caso de uso "Añadir comentario"

Añadir comentario	
Descripción	*Proceso para insertar en BD anotaciones sobre un determinado modelo.*
Actores	*Usuario*
Precondiciones	*Modelo cargado*
Flujo de eventos principal	*1. Click en "Menu" para mostrar menú contextual* *2. Selecciona la opción "Add comment"* *3. Escribe comentario* *4. Click en "Aceptar"* *5. Se muestra un mensaje breve de confirmación al usuario*
Flujo de eventos secundario	*Cancelación de la acción* *Si el usuario cancela la operación, finaliza la ejecución del caso de uso.*
Flujo de eventos alternativo	*Error de BD* *Si se produce un error en la capa de datos, se muestra el correspondiente mensaje.*
Poscondiciones	*Comentario insertado en BD*
Requerimientos no funcionales asociados	*Integridad de datos* *Asistencia al usuario* *Prevención de errores*

7.2.11 Caso de uso "Mostrar grid de imágenes"

Mostrar grid de imágenes	
Descripción	*Proceso para ver tabla con imágenes almacenadas.*
Actores	*Usuario*
Precondiciones	

Mostrar grid de imágenes	
Flujo de eventos principal	*1. El usuario accede al módulo "Gallery"* *2. Explora la tabla de thumbnails*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.2.12 Caso de uso "Ver imagen"

Ver imagen	
Descripción	*Proceso para mostrar una imagen a pantalla completa.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	*1. El usuario accede al módulo "Gallery"* *2. Explora la tabla de thumbnails* *3. Selecciona una imagen en miniatura* *4. Visualiza la imagen a pantalla completa*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	*Imagen a pantalla completa*
Requerimientos no funcionales asociados	*Usabilidad*

7.2.13 Caso de uso "Realizar búsqueda estándar"

Realizar búsqueda estándar	
Descripción	*Proceso para buscar contenidos.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	*1. El usuario accede al módulo "Search"* *2. Escribe el nombre del ítem a buscar en la caja de texto* *3. Click en "Search"*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	*Búsqueda realizada*
Requerimientos no funcionales asociados	*Prevención de errores*

7.2.14 Caso de uso "Realizar búsqueda con filtros"

Realizar búsqueda con filtros	
Descripción	*Proceso para buscar contenidos utilizando filtros.*
Actores	*Usuario*
Precondiciones	

Realizar búsqueda con filtros	
Flujo de eventos principal	*1. El usuario accede al módulo "Search"* *2. Escribe el nombre del ítem a buscar en la caja de texto* *3. Click en "Down"* *4. Selecciona el módulo de la app donde buscar* *5. Escoge la categoría objetivo* *6. Introduce la fecha de publicación* *7. Click en "Search"*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	*Búsqueda realizada*
Requerimientos no funcionales asociados	*Prevención de errores*

7.2.15 Caso de uso "Listar resultados"

Listar resultados	
Descripción	*Proceso para mostrar los resultados de búsqueda.*
Actores	*Usuario*
Precondiciones	*Búsqueda realizada*
Flujo de eventos principal	*1. Consulta la lista de resultados obtenida* *2. Selecciona un ítem de la lista* *3. Detalle*
Flujo de eventos secundario	*Sin resultados* *Si los criterios de búsqueda no retornan ningún resultado, se muestra un mensaje.*
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Asistencia al usuario* *Tiempos de respuesta*

7.2.16 Caso de uso "Consultar preguntas frecuentes"

Consultar preguntas frecuentes	
Descripción	*Proceso para ver respuestas a preguntas frecuentes.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	*1. El usuario accede al módulo "Docs"* *2. Click en el icono "FAQ"* *3. Selecciona panel desplegable* *4. Click en título para mostrar contenido* *5. Consulta información*
Flujo de eventos secundario	
Flujo de eventos alternativo	

Consultar preguntas frecuentes	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.2.17 Caso de uso "Consultar tema de ayuda"

Consultar tema de ayuda	
Descripción	*Proceso para consultar un tema de ayuda concreto.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	1. *El usuario accede al módulo "Docs"* 2. *Click en el icono "HowTo"* 3. *Consulta la lista de temas mostrada en pantalla* 4. *Selecciona un tema de ayuda* 5. *Consulta información mostrada*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.2.18 Caso de uso "Listar temas de ayuda"

Listar temas de ayuda	
Descripción	*Proceso para ver la lista de temas de ayuda.*
Actores	*Usuario*
Precondiciones	
Flujo de eventos principal	1. *El usuario accede al módulo "Docs"* 2. *Click en el icono "HowTo"* 3. *Consulta la lista de temas mostrada en pantalla*
Flujo de eventos secundario	
Flujo de eventos alternativo	
Poscondiciones	
Requerimientos no funcionales asociados	*Usabilidad*

7.3 Modelado de comportamiento: diagrama de clases

Los elementos de la aplicación se agrupan en diferentes paquetes, según su **ámbito y/o funcionalidad**:

- com.example.loader
- com.example.loader.adapter
- com.example.loader.ar
- com.example.loader.constants

- com.example.loader.data
- com.example.loader.repo
- com.example.loader.utils
- com.example.loader.workers

La estructura básica de los principales elementos de la app se desglosa en los siguientes apartados. Se puede encontrar una **descripción detallada** de los anteriores elementos en la **documentación Javadoc** adjunta al presente documento.

7.3.1 Paquete "com.example.loader"

Contiene las **actividades** (pantallas) de la app.

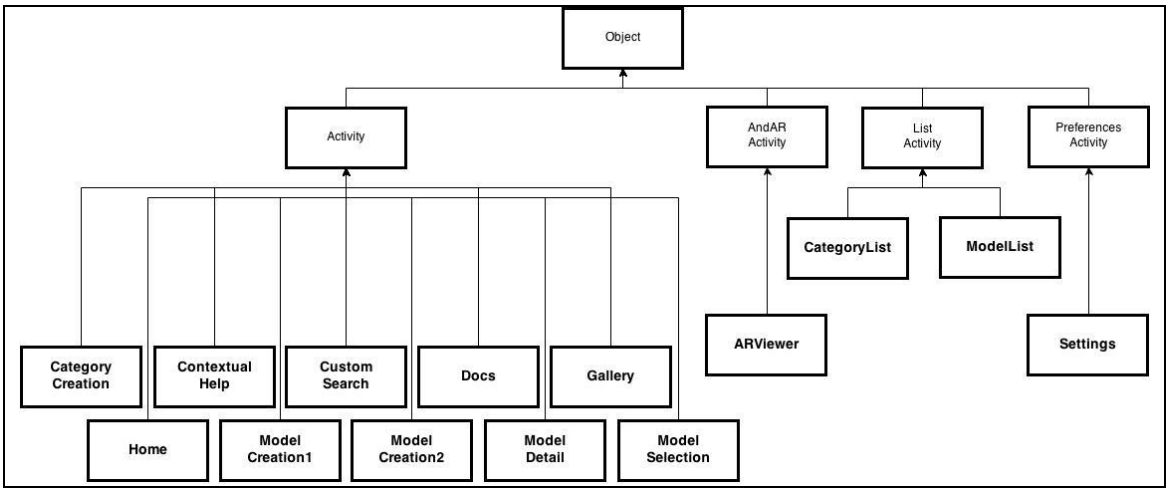

Figura 13. Detalle del paquete "com.example.loader"

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
ARViewer	*Visor de modelos en RA*
CategoryCreation	*Pantalla para creación de categorías*
CategoryList	*Listado de categorías*
ContextualHelp	*Ayuda contextual de la app*
CustomSearch	*Buscador global de contenidos*
Docs	*Documentación de la app*
Gallery	*Galería de imágenes con las capturas realizadas por el usuario*
Home	*Pantalla principal*
ModelCreation1	*Pantalla para creación de modelos (1)*
ModelCreation2	*Pantalla para creación de modelos (2)*
ModelDetail	*Detalle del modelo seleccionado*
ModelList	*Listado de modelos 3D*
ModelSelection	*Selección del modelo 3D a visualizar*
Settings	*Configuración del programa*

7.3.2 Paquete “com.example.loader.adapter”

Contiene los **adaptadores** de la app, es decir, aquellos elementos que vinculan orígenes de datos con componentes gráficos.

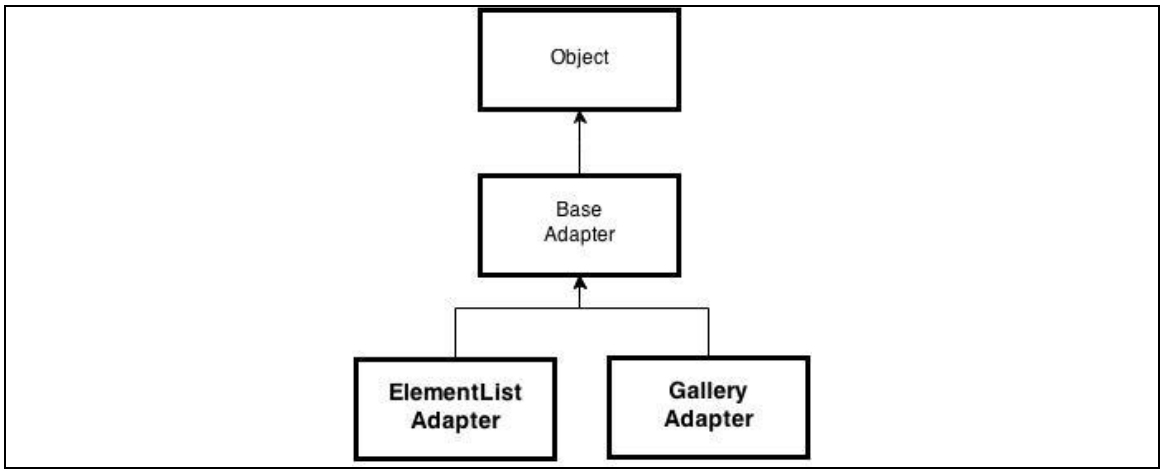

Figura 14. Detalle del paquete “com.example.loader.adapter”

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
ElementListAdapter	*Adaptador para mostrar listados de categorías y modelos*
GalleryAdapter	*Adaptador para mostrar imágenes en miniatura*

7.3.3 Paquete “com.example.loader.ar”

Contiene los **componentes del motor de RA** implementado.

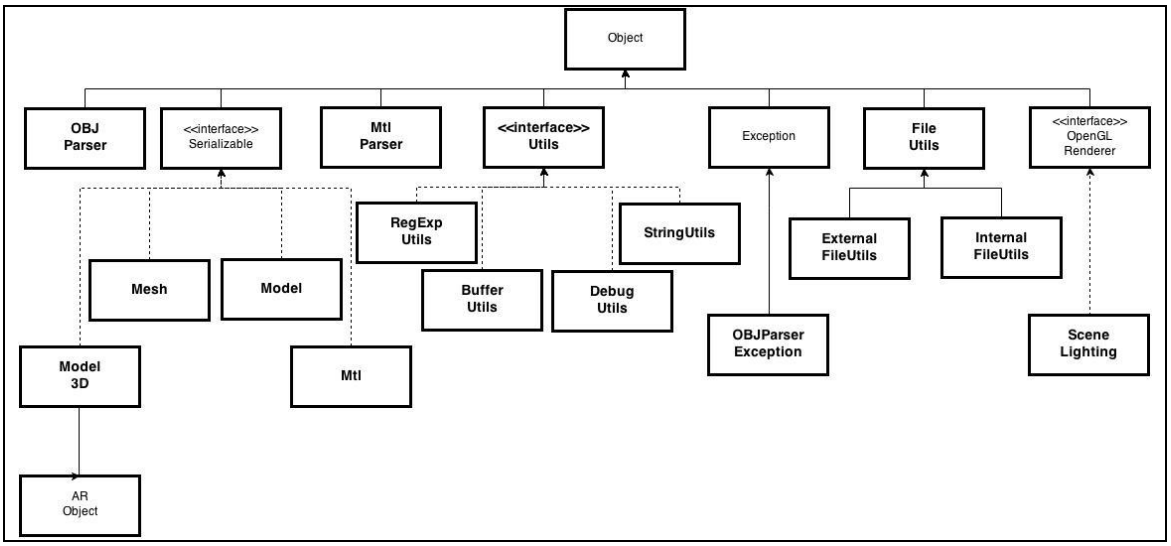

Figura 15. Detalle del paquete “com.example.loader.ar”

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
BufferUtils	*Utilidades para trabajar con buffers de memoria nativos*
DebugUtils	*Mensajes y trazas para debug del sistema*
ExternalFileUtils	*Utilidades para trabajar con ficheros almacenados en SD Card*
FileUtils	*Abstracción para trabajar con ficheros*
InternalFileUtils	*Utilidades para trabajar con ficheros almacenados como activos del proyecto*
SceneLighting	*Utilidades para iluminación de la escena*
Mesh	*Agrupación del conjunto de vértices y caras que componen un modelo 3D*
Model	*Encapsulación de la información básica de un modelo (posición, orientación, rotación y escala)*
Model3D	*Visualización en 3D de un modelo*
Mtl	*Material que compone la superficie de un modelo*
MtlParser	*Analizador de ficheros *.mtl*
OBJParser	*Analizador de ficheros *.obj*
OBJParserException	*Excepción personalizada para representar errores durante el análisis de los ficheros *.obj*
RegExpUtils	*Utilidades para trabajar con expresiones regulares*
StringUtils	*Utilidades para trabajar con cadenas de texto*
Utils	*Interface común para utilidades*

7.3.4 Paquete "com.example.loader.constants"

Contiene **constantes comunes** utilizadas por los módulos de la app.

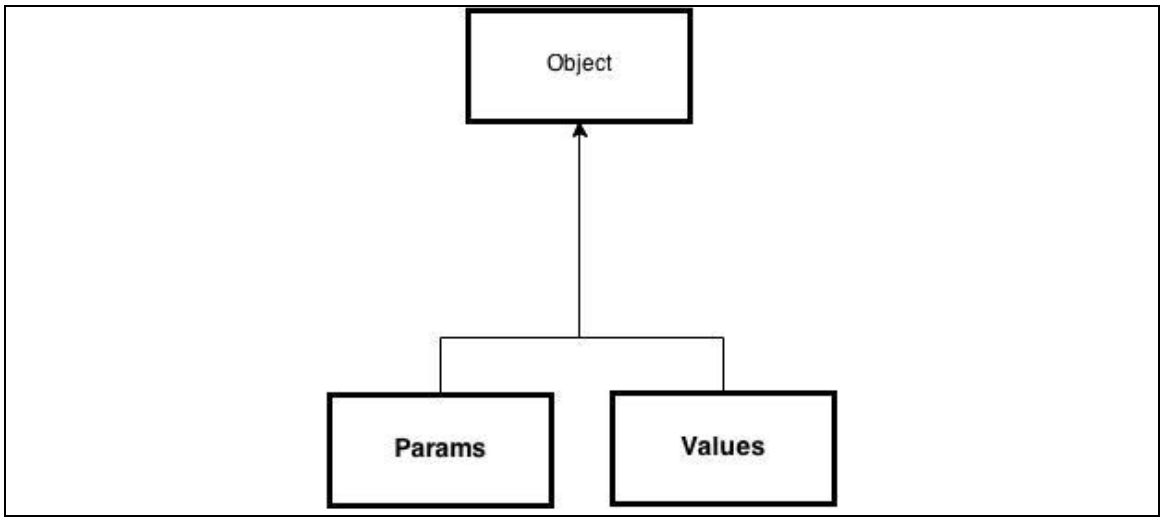

Figura 16. Detalle del paquete "com.example.loader.contants"

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
Params	*Nombre de parámetros intercambiados entre diferentes partes del programa*
Values	*Valores constantes globales*

7.3.5 Paquete "com.example.loader.data"

Contiene **abstracciones de los principales elementos** utilizados por el programa, como las categorías o los modelos. Dichas abstracciones realizan un mapping del elemento correspondiente en BD.

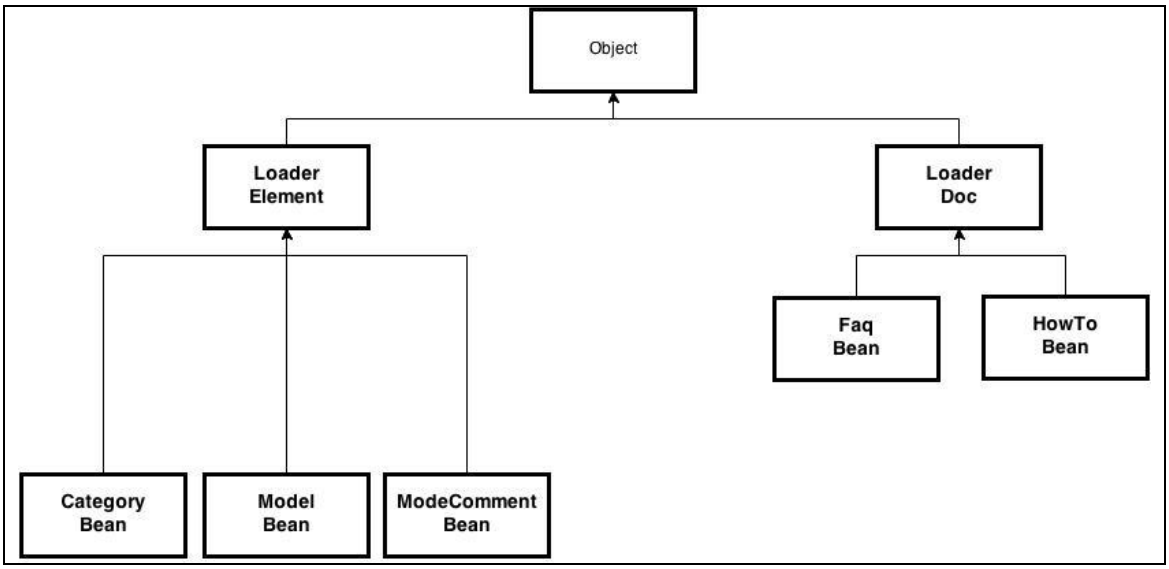

Figura 17. Detalle del paquete "com.example.loader.data"

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
CategoryBean	*Encapsulación de las principales propiedades de una categoría*
FaqBean	*Encapsulación de las principales propiedades de una pregunta frecuente*
HowToBean	*Encapsulación del funcionamiento de un parte concreta del programa*
LoaderDoc	*Superclase para encapsulaciones relativas a la documentación de la app*
LoaderElement	*Superclase para encapsulaciones relativas a los elementos de la app*
ModelBean	*Encapsulación de las principales propiedades de un modelo*
ModelCommentBean	*Encapsulación de las principales propiedades de un comentario acerca de un modelo*

7.3.6 Paquete “com.example.loader.repo”

Contiene los elementos que permiten crear y definir el **repositorio de datos** de la app.

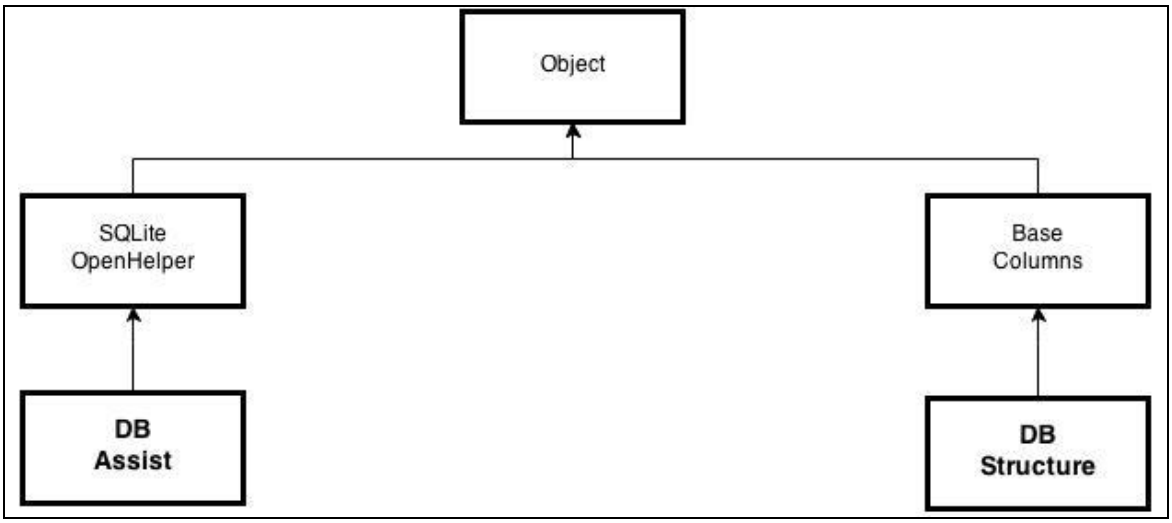

Figura 18. Detalle del paquete “com.example.loader.repo”

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
DBAssist	*Asistente de creación de BD*
DBStructure	*Conjunto de nombres para tablas y columnas de BD*

7.3.7 Paquete “com.example.loader.utils”

Contiene **utilidades compartidas** por los diferentes módulos del programa.

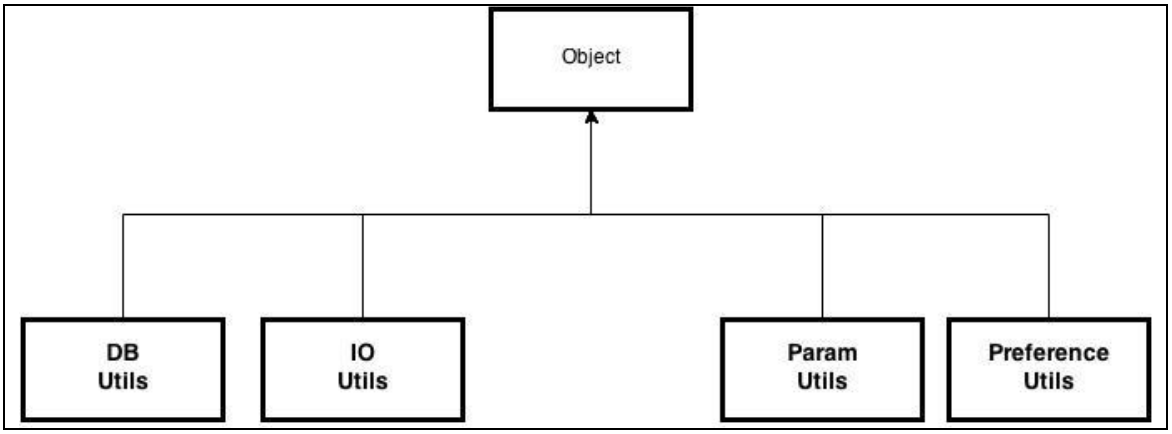

Figura 19. Detalle del paquete “com.example.loader.utils”

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
DBUtils	*Utilidades para manipulación y gestión de BD*
IOUtils	*Utilidades para operaciones de E/S*
ParamUtils	*Utilidades para manipulación de parámetros*
PreferenceUtils	*Utilidades para manipulación de configuración del programa*

7.3.8 Paquete "com.example.loader.workers"

Contiene clases **para realizar tareas en plano secundario**, de manera asíncrona y sin la intervención directa del usuario.

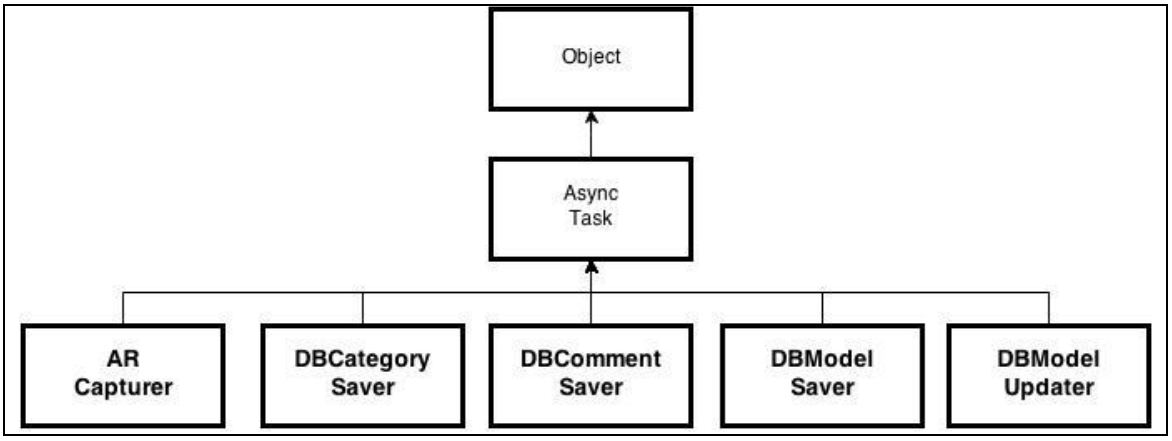

Figura 20. Detalle del paquete "com.example.loader.workers"

Incluye las siguientes **estructuras de datos**:

Clase/Interface	Descripción
ARCapturer	*Clase encargada de realizar capturas de pantalla*
DBCategorySaver	*Clase encargada de almacenar nuevas categorías en la BD*
DBCommentSaver	*Clase encargada de almacenar nuevos comentarios en la BD*
DBModelSaver	*Clase encargada de almacenar nuevos modelos en la BD y de copiar los ficheros físicos correspondientes en disco*
DBModelUpdater	*Clase encargada de actualizar la categoría de los modelos*

8. Estructura del sistema

8.1 Módulos de la aplicación

Para estructurar los contenidos de la app, se utilizan las siguientes **secciones**:

- **Elements**. Permite navegar por los **modelos 3D** del usuario, que se presentan agrupados en **categorías**.

- **Viewer**. Para visualizar un determinado modelo sobre un marcador físico.

- **Gallery**. Presenta un grid de miniaturas con las **capturas de pantalla** realizadas por el usuario.

- **Search**. Incluye un **buscador global de contenidos** sobre los diferentes módulos del programa.

- **Docs**. Contiene **tutoriales** sobre las características y el funcionamiento del programa.

- **Settings**. Posibilita la **configuración** de las diferentes opciones del programa.

8.2 Pantallas de la aplicación

Las principales **unidades de interacción** que incluye el programa se listan a continuación.

1. **Home**. Pantalla principal de la app, que permite acceder al resto de contenidos.

2. **Categories**. Presenta una lista con las diferentes agrupaciones de modelos.

3. **New category**. Permite crear una nueva categoría.

4. **Models**. Presenta una lista con los modelos contenidos dentro de una categoría concreta.

5. **New model**. Permite crear/subir un nuevo modelo.

6. **Model detail**. Incluye la información básica de un determinado modelo.

7. **Viewer**. Muestra el modelo 3D seleccionado sobre un marcador físico mediante realidad aumentada.

8. **Gallery**. Formada por un grid de imágenes en miniatura con las diferentes capturas realizadas por el usuario.

9. **Gallery (detail)**. Muestra una imagen a pantalla completa.

10. **Search**. Incluye un buscador con filtros para refinar los resultados obtenidos.

11. **Search results**. Presenta una lista de resultados relativos a la búsqueda realizada.

12. **Docs**. Incluye compendios de preguntas y respuestas sobre la app.

13. **Settings**. Permite activar o desactivar opciones relativas al funcionamiento del programa.

8.3 Navegación principal

8.3.1 Springboard

Página principal que actúa como lanzadera y permiten consultar el resto de módulos del programa **[9]**.

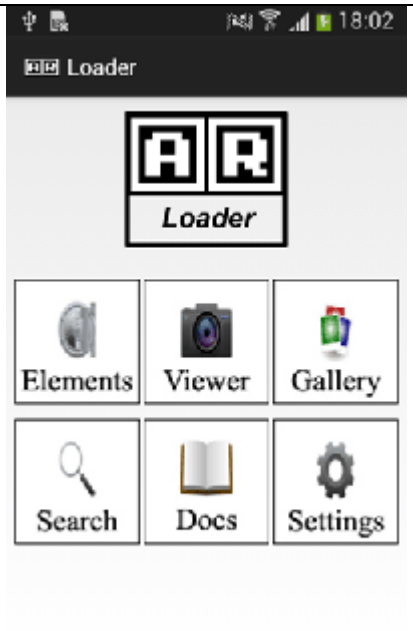

Figura 21. Springboard

8.4 Navegación secundaria

8.4.1 Tablas de datos

Internamente, los principales módulos de la app utilizan **tablas de una única columna**, sin cabecera, para presentar la información al usuario.

Esta estructura permite **combinar elementos de texto y elementos gráficos** en cada entrada, poniendo así el foco de atención del usuario en los contenidos de la app **[9]**.

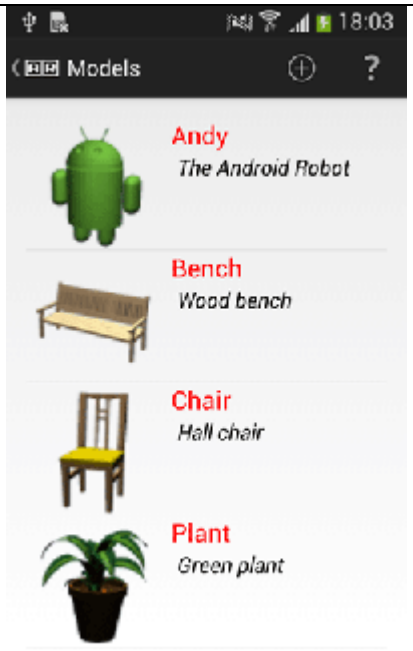

Figura 22. Tablas de datos

8.5 Otros recursos utilizados

8.5.1 Ayuda contextual

Todas las páginas contienen un **icono de ayuda**, situado en la parte derecha de la barra de acciones, que permite acceder a la documentación del programa.

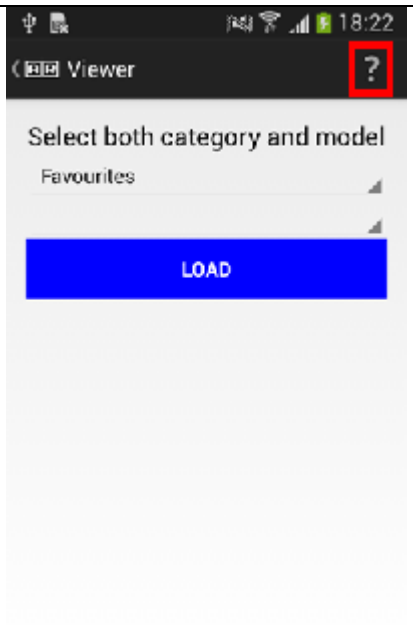

Figura 23. Ayuda contextual

8.5.2 Enlaces de retorno

Se incluye en la parte izquierda de la barra de acciones un **icono para regresar a la pantalla anterior**.

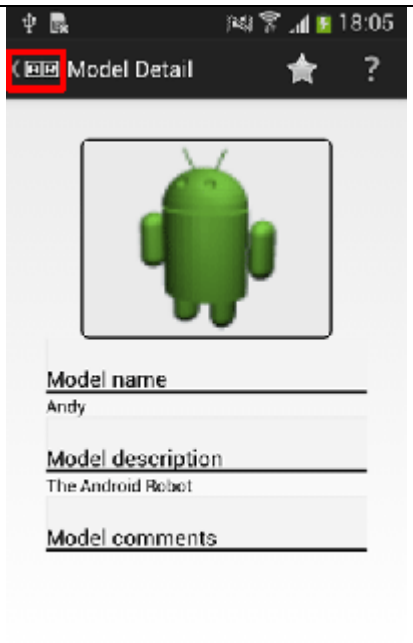

Figura 24. Enlaces de retorno

8.5.3 Foco automático

Las páginas que contienen **formularios de datos** establecen de manera automática el **foco en el primer campo** del mismo.

8.5.4 Hints

Los campos de formulario utilizan hints/**marcas de agua** para indicar al usuario el valor esperado.

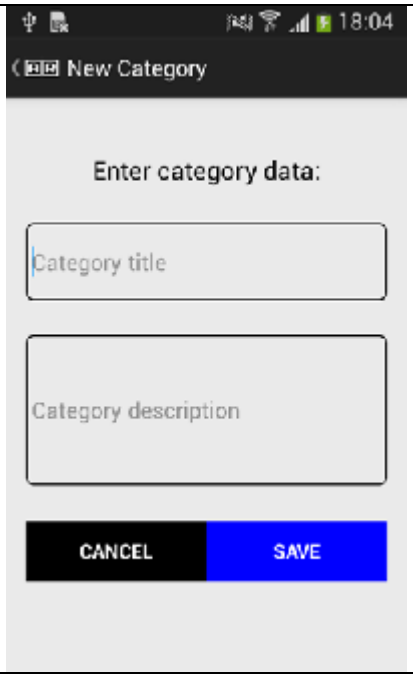

Figura 25. Hints

8.5.5 Textos alternativos

Los controles que componen la interfaz incluyen textos alternativos y descripciones para proporcionar **accesibilidad** al programa.

8.6 Patrones de diseño

8.6.1 Home

Utiliza el patrón de diseño **"Springboard"**.

Características:

- Contiene iconos para lanzar los principales módulos de la app.
- Incluye un widget de preview que indica las novedades y proporciona acceso directo a los contenidos.

Justificación:

- Permite acceder de manera rápida a los elementos de la app.
- Informa de manera visual y dinámica sobre las novedades.
- Prioriza los contenidos de la app.

8.6.2 Categories

Utiliza el patrón de diseño **"Headerless table"**.

Características:

- Utiliza tablas de una columna, sin título en la parte superior.
- Aplica diferentes tamaños de letra en cada fila, para establecer una jerarquía visual.
- Permite la inclusión de imágenes en miniatura.

Justificación:

- Posibilita la rápida búsqueda y selección de un elemento concreto.
- Proporciona un resultado global más rico en contenidos.

8.6.3 New category

Estructurada según un **diseño estándar para formularios**.

Características:

- Resalta el campo activo y las acciones principales mediante efectos.
- Minimiza el número de controles en pantalla.

Justificación:

- Facilita la labor del usuario gracias a los mecanismos incluidos (valores por defecto, marcas de agua, foco, etc.)

8.6.4 Models

Utiliza el patrón de diseño **"Headerless table"**.

8.6.5 New model

Estructurada según un **diseño estándar para formularios**.

8.6.6 Model detail

Utiliza el patrón de diseño **"Action Bar"**.

Características:

- Contiene una barra que incluye acciones relacionadas con el elemento actual.
- Cada acción se presenta de manera gráfica a través de un icono.

Justificación:

- Agrupa todas las posibles interacciones en un único contenedor.

8.6.7 Model viewer

Utiliza el patrón de diseño **"Option Menu"**.

Características:

- Muestra un menú ante la pulsación del botón físico del dispositivo.
- Incluye acciones relacionadas con el elemento actual.
- Cada acción se presenta de manera gráfica a través de un icono.

Justificación:

- Agrupa todas las posibles interacciones en un único contenedor.

8.6.8 Gallery

Utiliza el patrón de diseño **"Gallery"**.

Características:

- Incluye un buscador en la parte superior.
- Presenta un grid con imágenes en miniatura que enlazan al elemento correspondiente.

Justificación:

- Facilita la localización y el acceso a los elementos.
- Prioriza los contenidos de la app.

8.6.9 Gallery detail

Utiliza el patrón de diseño **"Option Menu"**.

8.6.10 Search

Utiliza el patrón de diseño **"Scoped Search"**.

Características:

- Presenta una caja de texto y un icono para lanzar el proceso de manera explícita.
- Contiene una serie de filtros ocultos inicialmente, que se muestran ante la acción del usuario.
- Cada filtro se representa mediante un control que facilita la introducción de datos.
- Se asignan valores por defecto automáticamente.

Justificación:

- Permite realizar una búsqueda estándar o una búsqueda personalizada, ajustándose a las necesidades del usuario.

8.6.11 Search results

Utiliza el patrón de diseño **“Search Results”**.

Características:

- Presenta el número total de resultados relativos a la búsqueda.
- Destaca el texto buscado por el usuario en cada entrada.

Justificación:

- Permite explorar de manera rápida y eficiente los resultados.

8.6.12 Docs

Utiliza el patrón de diseño **“Springboard”**.

8.6.13 FAQ

Utiliza el patrón de diseño **“Collapsible Panels”**.

Características:

- Presenta una serie de paneles interactivos que se pueden plegar/desplegar mediante un click en el título.
- Una vez desplegados, los paneles muestran la información correspondiente.

Justificación:

- Permite acceso puntual a la información, mostrándola únicamente cuando es necesaria.
- Minimiza la cantidad de texto mostrado en pantalla.

8.6.14 How To

Utiliza el patrón de diseño **“Headerless Table”**.

8.6.15 Settings

Estructurada según un **diseño estándar para formularios de preferencias**.

Características:

- Contiene una serie de opciones que permiten configurar el funcionamiento del programa.
- Cada opción se representa mediante un control que facilita la introducción de datos.
- Se asignan valores por defecto automáticamente.

Justificación:

- Permite introducir las preferencias de manera rápida y eficiente.
- Minimiza la posibilidad de errores durante la introducción de datos

9. Conclusiones

9.1 Grado de cumplimiento de los objetivos

El grado de cumplimiento de los objetivos planteados en la fase inicial del proyecto es, en líneas generales, **satisfactorio**. Se ha implementado un sistema, compuesto por una app para la plataforma Android y una base de datos SQLite, que presenta las características descritas en la introducción del presente documento.

Así, la **herramienta** desarrollada permite a los usuarios de dispositivos móviles **visualizar y manipular modelos 3D mediante un entorno de RA** basada en marcadores. Además, incorpora otras funcionalidades destacables, como la utilización de modelos personalizados o la visualización de los diferentes elementos en una galería de imágenes.

A pesar de todo, las **dificultades técnicas** de algunos aspectos del proyecto han acarreado **demoras en la planificación** que, en conjunto, han impedido implementar todos los módulos previstos durante la fase inicial.

Ante esta situación, siempre con el objetivo de cumplir las plazos de entrega establecidos, se ha priorizado el desarrollo de los módulos principales de la app, es decir, aquellos que componen el núcleo de la misma.

En consecuencia, el **módulo "Buscador"**, concebido para localizar elementos de la app, y el **módulo "Ayuda"**, formado por un conjunto de tutoriales para asistir al usuario, finalmente han quedado **fuera del ámbito del presente proyecto**.

9.2 Extensiones del sistema

La aplicación desarrollada permite la substitución de alguno de sus componentes o la **incorporación de nuevas funcionalidades**. Algunas de las posibles extensiones del sistema se listan a continuación:

- **Codificación de los módulos pendientes**. Como se ha comentado en el apartado anterior, los módulos "Search" y "Docs" no se han podido implementar.
- **Definición de marcadores personalizados**. Brindar al usuario la posibilidad de definir el marcador físico a utilizar para el reconocimiento de objetos.
- **Uso de múltiples marcadores simultáneos.** Modificar la lógica del programa para reconocer y mostrar varios elementos al mismo tiempo.
- **Incorporación de mecanismos de interacción natural.** Posibilitar la manipulación del modelo mediante comandos de voz, por ejemplo.

9.3 Valoración personal

El desarrollo del proyecto ha presentado numerosas dificultades, cuya resolución ha requerido la aplicación de diversos conceptos del plan de estudios, como la ingeniería de software, el diseño de interfaces, el desarrollo de aplicaciones o la gestión de proyectos.

Además, la implementación ha obligado a realizar un estudio exhaustivo de las tecnologías utilizadas, permitiendo, en consecuencia, ampliar notablemente los conocimientos sobre las mismas.

Por otra parte, la elaboración del proyecto ha permitido distanciarse de los contenidos básicos del plan académico para explorar otros elementos, como la realidad aumentada o la programación gráfica.

En conclusión, considerando todos los motivos expuestos anteriormente, la **valoración** del proyecto es **positiva**, tanto a nivel académico como a nivel personal.

10. Glosario

10.1 Listado de términos

- **Base de datos**. Repositorio que permite almacenar la información de un sistema mediante un conjunto de tablas o relaciones.
- **Diseño evolutivo**. Metodología de desarrollo de software que propone la implementación de los programas de manera incremental, es decir, añadiendo las diferentes funcionalidades sucesivamente.
- **Java**. Lenguaje de programación orientada a objetos desarrollado en los años 90 por Sun Microsystems, actualmente propiedad de Oracle.
- **Lenguaje Unificado de Modelado**. Notación conceptual que permite modelar las características de un programa aplicando los conceptos de la P.O.O.
- **OpenGL**. Especificación estándar para representar objetos en 2D/3D en el campo de la programación gráfica.
- **OpenGL ES**. Versión de la librería OpenGL optimizada para dispositivos móviles.
- **Programación Orientada a Objetos**. Paradigma de programación basado en el uso de clases y objetos.
- **Realidad Aumentada**. Técnica en el campo de la imagen por computador que permite incorporar elementos virtuales sobre la imagen del mundo real.
- **Realidad Virtual**. Técnica en el campo de la imagen por computador que permite crear entornos totalmente virtuales.
- **Requerimiento funcional**. Funcionalidad que debe incluir de una aplicación de software.
- **Requerimiento no funcional**. Aspecto a tener en cuenta durante el desarrollo de una aplicación de software.
- **SQLite**. Motor de base de datos optimizado para plataformas con escasos recursos, como, por ejemplo, los dispositivos móviles.
- **Structured Query Language**. Lenguaje de datos que permite administrar y manipular tanto las estructuras como el contenido de una base de datos.
- **Usabilidad**. Grado de facilidad de uso y/o aprendizaje ofrecido por una aplicación.
- **Wavefront OBJ**. Formato estándar para la definición de modelos estáticos en 3D.

11. Bibliografía

11.1 Bibliografía básica

[1] Mullen, Tony. **Realidad Aumentada. Crea tus Propias Aplicaciones.** 1ª Edición. Anaya Multimedia, 2012.

[2] Sommerville, Ian. **Ingeniería del Software.** 7ª Edición. Pearson Educación, 2005.

[3] Furth, Borko. **Handbook of Augmented Reality.** 1ª Edición. Springer, 2011.

[4] Fitzgerald, Elisabeth et al. **Augmented Reality and Mobile Learning.** 1ª Edición. The Open University, 2012.

[5] Kipper, Gregory. **Augmented Reality. An Emerging Technologies Guite to AR.** 1ª Edición. Elsevier, 2013.

[6] Craig, Allan et al. **Augmented Reality Software Kits for Smart Phones.** 1ª Edición. University of Illinois, 2011.

[7] Zukowski, John. **JAVA 2. J2SE 1.4.** 1ª Edición. Anaya Multimedia, 2003.

[8] MacWilliams, Asa et al. **Design Patterns for Augmented Reality Systems.** 1ª Edición. Institut für Informatik.

[9] Neil, Theresa. **Mobile Design Pattern Gallery.** 1ª Edición. O'Reilly, 2012.

11.2 Bibliografía electrónica complementaria

[e1] Verizon Wireless (2014). **Augmented Reality Apps. The Wave of the Future**. Disponible en:
https://insidersguide.vzw.com/entertainment/augmented-reality-apps/
(Junio de 2014)

[e2] Wikimedia Project (2014). **Wavefront .obj file**. Disponible en:
http://en.wikipedia.org/wiki/Wavefront_.obj_file
(Junio de 2014)

[e3] Apple Inc (2014). **Wikitude on the App Store**. Disponible en:
https://itunes.apple.com/en/app/wikitude/id329731243?mt=8
(Junio de 2014)

[e4] Google (2014). **Beyond Reality. Android Apps on Google Play**. Disponible en:
https://play.google.com/store/apps/details?id=com.BeyondReality.ARDemo
(Junio de 2014)

[e5] Google (2014). **Augmented Class! Android Apps on Google Play**. Disponible en:

https://play.google.com/store/apps/details?id=com.CreativiTIC.AugmentedClass
(Junio de 2014)

[e6] Google (2014). **DroidShooting. Android Apps on Google Play**. Disponible en:
https://play.google.com/store/apps/details?id=jp.co.questcom.droidshooting
(Junio de 2014)

[e7] BlackBerry (2014). **TAT The Astonishing Tribe**. Disponible en:
http://www.tat.se/
(Junio de 2014)

[e8] Apple Inc (2014). **Hallux Angles on the App Store**. Disponible en:
https://itunes.apple.com/gb/app/hallux-angles/id398338105?mt=8
(Junio de 2014)

[e9] Human Interface Technology Laboratory (2014). **ARToolkit Home Page.** Disponible en:
http://www.hitl.washington.edu/artoolkit/
(Junio de 2014)

[e10] ARToolworks (2014). **Mobile. ARToolworks.** Disponible en:
https://www.artoolworks.com/products/mobile/
(Junio de 2014)

[e11] NYARToolkit Project (2014). **NYARToolkit Project.** Disponible en:
http://nyatla.jp/nyartoolkit/wp/
(Junio de 2014)

[e12] Qualcomm (2013). **Qualcomm Vuforia.** Disponible en:
http://www.qualcomm.com/solutions/augmented-reality
(Junio de 2014)

[e13] BuildAR (2014). **BuildAR Pro.** Disponible en:
http://www.buildar.co.nz/
(Junio de 2014)

[e14] Georgia Institute of Technology (2014). **DART Homepage.** Disponible en:
http://ael.gatech.edu/dart/
(Junio de 2014)

[e15] ARMonkey Kit (2014). **ARMonkey Kit.** Disponible en:
http://armonkeykit.wordpress.com/
(Junio de 2014)

[e16] metaio GbmH (2014). **Metaio Overview**. Disponible en:
http://www.metaio.com/products/sdk/
(Junio de 2014)

[e17] Total Inmersion (2013). **Total Inmersion Community Website**. Disponible en:
https://community.t-immersion.com/
(Junio de 2014)

[e18] Google (2014). **AndAR Model Viewer. Android Apps on Google Play**. Disponible en:
https://play.google.com/store/apps/details?id=edu.dhbw.andarmodelviewer
(Junio de 2014)

[e19] Google (2014). **AR Model Viewer. Android Apps on Google Play**. Disponible en:
https://play.google.com/store/apps/details?id=com.sujewan.mv
(Junio de 2014)

[e20] Google (2014). **3D Loader Augmented Reality. Android Apps on Google Play**. Disponible en:
https://play.google.com/store/apps/details?id=com.Fractal.LoaderOBJAR
(Junio de 2014)

[e21] Apple Inc (2014). **Augmented 3DS File Viewer on the App Store on Itunes**. Disponible en:
https://itunes.apple.com/us/app/augmented-3ds-file-viewer/id622229455?mt=8
(Junio de 2014)

[e22] Google Inc (2014). **Android Developers**. Disponible en:
http://developer.android.com
(Junio de 2014)

Anexos

A1. Listado de recursos gráficos utilizados

Se incluye el listado de **recursos visuales** en la elaboración del proyecto.

Recurso	Autor	Procedencia	Status
Usuario focal	---	*Free Stock Photos*	*Free*
Fotomontaje	---	*Dreamstime*	*Copyright*
Gadgets	---	*Free Stock Photos*	*Free*
Oficina	---	*Free Stock Photos*	*Copyright*
Logo Apple	---	*Dreamstime*	*Copyright*
Innovación	---	*IEEE*	*Copyright*
Clips	---	*Free Stock Photos*	*Free*
Usuario secundario	---	*Free Stock Photos*	*Free*
Generación digital	---	*Dreamstime*	*Copyright*
Redes sociales	---	*Dreamstime*	*Copyright*
Logo Android	---	*Dreamstime*	*Copyright*
Libros	---	*Free Stock Photos*	*Free*
Videojuegos	---	*Free Stock Photos*	*Free*
Loading	---	*Dreamstime*	*Copyright*
Android Robot	---	*Google*	*Copyright*
Elements Icon	---	*Prezi.com*	*Copyright*
Viewer Icon	---	*Prezi.com*	*Copyright*
Gallery Icon	---	*Prezi.com*	*Copyright*
Search Icon	---	*Prezi.com*	*Copyright*
Docs Icon	---	*Prezi.com*	*Copyright*
Settings Icon	---	*Prezi.com*	*Copyright*
Models Icon	---	*Prezi.com*	*Copyright*
Category Icon	---	*Prezi.com*	*Copyright*
FAQ Icon	---	*Prezi.com*	*Copyright*
How To Icon	---	*Prezi.com*	*Copyright*
Bulb Icon	---	*Prezi.com*	*Copyright*
Help Module Picture	---	*United States Nuclear Regulatory Comission*	*Copyright*
3D models	---	*Turbosquid.com*	*Free*

A2. Listado de recursos de SW utilizados

Para finalizar, se indica también el listado de **recursos informáticos** utilizados en la implementación del proyecto.

Recurso	Autor	Procedencia	Status
ua.com.vassiliev.androidfilebrowser	*Alexey Vassiliev*	*GitHub*	*Apache License*

Firmado: Bernat Gómez Garcia