



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escuela Técnica Superior
de Ingeniería Informática

Proyecto Fin de Carrera
Ingeniería en Informática

UPV-MobARGuide

Aplicación Android de Realidad Aumentada para guía
interactiva de la UPV orientada a móviles

Autor:

Fernando Marcos Guillem Ortiz

Director:

José Antonio Lozano Quilis

Valencia, Febrero de 2012

TÍTULO:	UPV-MobARGuide. Aplicación Android de Realidad Aumentada para guía interactiva de la UPV orientada a móviles
AUTOR:	Fernando Marcos Guillem Ortiz
DIRECTOR:	José Antonio Lozano Quilis
DEPARTAMENTO:	DSIC

Resumen

Las aplicaciones móviles que usan geolocalización nos permiten obtener información detallada y filtrada sobre los lugares que nos rodean de la forma más rápida posible. La Realidad Aumentada unida a la geolocalización, no hace sino mejorar la experiencia de usuario a la hora de obtener dicha información.

La *Universitat Politècnica de València* (UPV) aloja en sus servidores gran cantidad de información sobre la misma, su distribución en el espacio, en edificios, departamentos, institutos, escuelas y una infinidad de servicios, cada uno con unos datos de contacto e información asociada. Toda esta información es accesible desde la web de la universidad, aunque esto no quiere decir que sea fácil ni rápida de obtener. Así pues, una aplicación móvil con realidad aumentada en la UPV facilitaría el acceso a esta información. Cada lugar a destacar tendrá un punto de información asociado y geolocalizado en el mismo sitio, de manera que al mirar a dicho punto, un usuario pueda obtener los datos relativos a ese lugar, lo cual es mucho más rápido y sencillo que la búsqueda en la web.

Hasta aquí, la aplicación se limita a mostrar información de una manera diferente. Sin embargo, también se le ha dotado de interactividad. La aplicación permite que los usuarios compartan información y creen sus propios puntos de interés, convirtiéndola por tanto en un medio social. Y consiguiendo así reunir dos conceptos ampliamente estudiados hoy en día, la realidad aumentada basada en geolocalización y las redes sociales, lo cual ya empieza a despegar como una nueva tendencia en el mundo tecnológico.

Palabras clave: realidad aumentada, geolocalización, interactividad, redes sociales, Android.

Resum

Les aplicacions mòbils que fan us de la geolocalització ens permeten obtenir informació detallada i filtrada dels llocs que ens envolten de la manera més ràpida possible. La Realitat Augmentada unida a la geolocalització, millora l'experiència de l'usuari quan tracta d'obtindre dita informació.

La *Universitat Politècnica de València* (UPV) conté als seus sevidors gran quantitat d'informació sobre si mateixa, la distribució de l'espai, dels edificis, departaments, instituts, escoles i una infinitat de serveis, cadascun amb les seues dades de contacte i informació associada. Tota aquesta informació es accessible des de la web de la universitat, encara que això no vol dir que siga fàcil ni ràpida d'obtindre. Així doncs, una aplicació mòbil amb realitat augmentada a la UPV permetria un accés més fàcil a aquesta informació. Cada emplaçament destacable tindrà un punt d'informació associat i geolocalitzat al mateix lloc, de manera que quan es mire a aquest punt, un usuari puga obtenir les dades relatives al emplaçament, el qual es molt més ràpid i senzill que cercar la web.

Fins ací, l'aplicació només es limita a mostrar l'informació d'una manera diferent. No obstant, també ha sigut dotada d'interactivitat. L'aplicació permet que els usuaris comparteixin informació i creen els seus propis punts d'interés, convertint-la per tant en un mitjà social. I aconseguint així, agabellar dos conceptes ampliament estudiats avui dia, la realitat augmentada basada en geolocalització i les xarxes socials, la qual cosa ja comença a prendre forma com una nova tendència al món tecnològic.

Paraules clau: realitat augmentada, geolocalització, interactivitat, xarxes socials, Android.

Abstract

Geolocation mobile apps allow us to quickly obtain detailed and filtered information of the surroundings. When adding Augmented Reality to the equation, the user experience is enhanced and hence, the process of obtaining such information.

The *Universitat Politècnica de València* (UPV) hosts a huge amount of data in its servers about itself, its space distribution, buildings, departments, institutes, schools and a great number of services, together with its metadata. All this information is accessible from the university web site. Although, it does not mean this process is easy nor fast. Thus, an augmented reality mobile app of the UPV would bring an information access easement. Each remarkable location shall have an associated point of information geolocalized within it, in a way that when a user looks at it, he can easily get all the related information to that point, which is way faster and simpler than the web search.

But this app does not just display the information that already exists in a different fashion, it has also been gifted with user interactivity. It allows users to share information and create their own points of interest, turning it into a social network. And therefore, it puts together two concepts widely studied nowadays, geolocation based augmented reality and social networks, which turns into a new field that is taking off and is becoming a new trend in the tech world.

Keywords: augmented reality, geolocalization, interactivity, social networks, Android.

Agradecimientos

Este proyecto se ha beneficiado de la gran guía que supone tener a José Antonio Lozano como director. Sus ideas disruptivas y su experiencia han sido de gran ayuda.

Además, quisiera agradecer a mis amigos el haberme apoyado durante estos meses de duro trabajo.

Por último, agradecer a mi familia su soporte. En especial a mis padres, ya que sin su gran esfuerzo por darme una buena educación hoy no podría escribir estas líneas.

Índice general

1. Introducción	5
1.1. Idea original	5
1.2. Objetivos	6
1.3. Descripción general	8
1.4. Estructura del documento	11
2. Antecedentes	12
2.1. Realidad Aumentada	12
2.2. Sistema operativo Android	28
2.3. Redes sociales	34
2.4. Aplicaciones de RA existentes para Android	35
3. Análisis	40
3.1. Metodología	40
3.2. Descripción detallada	41
3.3. Flujos de información	49
3.4. Herramientas y librerías	53
4. Diseño e implementación	57
4.1. Proceso de desarrollo	57
4.2. Arquitectura del servidor	57
4.3. Arquitectura de la aplicación	62
4.4. Interfaz de usuario	67
5. Evaluación y resultados	71
5.1. Casos de uso llevados a cabo	71
5.2. Evaluación del sistema	77
6. Conclusiones y trabajos futuros	79
6.1. Vicisitudes	79

ÍNDICE GENERAL	2
6.2. Conclusiones	80
6.3. Trabajos futuros	80
Bibliografía	85

Índice de figuras

1.1. Boceto de la vista en realidad aumentada.	8
1.2. Boceto de la vista del mapa de situación.	10
2.1. Continuo de Milgram.	13
2.2. Ejemplos de virtualidad aumentada.	13
2.3. Ejemplos de realidad aumentada.	14
2.4. Periscopio de Pratt. Imagen cortesía de la oficina de patentes de EEUU.	14
2.6. Sensorama.	15
2.5. Simulador de vuelo de Edward Link.	15
2.7. HMD desarrollado por Sutherland.	16
2.8. Marcador de realidad aumentada.	17
2.9. ARQuake.	17
2.10. HMD's opacos. Fotos de General Reality Corp.	20
2.11. HMD's semitransparentes.	20
2.12. Escenas del juego Parallel Kingdom.	24
2.13. Proyecto BMW Augmented Reality.	25
2.14. Aplicaciones de RA en la industria militar.	25
2.15. Aplicaciones sobre turismo.	26
2.16. Aplicaciones en el sector automovilístico.	26
2.17. Aplicaciones de RA en Marketing y Publicidad.	27
2.18. Señalización del fuera de juego en un partido de fútbol.	28
2.19. Arquitectura del SO Android.	29
2.20. Ciclo de vida de la componente Activity.	32
2.21. Cuota de mercado de los SO para dispositivos móviles. Imagen: NPD Group.	33
2.22. Ventas según tipos de teléfono. Imagen: NPD Group.	33
2.23. Imágenes de la aplicación Layar.	36
2.24. Imágenes del cielo con información sobre las estrellas en Star Chart.	37
2.25. Imágenes de la aplicación para iOS de la Universidad de West Virginia.	39

3.1. Menú de opciones de Android.	43
3.2. Perspectivas de la UPV en Google Maps.	45
3.3. Cover Flow de Apple en iTunes.	47
3.4. Diagrama de flujo de la aplicación.	50
3.5. Ejemplo de uso de la pila de Android número 1.	51
3.6. Ejemplos de uso de la pila de Android número 2.	51
3.7. Android-ActionBar.	55
3.8. Android-MapViewBalloons.	56
4.1. Topología de la red.	58
4.2. Diagrama Entidad-Relación de la base de datos.	60
4.3. Diagrama de clases en UML.	62
4.4. Dependencias del proyecto.	63
4.5. Árbol de directorios del proyecto.	64
4.6. Diagrama del componente menú principal.	65
4.7. Diagrama del componente vista en RA.	66
4.8. Diagrama del componente vista mapa.	66
4.9. Diagrama del componente perfil de usuario.	67
4.10. Bocetos de la interfaz gráfica 1.	68
4.11. Bocetos de la interfaz gráfica 2.	69
4.12. Bocetos de la interfaz gráfica 3.	70
5.1. Esquema del inicio de sesión.	72
5.2. Capturas del inicio de sesión.	73
5.3. Capturas de la realidad aumentada.	74
5.4. Capturas del mapa 1.	74
5.5. Capturas del mapa 2.	75
5.6. Capturas de la creación de POIs 1.	76
5.7. Capturas de la creación de POIs 2.	77

Capítulo 1

Introducción

*Un mar calmado no
hace buenos marineros
~ Proverbio inglés ~*

1.1. Idea original

Los tiempos en los que se hablaba de la realidad aumentada en dispositivos móviles como algo que estaría presente en el futuro cercano quedaron atrás. Porque ese futuro cercano ya es presente, y ya está al alcance de cualquiera el poder jugar con aplicaciones de realidad aumentada, ya sean basadas en marcas, en geolocalización o en reconocimiento de formas. Esto, unido al boom que ha habido en el desarrollo de aplicaciones en el que es más fácil que nunca desarrollar y vender un producto de software, ha hecho que la oferta haya crecido de forma considerable.

Uno de los efectos colaterales ha sido que cada vez más instituciones quieren tener un producto a su medida. Así pues, ya existen muchas aplicaciones de grandes corporaciones desarrolladas por terceros, en las que se puede acceder a todo tipo de servicios e información de dicha corporación. Y si hablamos de realidad aumentada basada en geolocalización, que es el tema que nos atañe, o geolocalización sin más, no sólo las grandes compañías entran en juego. El pequeño comercio también está presente en muchas aplicaciones, entrando así en un nuevo mundo en el que promocionarse y estar más cerca del consumidor.

Las instituciones educativas no son una excepción, y ya hay numerosas universidades de todo el mundo que se han subido al carro de las aplicaciones móviles, como por ejemplo la Universidad de West Virginia [48], que tiene su propia aplicación de realidad aumentada hecha con la API de Layar [2]. Trasladar los contenidos que actualmente se alojan en webs de universidades a otras plataformas como los móviles, ya sea en realidad aumentada o con otra tecnología, no es más que el siguiente paso en la evolución de los medios de información. Ya no es suficiente con tener una web en internet, también hay que estar en las redes sociales y en los dispositivos móviles, o también llamados *smartphones*.

A día de hoy, la UPV está siguiendo las pautas que marca la tecnología y ya se encuentra en redes sociales como *Facebook*, *LinkedIn* o *Twitter*. Pero todavía no existe ninguna aplicación móvil sobre ella, o

al menos ninguna que hayamos podido encontrar. Y es por eso que consideramos una buena idea el realizar un prototipo de aplicación móvil para esta universidad. Así pues, usar tecnología de vanguardia como es la realidad aumentada parece el enfoque más apropiado para este trabajo sobre el campus virtual. De esta forma, se podría identificar cualquier parte del campus sólo enfocar la cámara del teléfono hacia ese sitio, y obtener toda la información pertinente que puede ser de interés para el usuario. Como complemento a esta forma de ver la información, también existe la posibilidad de verla sobre un mapa, lo cual es un enfoque más clásico pero no por ello menos funcional.

Finalmente, puesto que la intención del proyecto es explorar nuevos campos de acción en la universidad, es preciso dotar a la aplicación de interactividad. De tal manera que los usuarios se integren en la aplicación como un medio social en el que intercambiar información, hacer propuestas o incluso hacer negocios. Este es un elemento diferenciador con respecto a otras aplicaciones del mismo corte pertenecientes a otras universidades y que puede suponer un incremento en el uso de la misma, yendo más allá de saber ante qué edificio se está o dónde está el departamento al que se debe ir.

1.2. Objetivos

Debido a las singularidades de este tipo de proyecto, el B, en el que el alumno trabaja junto al director en la elección de un tema de proyecto y éste se desarrolla desde cero (al contrario que los proyectos en los que el alumno ya elige una propuesta definida y se tiene que limitar a cumplirla), podemos aglutinar los objetivos en dos tipos.

Por un lado los del proyecto a desarrollar, en este caso una aplicación móvil, que tendrá que tener unos objetivos definidos que deba cumplir y se puedan verificar cuando esté finalizada.

Y por otro lado los objetivos del proyectando, por los cuáles ha pensado que emprender este tipo de proyecto sería bueno como parte de su formación académica. Dado que la confección de la idea se hace entre el alumno y el tutor, se pueden intentar incluir varios campos en un mismo proyecto en los que el alumno desee formarse puesto que considere que pueden ser útiles para su carrera.

1.2.1. Objetivos del proyectando

La elección de un proyecto final de carrera puede ser algo fácil o complicado dependiendo de las expectativas que se pongan en dicho trabajo. Si este trabajo sólo supone un trámite a la hora de conseguir el título para el alumno, entonces la decisión no debería ser muy complicada, escoger un proyecto sencillo. Si por el contrario el alumno quiere adentrarse en algún campo específico, adquirir nuevos conocimientos o explorar nuevas tecnologías, entonces la elección es más complicada. Hay una gran cantidad de proyectos ofertados por los departamentos de la universidad. Pero encontrar uno idóneo que cumpla tus expectativas, y que además no haya sido solicitado por nadie antes no es una tarea fácil.

En mi caso, no hallé satisfacción en ninguno de los proyectos ofrecidos por los departamentos, bien porque no acaparaban todas las ramas en las que quería adentrarme, o porque eran muy específicos en

cierto campo, o porque eran muy simples, o me gustaban pero ya estaban asignados, o por muchas otras razones que conformarían un largo etcétera . Considero que es importante recalcar este hecho, puesto que define la naturaleza del propio proyecto, en el que los requisitos y los objetivos los he definido yo mismo. Contando eso sí, con la inestimable ayuda y consejo de mi director J.A. Lozano.

La intención inicial era desarrollar algo que se pudiese manipular con las manos, que fuese interactivo y que suscitase curiosidad al potencial usuario. En este filtro podíamos incluir todo tipo de nuevas tecnologías que estuviesen de moda como mandos de la *Wii*, el *Kinect* de Microsoft, dispositivos móviles con cualquier sistema operativo, dispositivos ópticos, o gafas de realidad virtual entre muchos otros. El campo al que se aplicase dicha tecnología era incierto, pero desde luego el objetivo era involucrar al usuario.

Al final se opta por los dispositivos móviles, puesto que en el momento de esta decisión las aplicaciones móviles son un boom dentro del mundo del desarrollo de software y son uno de los elementos con un entorno de desarrollo más fáciles de conseguir. Además, tengo un gran interés en el desarrollo de aplicaciones móviles y es un mundo en el que quiero adentrarme. Y en esta plataforma se pueden desarrollar los campos que más interés me suscitan en el momento, que son:

- **Realidad aumentada:** Ya existen en el mercado diversas aplicaciones de realidad aumentada que usan la cámara del teléfono para este fin.
- **Geolocalización:** La mayoría de teléfonos actuales disponen de conexión con GPS que les permite obtener datos geográficos de su posición.
- **Redes sociales:** Son la gran moda de los últimos tiempos y todo pasa por ellas. Añadir una componente social al proyecto le dará un salto cualitativo importante.

Con todo esto, surge la idea de hacer una aplicación móvil de realidad aumentada geolocalizada sobre la UPV, en la que haya una componente social. Esto además, obliga a utilizar bases de datos en un servidor remoto que sirvan la información a la aplicación, con lo cuál se añade otro elemento nuevo sobre el que aprender a la lista.

1.2.2. Objetivos de la aplicación

El objetivo principal de la aplicación es la información. No sólo la información que desde un principio proporciona la base de datos de la aplicación, sino también el flujo de información que puede crearse gracias a la colaboración de los usuarios. Puesto que se usa realidad aumentada, el acceso a la información relacionada con un punto será mucho más rápido que su búsqueda en la web, ya que sólo hay que enfocar el teléfono a dicho punto.

Esta aplicación está destinada a dos tipos de personas. Por un lado, las personas familiarizadas con el entorno de la UPV, a los cuales puede que la información básica proporcionada por la aplicación les

sea familiar. Estas personas son las que principalmente llenarán la aplicación con nueva información que enriquecerá la visita de futuros usuarios. El objetivo entonces, es dotar a todos estos usuarios con la capacidad de crear contenido que podrá ser visualizado por el resto de la comunidad.

Por otro lado, la UPV recibe numerosos visitantes durante todo el año. Entre ellos podemos encontrar: ponentes de congresos, estudiantes de intercambio o estudiantes de instituto que van de visita informativa. El conocimiento de esta gente sobre la universidad será a priori reducido, y la información inicial que ofrece la aplicación por defecto sobre los distintos edificios, escuelas, departamentos, institutos y servicios les puede ser muy útil para orientarse y encontrar lo que buscan durante su visita, de forma que ésta sea mucho más amigable y dinámica. Estos usuarios sin embargo, al no tener un perfil dentro del sistema de la universidad, no se les permitirá acceder a la información creada por otros miembros de la comunidad, puesto que ellos no lo son.

1.3. Descripción general

La aplicación a desarrollar ofrecerá una vista en realidad aumentada donde se reconocerán las infraestructuras y establecimientos de la universidad, de forma que al enfocar la cámara del dispositivo hacia dicha estructura, el software la reconozca y muestre su información asociada, como se muestra en el boceto¹ de la figura 1.1.



Figura 1.1: Boceto de la vista en realidad aumentada.

Este reconocimiento se hará por geolocalización. Se dispondrá de una base de datos en un servidor remoto donde será almacenada toda la información con sus correspondientes coordenadas geográficas. Para tal fin, es necesario el diseño e implementación de dicha base de datos y de dicho servidor.

La información de cada objeto se concentrará en puntos de interés, POI de ahora en adelante. Estos estarán divididos por tipos en diferentes capas:

- **Escuelas:** Contendrá todas las escuelas que se encuentran en el campus de Vera, 11, con su respectiva información y contacto.

¹Nótese que la imagen no es más que un esbozo inicial antes de iniciar el desarrollo, y que en ningún caso tiene por qué ser igual al resultado final.

- **Edificios:** Infraestructuras con su nomenclatura oficial, de la 1A a la 9F. Al ser estructuras no tienen ningún tipo de información de contacto. Pero sí se describe qué alojan en su interior, ya sean escuelas, departamentos, institutos o servicios.
- **Departamentos:** Los 42 departamentos que actualmente hay en el campus de Vera. También con su información y contacto.
- **Institutos:** Los 15 que hay en la actualidad con sus respectivos datos.
- **Servicios:** Los más de 50 servicios que se encuentran en la UPV, también con sus datos.
- **Comunidad UPV:** En esta capa tendrán cabida todos los POI creados por los usuarios y que sólo serán accesibles por usuarios registrados.

Mediante el uso del GPS del dispositivo se obtendrá la posición del usuario y mediante el uso de la brújula se obtendrá la dirección en la que enfoca la cámara del dispositivo. Y por tanto, se podrán ubicar correctamente el resto de datos geoespaciales con respecto a la posición del usuario. Esta posición será actualizada periódicamente con el objetivo de que el funcionamiento sea preciso incluso si se está en movimiento.

Esto será de especial utilidad en el caso en el que el usuario esté perdido y necesite encontrar un punto en concreto del campus. En este caso, simplemente siguiendo la dirección en la que se encuentra el POI buscado se llegará al destino, ya que también es proporcionada la distancia que hay entre el usuario y cada POI en metros, junto con un radar con la situación de los puntos cercanos.

Cada uno de los mencionados puntos podrá ser examinado haciendo click sobre él, lo cual hará que la información asociada se muestre en la pantalla. Si el POI dispone de opciones de contacto, será posible:

- Llamar al teléfono de contacto, de manera que al realizar esta acción se abra la aplicación teléfono con el número ya marcado.
- Enviar un e-mail. En este caso se abrirá el gestor de correo predefinido por el sistema o una ventana que nos permita elegir uno si no hay uno ya predefinido, y el campo del destinatario ya estará rellenado con el e-mail del POI.
- Visitar la web que figura en la información del POI. La aplicación abrirá el navegador predefinido o una ventana con las opciones a elegir si no hay uno por defecto, y irá directamente a la URL de la web del punto.

1.3.1. Mapa

En cualquier momento se podrá acceder a un mapa donde se indicará la situación del usuario y la situación de los POI que tiene alrededor, como se muestra en el boceto² de la figura 1.2.

²Nótese que la imagen no es más que un esbozo inicial antes de iniciar el desarrollo, y que en ningún caso tiene por qué ser igual al resultado final.



Figura 1.2: Boceto de la vista del mapa de situación.

Será posible hacer zoom sobre un punto del mapa y inspeccionar los puntos de interés al igual que en la vista con realidad aumentada, ofreciendo las mismas opciones de contacto que en ésta.

En este caso también se puede usar esta vista como una guía para llegar a un punto en concreto, ya que la posición del usuario también aparecerá dibujada y será actualizada en pequeños intervalos de tiempo para que el usuario no pierda la perspectiva de su situación.

1.3.2. Comunidad UPV

Esta capa será el medio social en el que los usuarios podrán comunicarse y publicar distintas informaciones. A los POI que decidan crear se les podrá adjuntar una foto, que será mostrada cuando se inspeccione tal POI en alguna de las vistas de la aplicación. Para ello se deberá rellenar un formulario con los distintos campos de información, seleccionar una ubicación para el punto en el mapa, y adjuntar una imagen si se considera necesario.

A la hora de elegir la ubicación de un punto, es conveniente pensar en cuál es el sitio propicio para situarlo:

- Ejemplo 1: Enfocar el edificio en el que se encuentra un departamento, i.e.: DSIC, permitirá ver ofertas de becas ofrecidas por dicho departamento.
- Ejemplo 2: Enfocar el edificio de una escuela, i.e.: ETSINF, permitirá ver ofertas de PFC, de conferencias o información de acontecimientos pertenecientes a dicha escuela.
- Ejemplo 3: Enfocar a un puesto de bicicletas permitirá ver notas de personas interesadas en comprar, vender o intercambiar una bicicleta.
- Ejemplo 4: Enfocar pisos de los alrededores permitirá ver ofertas de pisos compartidos de estudiantes, geolocalizados en el propio lugar del piso.
- Ejemplo 5: Enfocar la biblioteca o la librería, permitirá ver ofertas de compra-venta de libros y otro material didáctico.

- Ejemplo 6: Enfocar una cafetería permitirá ver puntos de contacto para la compra de entradas a conciertos o fiestas, ya que normalmente éste es el lugar en cuyas paredes cuelgan ese tipo de anuncios.

Por último, puesto que dejar que el usuario cree información sin control puede desembocar en un colapso del sistema, también deberá rellenar una fecha a partir de la cuál el POI dejará de ser válido y por tanto de ser mostrado al resto de la comunidad. Si tal campo no es rellenado, el sistema asignará un periodo de tres meses desde la fecha de creación. Además, el número de puntos que un mismo usuario pueda crear simultáneamente estará limitado.

1.4. Estructura del documento

Esta sección pretende describir los contenidos que se encontrarán en ese documento. Está dividido en seis capítulos, que juntos forman una visión completa del desarrollo del proyecto y su entorno. El primer capítulo, en el que se encuentra esta sección, otorga una perspectiva general del proyecto. Da información en términos generales sobre la descripción y los objetivos del proyecto.

El segundo capítulo relata el estado de la situación actual en el campo en el que se desarrolla el proyecto, de las tecnologías ya existentes en las que está basado. Lo cual da pie al capítulo siguiente, donde se analizan todas las opciones de desarrollo y se especifica qué se va a hacer y cómo, con qué herramientas.

Una vez especificados los detalles del proyecto claramente y las herramientas y procesos a seguir, se pasa al diseño y la implementación. El cuarto capítulo hace hincapié en todos los puntos clave del diseño y la implementación, desde el servidor hasta la interfaz de usuario de la aplicación.

Pero el desarrollo del proyecto no acaba con la implementación, hay que evaluarlo. Por ello en el capítulo de resultados se exponen las pruebas realizadas y los resultados obtenidos, adjudicando una evaluación a los mismos. Y basadas en esos resultados se pueden por fin obtener conclusiones, que forman parte del último capítulo, el sexto. También figuran en este capítulo las vicisitudes del proyecto y los posibles trabajos futuros relacionados.

Capítulo 2

Antecedentes

2.1. Realidad Aumentada

Para comprender el concepto de realidad aumentada, primero es necesario clarificar el concepto de realidad virtual, ya que ésta es anterior y uno de los pilares de la realidad aumentada. Así pues, según Burdea et al. en [10] la realidad virtual es una simulación en la cual se usan gráficos por computador para crear un entorno realista. Además, este mundo sintético no es estático, sino que responde a las acciones del usuario. Es decir, existe una interactividad en tiempo real, lo cual es un punto clave en esta tecnología.

La realidad aumentada aparece más tarde dentro del mismo campo de estudio, aunque en principio sólo se la asocia al uso de *Head Mounted Displays* (HMD), que son dispositivos que se montan en la cabeza del usuario, obligándolo a mirar a una pantalla que acapara todo su campo visión. De esta manera se facilita la inmersión en mundos virtuales. Estas son algunas de las definiciones:

- Janin et al. [22] definen la realidad aumentada como un HMD que permite al usuario ver su entorno real con una superposición de gráficos sobre el mismo.
- Según Beaudouin-Lafon [8], el mundo real no puede ser reproducido con exactitud por medio de las técnicas de computación, por lo que la realidad aumentada se basa en el mundo real para construir escenarios con añadidos gráficos generados por computador.

Como éstas, existen otras definiciones basadas en el uso de HMD, y otras más generalistas. Pero hay una que cabe destacar, ya que es referenciada a menudo en gran cantidad de artículos y libros. Es la definición de las realidades en el **Continuo del Milgram** [26][27]. Como se puede observar en la figura 2.1, es posible establecer una clasificación del tipo de realidad dependiendo de la cantidad de imagen sintética de la que se dispone, teniendo como extremos el mundo completamente real y el mundo completamente sintético.

Dejando a parte los extremos, queda un espacio de realidad mixta en el que la imagen real y la sintética coexisten en el entorno. Dentro de esta realidad se pueden definir dos sub-realidades que es

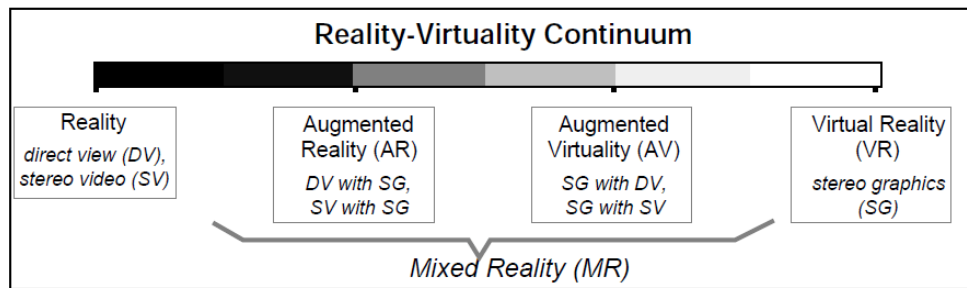
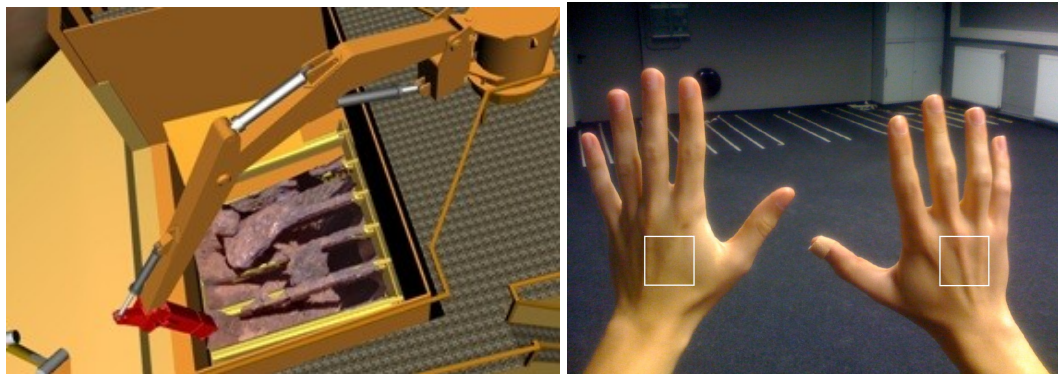


Figura 2.1: Continuo de Milgram.

importante diferenciar:

- **Realidad Aumentada**, que es la que nos ocupa y se define como entorno real al que se le añade imagen sintética con el fin de mejorarlo.
- **Virtualidad aumentada**, que en contraposición a la anterior, se define como un entorno principalmente virtual que puede ser aumentado con objetos o imágenes reales, como se puede apreciar en la figura 2.2.



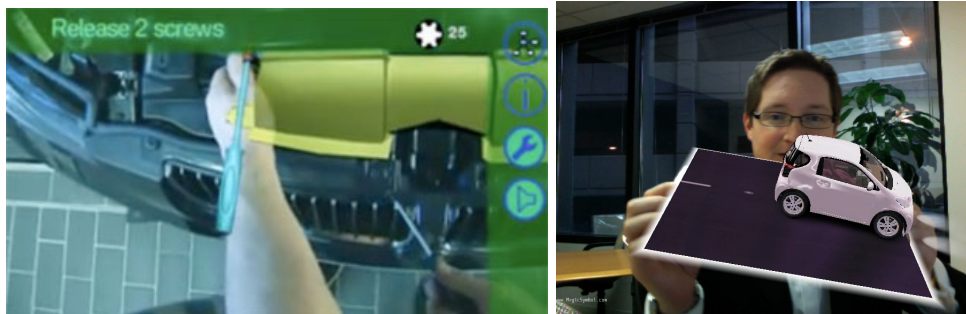
(a) Proyecto Rockbreaker [43]

(b) Manos reales en un entorno virtual [42]

Figura 2.2: Ejemplos de virtualidad aumentada.

Tantas definiciones pueden hacer que el concepto no sea completamente cerrado, dejando libertad para darle nuevos giros al mismo con nuevos tipos de aplicaciones. Sin embargo, sí que se pueden definir unas características básicas que cualquier definición de realidad aumentada incluye, como lo hace Portalés en [35]:

- Mezcla de lo real y lo virtual.
- Interactividad en tiempo real.
- Registro tridimensional.



(a) Proyecto BMW Augmented Reality [9]

(b) Proyecto IQ Reality de Toyota [47]

Figura 2.3: Ejemplos de realidad aumentada.

2.1.1. Antecedentes históricos

Aunque la realidad aumentada es un concepto relativamente nuevo como se puede ver por el año en que se propusieron las primeras definiciones, hay una larga investigación detrás, que se remonta hasta principios del siglo XX. A continuación, algunos de los hechos clave que conforman los antecedentes de la realidad aumentada y en general, de las realidades mixtas¹ y virtuales. Algunos de ellos los podemos encontrar descritos en más profundidad en [39], por Sherman et al.

1916 Albert B. Pratt patenta un sistema de periscopio que se monta en la cabeza del usuario como un casco (figura 2.4). Este artilugio es considerado el primer HMD² de la historia.

1929 Edward Link desarrolla un simulador de vuelo mecánico para entrenar a pilotos de manera estacionaria. Con este sistema el piloto podía aprender a volar y a manejar los instrumentos de los aviones de la época. Como se puede apreciar en la figura 2.5, era una plataforma muy rudimentaria.

1954 Fred Fawler inventa el Cinerama. Un cine con pantallas curvadas de 180° que conseguían dotar de profundidad a las proyecciones, con lo que se conseguía un efecto visual que hacía parecer las imágenes más realistas.

1956 Morton Heilig desarrolla Sensorama inspirado por Cinerama. Constaba de un vídeo grabado en estereoscopia, sonido

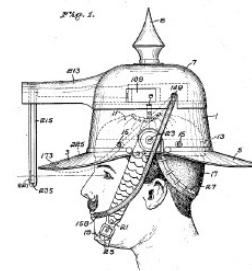


Figura 2.4: Periscopio de Pratt. Imagen cortesía de la oficina de patentes de EEUU.

¹La realidad mixta es un entorno en el que coexisten la imagen real y la sintética, cuyo espectro abarca la realidad aumentada.

²Head Mounted Display

estéreo, una silla vibradora, viento y olor. Con este sistema se pretendía llegar a diferentes sentidos, no sólo la vista. Aunque no era interactivo, con lo cual la inmersión no era completa. Figura 2.6.



Figura 2.6: Sensorama.

1960 Morton Heilig patenta una televisión estereoscópica similar a lo que conocemos como HMD. Ésta también llevaba un sistema que generaba olores.

1965 Ivan Sutherland explica el concepto de *ultimate display* durante su presentación en el congreso de la Federación Internacional de Procesado de Información (IFIP). Según Sutherland [44], por medio de estímulos visuales y dispositivos hápticos el usuario podría interactuar con objetos en un mundo sintético que no tendría por qué seguir las leyes de la física.

1968 En [45] Sutherland describe su desarrollo de un HMD estereoscópico en la Universidad de Harvard. En él utiliza tubos de rayos catódicos en miniatura para mostrar



Figura 2.5: Simulador de vuelo de Edward Link.

imágenes diferentes a cada ojo. Figura 2.9.

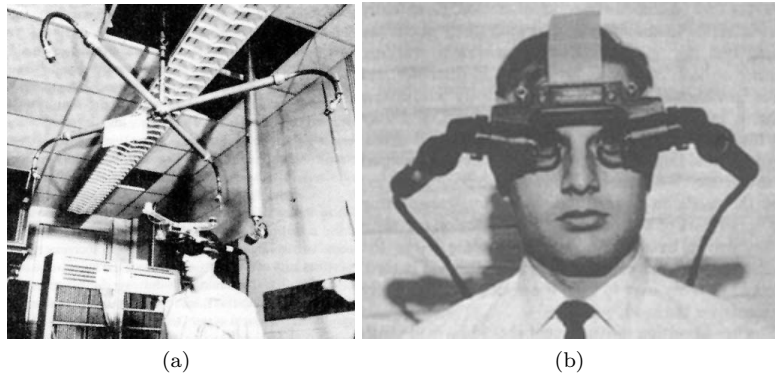


Figura 2.7: HMD desarrollado por Sutherland.

1981 Bajo la dirección de Tom Furness en la base militar de Wright Patterson, se consigue una versión operacional de *Super Cockpit*, que provee a los pilotos de realidad aumentada. El casco de vuelo consta de un visor por el que el piloto puede ver información adicional de su avión cuando mira a puntos clave de éste. Por ejemplo, si mira a las alas (donde están los misiles), el sistema le proveerá información sobre el número de misiles aún disponibles.

1989 La compañía VPL anuncia un sistema completo de realidad virtual llamado RB-2, siendo ellos los que acuñan el término Realidad Virtual.

En el mismo año Autodesk Inc. anuncia su proyecto *CyberSpace*, un programa de PC para crear mundos en 3D.

1991 Virtual Research Systems Inc. lanzan el casco de vuelo VR-2. Es el primer HMD fiable con un precio inferior a los 10000\$, lo cual lo populariza mucho entre los centros de investigación universitarios, expandiendo así este campo de estudio.

1992 La realidad virtual proyectada es introducida como alternativa a la basada en HMD en la conferencia de gráficos por computador SIGGRAPH de Chicago. La mayor atracción del acontecimiento es *CAVE*. Un sistema de realidad virtual basado en paneles retroproyectados que envuelven al usuario en un cubo semicerrado, dándole así la sensación de estar en un habitáculo virtual.

Ese mismo año Caudell y Mizell [11] acuñan el término *realidad aumentada* para referirse a los sistemas de visualización que pueden añadir imagen sintética a la imagen real. Estudian la aplicación de esta tecnología a la fabricación industrial, usándola como ayuda visual en el proceso de montaje.

Y en un laboratorio de las fuerzas aéreas de EEUU, L.B. Rosenberg desarrolla uno de los primeros sistemas funcionales de realidad aumentada.

1994 Milgram y Kishino definen el anteriormente mencionado Continuo de Milgram en [26].

1995 Rekimoto y Nagao[34] presentan los primeros marcadores para realidad aumentada de matriz bidimensional (figura 2.8).

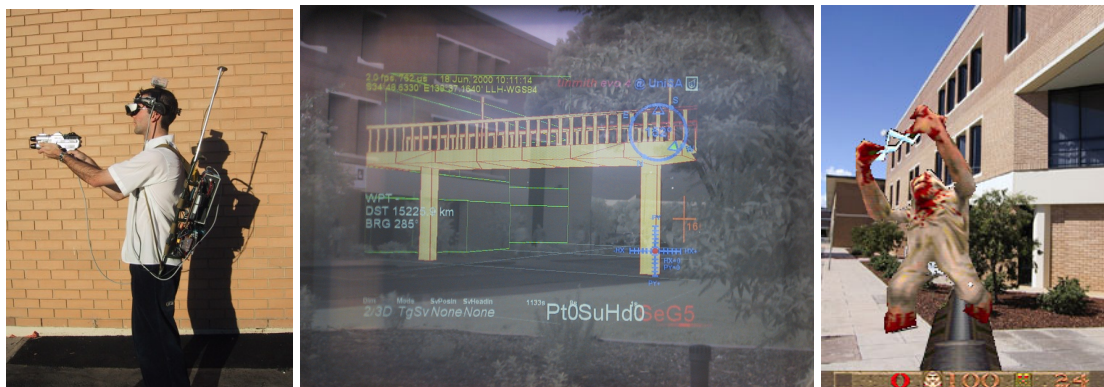
1999 Kato et al. [25] desarrollan ARToolkit. Una librería en código abierto para el desarrollo de aplicaciones de realidad aumentada. Esto ayuda a la RA a establecerse como un campo sólido y empieza a haber congresos dedicados a ella.

2000 Julier et al. [23] presentan BARS (Battlefield Augmented Reality System), un sistema de realidad aumentada aplicada al campo de batalla. Consta de un HMD conectado a una mochila donde se encuentran las herramientas de proceso de datos. Entre esas herramientas de proceso se encuentran: GPS, brújula y sensores de aceleración. Éste es entonces, uno de los primeros sistemas de realidad aumentada móvil basados en geolocalización.



Figura 2.8: Marcador de realidad aumentada.

Además, se publica ARQuake, una versión móvil en realidad aumentada del popular juego de ordenador Quake. En esta nueva versión los enemigos aparecen en el escenario real en el que estemos físicamente situados (ver figura 2.9c). El equipamiento necesario para jugar consta de un HMD con brújula, para determinar la dirección en que se mira, una mochila con un ordenador portátil y una antena GPS, y una pistola conectada al ordenador. Como se puede apreciar en la figura 2.9a el equipo es algo aparatoso.



(a) Equipamiento para jugar a ARQuake

(b) Escena del juego

(c) Enemigo virtual en escenario real

Figura 2.9: ARQuake.

2002 Kalkusch et al. [24] presentan una aplicación para guiar a un usuario por el interior de un edificio. Para ello hace falta poner marcas en las paredes del edificio, de forma que la máquina que porta el usuario

sepa el lugar donde se encuentra gracias al reconocimiento de esas marcas con ARToolkit. El sistema tiene almacenados los planos del edificio y el lugar en el que está situada cada marca. De esta forma, sabiendo la posición en la que está el usuario y a la que quiere dirigirse, se pueden aplicar algoritmos de búsqueda del camino más corto para resolver el problema y guiarlo por medio de información en un HMD.

2005 Se portan las librerías de ARToolkit al sistema operativo para teléfonos móviles Symbian. Gracias a esto empiezan a surgir aplicaciones de realidad aumentada para dispositivos móviles.

2008 La empresa Mobilizy lanza Wikitude [28] para el sistema operativo Android. Por medio del uso de la posición GPS, la brújula y el acelerómetro del dispositivo, la aplicación es capaz de mostrar información de diversas fuentes sobre elementos urbanos sobre esos mismos elementos.

2009 Al igual que pasase anteriormente con Symbian, se portan las librerías de ARToolkit a Adobe Flash, permitiendo así el uso de RA en navegadores web con contenidos Flash.

2.1.2. Dispositivos

Los sistemas de realidad aumentada necesitan de ciertos dispositivos de hardware para un mínimo funcionamiento. Vian los clasifica en los siguientes tipos en [16]:

1. Elemento capturador: Es el encargado de captar la imagen del mundo real para que esta sea procesada por la aplicación de RA. Será una cámara que dependiendo del tipo de sistema tendrá unas condiciones específicas para que el funcionamiento sea óptimo. No es necesario que esté integrada con el resto de elementos del sistema en una misma pieza de hardware, sino que puede ser una cámara sencilla que a través de un cable envía la información a la máquina de proceso. Aunque hoy en día es muy común que los dispositivos que RA tengan todos los elementos integrados en un mismo aparato.
2. Elementos de situación: Son las variables que determinan la posición de los objetos a procesar por el software de RA. Dependiendo del tipo de sistema pueden usarse diferentes elementos:
 - Marcadores como el de la figura 2.8. Su uso implica que el sistema ha de conocer de antemano todos los marcadores que sean susceptibles de ser reconocidos. Requiere el uso de espacio en memoria para guardar estos datos, lo cual puede ser considerado un inconveniente.
 - GPS, brújula y sensores de aceleración. Permiten conocer la posición de un objeto en el espacio geográfico, longitud y latitud, la dirección en la que está enfocado y su inclinación. Estos elementos son utilizados en sistemas que usan la geolocalización como fuente de información. Ver apartado 2.1.3.

- Ninguno en concreto. Hay sistemas de RA que no necesitan de elementos de situación pues actúan en base a reconocimiento de formas. En este caso el elemento procesador hace todo el trabajo.
3. Elemento procesador: El encargado de recoger los datos de entrada provenientes del elemento capturado y de los elementos de situación y generar una respuesta en base a las políticas de actuación definidas por el software de RA utilizado. Así pues, se trataría de un ordenador o algún dispositivo electrónico capaz de procesar todos estos datos donde se haya instalado el sistema de realidad aumentada a utilizar. Estas máquinas pueden hacer uso de librerías de terceros para procesar esta información, como por ejemplo el anteriormente mencionada ARToolkit creado por Kato et al.
 4. Elemento de muestreo: Por último, es necesario un dispositivo que muestre el resultado del procesado de imagen hecho por el elemento procesador, donde se verá la mezcla de elementos reales y elementos sintéticos. Igual que ocurre con el elemento capturador, dependiendo del tipo de sistema las condiciones pueden variar, pudiendo usar desde un simple monitor hasta un complejo HMD.

Después de una clasificación generalista de los elementos que intervienen en este proceso, aún es importante recalcar algunos elementos concretos que se pueden englobar en los ya descritos. Estos son a o han sido de especial relevancia en el desarrollo de este campo.

Head-Mounted Displays Los HMD han sido usados desde mucho tiempo en el campo de la realidad virtual, concretamente desde los años 60. Normalmente se conectan a algún tipo de unidad de procesado, la cual después de reconstruir la imagen la envía al HMD para que la proyecte ante el usuario. Suelen llevar una cámara incorporada, puesto que de este modo la imagen a procesar está tomada desde el punto de vista del usuario. A parte de la cámara también llevan detectores de movimiento, que miden la posición y orientación de la cabeza del usuario en pequeños intervalos de tiempo.

Se pueden distinguir dos grandes tipos de HMD:

- **Opacos:** Encapsulan totalmente la cabeza del usuario, de manera que sus ojos no pueden ver más allá de la pantalla del propio dispositivo (figura 2.10). Bien sean utilizados para realidad virtual o para realidad aumentada, el HMD opaco mostrará el vídeo que le llega desde la unidad de proceso. En el caso de la RA, este vídeo contendrá ya la mezcla de imagen real con imagen sintética. Aunque el procesar los datos de imagen es costoso en cuanto a tiempo y recursos, por lo que lo que la parte de imagen real que vea el usuario en su pantalla puede que esté retrasada un par de segundos respecto a la realidad.

Otro inconveniente de estos dispositivos es que suelen ser muy pesados, y a día de hoy cada vez están más en desuso por lo que no tienen un gran éxito comercial a parte de en los centros de investigación.

- **Semitransparentes:** Al contrario que los anteriores, los de este tipo no restringen la visión del usuario, ya que constan de lentes semitransparentes. Esto quiere decir que se puede ver a través de ellas, pero sin embargo también son capaces de mostrar imagen en ellas que se superpone a la imagen real del entorno. Normalmente tienen un pequeño proyector acoplado que se encarga de dibujar sobre la lente los contenidos sintéticos. Además, estos dispositivos suelen ser más ligeros como se puede ver en la figura 2.11a.

En este caso no hay ningún desajuste entre la imagen real y la que percibe el usuario, puesto que la está viendo en todo momento. Mejora entonces el tiempo de reacción y hace que este tipo de HMD sea usado en trabajos críticos en este aspecto como es pilotar un avión de combate. En la imagen 2.11b se puede ver un HMD Cobra, un avanzado sistema binocular que puede mostrar por pantalla toda la información de la navegación del avión y de sus armas de combate. Es la evolución natural del pionero *Super Cockpit*, anteriormente mencionado en la sección 2.1.1.

Por último, los HMD semitransparentes con RA han sido elementos de ficción durante mucho tiempo. Elementos de ficción literalmente, puesto que existen gran cantidad de producciones audiovisuales que han hecho uso de ellos, aunque este uso fuese ficticio. En la imagen 2.11c una muestra de ello.

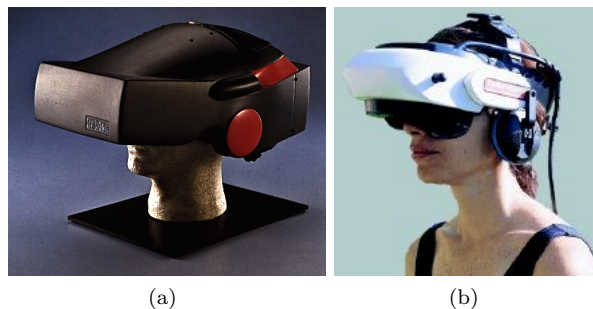


Figura 2.10: HMD's opacos. Fotos de General Reality Corp.



(a) HMD semitransparente ligero. (b) HMD Cobra para pilotos de combate. (c) HMD de ficción en la serie de animación de los 90 Dragon Ball Z.

Figura 2.11: HMD's semitransparentes.

Dispositivos hápticos Son dispositivos habilitan la interacción manual con entornos virtuales o con entornos remotos teleoperados [13]. Se usan para tareas que normalmente necesitarían manipulación manual, pero en vez de eso recogen la acción motora de las manos y la procesan para generar los gráficos pertinentes. Aunque siguiendo una definición más genérica, dispositivos hápticos son cuales quiera que se operen con las manos, como por ejemplo un teclado o un ratón.

Juegan un papel clave en los entornos de realidad mixta, pues gracias a ellos se consigue interactividad entre el usuario y la máquina más allá de la posición en la que se encuentra. Por tanto, son un elemento esencial que participa en la inmersión en entornos virtuales.

Head-Up Displays Se considera un HUD cualquier objeto transparente que muestre información sobre él y además permita ver lo que hay detrás. Se ajusta perfectamente a las características de la realidad aumentada, pues proyecta imágenes sintéticas sobre la visión real del entorno.

Sus principales aplicaciones se encuentran en los aviones militares y en el sector automovilístico. En ambos casos permitiendo acceder a información sin apartar la vista del recorrido y evitando así posibles accidentes.

Dispositivos móviles La realidad aumentada siempre ha ido por el cauce de la investigación y nunca se había prodigado demasiado en el ámbito de consumo hasta la llegada de los dispositivos móviles, ya sean teléfonos o tabletas. Con estos se ha pasado de utilizar elementos costosos y aparatosos a elementos comunes del día a día con un precio asequible y que constan de muchas otras funcionalidades. Es decir, no están hechos con el único fin de ser un sistema de realidad aumentada, sino que las aplicaciones de RA son sólo una opción de las muchas que ofrecen.

Además, el hardware de los dispositivos móviles ha sufrido un avance considerable en los últimos años, consiguiendo una importante mejora en su capacidad de proceso. Hasta tal punto esto es así que los dispositivos móviles de hoy tienen capacidades de proceso similares a los ordenadores de sobremesa de hace sólo un par de años. Por tanto, el procesado de imagen ya no es un problema que no se pueda sobrellevar.

Todos estos aspectos suponen grandes ventajas respecto a los dispositivos no móviles, lo cual también es una causa de su gran expansión.

2.1.3. Clasificación de sistemas de RA

Según el método de obtener la información se pueden clasificar los sistemas de RA en diferentes tipos. Estos tipos coinciden con los mencionados en la sección 2.1.2 sobre los elementos de situación que proporcionan información de entrada al sistema. Aunque todos tienen el mismo propósito, que es hallar la posición donde se ha de mostrar imagen sintética por encima de la real:

- Sistemas basados en el reconocimiento de marcas: Con el fin de hallar el lugar donde se ha de

mostrar imagen sintética, el software busca marcas conocidas durante su procesado. Marcas que existan en su base de datos y que llevarán asociada algún tipo de información a mostrar. Estas marcas suelen ser las clásicas matrices 2D como la de la figura 2.8.

- Sistemas basados en el reconocimiento de formas: En este caso no se buscan marcas determinadas sino formas conocidas. La investigación en los algoritmos de reconocimiento de formas está muy avanzada y se pueden reconocer formas concretas que el sistema asocie a ciertos datos. La base por tanto, es la misma que con los marcadores. Simplemente acapara una cantidad de formas mayor, ya que no se restringe a formas concretas.
- Sistemas basados en la geolocalización: Son los más usados en dispositivos móviles, debido a las especificaciones técnicas de los mismos. Utilizan el GPS para obtener la posición geográfica, y la brújula y el acelerómetro para la dirección e inclinación. Tienen varias ventajas sobre los basados en marcas. Por un lado el procesado es menos costoso en cuanto recursos. Y por otro, son más apropiados para exteriores, ya que al usar el GPS pueden posicionar imágenes sintéticas extremadamente lejos del usuario y siempre estar en la posición correcta.

Aunque también tienen sus inconvenientes con respecto a los otros sistemas mencionados. Los GPS no suelen funcionar dentro de espacios cerrados o su rendimiento suele ser bajo, por lo que su uso se limita a espacios abiertos. Por otro lado los GPS convencionales tienen un coeficiente de error de metros, con lo cual la imagen sintética pueda aparecer desplazada sobre su posición.

Aunque esta no es la única manera de clasificar los tipos de realidad aumentada. A continuación algunas clasificaciones alternativas propuestas por Portalés en [35]:

- Según el entorno físico se pueden clasificar como cerrados o al aire libre. Esto influirá especialmente en la forma de adquirir la información, bien sea por marcadores, por reconocimiento de formas o por geolocalización.
- Según su extensión pueden ser locales o ubicuos. En los primeros el espacio está acotado por el sistema. En los segundos no hay límite, la respuesta de la aplicación dependerá de la posición sea cual sea.
- Según la movilidad se pueden distinguir entre móviles y espaciales. En el primero los elementos de situación no son fijos sino que pueden variar y moverse. En el segundo, los elementos pertenecen al mundo físico real y son estáticos.
- Según el número de usuarios que participan pueden ser individuales, si sólo participa uno, o colaborativos, si hay interactividad.
- Según el tipo de colaboración, presenciales o remotos.

A parte de las clasificaciones propuestas, siempre se pueden crear algunas nuevas basándose en otros rasgos que puedan afectar a estos sistemas.

2.1.4. Campos de aplicación

La realidad aumentada ya ha dejado de ser un mero objeto de investigación y ha entrado en acción. Las mejoras en su desarrollo y los avances tecnológicos han propiciado que pueda ser aplicada en multitud de campos, probando ser muy útil en algunos casos. En otros, aún se encuentra en la condición de operacional pero en pruebas, aunque es probable en el futuro cercano consiga asentarse en estos campos. Y también es más que probable que se empiece a aplicar en campos nuevos, abarcando así un espectro de mercado mucho mayor.

Entretenimiento La aparición de la RA en este campo se inicia con el anteriormente mencionado ARQuake que fue lanzado en el año 2000. La mejora en desde entonces ha sido sustancial, puesto que ya no hace falta acarrear una pesada mochila que porte todo el sistema para poder jugar. Este ámbito ha sido invadido por los teléfonos de última generación y ha facilitado su acceso por tanto, al gran público.

La propia naturaleza de la RA hace que en estos juegos el usuario sea protagonista físicamente de la acción del mismo, ya que el escenario es el mundo real. Para los desarrolladores ha supuesto una manera completamente nueva de hacer juegos. Algunos de ellos ya están en las listas de los más descargados, como *Parallel Kingdom*, un juego de rol multijugador en el que el usuario es un personaje que se mueve libremente por el escenario, convirtiendo la ciudad en un mapa al estilo de los RPG³ clásicos. Ver figura 2.12.

También se han desarrollado juegos para plataformas específicas a ese fin, como la Nintendo DS o la PSP. La primera consta de títulos como *Ghostwire*, en el que aparecen fantasmas sobre la imagen real del entorno obtenida por la cámara del dispositivo. En el caso de la PSP además, hay juegos de producción española como *Invizimals*. Un juego donde se coleccionan pequeñas criatura virtuales para después luchar con ellas. Algo que quizá recuerde a juegos como *Pokemon*, con el añadido por supuesto de la realidad aumentada.

Educación Es uno de los ámbitos en los que tiene más sentido implantar la realidad aumentada, ya que facilita y ensalza la actividad pedagógica. Uno de los sistemas más habituales son los libros para niños en RA, que contienen marcadores en las páginas que el procesador convertirá en objetos virtuales animados. Esto desemboca en una mezcla de educación y entretenimiento para niños que es muy positiva. Se ha acuñado un término específico para esta campo que es *Edutainment*, del inglés Education + Enterteinment.

Así pues, en este concepto se pueden incluir todo tipo de juegos educativos para niños como rompecabezas, puzzles, tarjetas o juegos de mesa, todos en RA. Un ejemplo de esto es el proyecto *Realitat*

³Acrónimo de Role-Playing Game.



Figura 2.12: Escenas del juego Parallel Kingdom.

llevado a cabo por el grupo de investigación LabHuman de la UPV, que pretende fomentar y facilitar el estudio a los niños de la Comunidad Valenciana.

Medicina La RA no está tan presente en la ejecución de la medicina como en su aprendizaje, y es que es muy útil para la formación en campos como la cirugía. Igual que en la industria un dispositivo de RA puede guiar a un mecánico en el cambio de una pieza de coche, también puede guiar a un estudiante de medicina en cómo operar a un paciente. Con este objetivo surge ARMed [6], que pretende formar a nuevos cirujanos por medio de la realidad aumentada. Reconociendo las formas del cuerpo humano, el sistema podrá mostrar lo que hay en el interior de cada punto de cuerpo antes de que el cirujano use el bisturí.

Sielhorst et al. [40] presentan un simulador de partos en realidad aumentada para el entrenamiento de futuros ginecólogos. El sistema también cuenta con dispositivos hápticos que harán las veces de forceps durante las simulaciones.

Psicología La RA en este campo está focalizada principalmente en el tratamiento de fobias. Para ello, el método habitual es someter a los pacientes a estímulos controlados de la fobia a tratar en un HMD. De esta forma el doctor puede controlar el estado del paciente y evitar que entre en pánico. Entre las fobias para las que se han desarrollado programas destacan la acrofobia⁴, la aracnofobia⁵, la claustrofobia⁶ o la entomofobia⁷.

⁴Fobia a las alturas.

⁵Fobia a las arañas.

⁶Fobia a los espacios cerrados

⁷Fobia a los insectos

Industria, Manufactura Azuma menciona en 1997 [7] el posible uso de la realidad aumentada en este campo, facilitando así la fabricación y mantenimiento de maquinaria compleja como pueden ser coches, barcos, aviones o camiones. Los sistemas de RA en este campo son de gran ayuda para ingenieros, técnicos y mecánicos, ya que les proveen de una ayuda visual sobre cada pieza de las máquinas que operan.

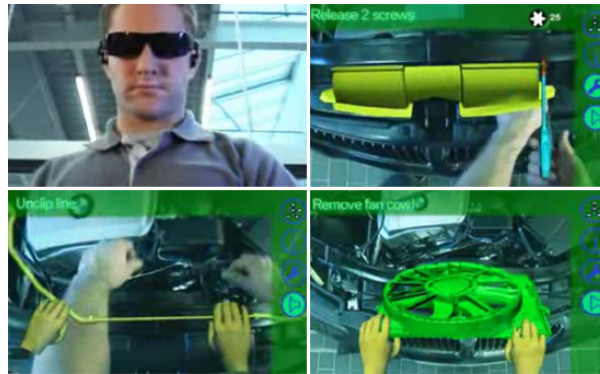
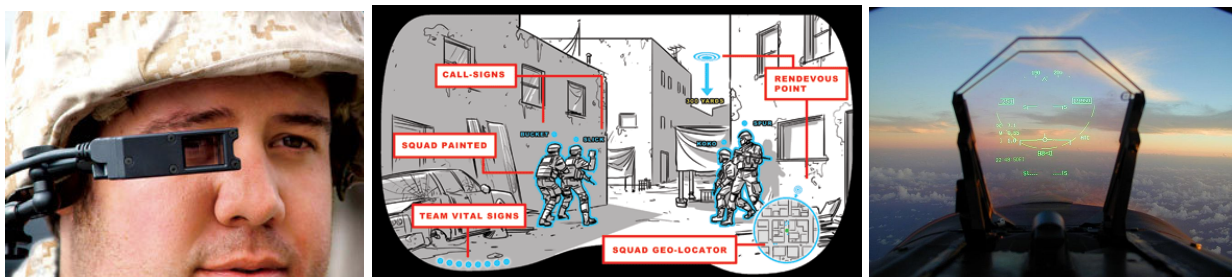


Figura 2.13: Proyecto BMW Augmented Reality.

Como ejemplo de ello en la figura 2.13 aparece el proyecto de realidad aumentada de BMW para facilitar la labor de sus operarios [9].

Militar Como sucede en muchos otros campos de investigación, la industria militar es a veces uno de los grandes impulsores de la ciencia. Y a veces, los productos para el gran público son derivados modestos de investigaciones militares. Esto explica el por qué del avance de la RA en este sector respecto a otros no tan desarrollados.



(a) HMD semitransparente militar. (b) Visión conceptual militar en RA del campo de batalla. (c) HUD en aviones de combate.

Figura 2.14: Aplicaciones de RA en la industria militar.

Los mayores avances se encuentran en los HMD y HUD. Los primeros ya han sido mencionados en la sección 2.1.2 como ayuda a los pilotos de combate, dándoles información relativa al estado de sus aviones. Pero la tecnología va más allá y ya se fabrican HMD para soldados de infantería, con el objetivo de que tengan información de primero mano sobre el entorno hostil en el que se encuentran (figura 2.14b). El diseño de estos dispositivos suele basarse en HMD semitransparentes, ya que son más ligeros y fáciles de

acoplar a un casco de soldado como se ve en la figura 2.14a.

En cuanto a los HUD, ya se implantan en aviones de combate (figura 2.14c) y otros vehículos, proporcionando así información sobre los mismos y sobre el entorno, igual que hacen los HMD.

Turismo Éste es un ámbito principalmente dominado por los dispositivos móviles donde se hace uso de sistemas de RA basados en geolocalización. Tiene dos grandes destacados y dominadores, que son Layar y Wikitude (figura 2.15).

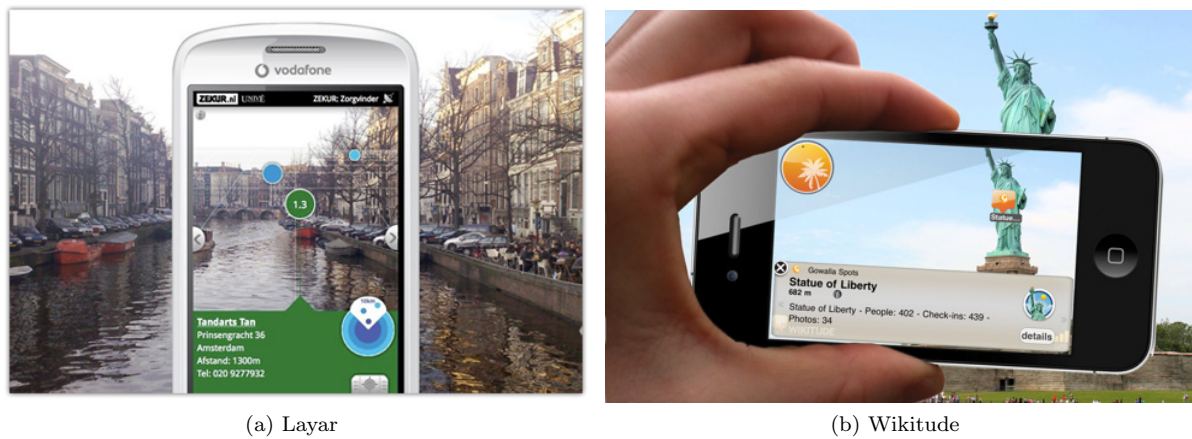


Figura 2.15: Aplicaciones sobre turismo.

Ambos están diseñados de manera que se pueda añadir nueva información al sistema de forma colaborativa, utilizando la técnica conocida como *crowdsourcing*. Así, cualquier desarrollador ajeno a las compañías desarrolladoras de estas aplicaciones puede añadir nuevos datos y completar la descripción turística de cualquier lugar de la Tierra.

Sector automovilístico Es un campo en el que los sistemas de RA aún son muy jóvenes y la mayoría están en fase de pruebas. Parte de la causa es aa complejidad técnica que conllevan, ya que la mayoría trabajan sobre parabrisas de coches que actúan como HUD, y su fabricación todavía no se ha extendido.

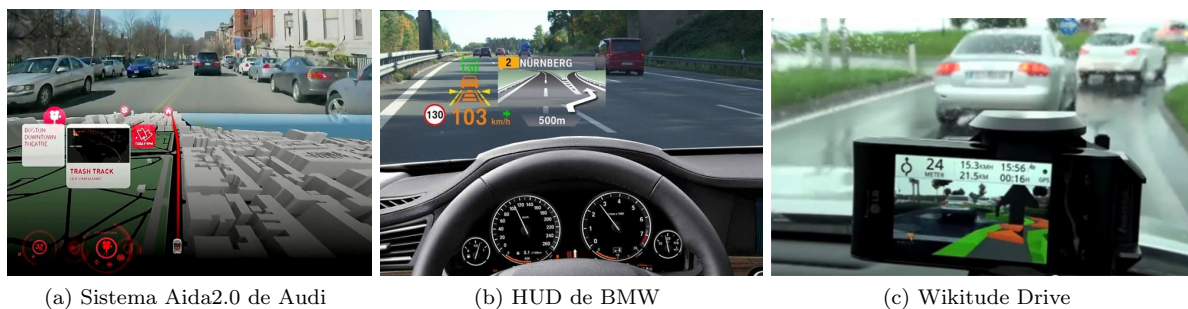


Figura 2.16: Aplicaciones en el sector automovilístico.

Algunas firmas como Audi, BMW, Peugeot o Pioneer ya está trabajando en sus coches con sistemas HUD que pretenden sustituir a los navegadores GPS convencionales, pues son más seguros al no tener que apartar la vista de la carretera.

Mientras tanto ha salido al mercado alguna opción novedosa como Wikitude Drive para dispositivos Android, que hace la misma función que un navegador GPS. Con la diferencia de que muestra la carretera por la pantalla del dispositivo, eliminando así el peligro de mirar al navegador y no ver lo que ocurre en la calzada.

Marketing y Publicidad Algunas compañías han puesto de moda el promocionar sus productos con RA por medio de internet. Para ello normalmente sólo hace falta una cámara web conectada a un ordenador y visitar la web de la compañía en cuestión.

Algunos ejemplos llamativos son los de la firma de relojes Tissot, que permite al usuario probarse cualquier reloj de su colección sin moverse de casa. O la de la firma de gafas de sol Ray-Ban, que propone el mismo concepto pero con gafas.



(a) Probador en RA de Tissot.

(b) Probador en RA de Ray-Ban.

(c) Transformación en Iron Man.

Figura 2.17: Aplicaciones de RA en Marketing y Publicidad.

Otra propuesta interesante es la de los productores de la película Iron Man, que para promocionar su estreno en los cines se propusieron como misión transformar a todo el que quisiese en el héroe de la película.

Arte En el mundo digital actual el arte digital también existe, y la realidad aumentada es una nueva herramienta para los artistas. Se pueden combinar diversos elementos de RA en una instalación artística en museos o espacios dedicados a tal fin, consiguiendo unos trabajos mucho más llamativos que las ya habituales producciones audiovisuales clásicas.

Por el Ars Electronica Museum of the Future de Linz (Austria) ya han pasado varias obras de este tipo y parece que la tendencia sigue en aumento.

Cine y Televisión La RA en este ámbito está muy extendida, aunque quizá no se la suele llamar así o no se sea consciente de que lo que se está viendo es realidad aumentada. El ejemplo más claro quizá sean

los *Chroma Keys* que se usan en los sets de TV para por ejemplo, dar las noticias del tiempo. Ese famoso panel verde no es más que una marca de RA que es sustituida por imagen sintética.

También en la TV se usa la RA en las retransmisiones deportivas, normalmente para indicar aspectos de la estrategia de juego. En el fútbol es utilizado a menudo para señalar los fuera de juego en las repeticiones a cámara lenta, como se puede ver en 2.18.

Por último, en el cine a menudo aparecen personajes fruto de los gráficos por computador. Cuando estos personajes se introducen en un vídeo con escenas del mundo real se convierten en RA.

A menudo se usan marcas para la introducción de estos personajes en el entorno, aunque con técnicas algo diferentes como la *motion capture* o captura de movimientos. Así han sido rodados personajes famosos como Golum en *El Señor de los Anillos*, o muchas escenas de la película *Avatar*.



Figura 2.18: Señalización del fuera de juego en un partido de fútbol.

2.2. Sistema operativo Android

Android es un sistema operativo de Google orientado a dispositivos móviles y representa la incursión de la compañía en el mercado móvil. Aunque el sistema en sí no fue un proyecto iniciado por Google, sino por la compañía Android Inc. que posteriormente sería comprada por Google en 2005. Así pues, el 23 de Septiembre de 2008 Google lanza la primera versión de este sistema operativo, que está basado en una versión del Kernel 2.6 de Linux modificado. El sistema actual ya funciona sobre teléfonos y tabletas.

Siguiendo la filosofía Linux, la licencia de distribución de este SO es Apache 2.0, es decir, es software libre. Además, permite acceder al código fuente y modificarlo, lo cual permite que las compañías fabricantes de teléfonos y tabletas puedan retocar el sistema a su gusto para una mejor adaptación con su hardware. Y lo que es más, también hay desarrolladores independientes que lanzan sus propias ROMs de Android con Kernels modificados diferentes al original, con el fin de optimizar su funcionamiento.

Pero el desarrollo del sistema no es sólo asunto de Google, sino que existe un conglomerado empresarial que se ocupa de ello bajo el nombre *Open Handset Alliance* (OHA). Está integrado por empresas como Intel, Texas Instruments, Samsung, LG, Nvidia o HTC entre otros. Su objetivo es desarrollar estándares abiertos en el campo de la telefonía. Y el sistema Android, que es software libre, es uno de sus estandartes.

Entonces, Android pretende reunir en una misma plataforma todos los elementos necesarios para crear un entorno de desarrollo y control de aplicaciones integrado. Para ello, se usa Java como lenguaje de desarrollo y se proporciona un SDK a la comunidad de desarrolladores para facilitar el máximo posible el trabajo en esta plataforma. También se proporcionan APIs desarrolladas por Google y otras empresas y un plugin para el IDE⁸ Eclipse que contiene todas las herramientas necesarias para el desarrollo. Por

⁸Integrated Development Environment. Es una herramienta de desarrollo de software muy común que suele contener todo lo necesario para el trabajo con ciertos tipos de lenguajes y en ciertos tipos de aplicaciones, como es el caso de Android.

tanto, en principio cualquier dispositivo Android puede ejecutar cualquier aplicación, sea de la marca que sea, pues son multiplataforma.

Como es normal, las librerías del sistema se van actualizando y aparecen nuevas funciones. El uso de nuevas funcionalidades puede provocar que las aplicaciones que las contienen no se puedan ejecutar en dispositivos con versiones antiguas del sistema operativo, lo cual es normal en el ciclo de vida del software.

2.2.1. Arquitectura

Android está constituido por una pila de software que incluye todo lo necesario para su funcionamiento. En esta pila se pueden destacar 5 capas principales como se puede ver en la figura 2.19, unas dependientes de las otras:

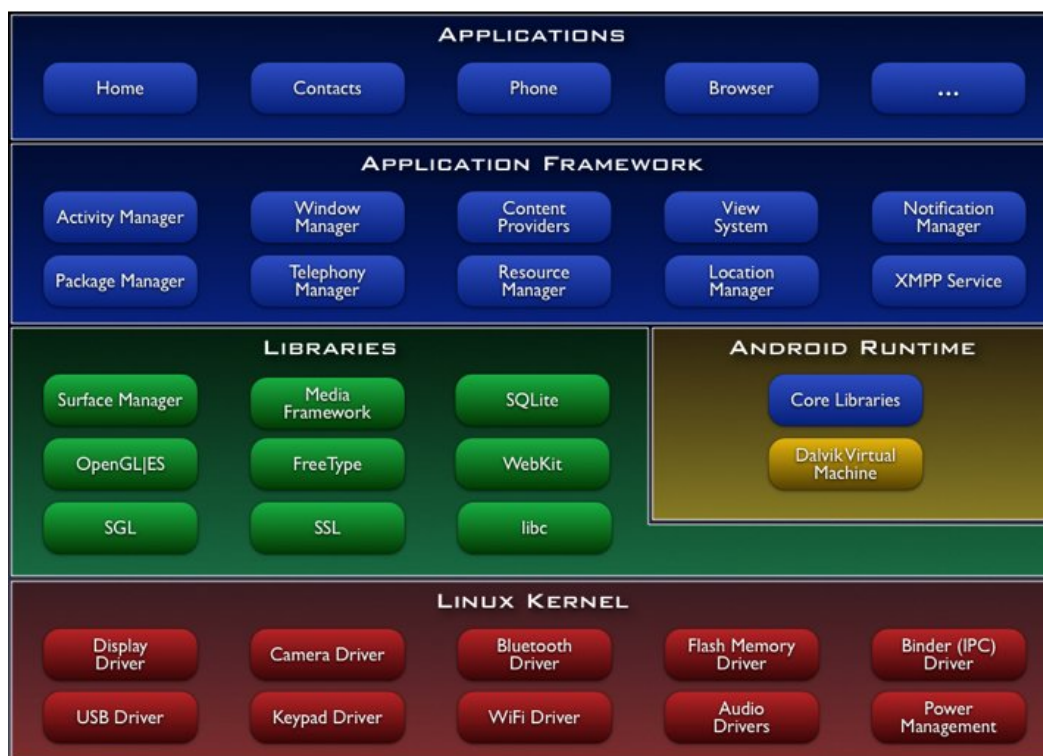


Figura 2.19: Arquitectura del SO Android.

1. **Núcleo Linux:** Como ya se ha mencionado, el núcleo de android es el Kernel 2.6 de Linux, que se encarga de los servicios base del sistema: seguridad, gestión de memoria y procesos, pila de red y controladores. En cuanto a los controladores, esta capa contiene todos los necesarios para cualquier hardware que puedan llevar los dispositivos Android. Los controladores de hardware propietario deben ser provistos por los fabricantes.
2. **Entorno de ejecución de Android:** Aquí se incluyen la mayoría de librerías necesarias para

la ejecución de Java y la máquina virtual Dalvik, que ejecuta el código Java. Dalvik no es la máquina virtual de Java común, ya que ha sido especialmente desarrollada para su funcionamiento en este tipo de dispositivos, que tienen menor capacidad de proceso y menos memoria. La diferencia principal estriba en que, al contrario que la máquina virtual estándar de Java que trabaja sobre *bytecode*, **Dalvik** transforma el código fuente en uno más eficiente antes de ejecutarlo. En este proceso crea archivos *.dex*, que son los que ejecutará. De esta manera reduce la carga de procesado.

3. **Librerías:** Set librerías C/C++ que son usadas por algunos de los componentes del sistema. Una de ellas es la librería **SQLite**, que provee un marco de trabajo para bases de datos, y que dada la naturaleza del este proyecto será necesario utilizar.
4. **Marco de trabajo de aplicaciones:** Conjunto de herramientas de desarrollo de aplicaciones para el sistema. Contiene diversas API que son comúnmente usadas por los desarrolladores. Google actualiza su API oficial cada cierto tiempo con nuevas mejoras. Al desarrollar para este sistema, hay que tener en cuenta que si se usa una API n , esa aplicación no podrá ser instalada en dispositivos con una versión del framework $m < n$.
5. **Aplicaciones:** Es la capa más alta y depende por tanto de todas las de abajo. Por defecto incluye algunas aplicaciones que proporcionan la funcionalidad básica del teléfono o tableta. Las aplicaciones que instale el usuario estarán en el mismo nivel.

2.2.2. Componentes

Las aplicaciones Android están compuestas por cuatro tipos de componentes. No es necesario que una aplicación contenga todos los tipos, normalmente algunos tipos de componentes abundan y otros escasean.

Todos ellos han de ser declarados en el manifiesto de Android, un fichero en XML (`androidManifest.xml`) que contiene todas las referencias a los componentes utilizados y a los permisos de seguridad que necesita la aplicación para funcionar.

Activity Son el componente principal de la interfaz gráfica de Android. Se podría decir que una Activity o actividad equivale a una pantalla, vista o ventana de la aplicación. Por tanto, cada actividad contiene la definición de su interfaz gráfica en ficheros XML. No hay que confundir las actividades con las ventanas emergentes o notificaciones del sistema, que son simplemente eso, notificaciones.

Además, cada una de ellas supone un proceso diferente, por lo que los datos son independientes del resto de actividades. Cuando se pasa de una actividad a otra, los datos generados por la primera no desaparecen, ya que la actividad queda pausada a la espera de que el hilo de ejecución retorne a ella. De tal forma que forma que lo que se obtiene es una pila de actividades, unas descendientes de otras, que son pausadas cuando se apilan.

El medio por el que se pasa de una a otra son los **Intent**, que son objetos que llaman a otras actividades o a otras aplicaciones, dejando en pausa la actividad desde la que se lanzan.

Service Los servicios son procesos que se ejecutan sin necesidad de una interfaz gráfica. Suelen realizar tareas en segundo plano para proveer datos a las actividades que están en primer plano. De esta manera se evita la sobrecarga de las actividades en primer plano y se previenen posibles ralentizaciones en la interfaz de usuario.

Cualquier aplicación que necesite descargar grandes cantidades de datos de internet necesita volcar estas descargas en hilos de ejecución diferentes al de la interfaz, y los servicios se pueden encargar de esta tarea. Esto no quiere decir que cualquier hilo de ejecución diferente del principal necesite de un servicio, ya que también se pueden crear hilos secundarios desde el hilo de la interfaz gráfica, pero los servicios están desarrollados específicamente para eso.

Broadcast Receiver Este componente no necesita que la aplicación esté funcionando para llevar a cabo su tarea. Se encarga de recibir y detectar notificaciones que afectan a la aplicación a la que pertenece. Por ejemplo, cuando se recibe un e-mail, el sistema avisa de que el gestor de correo contiene un nuevo mensaje no leído. Para ello, ha habido un proceso que ha estado enviando consultas periódicamente al servidor de correo para saber si había nuevos mensajes.

El mismo ejemplo se puede aplicar a cualquier tipo de aplicación que pueda recibir notificaciones, como por ejemplo las famosas twitter y facebook.

Content Provider Son el mecanismo que permite compartir datos entre diferentes aplicaciones. Gracias a ellos no se compromete la seguridad de los datos, ya que no se revelan detalles como su almacenamiento interno, su estructura o sus datos de creación.

El sistema ya proporciona algunos de estos mecanismos, de forma que por defecto con el paquete *android.content* se pueden compartir contactos, mensajes de texto, fotos, vídeo y audio. Si una aplicación requiere compartir otro tipo de datos que no se encuentre entre estos, siempre se puede definir un nuevo proveedor de contenidos que lo habilite.

2.2.3. Ciclo de vida de las aplicaciones

Las aplicaciones tienen sus propios procesos y estos no son compartidos con otras. Estos procesos se inician y finalizan según necesite la aplicación. Aunque esto no siempre es así, ya que el sistema puede decidir matar procesos que no están en primer plano en un momento dado porque necesita más recursos para otras operaciones. Es decir, el juez final que determina el tiempo de vida de los procesos es el propio SO.

Los ciclos de vida son dependientes del tipo de proceso, que a su vez depende del tipo de componente el que se ejecuta. El más importante de estos como cabe esperar, es *Activity*, ya que este tipo de procesos

son el hilo de ejecución principal de las aplicaciones. En la figura 2.20 se puede ver un diagrama de flujo con sus posibles estados.

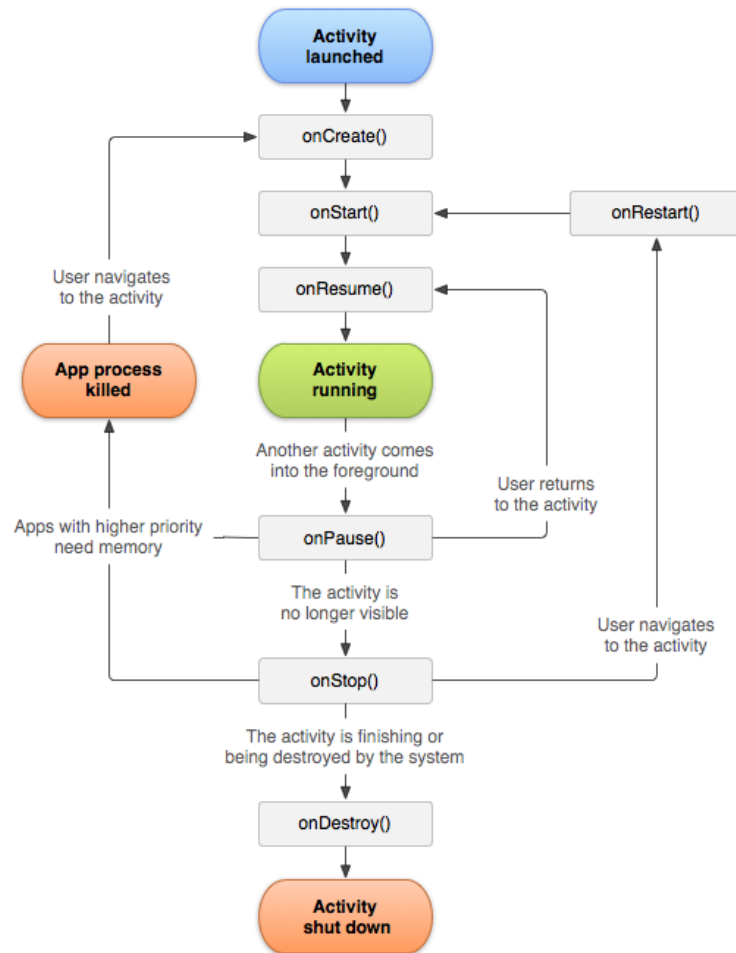


Figura 2.20: Ciclo de vida de la componente Activity.

Algunos de los estados no pueden ser matados, es decir, que el sistema no puede finalizarlos aunque necesite más memoria para otra tarea, tiene que esperar a que acaben. Según la documentación de la web de desarrolladores de Android [17], los estados que no pueden ser abortados son:

- onCreate
- onRestart
- onStart
- onResume

2.2.4. Datos de mercado

La guerra por el el trono de los sistemas operativos móviles tiene dos claros contendientes, Android y iOS de Apple, siendo el primero el que ha estado por encima en 2010 y 2011. En la gráfica 2.21 se puede apreciar un incremento de cuota de mercado de iOS y una disminución de la de Android, aunque el mayor perjudicado ha sido RIM, el fabricante de BlackBerry, que en los últimos años ha perdido casi un 20 % de cuota.

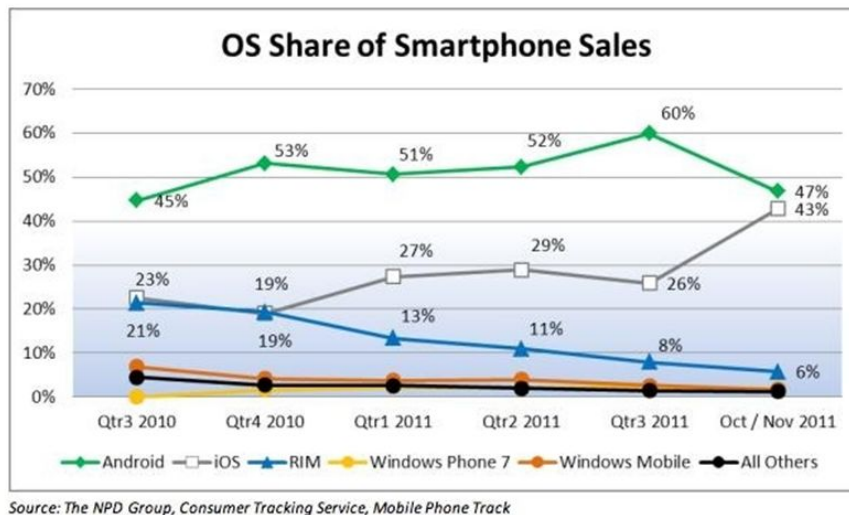


Figura 2.21: Cuota de mercado de los SO para dispositivos móviles. Imagen: NPD Group.

A pesar de su bajada Android sigue siendo el líder del mercado, por lo que utilizarlo como plataforma para desarrollar este proyecto es una buena decisión. Permitirá llegar al mayor número de gente posible, puesto que es la opción mayoritaria. Además, los teléfonos inteligentes ya son mayoría en el mercado como se puede ver en la gráfica 2.22. Los teléfonos simples están siendo desterrados y más de la mitad de los teléfonos en el mundo son considerados de última generación.

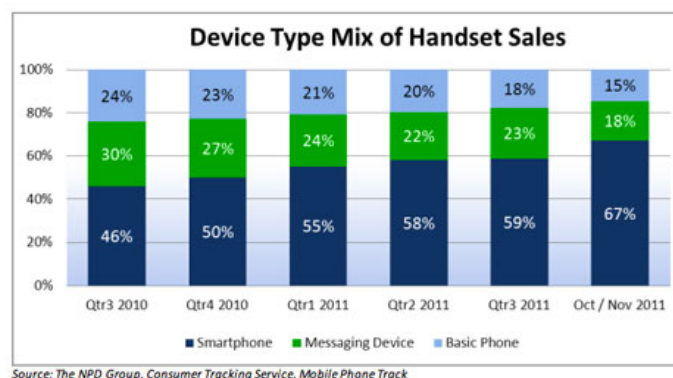


Figura 2.22: Ventas según tipos de teléfono. Imagen: NPD Group.

Por último, según un estudio llevado a cabo por comScore Mobilens [12], el porcentaje de uso activo

de las aplicaciones descargadas en dispositivos móviles en Noviembre de 2011 asciende al 44.9%. Esta cifra puede considerarse muy positiva dado lo incipiente que es el mercado y que este porcentaje no ha dejado de crecer. La medición anterior tomada en Agosto del mismo año era del 41.6%.

2.3. Redes sociales

La aparición de las redes sociales y la popularidad que han conseguido en los últimos años ha cambiado el ecosistema de internet, transformándolo en uno más colaborativo. Millones de personas navegan por la red, y muchas de ellas participan en las redes sociales. Pero lo más asombroso es la gran masa de contenidos que son creados por los usuarios de las redes sociales. Este fenómeno forma parte de lo que ha sido acuñado como la **Web 2.0**, un internet en el que todo el mundo contribuye.

Se puede decir entonces, que los servicios sociales en internet han cambiado la forma de comunicarse, entretenerse y hasta la forma de vivir de las personas. Y es que son la razón de que una gran masa de gente se haya convertido en habitual de internet, cuando antes de este fenómeno la web no les parecía tan atractiva como para pasar su tiempo en ella. Además ya no sólo consumen contenidos, sino que los crean. Es lo que ya se conoce como **prosumidor**, de productor+consumidor, ya que ahora además crean contenidos, creándose así un entorno en el que todos crean y consumen.

Tapscott et al. toman algunos de estos conceptos para definir los principios de las redes sociales [46]:

- La apertura. Muchas de las redes sociales publican API's para que desarrolladores independientes creen nuevos componentes para ellas, beneficiándose así del trabajo colectivo.
- La interacción entre iguales.
- Compartir.
- El ámbito global.

De estos principios se pueden extraer algunos conceptos interesantes. El primero es el de la **inteligencia colectiva**, que hace referencia al poder de creación de las masas en la red. Cualquier persona en cualquier parte del mundo puede tener algún conocimiento que sea muy útil para otras personas, y compartirlo en la web es la mejor forma de generar conocimiento común. Si cada persona hace esto, la red se convierte en una base de datos de casi infinita extensión donde se puede encontrar casi cualquier cosa que se necesite.

Por otro lado, también es relevante remarcar que nada en las redes sociales valdría la pena sin participación. Éstas están cimentadas sobre la participación de los usuarios, y sin ellos no tienen sentido. Por lo que el primer paso para que una red social tenga éxito es que tenga un mínimo número de usuarios capaces de generar contenido para el resto. Sí, el resto, porque a pesar de la inteligencia colectiva y la actividad social, sólo un 10% de los usuarios crean contenido habitualmente. El resto se limita a consumir

estos contenidos. Puede parecer un porcentaje bajo, pero está comprobado que es suficiente para mantener viva un red social. De hecho, la red *Facebook* tiene unos números similares.

Por último, el que haya tales cantidades de gente reunida en un mismo lugar virtual, supone una gran oportunidad para empresas que quieren promocionarse y darse a conocer en nuevos mercados. Desde el inicio del boom de las redes sociales, se puede apreciar un cambio en el que cada vez más las empresas están más involucradas en ellas, y los usuarios lo aceptan como una parte más de su sociedad virtual.

2.3.1. Comunicación y acercamiento al poder

Pero si hay algo que realmente ha cambiado, son las distancias entre el pueblo llano y la gente con poder. Quizá sea exagerado proponerlo de esta manera, pero es la forma más fácil de entenderlo. Por ejemplo, antes de que existiesen redes como **Twitter**, que el ciudadano medio asalariado pudiese hacerle llegar un mensaje a un político local o nacional, era una tarea ardua. Hoy sólo le hace falta buscar el perfil de esa persona en Twitter y enviarle un mensaje citándolo. Y esto es aplicable a cualquier ámbito, no sólo a políticos. Los actores, deportistas, músicos, cineastas, periodistas, empresarios y otros muchos, abundan en las redes sociales y son todos ellos por fin accesibles para el gran público. Las redes como Twitter han supuesto una desintermediación entre el poder y el ciudadano.

Trasladando este ejemplo a la universidad, también se podría poner al alcance del cualquier alumno el contacto con un profesor, catedrático o incluso el rector de la universidad. Porque de la interactividad de las redes sociales se benefician todos. Todos encuentran contenidos que consumir. Y está es una de las razones por las que dotar de interactividad a este proyecto es el movimiento acertado.

2.4. Aplicaciones de RA existentes para Android

EL catálogo de aplicaciones de realidad aumentada para android es muy extenso, y la intención de este apartado no es repararlo todo. Es más bien, destacar las que tienen alguna relevancia en el ámbito o son cercanas al concepto de aplicación que se pretende en este proyecto.

2.4.1. Clásicas

Las aplicaciones clásicas y más conocidas de esta plataforma están basadas en geolocalización y proporcionan información del entorno. De ahí que como ellas mismas se autodefinan como *reality browsers*, navegadores de la realidad. Los dos grandes contendientes son Layar y Wikitude, aunque existen más aplicaciones que vale la pena mencionar.

Layar Una de las primeras en aparecer. Como su propio nombre indica, la aplicación se basa en capas. Capas que el usuario puede intercambiar para elegir qué se sobrepone sobre la realidad en su dispositivo. De esta manera, si el usuario está interesado en ver monumentos históricos cercanos a su posición, puede elegir una capa de monumentos históricos que se los muestre.

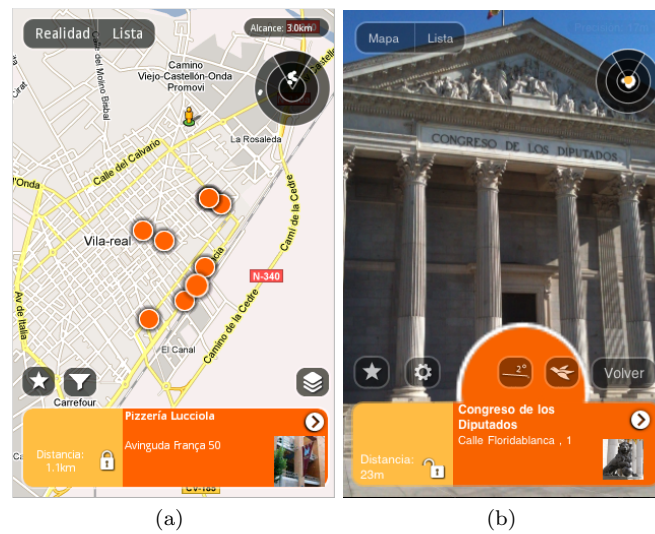


Figura 2.23: Imágenes de la aplicación Layar.

El número de capas disponibles es enorme y muy variado. Además, es posible añadir capas al repositorio de la aplicación. Aunque quizá esto no es algo orientado al usuario medio, ya que requiere de ciertos conocimientos técnicos. Algunas universidades ya tienen sus capas disponibles en Layar, algunas de ellas españolas, como la Universitat Jaume I o la Universitat d'Alacant.

Layar ha sido barajada como opción para mostrar la UPV en realidad aumentada. El problema es que al crear una capa simplemente se proporcionan datos, pero no se puede definir una funcionalidad interactiva con ellos, al menos en el momento en que se barajó como opción. Para solventar ese problema, Layar también tiene una API llamada *Layar Player*, que permite embeber el motor de RA de Layar en aplicaciones de terceros, posibilitando así cualquier otra funcionalidad no definida en la aplicación original. Esta habría sido una gran opción para el desarrollo de este proyecto de no haber sido porque sólo está disponible para iOS.

Wikitude Nace como una base de datos geolocalizada de wikipedia. La idea original era mostrar en cada lugar los artículos de wikipedia relacionados. Este hecho ya permite que hayan más de 350.000 puntos disponibles a lo largo y ancho del plante.

Sin embargo la aplicación ha evolucionado hacia un sistema similar al que usa Layar, posibilitando cambiar de capa y ver otro tipo de datos. Incluso ya existen capas sociales como la de twitter, que muestra todos los *tweets* que se han creado cerca del usuario, lo cual se acerca mucho al concepto que se quiere llevar a cabo en la UPV con este proyecto.

Wikitude además, proporciona herramientas de desarrollo al igual que Layar, para que su motor de RA puede integrar en aplicaciones de tercero. En este caso, sí que está disponible para Android, y ha sido una de las herramientas utilizadas para la realización de este proyecto.

Weather Reality Curioso concepto de RA en el que se muestra la climatología de los lugares cercanos. Permite saber en qué dirección no hay que ir porque hay tormenta, o hacia donde hay que dirigirse si se buscan los rayos del sol.

También permite tomar fotos del estado del tiempo en el lugar en el que se está y compartirlas con el resto de usuarios, para que tenga información visual a parte de los datos en texto.

Star Chart y Google Sky Map Otra curiosa forma de aplicar los conceptos de RA, y es que gracias a los instrumentos de los dispositivos móviles actuales como el GPS, la brújula y sus múltiples sensores, se puede saber hacia qué punto del cielo se está mirando. Basándose en esa premisa, estas aplicaciones son capaces de mapear el cielo y localizar las estrellas que se están observando, proporcionando información sobre ellas en la pantalla.

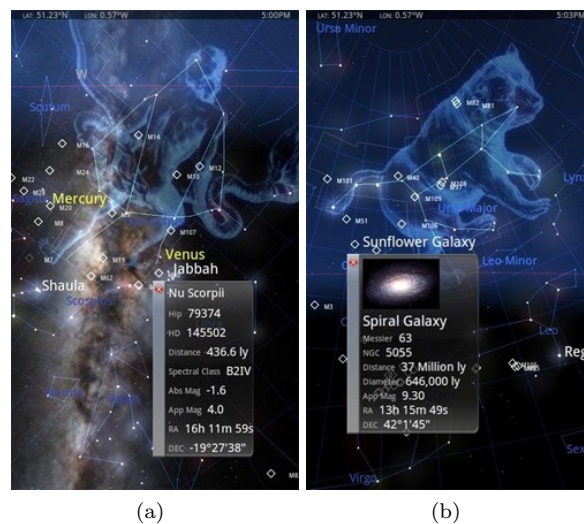


Figura 2.24: Imágenes del cielo con información sobre las estrellas en Star Chart.

2.4.2. Componente social

Empiezan a surgir cada vez más aplicaciones de RA con esta componente. Se pueden distinguir dos tipos principales de aplicaciones sociales en RA. Por un lado las que son una red social en sí. Es decir, el usuario tiene una cuenta en esa aplicación y por medio de ella comparte contenidos con el resto de usuarios. En este tipo se pueden incluir aplicaciones como *Sekai Camera* o *FriendsRadar*.

Por otro lado están las aplicaciones que no son redes sociales per se, sino que conectan las cuentas que el usuario tiene en otras plataformas sociales, como pueden ser twitter, LinkedIn o facebook. Así, se geolocalizan las actividades de los usuarios en estas redes, haciendo posible que el resto de la comunidad los visualice si pasa por ese lugar. Dentro de este tipo hay aplicaciones como *TweepsAround*, que permite la actividad geolocalizada en twitter.

2.4.3. Comercio

Éste es otro aspecto que se pretende incluir en este proyecto, facilitando a los usuarios un espacio donde poder comerciar. Aunque las aplicaciones de este ámbito no están tan enfocadas en el aspecto social como en el de ofertas por proximidad. La mayoría de ellas ofrecen información en RA sobre comercios cercanos al usuario, ofreciendo incluso promociones por encontrar los comercios de ese modo. Se pueden destacar aplicaciones como *vouchAR*, *Fast Food Reality* o *Fast Food Reality*. La primera es un buscador de ofertas en comercios locales y la segunda simplemente facilita la labor de encontrar restaurantes de comida rápida.

2.4.4. Orientación

Otro campo donde hay interesantes aplicaciones. *Car Finder AR* propone etiquetar el lugar donde se ha dejado aparcado el coche para que luego la aplicación guíe al usuario hasta su vehículo sin problemas. Tiene modos de realidad aumentada y de mapa, para que la experiencia sea lo más fácil posible.

AR Compass se propone ayudar a los menos hábiles a la hora de orientarse, mostrándoles una brújula en realidad aumentada que les marca en todo momento que dirección están siguiendo. Además, si se mira el mapa, este se orienta en la dirección en la que está mirando el usuario, facilitando así la búsqueda del camino a seguir.

2.4.5. Trabajo en otros centros de enseñanza

Ya hay algunas universidades que cuentan con su propia aplicación de realidad aumentada. Entre ellas, la ya antes mencionada Universidad de West Virginia, que tiene disponible una aplicación para iOS. Esta aplicación permite ver en RA la clásica información sobre el campus, como edificios, escuelas, laboratorios o departamentos. Pero además está conectada con la intranet de la universidad, con lo cual la experiencia es mucho más completa. Gracias a ello se puede obtener información en tiempo real sobre los servicios de la universidad, i.e.: cuántos ordenadores libres hay en la biblioteca en un momento determinado o en qué aulas se imparten las clases en las que el usuario está matriculado. Con este paso hacia la RA, se consigue un nuevo concepto de intranet y servicios de la universidad.

Se trata de un enfoque bastante peculiar, ya que en este caso el mejor lugar para obtener información sobre la universidad ya no es su web, sino su aplicación móvil. Como bien dice Olsen en [48], es extremadamente inusual tener el gran ecosistema móvil que ellos tienen.

Como bien se puede apreciar en la figura 2.25, la aplicación se parece mucho a Layar. El motivo es que su desarrollo incluye *Layar Player*, el SDK de Layar para iOS que proporciona su motor de RA para aplicaciones de terceros. Este es un proceso costoso, ya que conlleva el desarrollo completo de una aplicación, el cual no es trivial.

Aunque hay otra opción que es más comúnmente usada en las universidades y más sencilla, que es crear una capa de Layar. El concepto es muy parecido, la diferencia estriba en que al crear una capa para

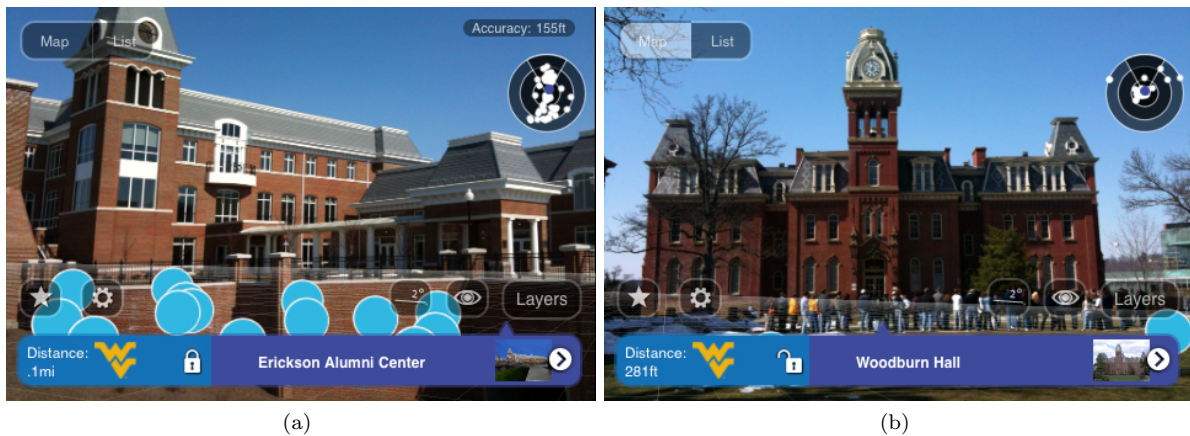


Figura 2.25: Imágenes de la aplicación para iOS de la Universidad de West Virginia.

Layar no se crea una aplicación propia, sino un complemento para otra aplicación. Esto implica que la funcionalidad quede reducida a la que ofrezca Layar en la producción de capas de realidad aumentada, y evidentemente la integración con la intranet de la universidad no es una de ellas.

Por otra parte también hay universidades que prescinden de usar herramientas ya existentes, como Layar o Wikitude, y desarrollan su propia aplicación desde cero. Este es el caso de la Universidad de Paderborn [21], en Alemania, que ha desarrollado una aplicación de RA como guía del campus. Aunque lo han hecho con marcas en vez de geolocalización, lo cual no es el enfoque más apropiado de cara a un incremento en los datos a mostrar, ya que para cada punto hay que crear una marca. Pero prueba que no hay una única manera de plantear el reto de “aumentar” los campus de las universidades, existen más opciones a parte de las dos grandes plataformas de RA, Layar y Wikitude.

Capítulo 3

Análisis

3.1. Metodología

Dentro de la ingeniería del software existen diferentes metodologías de desarrollo, que según el tipo de proyecto pueden ser más o menos adecuadas. Éstas son algunas de las más importantes descritas por Sommerville en [41], entre las que está la usada en este proyecto.

- **Cascada:** Desarrollo lineal dividido en fases secuenciales que incluyen análisis, diseño, implementación, testeo, integración y mantenimiento.
- **Prototipado:** Basado en la creación de prototipos incompletos de diferentes partes del proyecto.
- **Incremental:** Se puede definir como una serie de cascadas, donde para cada componente del sistema se sigue el modelo de cascada, pero no se realiza con todos los elementos simultáneamente.
- **Espiral:** Se focaliza en el análisis del riesgo. Para asegurar que todos los requisitos se cumplen se crea un círculo de ejecución de las fases de planificación, análisis, evaluación y desarrollo, repitiéndolas una y otra vez hasta el final.
- **RAD:** Desarrollo rápido de aplicaciones (Rapid Application Development). La clave es conseguir una calidad aceptable en poco tiempo para entregar cuanto antes. Da más importancia al desarrollo de negocio que al proceso de ingeniería, que puede ser modificado en posteriores actualizaciones.

Antes de explicar cuál es la metodología usada, hay que enfatizar que este proyecto es un prototipo, no es una aplicación lista para entrar en el mercado. Pero hay muchos tipos de prototipos, más y menos desarrollados, por lo que también hay que matizar que es un prototipo muy cercano a un producto final, ya que el principal impedimento para que no se considere como listo para el mercado es que utiliza una base de usuarios ficticia. Además, debido a la falta de recursos el servidor utilizado es la propia máquina donde se ha desarrollado el proyecto, es decir, mi portátil. De cara a la salida al mercado habría que contratar un servicio de alojamiento donde establecer la base de datos remota, haciendo una copia de la ya existente.

Considerando entonces que es una aplicación que se encuentra en los límites del prototipado, se obviará la metodología de prototipado, ya que aunque tenga el mismo nombre no ha sido el proceso seguido. El proyecto ha seguido una metodología incremental, dividido en componentes cada uno de los cuales ha seguido un pequeño proceso en cascada.

El hecho de que la definición de esta metodología indique que cada componente tendrá las fases de un desarrollo en cascada no tiene por qué acatarse literalmente. En algunos componentes hay ciertas fases que son más exhaustivas que otras, y tampoco están aislados los unos de los otros. Los componentes se tienen que integrar entre sí, por lo que sí que ha de existir una interacción entre sus fases y en cierto modo han de ser iterativas.

3.2. Descripción detallada

La descripción detallada de la aplicación pretende explicar cuáles van a ser los modos de funcionamiento de ésta. Para ello van a ser explicadas las diferentes vistas o pantallas en las que se puede actuar y cuáles son sus opciones, relacionándolas con lo ya explicado en la descripción general de la sección 1.3. Por tanto, ésta es una explicación del “qué”. Qué se va a llevar cabo y qué herramientas o algoritmos harán falta para ello.

En la aplicación podrán distinguirse dos tipos de actividades, las de visualización y las de manipulación o acción. Las de visualización son las que muestran información, bien sea en realidad aumentada o bien sobre un mapa. Y las de manipulación o acción el resto, en las que se trabaja con la información que en las de visualización es mostrada. El término actividad no se refiere a un campo de acción en sí, sino a lo que en Android se conoce como *Activity*, que ha sido previamente explicado en el punto 2.2.2. Cada una de estas actividades supone una vista o ventana de la aplicación, y por tanto una interfaz gráfica. A continuación se detalla el funcionamiento de la aplicación por cada actividad, que es una buena manera de diferenciar los procesos de acción y funcionalidades que se ofrecen.

3.2.1. Inicio de sesión

Es la primera pantalla que se ve al inicio de la aplicación. Sin embargo no quiere decir que sea la primera actividad en ejecutarse, ya que esta será llamada desde la actividad principal, que será la del menú. Y se lanzará si y sólo si no hay ya almacenadas unas credenciales de una sesión anterior en la memoria. En tal caso no se requeriría al usuario que volviese a introducir su DNI y PIN, ahorrándole este a menudo incómodo trámite.

Entonces, esta sencilla vista necesita del usuario varias cosas:

- **DNI:** Puesto que es una aplicación de la universidad, y el propósito de la aplicación final sería integrarla completamente en el ecosistema de datos de la UPV, se requerirá que el inicio de sesión se haga con las credenciales oficiales de la universidad. No hay nombres de usuario como en redes

sociales al uso, sino que se usa el DNI como identificación personal. Esto puede incitar a pensar que de esta manera se cierran las puertas a los usuarios extranjeros, pero nada más lejos de la realidad. Los miembros extranjeros de la UPV, ya sean alumnos, docentes o personal de servicios son dotados con un DNI de uso interno que sólo sirve para el ámbito universitario. Éste es ligeramente diferente al DNI común puesto que contiene más caracteres, pero igualmente válido para iniciar sesión en la intranet de la UPV, que es de lo que se trata.

- **PIN:** Es la otra parte de la identificación del usuario en el inicio de la sesión de la UPV. Obviamente, y dado que esta aplicación es un prototipo, estos datos no son reales sino que se trabaja sobre datos ficticios que se han creado para hacer pruebas.
- **Almacenar credenciales:** Una casilla de verificación o *checkbox* con el fin de guardar las credenciales, de manera que el usuario no tenga que introducirlas la próxima vez que inicie la aplicación. Al marcar esta opción se guardarán estos datos en la memoria de datos de la aplicación, la cual es sólo accesible por el código fuente de la misma. Este procedimiento es totalmente seguro y recomendado en la documentación de Android.

Como los datos de sesión que se usan no son reales, hay que sacarlos de algún otro lugar. Con este fin se incluirá una tabla con usuarios en la base de datos remota, hacia la cual irán dirigidas las consultas de verificación de las credenciales desde la aplicación.

Llegado el momento de elevar este prototipo al nivel de aplicación disponible para cualquier usuario, habría que integrar el inicio de sesión con el sistema de la UPV, o bien usar el que ya se existe en el prototipo y habilitar algún método para el registro de usuarios. En este último caso, si aún así se quisiese limitar el acceso a la comunidad y restringirlo a las personas pertenecientes a la universidad, habría que usar un proceso de verificación que incluyese la comprobación por e-mail, siendo éste perteneciente al dominio de la universidad.

3.2.2. Menú inicial

Es la pantalla principal de la aplicación. Desde ella se tendrá acceso a todas las opciones que ofrece el sistema.

Por tanto, debe proveer formas de acceso a las siguientes pantallas:

- **Inicio de sesión:** Si en un momento dado, después de que el usuario haya estado usando la aplicación sin iniciar sesión, cambia de opinión y necesita identificarse, deberá poder acceder a ello desde aquí. Pero dado que no es una opción principal, el modo de acceder no sería mediante un botón en la pantalla, sino mediante un botón en un menú de opciones clásico de Android como el de la figura 3.1.

- Preferencias: El menú principal debe ser el lugar desde el que poder acceder al menú de preferencias, como se hace en la mayoría de aplicaciones. Se puede decir que es un estándar en Android. Al igual que la anterior, su acceso deberá ser por un botón en el menú de opciones de Android y no por un botón en pantalla.
- Perfil de usuario: Una de las opciones principales de la aplicación. Aunque sólo se debe permitir el acceso a esta pantalla si el usuario ha iniciado sesión. Al contrario que las anteriores, esta opción sí requiere de algún tipo de botón en este menú principal.
- RA: Es el foco principal de la aplicación, ha de poder accederse desde el inicio. Para ello se mostrará una lista con las capas de RA disponibles en la aplicación. Pulsar cualquiera de ellas dará acceso a su vista en RA. Tiene sentido que esto sea así puesto que es la actividad principal de la aplicación.
- Mapa: Otra de las principales opciones, también ha de poder accederse desde el menú de inicio. Por tanto también deberá poseer un botón en la pantalla para tal fin.



Figura 3.1: Menú de opciones de Android.

Que sea la pantalla principal no quiere decir que sea la primera en visualizarse al inicio, ya que la que lo hará será la pantalla de inicio de sesión. En el caso de que la aplicación tuviese ya unas credenciales previamente almacenadas, entonces el menú principal sí que sería la primera actividad que se muestra al iniciar la aplicación.

3.2.3. Opciones de configuración

La mayoría de opciones son ofrecidas en tiempo real durante la ejecución de la aplicación, por lo que no se necesita de un largo menú de preferencias a configurar. Esto es, las actividades principales, como son la vista en RA, el mapa y el perfil de usuario, ofrecen sus opciones en un menú. Entonces, las únicas opciones que quedan para la configuración son el borrado de credenciales guardadas en la aplicación y la información sobre la aplicación, lo que comúnmente se conoce como “Acerca de...”.

En cuanto al borrado de credenciales, el usuario puede requerir de esta opción por motivos diversos, y es necesario que pueda decidir borrar sus datos igual que decidió guardarlos.

3.2.4. Vista en realidad aumentada

Mostrará en realidad aumentada la capa seleccionada desde el menú principal. Para ello se leerán los datos sobre los puntos a mostrar y se situarán en su posición geográfica, dejando al usuario rodeado de ellos. Se podrán inspeccionar dichos POI, de manera que al hacer click sobre ellos se muestre en la parte inferior de la pantalla una pequeña ventana con la información relativa al punto. Uno de los datos proporcionados será la distancia existente entre el usuario y el punto. Además, si entre dichos datos se encuentra la información de contacto a dicho punto de información, la aplicación proveerá los medios necesarios para abrir con dichos datos las aplicaciones pertinentes. Es decir, si hay un teléfono disponible la aplicación permitirá llamar a dicho teléfono, para lo cual abrirá la aplicación Teléfono del dispositivo con el número ya marcado. Lo mismo sucedería con otros datos como el e-mail o la página web. En el caso del primero, la aplicación permitirá abrir el gestor de correo del dispositivo con la pantalla de enviar un correo y con el destinatario ya rellenado con el e-mail del POI. En el otro caso, la aplicación permitirá abrir el navegador de internet con la web en cuestión.

Asimismo, gracias al sistema de funcionamiento de Android, en el que las actividades son pausadas y apiladas cuando una nueva se abre a partir de otra, siempre se puede volver a la aplicación original. Esto es, si el usuario decide contactar con uno de los POI mostrados en RA, y por tanto es llevado fuera de la aplicación, si luego se echa atrás y quiere volver a donde estaba, sólo necesita pulsar el botón de retroceso. Al hacer esto, se destruiría la actividad en la que estuviese, y Android sacaría de la pila la última actividad que fue apilada, que pertenecería a la aplicación de este proyecto.

En cuanto a la visión de la RA en sí, se mostrará el título o nombre de cada POI junto con un icono, de forma que sólo con verlo ya se tenga una idea de lo que es y no sea necesario inspeccionarlo. Clarificar el espacio visual es muy importante en este tipo de sistemas, ya que si no puede ser contraproducente. Tener muchos iconos en pantalla y no saber qué son o no poder averiguarlo fácilmente no contribuye a una interfaz usable. Por ende, otro añadido a este fin es el poder seleccionar a placer la cantidad de información que es mostrada por pantalla. La forma más eficaz de hacerlo en este contexto es por medio de las distancias. Dejar al usuario elegir el radio de extensión en el que quiere ver información, siendo él el centro del mismo.

Por último, se proveerá un radar que muestre la situación de los puntos que rodean al usuario dentro del rango especificado. Esta funcionalidad es ya un clásico en la mayoría de aplicaciones de RA basadas en geolocalización que existen en el mercado, y ha probado ser muy útil de cara a la orientación del usuario. Por ello debe ser un elemento indispensable.

3.2.5. Vista mapa

De forma similar a lo que sucede en la vista de RA con su radar, esta vista también mostrará la situación de los POI pero en un mapa. Para ello se trabajará sobre las librerías de *Google Maps*, que son nativas de Android y tienen una alta integración con el sistema. Esto permitirá dibujar sobre el mapa los

distintos iconos que representen los POI de la capa a mostrar e incluso el contorno espacial que ocupan los mismos.

Hay puntos que son muy concretos cuya localización se reduce a un punto geográfico concreto. No obstante, hay otros que tienen una extensión espacial más grande y un simple icono no es apropiado para acotarlos. Este puede ser el caso de las escuelas, algunas de las cuales pueden extenderse a lo largo de varios edificios. Así pues, la mejor solución en estos caso es dibujar el contorno de los mismos además del punto en sí sobre el que se puede hacer click.

Pero la información no es lo único que hay que situar sobre el mapa. También es necesario representar al usuario, puesto que sin saber su posición no es posible una buena orientación. Esto requiere el uso del GPS de forma periódica para localizar al usuario y sus movimientos. Los POI no se pueden mover, pero el portador del dispositivo sí, por lo que este proceso ha de ser constante. En Android hay dos maneras de obtener la posición de un usuario. Una es por satélite y la otra por triangulación de redes inalámbricas. Ninguna de las dos es 100% exacta ni está asegurado que puedan funcionar siempre. En espacios cerrados la recepción de datos de GPS por satélite es muy mala, por lo que es mejor utilizar las redes móviles, que son capaces de aproximar una posición. Y en espacios abiertos, la localización por satélite es mejor a priori. Todo esto conlleva que se necesario el uso de ambos métodos. Es más, como no son excluyentes, se pueden usar los dos a la vez y escoger la información más fiable.



Figura 3.2: Perspectivas de la UPV en Google Maps.

Habitualmente el usuario de la aplicación se encontrará dentro de la UPV, pero esto no tiene por qué ser así siempre. Pueda darse el caso en el que lo que esté haciendo sea buscar el campus, estando por tanto fuera de él. El mapa por defecto mostrará la vista aérea de la UPV, pero se puede mover y navegar libremente por él, ya que se trata de *Google Maps*. El icono del usuario será dibujado igualmente, aunque no quede dentro del espacio de la universidad. Y para encontrarlo, existirá un botón que centre el mapa en la ubicación del usuario, permitiendo así conocer ambas posiciones. De igual manera, para no tener que desplazar el mapa manualmente hasta la zona de la universidad otra vez, se dispondrá de otro botón que centre el mapa en dicho espacio. Estos dos botones pertenecerán a un menú de opciones,

ya que así se mantiene la pantalla libre de objetos molestos.

Aún hay dos opciones más. Una será poder cambiar la vista entre satélite y mapa, como es habitual en *Google Maps*. La vista mapa es una representación gráfica de la realidad y la vista satélite es un compuesto de fotos del espacio tomadas desde un satélite y escaladas en consecuencia. Para aclarar las dudas que puedan suscitar este concepto, la imagen 3.2 muestra la UPV en ambas perspectivas.

Por último y no menos importante, será posible cambiar la capa que es está viendo. Como ya se comentó en la descripción general de la sección 1.3, la aplicación consta de seis capas visualizables que clasifican los POI según su categoría. Por tanto, también se dispondrá de un botón que al pulsarlo despliegue una lista con las capas disponibles para que el usuario pueda elegir la que desea ver.

3.2.6. Inspección y acción sobre objetos de realidad aumentada o del mapa

Como se indica en los dos puntos anteriores, es posible inspeccionar los puntos de información y en ocasiones realizar alguna acción sobre ellos. Aunque no especifica cómo se instará al usuario a que realice tal acción, y ese es el propósito de este apartado.

Cuando un POI sea seleccionado, se mostrará una pequeña ventana con cierta información. Y cuando esta ventana sea pulsada, se estará indicando que se desea realizar algún tipo de acción sobre ese punto. Esto causará que se abra una nueva actividad con un diálogo y una imagen de fondo. La imagen sólo se verá si existe una, con lo cual esto sólo es aplicable a la capa de *Comunidad UPV*, donde los usuarios pueden crear nuevos puntos y adjuntarles una imagen.

En cuanto al diálogo, la aplicación comprobará qué datos de contacto están disponibles en ese punto y los incluirá en una lista desplegable. Como ya se ha mencionado repetidas veces en este documento, estos pueden ser tres: teléfono, e-mail y página web. Y en cada uno de los casos al hacer click sobre él se abrirá la aplicación pertinente para establecer ese contacto. En el caso del teléfono se abrirá la aplicación Teléfono con el número marcado. En el caso del e-mail se abrirá el gestor de correo con el e-mail como destinatario. Y en el caso de la web se abrirá el navegador y cargará dicha dirección.

En el caso de que no haya una aplicación determinada para realizar las mentadas acciones, Android siempre provee un diálogo en el que permite elegir una de las aplicaciones disponibles. Esto es, si en el sistema hay más de un navegador instalado, Android permitirá elegir con cuál de ellos se desea visitar la web del punto que se ha inspeccionado. Es algo que incluye el propio sistema y que no depende de la aplicación de este proyecto, pero es menester describir su funcionamiento, pues afecta directamente al proyecto.

3.2.7. Perfil de usuario

Será accedido desde el menú principal y proporcionará información sobre los puntos creados por el usuario. Con el fin de hacerlo llamativo a la vez que funcional, y como lo normal será que los POI creados por el usuario tengan foto, la forma de navegar por los puntos creados será mediante un *Cover Flow*.

Ésta es una interfaz muy de moda actualmente gracias a Apple, que la ha popularizado por medio de sus navegadores de archivos y reproductores de música, como muestra el ejemplo de la figura 3.3. Se trata de una tira de imágenes de las cuales hay una en primer plano en el centro, y se puede desplazar todo el conjunto bien arrastrando con el dedo si es un dispositivo táctil, o bien con el ratón. El original de Apple además simula un escenario en 3D sintético, lo cual lo hace más atractivo visualmente.



Figura 3.3: Cover Flow de Apple en iTunes.

Los dispositivos con iOS disponen de tal elemento por defecto, con lo cuál incluirlo en una aplicación es sencillo. En Android en cambio tal objeto no existe, así que es necesario crearlo con las herramientas que proporciona el sistema. Esto es perfectamente posible, y se puede conseguir algo funcional, aunque quizá no tan vistoso como el de la compañía de Cupertino.

Entonces, la pantalla constará del mencionado Cover Flow donde se mostrarán las imágenes de los POI creados por el usuario. Y la imagen que esté en el centro, y por tanto seleccionada, será la que ofrezca su información. Una tabla con los datos de cada POI será mostrada cada vez que se manipule la tira de imágenes del Cover Flow. Será un mecanismo muy funcional y amigable para el usuario.

Para finalizar, también será posible acceder a la creación de nuevos puntos de interés. Existirá un menú con un botón para tal fin, que llevará al usuario a una actividad específica para ello.

3.2.8. Creación de puntos de información

Ésta es la actividad a la que se refiere el punto anterior cuando se pretende crear un nuevo POI. La forma más sencilla de explicar esta actividad es con una palabra, formulario. Porque eso es lo que se necesita, un conjunto de campos a rellenar por el usuario donde se describa todos los datos el ítem a crear. Los campos serán:

- **Título** del POI. Por supuesto es un campo obligatorio. No se podrá enviar el formulario si este campo no está relleno.
- **Descripción** del mismo. También es un campo obligatorio.
- **Imagen**: Se podrá adjuntar una imagen para que el resto de los usuario la vea cuando inspeccionen

dicho POI. Para adjuntar la imagen se dispondrá de dos métodos. Uno en el que se abra la galería de imágenes de Android y se seleccione una de ellas. Y otro en el que se abra la aplicación Cámara y se saque una foto que será la que se adjuntará. Ambos son procesos estándar en el sistema, por lo que se hará uso de ellos.

- **Ubicación:** Las localizaciones con las que tratará el sistema son coordenadas, latitud y longitud. Y no se espera que el usuario sepa las coordenadas en las que quiere posicionar su POI, por lo que la ayuda de un mapa es necesaria. Por tanto se establecerá la ubicación por medio de un mapa en otra actividad.

Es un campo esencial, y por tanto obligatorio.

- **Teléfono** de contacto para tal POI.
- **E-mail** de contacto para tal POI.
- **Web** de contacto e información sobre tal POI. Al menos un tipo de contacto deberá ser establecido, ya sea teléfono, e-mail o web. Al enviar el formulario se comprobará que al menos uno de estos campos ha sido rellenado.
- **Fecha:** Los POI creados por un usuario no pueden ser mostrados siempre, ya que lo más lógico es que estos queden desfasados en cierto periodo de tiempo. Además, algunos de los POI que el usuario cree pueden hacer referencia a un día en concreto en el que algún acontecimiento tomará forma. Este campo permite determinar esa fecha y hará que el POI sólo sea mostrado hasta tal día. Además, sólo se podrán crear puntos a 3 meses vista, el cual será el tiempo máximo que se muestre cada uno.

Por otro lado, este no es un campo obligatorio, por lo que el usuario no tiene por qué rellenarlo. Sin embargo, esto no significa que por esa razón el POI pueda permanecer más de tres meses a la vista del resto de usuarios. Si no se rellena ese campo, se tomará la fecha de creación y se le añadirán tres meses, obteniendo así la fecha límite. Se trata entonces de una fecha de caducidad que permite que el sistema no se colapse y que la información no quede desfasada.

Al enviar el formulario se comprobará que los campos que se han descrito como obligatorios estén rellenos, y si no lo están se avisará pertinentemente. Si lo están, entonces se contactará el servidor remoto para enviar la información del nuevo POI. Al volver al perfil de usuario, el nuevo POI estará visible.

3.2.9. Selección de la geolocalización de un punto creado por el usuario

Esta es la actividad que permite seleccionar la ubicación de un nuevo POI como se ha mencionado en el punto anterior. Para ello se hará uso una vez más de los mapas de Google Maps y sus librerías. El mapa mostrará la vista aérea de la UPV con un marcador en el centro que marcará la ubicación del

futuro POI. Para cambiarlo de lugar bastará con arrastrarlo con el dedo por el mapa hasta dejarlo en la posición deseada. Una vez conseguido esto, habrá que fijar la ubicación por medio de un botón que la enviará de vuelta al formulario de creación de nuevos POI.

Al volver al formulario, gracias al flujo de actividades de Android, los campos que ya habían sido rellenados seguirán igual, ya que la actividad habría sido pausada, que no destruida. Es decir, cuando se accede al mapa de selección de la ubicación, la actividad del formulario queda pausada y apilada en el entorno de ejecución del sistema. Cuando la actividad de selección de la ubicación finaliza, esta envía el resultado a la última actividad en la pila, que es el formulario, y ésta se reanuda en el mismo estado que estaba cuando se pausó.

3.3. Flujos de información

Esta sección pretende ilustrar la descripción de la sección anterior, proporcionando un diagrama de flujo entre las actividades que explique su estructura. Este diagrama es de la figura 3.4, que contiene todas las actividades mencionadas en en la sección anterior y la interacción entre ellas.

Se puede apreciar que al inicio de la aplicación se procesa la actividad del menú de inicio, y dependiendo de si hay credenciales en memoria o no, se redirige al inicio de sesión. Después de esto, la aplicación puede estar en dos estados, con sesión iniciada o sin sesión iniciada. Desde el primero se pueden acceder a todo el resto de pantallas. En cambio desde el segundo no, ya que al perfil de usuario sólo se puede acceder si se ha iniciado sesión.

Asimismo, se indica cuándo se descargan datos de la base de datos en el servidor remoto, que es al acceder a las vistas de RA y del mapa. Y también se muestra la comunicación inversa, es decir, cuando se guarda algo en las bases de datos, tanto interna como remota, que es cuando se crean nuevos POI. En el próximo capítulo, el de diseño e implementación, se profundizará en la interfaz que se usa para establecer estas comunicaciones con el servidor remoto que tan importantes son.

En cuanto a las salidas de la aplicación, están coloreadas en amarillo oscuro y se producen cuando la aplicación llama a otra por algún motivo. Los motivos son los ya explicados anteriormente, realizar una acción de contacto o adjuntar una foto a un nuevo POI. Y como ya se recalca también anteriormente, es posible retornar a la aplicación después de salir de ella gracias al sistema de pilas que utiliza Android. Para clarificar este concepto se proponen varios flujos de ejemplo de la aplicación, que son:

- Figura 3.5: Cuando se crea un nuevo POI, hay un momento en el que se ha de pausar la actividad del formulario para seleccionar la ubicación del punto. En ese momento, la actividad es apilada para que se pueda ejecutar la siguiente, pero ninguno de los campos que habían sido rellenados se borran, pues se quedan guardados en la memoria.
- Figura 3.6: Cuando se inspecciona un POI en la vista de RA, ésta se apila y se abre la actividad de inspección. Cuando se selecciona una opción del diálogo que ofrece esta última, se sale de la

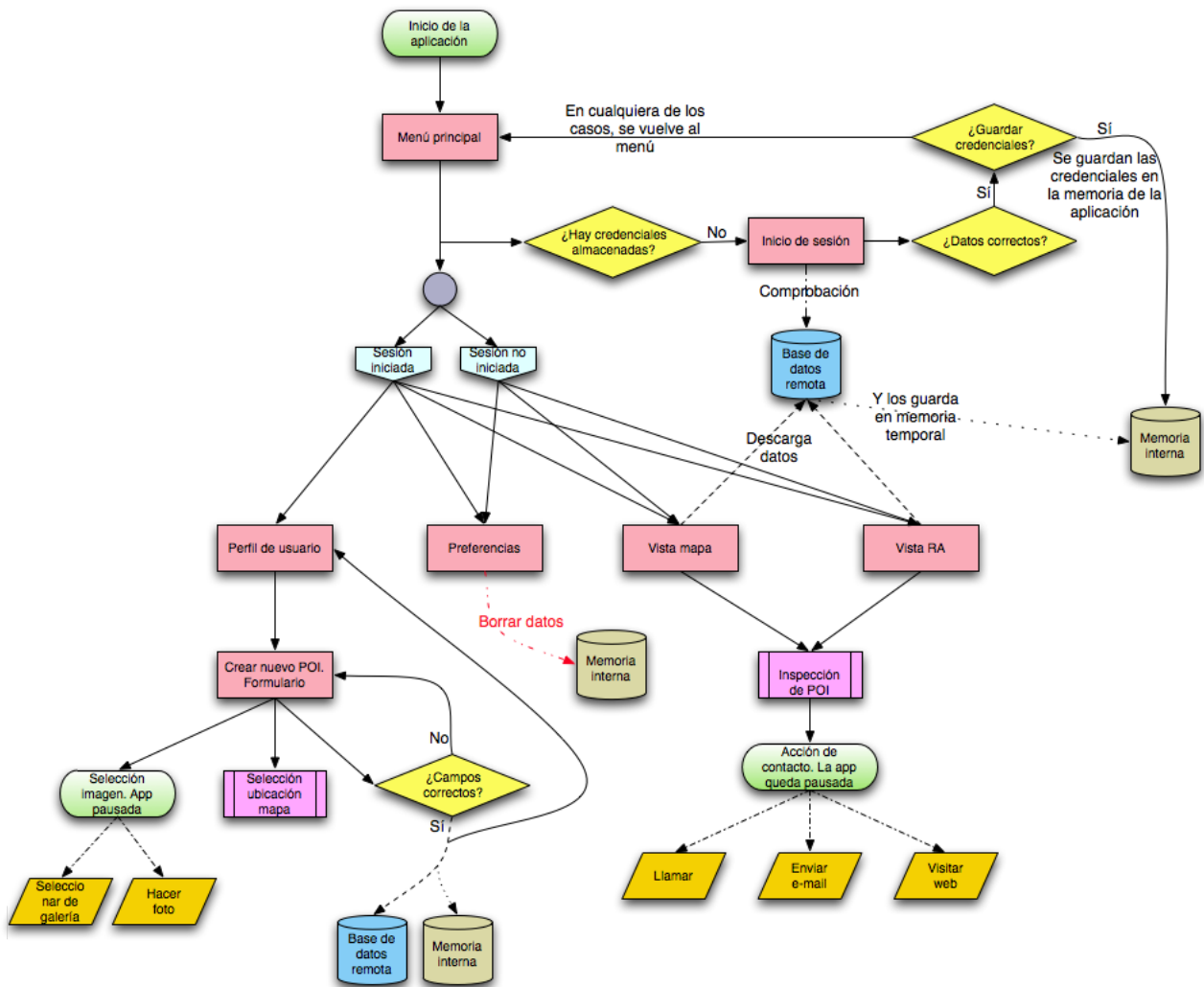


Figura 3.4: Diagrama de flujo de la aplicación.

aplicación y se apila la última actividad de la aplicación. En el ejemplo de la figura, se ve como se realiza la acción de llamar con su consecuente proceso. Al volver a la vista de RA que estaba apilada, las actividades que estaban por encima en la pila finalizarán y la vista en RA volverá a mostrar la misma capa de puntos que mostraba antes de ser pausada.

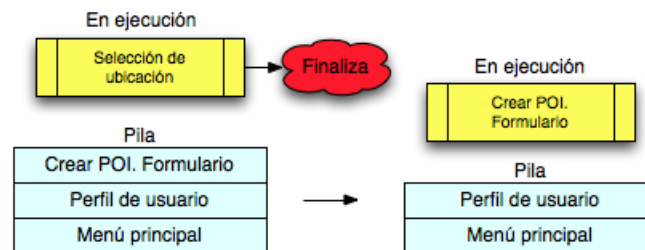


Figura 3.5: Ejemplo de uso de la pila de Android número 1.

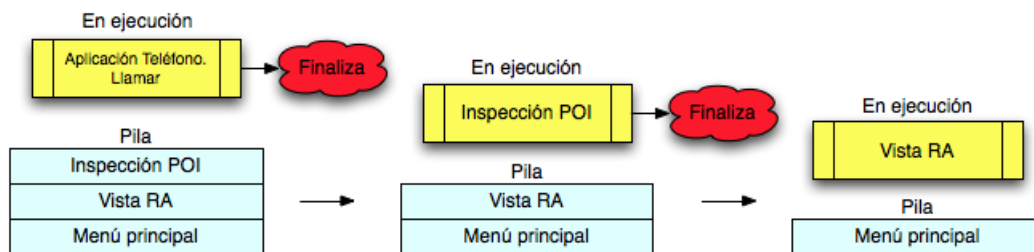


Figura 3.6: Ejemplos de uso de la pila de Android número 2.

3.3.1. Flujo de datos desde el servidor

Las consultas que se realizan al servidor la mayoría de las veces esperarán como respuesta un gran volumen de datos, por lo que el análisis de la situación no es trivial. No basta con enviar todos los datos a la vez, aunque ésta sea la opción más fácil a la hora de programar.

Tampoco tiene sentido el sobrecargar las vistas con demasiados POI, pues puede crear confusión en el usuario. Por este motivo, se ha establecido un límite de 50 POI simultáneos. Pero entonces surge varias cuestiones. ¿Qué puntos son los elegidos para ser mostrados? ¿Quiere decir esto que el resto de POI no serán descargados?

Todos los puntos serán descargados en hilos diferentes al principal, hilos secundarios o como se suele decir en inglés *background threads*. Y todos ellos serán guardados en memoria en la base de datos interna de la aplicación, siempre y cuando haya suficiente espacio. Si no lo hay, la aplicación hará limpieza y borrará datos, lo cual no es un problema pues los datos locales son solo una copia de los datos del servidor. Entonces, lo que cambia, ¿qué es? Lo que cambia es la prioridad en la descarga y en la visualización.

Estando el límite en cincuenta, habrá que priorizar este número de POI para que sean descargados primero. El elemento determinante para ello será su proximidad al usuario, ya que lo más interesante

para el usuario será lo que tenga más cerca, lo que tenga opción casi de tocar. Y esto mismo ocurrirá a la hora de mostrarlos, es decir, si la aplicación tiene en memoria doscientos POI, tendrá que priorizar y procesar un cuarto de ellos solamente. Para ello será necesario utilizar fórmulas matemáticas que tengan en cuenta las distancias en un plano curvo, pues la superficie de la tierra es una esfera.

Existen varios métodos para el cálculo de la navegación astronómica:

- **Pitagórico:** Basado en la relación euclídea entre los tres lados de un triángulo rectángulo. Considera que la tierra es plana, por tanto es el menos eficaz.
- **Ley esférica de los cosenos:** Teorema que relaciona los lados y ángulos de triángulos esféricos, de forma análoga a la que lo hace la ley de los cosenos en trigonometría bidimensional.
- **Haversiano:** Es el segundo más certero y el más robusto ante errores en coma flotante. Tiene un coste de proceso más que aceptable.
- **Fórmulas de Vincenty:** Relación de dos métodos iterativos usados en Geodesia para calcular las distancias entre puntos de un esferoide. Es el que proporciona resultados más certeros, pero tiene un alto coste de computación que no merece la pena pagar para el fin para el que se necesita.

El método más apropiado y además recomendado para estos casos, como sugiere Rubin en [37], es el de la **fórmula de Haversine** o del semiverseno. Combina fiabilidad en los resultados, robustez y un buen tiempo de proceso.

Calcula la distancia entre dos puntos en un esferoide, la Tierra en este caso. Y como es obvio, son necesarias las coordenadas de ambos puntos, latitud y longitud. Así pues, para dos puntos de una esfera de radio R , con latitudes φ_1 and φ_2 , con una distancia entre ellas $\Delta\varphi = \varphi_1 - \varphi_2$, y diferencia de longitudes $\Delta\lambda$, las distancia d entre ellas es:

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\Delta\varphi) + \cos(\varphi_1)\cos(\varphi_2)\text{haversin}(\Delta\lambda)$$

donde:

$$\text{haversin}(\theta) = \frac{\text{versin}(\theta)}{2} = \sin^2\left(\frac{\theta}{2}\right)$$

$$\text{versin}(\theta) = 1 - \cos(\theta) = 2\sin^2\left(\frac{\theta}{2}\right)$$

Una vez se conoce la manera de calcular esta distancia, hay que repetirla con todos los POI y con la ubicación del usuario. Las cincuenta menores distancias serán el resultado de los puntos más cercanos al usuario. Esto se puede traducir en un lenguaje de consultas como SQL de la siguiente manera:

Listing 3.1: Descriptive Caption Text

```
1 SET @lat = 39.48265;
2 SET @longi = -0.34694;
```

```
3 SET @dist = 0.3;
4
5 SELECT *, X(location) as longitud, Y(location) as latitud,
6 6378 * 2 * ASIN(SQRT(POWER(SIN((@lat - abs(X(location))) * pi()/180 / 2),
7 2) + COS(@lat * pi()/180 ) * COS(abs(X(location)) *
8 pi()/180) * POWER(SIN((@longi - Y(location)) *
9 pi()/180 / 2), 2) )) as distance
10 FROM poi
11 having distance < @dist
12 ORDER BY distance limit 50
```

3.4. Herramientas y librerías

Hay multitud de herramientas y librerías que pueden ser útiles para un proyecto de estas características, y como no se trata de reinventar la rueda, lo más acertado es hacer uso de aquellas que supongan una ventaja respecto a un desarrollo propio. Se puede hablar entonces del equivalente a un *mashup* en el desarrollo web, donde se integran todo tipo de librerías abiertas en el desarrollo.

3.4.1. Android

Poco queda decir sobre esta plataforma que no se haya mencionado ya en la sección 2.2, donde se ponen a la vista las virtudes del sistema. Aunque se pueden citar las razones principales por las que es una buena elección para este proyecto:

- Es software libre, abierto a la revisión de cualquiera.
- Tiene la mayor cuota del mercado móvil. Casi la mitad de los dispositivos móviles en el mundo son Android. Y aunque el índice de beneficios por aplicaciones descargadas sea favorable a iOS, en principio no preocupa pues esta aplicación sería gratuita.
- Se desarrolla con Java, un lenguaje muy conocido en la comunidad de desarrollo software.
- Gran cantidad de APIs de libre distribución disponibles.
- Gran variedad de dispositivos móviles donde elegir.
- Google tiene una política mucho más laxa que Apple en cuanto a la publicación de aplicaciones en el Android Market.

3.4.2. MySQL y PHP

La aplicación necesita de un servidor remoto que le sirva datos para poder funcionar, y existen muchas opciones disponibles para desarrollar uno. Para el tipo de peticiones que se van a realizar, las cuales no serán especialmente complicadas, PHP es la mejor opción.

- Es software libre.

- Es sencillo de poner en funcionamiento. Con poco código se pueden hacer muchas cosas.
- Orientado a objetos.
- Se puede complementar con muchas librerías de libre distribución.
- Tiene una gran comunidad detrás que da soporte y documentación.
- Buena integración con MySQL.

Si bien la decisión de usar PHP parece sencilla, la del sistema de bases de datos que usar no lo es tanto. Se presentan muchas más opciones y hay varias muy recomendables. Si hiciésemos una selección y dejásemos dos, éstas serían MySQL y PostgreSQL. Porque los dos son software libre. Las dos dan soporte para Mac OS X, que es donde va a ser instalado el servidor. Tienen extensiones espaciales, para manejo de datos geográficos. Y tienen buenas interfaces gráficas.

Ahora bien, ¿por qué MySQL en vez de PostgreSQL? Si se investiga en internet ambos sistemas, en seguida se llega a la conclusión de que PostgreSQL es más completo en cuanto a funciones y opciones. ¿Quiere decir esto que sea mejor? No. MySQL es algo más sencillo en cuanto a su arquitectura, pero esto también lo hace más sencillo de usar. Es más rápido en consultas en bases de datos sencillas y además es más amigable para el usuario, más sencillo de aprender. La conclusión que se puede sacar de esto es que PostgreSQL es la mejor elección si se trata de un sistema crítico, con grandísimas cantidades de datos. Si en cambio lo que se necesita es algo más estándar, que sea funcional pero sin ser complicado, y que sea rápido y eficaz, la mejor elección es MySQL.

La aplicación de este proyecto encaja más con la segunda descripción, por lo que MySQL es la mejor opción. Además, como usuario ya tengo alguna experiencia con él, con lo cual la curva de aprendizaje disminuye. Como complemento, hay muchas y muy buenas interfaces gráficas para MySQL. La elegida para éste sistema es *MySQL Workbench*, que está desarrollada por la misma empresa, y que se integra perfectamente con el entorno de Mac OS X. La razón de utilizar Mac OS X es que es la única máquina disponible para tal fin, así que se puede considerar un requisito.

3.4.3. API de Google Maps API

Las razones por las que usar la API de Google Maps parecen más que obvias. Es el gran dominador del mercado comercial en cuanto a librerías de mapas se refiere, y además sigue siendo Google. Es decir, su integración con Android está garantizada.

Pocos rivales aparecen en este campo. Uno que está afianzándose es *Bing Maps* de Microsoft, que ya tiene un SDK para Android. Pero el hecho de que sea algo muy reciente y que no sea de Google no incita a tomar el riesgo de desarrollar con él. Por tanto Google Maps es el claro ganador.

3.4.4. API de Wikitude

Esta es la herramienta más determinante en el desarrollo de esta aplicación, ya que proporciona un motor de realidad aumentada capaz de tomar como entrada puntos de información y ofrecer opciones de actuación sobre ellos. Y esto es básicamente lo que necesita la aplicación, por lo que desarrollar algo similar sería de necios.

Aunque el hecho de ser esencial, no implica que sea la única. Hay otras librerías disponibles para trabajar con realidad aumentada en Android, pero no son tan robustas. Entre ellas hay proyectos de software libre con repositorios abiertos en Google Code y Github como pueden ser Droid-AR [1] o Android-AR-Kit [20]. Aunque estos no están tan orientados la navegación de la realidad como lo está Wikitude, ya que se centran más en el trabajado con marcas de RA.

Por otro lado está Layar, para el que se pueden crear capas de POI. Esto se descarta igual que se descarta hacer lo mismo pero en Wikitude, pues no ofrece tanta libertad de desarrollo, ya que la funcionalidad queda muy acotada. En la sección 2.4.1 se habla de Layar Player, cuya funcionalidad es similar a la que ahora mismo ofrece la SDK de Wikitude, y podría haber sido la opción elegida de no ser porque sólo está disponible para iOS. De hecho, contacté con los desarrolladores de Layar para saber si la versión para Android sería publicada en el futuro cercano. La respuesta fue que estaban trabajando en ello, pero que podrían tardar un año en publicarla, por lo que había que descartarla. Por tanto queda Wikitude solamente, que no es perfecta, pero cumple muy bien con lo que se requiere para esta aplicación.

3.4.5. Extensiones

En este documento se ha repetido hasta la saciedad que Android es un sistema libre y que tiene una gran comunidad detrás desarrollando para él todo tipo de librerías y extensiones o *plugins* con licencia GPL¹. Así que tomando partido en ello y aprovechándolo, se hará uso de algunos proyectos encontrados en la web, concretamente en Github, que mejoran la interfaz gráfica de Android.



Figura 3.7: Android-ActionBar.

El primero es el Android-actionbar desarrollado por Johan Nilsson [30], que da su bendición a todo aquel que desee usar su creación en sus proyectos. Se trata de un objeto que es una barra de acción customizable. La barra de acción es el elemento superior de la pantalla que suele contener el título de la aplicación y ocasionalmente algún botón. Este elemento no se introduce por defecto en el SDK de Android hasta la versión 3.0, que va dirigida a tabletas. Este proyecto se desarrolla en la versión 2.2,

¹GNU General Public License. Licencia creada por la Free Software Foundation orientada a proteger la libre distribución, modificación y uso de software

que es la versión estable más avanzada al inicio del mismo, y no tiene este tipo de objetos. Por lo que el proyecto de Nilsson de es gran ayuda.

El otro es de Jeff Gilfelt, Android-MapViewBalloons [14], que soluciona la tesitura de incluir marcadores en Google Maps que al pulsarlos abran una ventana con su información. Google Maps para Android no dispone de esa opción aún, y esta extensión resuelve el problema por completo.

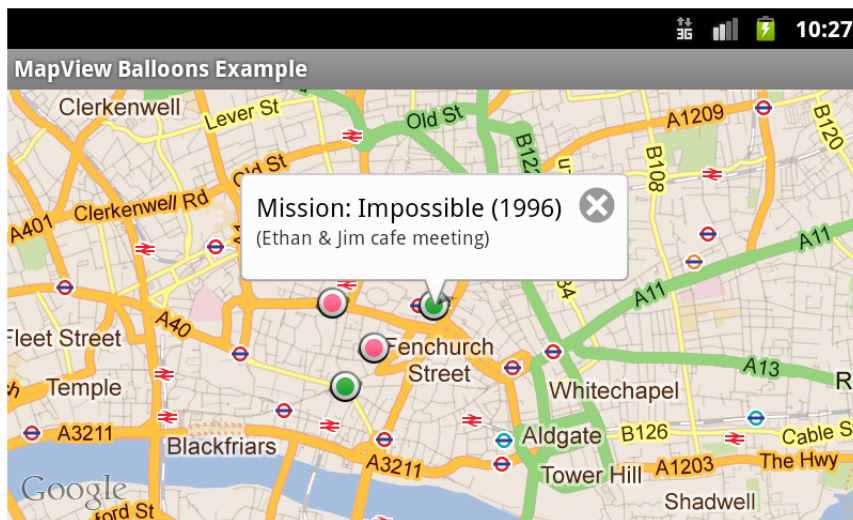


Figura 3.8: Android-MapViewBalloons.

Capítulo 4

Diseño e implementación

4.1. Proceso de desarrollo

Como ya se ha relatado en la sección 3.1, la metodología utilizada durante todo el proyecto ha sido la incremental. Para ello se ha dividido en proyecto en componentes, para cada uno de los cuales se realizará un desarrollo en cascada y luego se integrarán. Estos son:

1. Base de datos MySQL.
2. Parte PHP del servidor.
3. Esqueleto de la aplicación, que aloje todo tipo de menús e interfaces gráficas.
4. Vista en realidad aumentada.
5. Vista mapa.
6. Parte de gestión y creación de contenidos por el usuario.

Nótese que los dos primeros pertenecen al desarrollo de la parte del servidor, y el resto pertenece a la parte del cliente, es decir, la aplicación Android. Esto propicia que la integración entre cada conjunto sea más sencilla, y que por tanto su diseño e implementación sean explicados dentro de las mismas secciones. Las secciones siguientes hacen referencia a esos dos grandes grupos, cliente y servidor, y sus subsecciones a los componentes en sí de cada una.

4.2. Arquitectura del servidor

El servidor consta de dos partes a desarrollar: la base de datos, y la interfaz del servidor que hace de enlace entre la base de datos, el sistema de ficheros y la aplicación Android. Como ya se anticipaba en el capítulo anterior, la base de datos estará desarrollada en MySQL y los scripts del servidor en PHP. Todo ello alojado en un servidor de prueba, un MacBook con procesador Intel Core 2 Duo a 2.4GHz, 2GB de memoria DDR2 SDRAM a 667MHz, y más de 5GB de espacio de almacenamiento.

Dados estos escasos recursos, y que tampoco se dispone de un dominio web, el servidor estará montado en una red local LAN, con una IP local. Y para poder contactar con él será necesario estar dentro de la misma red, es decir, tanto los dispositivos móviles como el portátil con el servidor deberán estar conectados en la misma red de área local para poder intercambiar información. La figura 4.1 muestra el caso en el que ambos están conectados a la red creada por un mismo router, y por tanto en la misma red. Las IP's usadas para ilustrar la topología de la red son ficticias y sólo se usan como ejemplo.

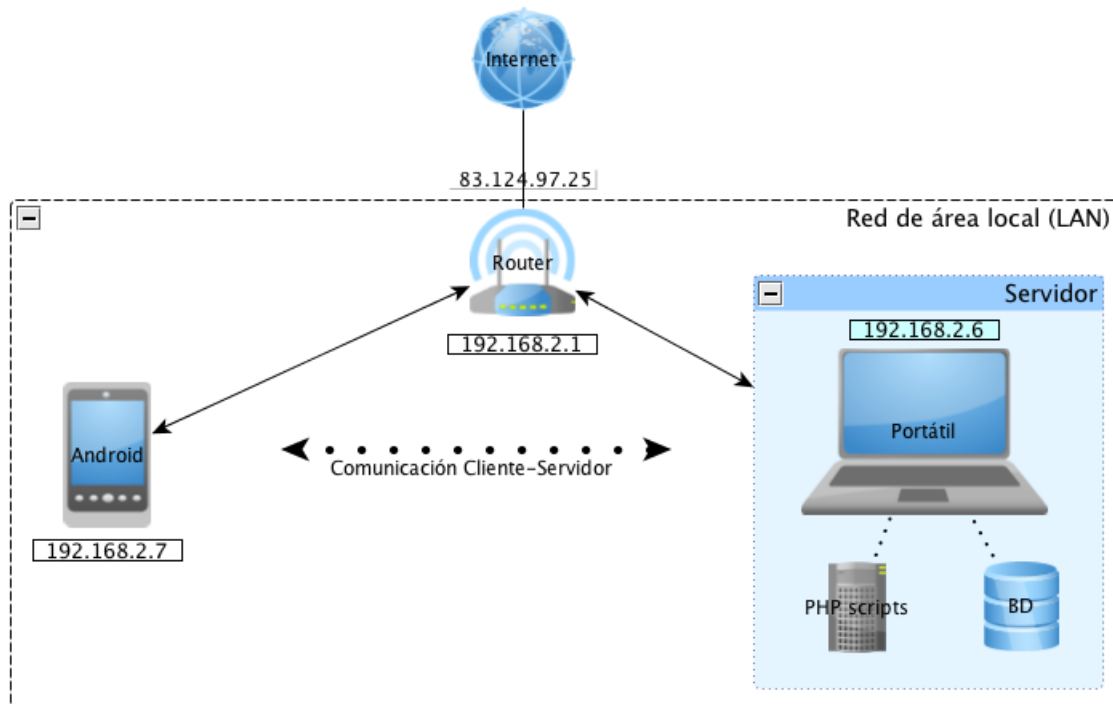


Figura 4.1: Topología de la red.

Hay algunas implicaciones en este aspecto, y es que a la hora de testar la aplicación en el escenario requerido, la UPV, también hay que llevar el servidor. Ambos deberán estar conectados por tanto a la red de la universidad para poder funcionar, de nombre *UPVNET2G*. Aunque gracias a que la aplicación guarda datos temporalmente, el campo de acción no se reduciría demasiado si hubiese un fallo en el servidor o si simplemente no estuviese en la misma red durante un momento dado.

4.2.1. Base de datos MySQL

La base de datos ha sido diseñada partiendo de la premisa de que la aplicación está sujeta a posibles ampliaciones y trabajos futuros. Desde un principio se han tenido en cuenta cuáles podían ser para incluir las tablas necesarias en la base de datos y que luego no supusiese un problema. De no ser así, quizá más tarde se incurriría en problemas de relación entre entidades, pues sería más difícil adaptar el sistema. Aunque los posibles trabajos futuros se comentarán más adelante en el capítulo 6.

En cuanto a la parte del diseño que sí es efectiva, se pueden distinguir seis tablas. La restante sería parte de posibles ampliaciones:

- **capas:** Los POI están clasificados en seis tipos de capas. En principio los valores originales no serán cambiados y no se espera añadir nuevas filas, aunque es posible hacerlo.
- **poi:** Puntos de interés. Contienen una referencia a la capa a la que pertenecen y ocasionalmente al usuario que los creó. Hay dos tipos, los proporcionados por el sistema por defecto y los creados por usuarios.
- **files:** Archivos que pueden ir enlazados a un POI. En principio sólo se procesan imágenes, pero se podría ampliar a otro tipo de ficheros, como vídeos, archivos de audio o documentos en pdf.
- **usuarios:** Tabla con los usuarios pertenecientes a la comunidad UPV.
- **escuelas:** Determina a qué escuela pertenece un usuario si es alumno.
- **departamentos:** Determina a qué departamento pertenece un usuario que es PDI¹. Puede suceder que el usuario no pertenezca a ninguno, igual que pasa con las escuelas. Las relaciones entre estas tablas no son restrictivas, no es necesario que existan.
- **mensajes:** Parte de lo que podría ser una ampliación del sistema. Que se puedan dejar mensajes asociados a los elementos en realidad aumentada. El mensaje iría asociado a un POI de una capa, y guardaría la fecha y el autor.

La figura 4.2 muestra las relaciones entre estas tablas en un diagrama entidad-relación hecho con el software de MySQL, *MySQL Workbench*. En ella se pueden ver si las relaciones entre las tablas son identificativas o no, como ya se ha descrito.

La base de datos almacenará desde el inicio todas las capas, todos los POI relativos a escuelas, departamentos, institutos, edificios y servicios, y algunos usuarios de prueba. El número inicial de puntos es por tanto superior a los doscientos, lo cual ya supone una buena dosis de información.

En cuanto a la conexión a la BD, se realizará mediante un socket, ya que están en la misma máquina todos los componentes del servidor.

Por último, es importante incidir en el tipo de almacenamiento usado en la tabla “files”. A la hora de guardar ficheros en una base de datos existen dos alternativas: guardarlos dentro de la BD como una variable *BLOB*², o guardar en la BD las referencias a la ruta donde se encuentran los archivos en el sistema de ficheros. Hay gran diversidad de opiniones respecto a este tema, incluso se han escrito artículos científicos al respecto, como el de Sears et al. [38]. Pero hay un argumento que se impone habitualmente y que tiene bastante lógica; las bases de datos se diseñaron para guardar datos y los sistemas de ficheros

¹Personal de Docencia e Investigación.

²Binary Large Object. Elemento utilizado en las bases de datos para guardar conjuntos de datos de gran tamaño.

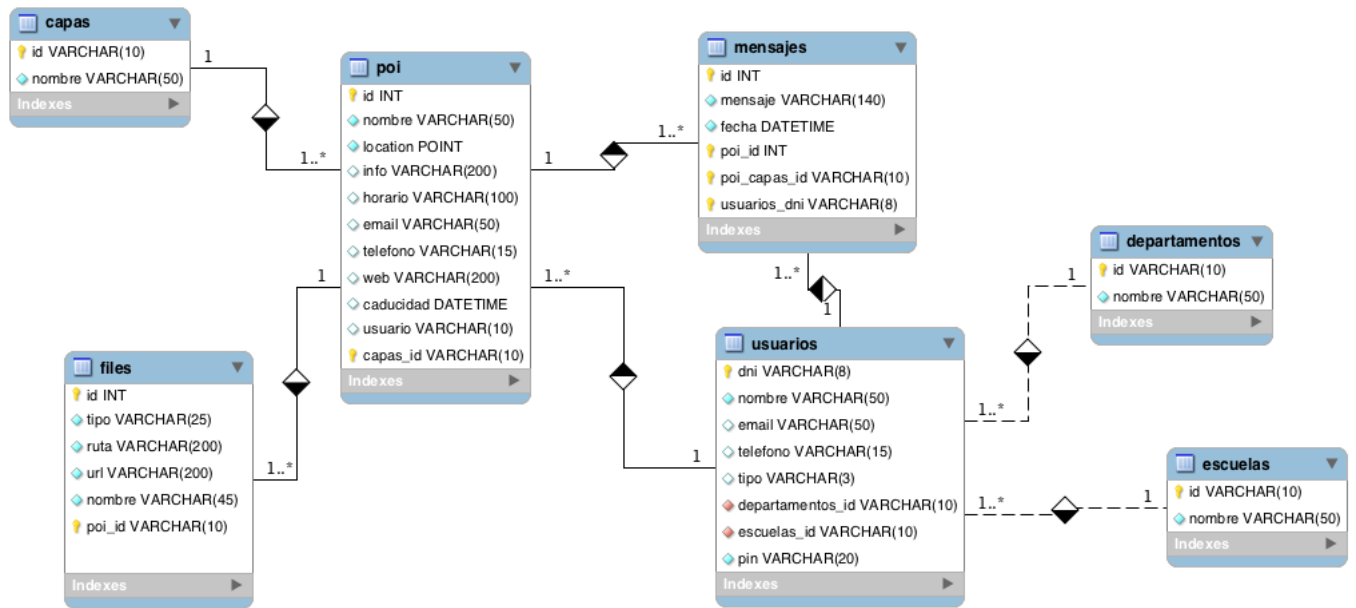


Figura 4.2: Diagrama Entidad-Relación de la base de datos.

se diseñaron para guardar ficheros. Y cierto es que es más práctico seguir esta corriente, especialmente en este proyecto donde ésta permitirá revisar las fotos subidas más fácilmente. La utilidad es el poder identificar con más rapidez fotos subidas por los usuarios que no cumplan con cierto decoro o sean ilegales.

4.2.2. Interfaz PHP del servidor

La interfaz del servidor en PHP se desarrolla siguiendo metodología **REST** (Representational State Transfer). Así que se puede hablar de un servidor RESTful. REST no es un lenguaje ni una librería, son un conjunto de principios arquitectónicos para diseñar servicios web como el presente. Según Richardson y Ruby [36], se pueden definir cuatro principios fundamentales dentro de esta metodología:

- Los métodos HTTP se han de usar de manera explícita. Esto es:
 - **GET**: Obtener recursos del servidor.
 - **POST**: Crear recursos en el servidor.
 - **PUT**: Actualizar un recurso del servidor.
 - **DELETE**: Eliminar un recurso del servidor.
- Las URL tomarán forma de rutas de directorios. Ayuda a hacerlas más intuitivas, pues se asemejan a un árbol de directorios.
- Las respuestas se transfieren en XML, en JSON o en ambas codificaciones.

- No mantiene estado.

Ahora bien, estos principios no son un dogma, no hace falta ceñirse a ellos estrictamente. Pueden tomarse como guías de buena praxis, y utilizarlos en la medida que convengan al proyecto. REST es sólo una alternativa a SOAP o WSDL, y hay que considerar cuál es la más apropiada. En este caso, la flexibilidad que aporta REST al construir una API es la mejor opción, así que los diseños están hechos siguiendo algunos de sus principios.

La API irá en una carpeta con el mismo nombre, dentro del directorio *htdocs*, que es la puerta de acceso al servidor Apache por medio del puerto 80. La instalación de Apache es anecdótica, ya que está incluida en un paquete software listo para usar llamado *MAMP*. La API contendrá una definición de una clase REST, y dos recursos: capas y pois. Cada vez que llegue una petición al servidor, se creará un objeto de la clase REST para el recurso solicitado con la finalidad del método HTTP usado. Aunque es mejor explicar este comportamiento con algunos ejemplos:

- <http://158.42.15.129:80/api/capas> Si el método HTTP usado fuera GET, pediría un recurso capas entero, es decir, que se devuelvan todas la capas.
- <http://158.42.15.129:80/api/pois?id=4> Usa el método GET para pedir el POI con valor de “id” igual a 4.
- Si en el caso del recurso “pois” se utilizase el método PUT o POST, el diseño REST entendería que se pretende crear un nuevo POI, y tomaría los datos enviados (i.e.: \$_POST) para crearlo.

La respuesta a estas peticiones estará formada por un código de respuesta HTTP, i.e.: 200 = OK; el *content-type*, que en esta API será siempre JSON; y el propio objeto o lista de objetos JSON que contiene la respuesta en sí. JSON es un formato ligero para el intercambio de datos basado en Javascript que no requiere el uso de XML. Es por ello mucho más sencillo y rápido de producir. Además las librerías PHP cuentan con métodos de codificación JSON por defecto.

Los recursos por tanto, consultarán la base de datos para dar una respuesta a la petición HTTP, y si la consulta a la BD es exitosa la codificarán en JSON y la enviarán como respuesta a la dirección IP de la aplicación Android de donde surgió. Aunque hay algunos pasos previos antes de la codificación JSON. Uno de ellos es codificar los campos de texto en UTF-8³, ya que el alfabeto español contiene símbolos como las tildes que sólo son reconocidas por ciertas codificaciones.

Además, si hay que enviar una imagen, ésta no se puede enviar como una cadena de caracteres tal cual. Antes de codificarla en el objeto JSON hay que codificarla en base 64, que se utiliza habitualmente para codificar imágenes que se enviarán como JSON. Android también la utiliza y deberá decodificarla al recibirla para poder crear una imagen.

³8-bit Unicode Transformation Format. Formato de codificación de caracteres que utiliza símbolos de longitud variable.

4.3. Arquitectura de la aplicación

En la sección 3.1 se ha descrito la metodología utilizada como incremental, es decir, no se diseñará toda la aplicación a la vez, sino que se dividirá en componentes que se diseñarán de forma independiente y luego se integrarán. Aunque para explicar su diseño es mejor hacerlo al revés, primero el conjunto general y luego parte por parte. Para ello se ha llevado a cabo un proceso de ingeniería inversa con herramientas del IDE Eclipse a partir del código fuente para conseguir un diagrama de clases UML general en el que se incluyan todos los componentes.

UML es un lenguaje de modelado de sistemas de software (Unified Modeling Language), que se utiliza para diseñar y documentar los mismos. Consiste en una serie de diagramas que describen la arquitectura de una aplicación, lo cual ayuda a conseguir una mayor abstracción a la hora de entender los conceptos. Uno de los diagramas más usados y más ilustrativos es el diagrama de clases, que describe la clases del sistema junto con sus atributos, métodos y relaciones entre ellas. El diagrama de clases de la figura 4.3 muestra la arquitectura de la aplicación. Por razones de espacio no se incluyen ni atributos ni métodos, puesto que hay centenares de ellos y no caben en la imagen.

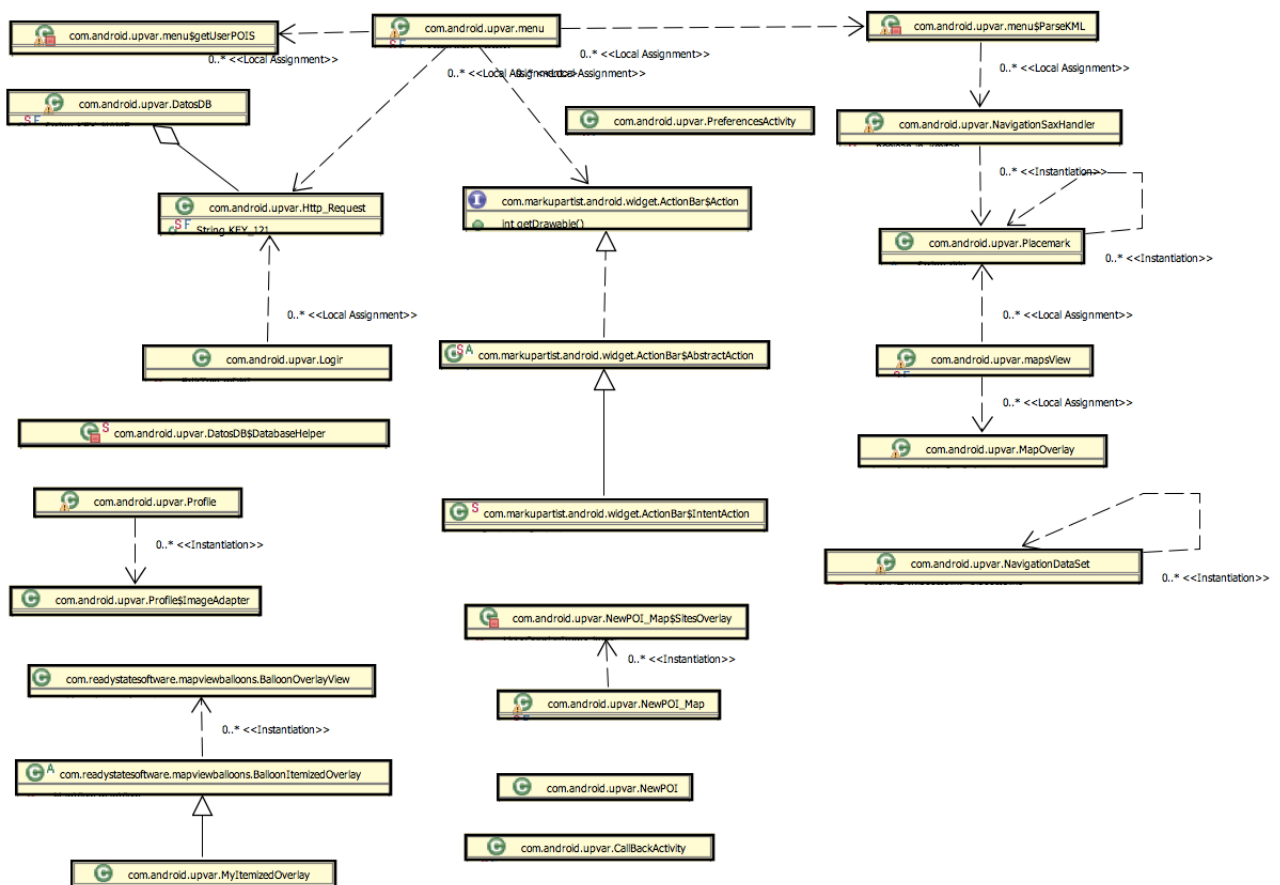


Figura 4.3: Diagrama de clases en UML.

El diagrama de clases sin embargo, no muestra as dependencias entre las actividades de Android. También son clases, pero no existen unos criterios a la hora de modelar este tipo de objetos que aún no están estandarizados en UML. Para entender la dependencia de clases, hay que volver a mirar a la figura 3.4 de la sección 3.3, donde se exponían los flujos de datos entre actividades, y contrastarlo con las clases proporcionadas.

Es posible también apreciar que no se encuentra ninguna clase relativa al uso del SDK de Wikitude. Esto es debido a que la librería está en formato *jar*, y por tanto pasa a ser una dependencia, como muestra la figura 4.4. También quedan reflejadas las dependencias relativa al uso del protocolo HTTP para comunicarse con el servidor remoto. El hecho de que no aparezcan como dependencias los otros proyectos de software libre utilizados, indica que estos no se han incluido como librerías *jar*, sino como proyectos librería. Estos son proyectos diferentes con sus propios directorios, a los que el proyecto hace referencia para hacer uso de sus clases.

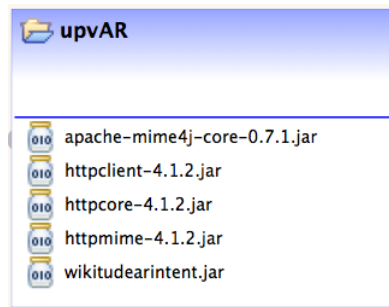


Figura 4.4: Dependencias del proyecto.

Antes de pasar a explicar cada componente, es interesante saber cuál es la estructura de un proyecto como éste, cómo se organizarán sus ficheros. La figura 4.5 es una captura de pantalla del árbol de directorios del proyecto en Eclipse, que sirve de ejemplo para retratar los directorios principales de éste:

- `src`: aloja el código fuente java de la aplicación.
- `gen`: Android genera algunos ficheros automáticamente que guardan referencias entre el código y los recursos. El más importante el *R.java*, que contiene referencias a todos los recursos del proyecto.
- Librerías varias: Se encuentran las pertenecientes a Google y el resto de librerías referenciadas.
- Proyectos librería: También se guarda un enlace a ellos dentro del proyecto.
- `res`: son los recursos del sistema. Pueden incluir imágenes, archivos de la interfaz gráfica, valores de variables, audio o vídeo entre otros muchos.
- `AndroidManifest`: Describe el nombre, versión, derechos de acceso, librerías referenciadas, permisos de ejecución y actividades de la aplicación.

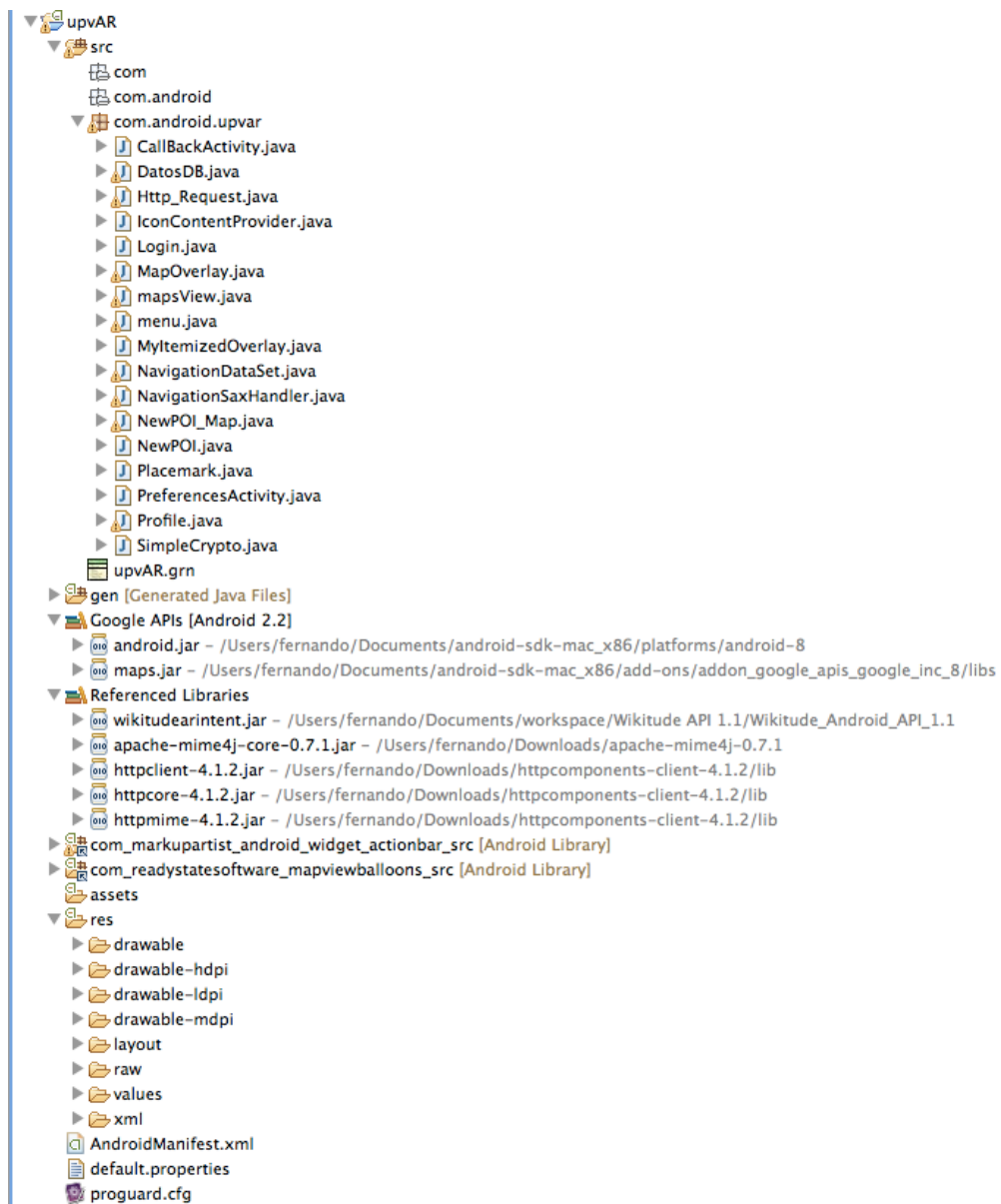


Figura 4.5: Árbol de directorios del proyecto.

En cuanto a los componentes, se pueden destacar 4: el menú de inicio, la vista en RA, la vista mapa y el perfil de usuario.

4.3.1. Menú principal

El menú principal da acceso al resto de actividades de la aplicación, por lo tanto contiene los elementos necesarios para acceder a ellas. Pero ésta no es la única tarea que realiza. También lleva a cabo algunos procesos secundarios relativos a la gestión de datos. La figura 4.6 muestra tres hilos diferentes, uno de los cuales es el principal, el de la interfaz gráfica.

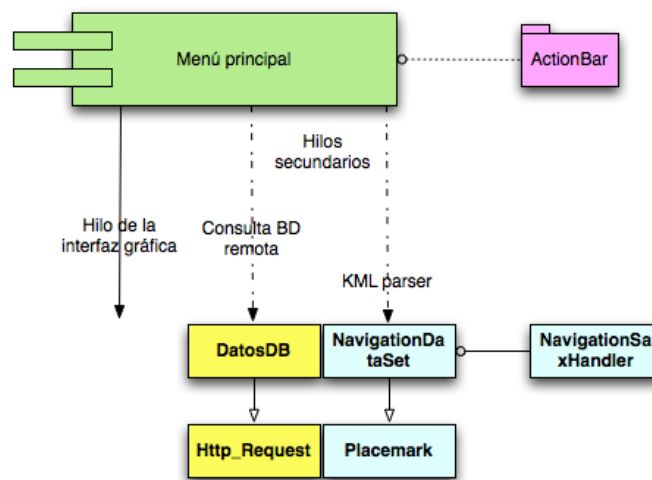


Figura 4.6: Diagrama del componente menú principal.

Uno de los otros dos hilos se encarga de comprobar si hay POIs en la base de datos local de Android. Si no los hay, lanzará una petición HTTP al servidor para descargar los puntos creados por el usuario en primera instancia. De esta manera, en el momento en que se quiera acceder a esta información se evita tener que hacerlo en el hilo principal de ejecución y por tanto mejora el aspecto visual al no haber ralentizaciones en la interfaz.

El hilo restante se dedica a parsear ficheros KML⁴. Como ya se comentó anteriormente, en algunas capas de la vista mapa no es suficiente con un punto que marque el lugar, pues el espacio que enmarca tal punto puede extenderse por varios edificios. Hace falta una forma de delimitar ese espacio, y una manera de hacerlo es dibujar el contorno. La aplicación web de Google Maps permite realizar estos planos y luego exportarlos en formato KML para su posterior parseo (análisis sintáctico). De forma que para las capas que necesiten de este extra, se incluirán ficheros KML en los recursos de la aplicación para que puedan dar la información de representación necesaria en el mapa. Y dado que estos archivos representan información estática que no va a cambiar a lo largo del tiempo, es más sencillo utilizar KML que crear complejas tablas en la base de datos.

⁴Keyhole Markup Language. Lenguaje de representación de datos geográficos en tres dimensiones basado en XML.

4.3.2. Vista en RA

La parte principal de este componente es el motor de realidad aumentada, que en cierto modo actúa como una caja negra. Este es el motivo por el que en la figura 4.7 Wikitude aparece dentro de la propia clase, y no como una clase externa como sucede por ejemplo con la barra de acciones (ActionBar).

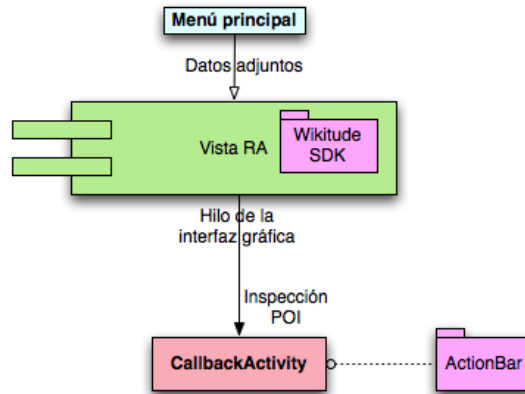


Figura 4.7: Diagrama del componente vista en RA.

Cuando se inicia esta actividad, se le han de pasar como parámetros la información a representar y las acciones que se pueden llevar a cabo. Posteriormente, si se realiza alguna acción sobre los POI expuestos, se utilizará la clase CallbackActivity para realizar las tareas pertinentes, la cuál sí que es una actividad normal de la aplicación y no una caja negra.

4.3.3. Vista mapa

El componente mapa ha de recibir los datos KML obtenidos en el hilo secundario iniciado en el menú principal. Estos datos estarán dentro de un objeto NavigationDataSet que se pasará como parámetro, como muestra la figura 4.8. Luego deberá procesar esta información junto con la del resto de POI, que si no están ya en memoria habrá que descargarlos del servidor.

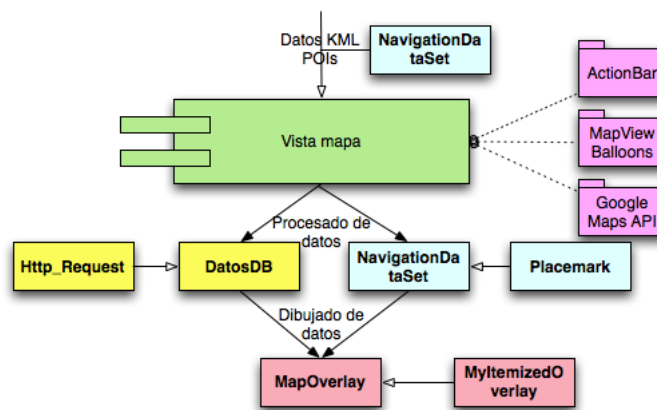


Figura 4.8: Diagrama del componente vista mapa.

Una vez procesada la información se inicia el proceso de dibujado, que pondría sobre el mapa proporcionado por Google Maps los POI requeridos junto con la posición del usuario obtenida mediante el GPS. Para el muestreo y dibujado de estos elementos son necesarias dos librerías: Google Maps y MapViewBalloons; que están referenciadas en este componente igual que ActionBar.

4.3.4. Perfil de usuario

Este componente tiene dos funciones principales. Una es mostrar los POI creados por el usuario, para lo cual necesita el acceso a dicha información. Si no está en memoria se realizará una petición al servidor para conseguirla.

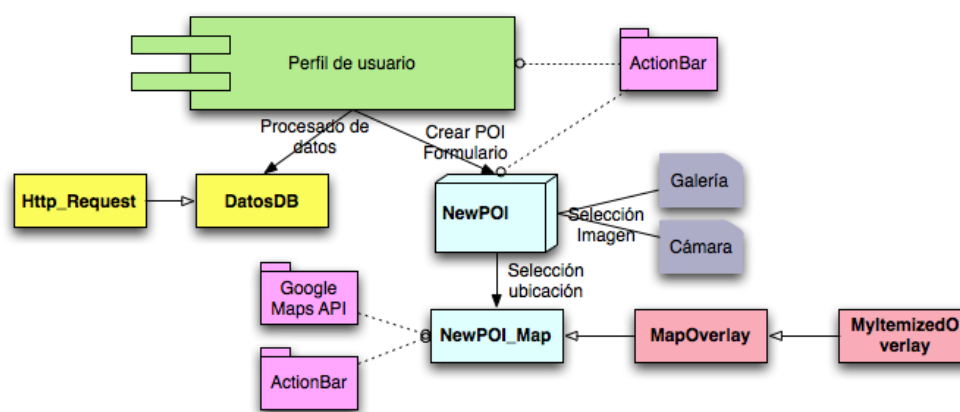


Figura 4.9: Diagrama del componente perfil de usuario.

La otra función es la de crear nuevos puntos de interés. Se realizará en otra actividad llamada NewPOI, que es dependiente de la actividad principal del perfil de usuario como muestra la figura 4.9. Ésta a su vez requerirá de otras clases, necesarias para poder rellenar el formulario de creación de puntos.

Para seleccionar la imagen adjunta al POI necesitará de aplicaciones externas que faciliten estos datos: la galería o la cámara. Y para seleccionar la ubicación del POI será necesario un nuevo mapa, aunque simplificado, ya que no necesita de la misma funcionalidad que el de la vista mapa. Pero sí hará uso de clases como MapOverlay, necesarias para dibujar marcadores.

4.4. Interfaz de usuario

Pronto o tarde el diseño llega a la parte gráfica, y es algo que no hay que pasar por alto. No sirve de nada tener un back-end muy trabajado si luego la interfaz gráfica no es buena y el usuario no está cómodo usándola. Es necesario realizar los diseños de antemano, aunque sea a lápiz en un papel, para clarificar los conceptos y poder conseguir la mejor experiencia de usuario.

La página web de desarrolladores de Android ofrece útiles consejos sobre buenas prácticas en el diseño de sus interfaces. Aunque está más dedicado al diseño gráfico en en sí que a la usabilidad y amigabilidad.

Hay que centrarse en hacer un diseño sencillo, que el usuario pueda aprender sin ningún tipo de ayuda, y que sea atractivo visualmente.

Los diseños de la interfaz de esta aplicación se hicieron en un principio a mano, a lápiz. Por supuesto estos no fueron los definitivos. Después de modificaciones varias se llegó a un diseño final, que es el que se ha esbozado en el ordenador con un software específico para ese fin, Balsamiq Mockups. Las figuras de esta sección son por tanto el resultado de repetidos cambios en la interfaz gráfica.



Figura 4.10: Bocetos de la interfaz gráfica 1.

Y aún siendo estas imágenes los bocetos finales, esto no quiere decir que la interfaz desarrollada sea exactamente igual, ya que puede haber estado sujeta a cambios como consecuencia de la experiencia del usuario. Y es que el diseño conceptual no tiene por qué ser correcto. Después de su uso pueden aparecer nuevas cuestiones que no se habían planteado. En este caso los cambios se reducen a los colores de la aplicación, para los que se ha seguido la guía de estilos de la UPV [49], y algunos iconos de los botones.

La figura 4.10a muestra el menú principal. En él hay una lista de opciones con los nombres de las capas, que dan acceso a la vista en realidad aumentada con los POI de la capa seleccionada. En la barra de acción también se presentan las opciones del mapa y del perfil de usuario. La razón de que estén posicionadas a primera vista y no en un menú de opciones es porque son partes principales de la aplicación, y se pretende que el usuario las vea cuanto antes. Abajo se puede ver el menú de opciones que da acceso al panel de preferencias y la inicio o salida de sesión, según el estado de la sesión en ese momento.

La pantalla de inicio de sesión es de las más sencilla, pues cuenta con un par de cuadros de texto, dos botones para aceptar o cancelar, y un checkbox para recordar los datos. Figura 4.10b.

El panel de configuración, o preferencias, también es de las más sencillas, pues sólo cuenta con dos opciones. Ambas opciones lanzarán un diálogo al pulsarlas. Figura 4.10c.



Figura 4.11: Bocetos de la interfaz gráfica 2.

En el segundo conjunto de figuras, el 4.11, se encuentran los bocetos de la vista en realidad aumentada, 4.11a, la vista mapa, 4.11c, y la inspección de POIs, 4.11b. La primera no tiene ningún diseño a parte de la barra de acción, ya que al ser la parte de Wikitude lleva su propio diseño embebido.

La de inspección es una pantalla en blanco que mostrará una imagen si el POI la tiene, y abre un diálogo donde se especifican las acciones disponibles.

En cambio la del mapa sí que es más compleja que las dos anteriores, ya que hay muchos más elementos. El mapa contiene marcadores para los POI y un icono en forma de persona que indica la posición del usuario. El menú de opciones presenta más posibilidades: cambiar entre vista mapa y satélite, cambiar de capa, centrar la vista en la posición del usuario, centrar la vista en la UPV y volver al inicio.

El perfil de usuario (figura 4.12a) tiene en la parte superior el famoso Cover Flow, donde se muestran



Figura 4.12: Bocetos de la interfaz gráfica 3.

las imágenes de los POI creados por el usuario. Y debajo se encuentra una tabla con los datos del POI en primer plano. El menú de opciones sólo contiene un botón que es el de crear un nuevo punto, que nos lleva a la figura 4.12b, el formulario. En él se encuentran todos los campos de texto a rellenar, y botones para adjuntar una imagen y para elegir una ubicación. Este último a su vez redirige la aplicación a la figura 4.12c, donde se selecciona la posición moviendo un marcador. Cuando se ha decidido se usa el botón “Fijar ubicación” para guardarla.

Capítulo 5

Evaluación y resultados

5.1. Casos de uso llevados a cabo

Para ver los resultados se han probado todas las funcionalidades de la aplicación. Se ha de comprobar que todo lo descrito en los capítulos anteriores funciona correctamente y que por tanto se han alcanzado los objetivos. Por ello se van a describir algunos casos de uso llevados a cabo que explotan las bondades de la aplicación.

Estos se pueden considerar tests de la aplicación, de los cuales se emitirá una evaluación según sus resultados. Los casos de prueba propuestos no son tests aleatorios, sino que pretenden comprobar el funcionamiento de las partes principales de la aplicación, de forma que la evaluación se completa. Estas partes son:

- Inicio de sesión
- Realidad aumentada
- Mapa
- Creación de POI

Cada una de estas partes es evaluada individualmente en los siguientes apartados, explicando cuál ha sido el proceso seguido, los resultados obtenidos y el veredicto.

Los resultados presentados no tienen por qué ser producidos por las primeras pruebas, ya que hay elementos que pueden fallar. Son los últimos que se han hecho, una vez se han corregido todos los posibles fallos que se hayan podido encontrar.

5.1.1. Inicio de sesión

La aplicación ha de permitir iniciar sesión a los usuarios registrados y guardar sus datos si lo desean. Asimismo, también ha de permitir borrar los datos guardados en cualquier momento desde el menú de configuración. Este comportamiento está esquematizado en la figura 5.1.

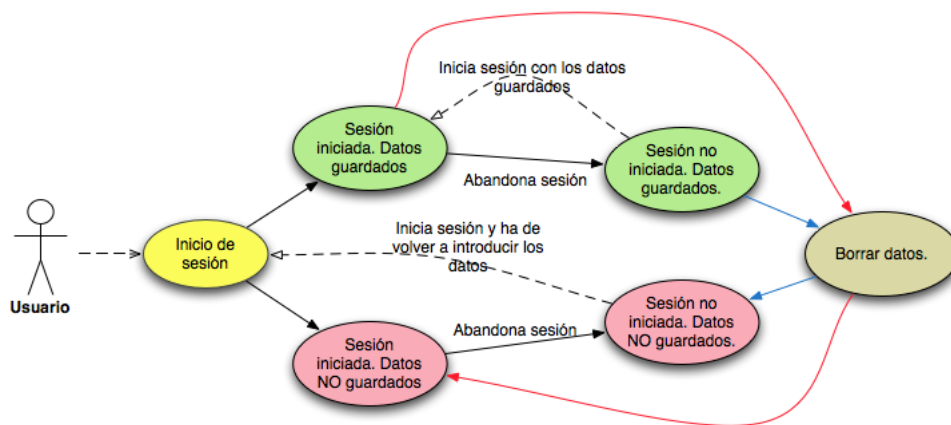


Figura 5.1: Esquema del inicio de sesión.

Para comprobarlo, se ha probado a iniciar sesión con un usuario y recordar sus datos. A continuación abandonar la sesión y volverla a iniciar sin necesidad de introducir el DNI y el PIN, pues ya está guardados en la memoria. Y por último, ir al panel de configuración (figura 5.2b), borrar los datos, e intentar el mismo proceso otra vez, estando obligados entonces a introducir los datos.

Después de repetir estas acciones una y otra vez el funcionamiento sigue siendo correcto, por lo que se puede calificar esta prueba como satisfactoria.

El lugar donde se guardan estos datos es la memoria interna de la aplicación, la cual es un lugar seguro pues no es accesible desde ninguna otra aplicación, y por tanto, de querer borrarlos sólo se puede hacer desde la aplicación que los ha creado. Aunque también existe otra opción, que en ningún caso trastoca el funcionamiento de este proceso. Desde el gestor de aplicaciones de Android se pueden borrar todos los datos de una aplicación, que en este caso incluiría las credenciales y los POI guardados en caché. Si esta opción se utilizase, la aplicación reconocería que no hay credenciales guardadas y volvería a pedir las, es decir, que el funcionamiento seguiría siendo correcto.

Resultado: Totalmente positivo.

5.1.2. Realidad aumentada

Para testar la realidad aumentada se ha de navegar por las distintas capas y comprobar que los POI están situados correctamente en su posición geográfica y que son accesibles para su inspección.

Como caso de prueba se va a buscar un punto concreto, la ETSINF¹, y se va inspeccionar. Una vez aparezca el menú de acciones se seleccionarán todas ellas para comprobar su funcionamiento. Una vez más, las acciones disponibles son:

- Llamar
- Enviar e-mail

¹Escuela Técnica Superior de Ingeniería Informática

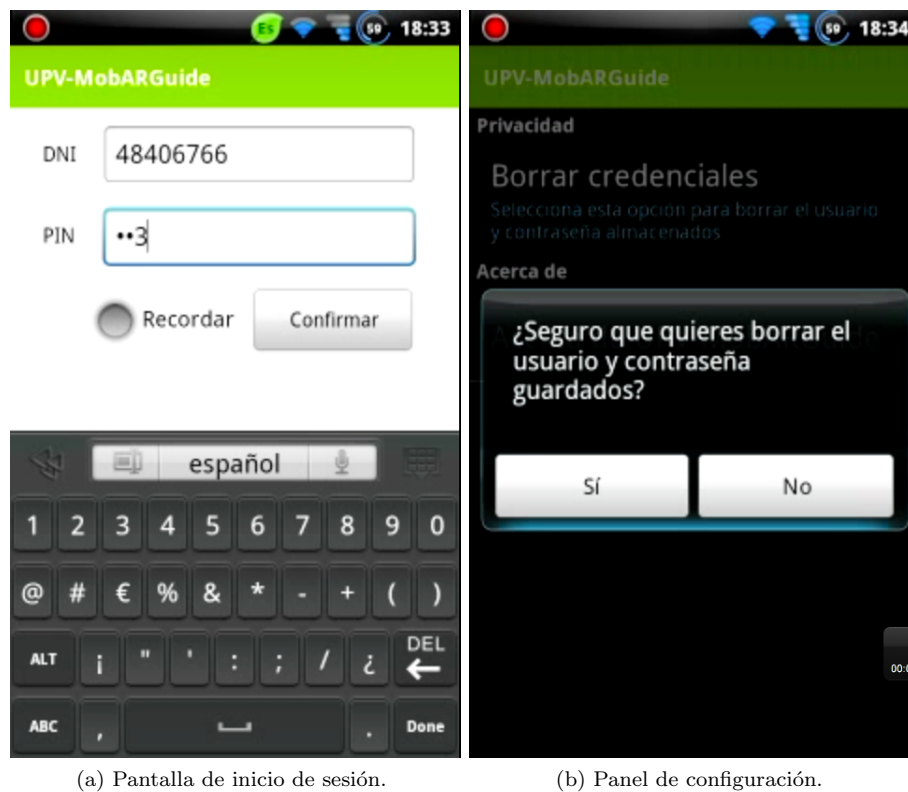


Figura 5.2: Capturas del inicio de sesión.

- Visitar la web

Así pues, en la imagen 5.3a se puede apreciar la vista del POI buscado en realidad aumentada. En la parte baja de la pantalla hay un diálogo con información sobre dicho punto, el cual al pulsarlo abrirá la pantalla de acciones, la figura 5.3b. Una vez seleccionada una de ellas se abrirá la aplicación pertinente para realizarla.

Se han ejecutado todas las acciones satisfactoriamente. Aunque hay ciertos factores que pueden afectar al desempeño de una de ellas, que es el posicionamiento. La posición en la que aparecerán los POI tiene una gran dependencia del hardware del dispositivo, ya que la aplicación dibuja en función de las variables que otorgan el GPS, la brújula y los acelerómetros. Por tanto, un mal funcionamiento o mala calibración de estos afecta enormemente al sistema.

Durante las pruebas se han dado casos en los que por ejemplo la brújula no estaba bien calibrada, por lo que los iconos estaban desplazados hacia un lado. También ha habido algún problema aislado con la precisión de los sensores de aceleración, que al dar una inclinación muy baja los iconos aparecían a ras de suelo en vez de por la parte alta de la pantalla. Sin embargo estos son casos aislados. Como el funcionamiento de estos elementos es ajeno a la aplicación, sólo queda confiar en su buen estado y calificar el software como satisfactorio, pues cumple su función.



(a) ETSINF en RA.

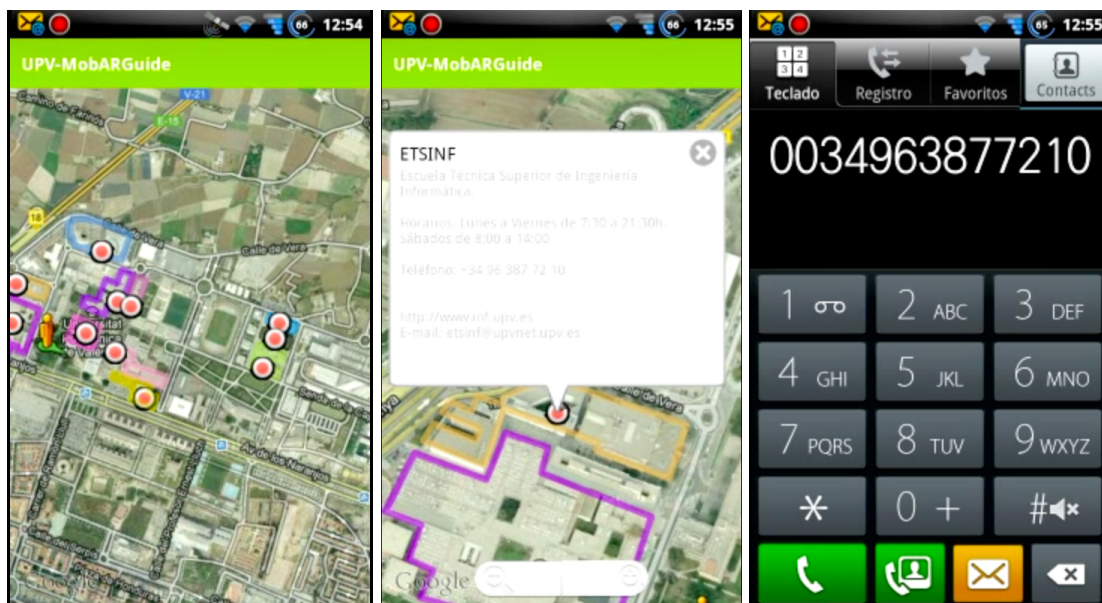
(b) Inspección de POI.

Figura 5.3: Capturas de la realidad aumentada.

Resultado: Parcialmente positivo.

5.1.3. Mapa

Al igual que en la vista en realidad aumentada, también se ha de comprobar que los POI están bien posicionados en el mapa, que se pueden inspeccionar y que se pueden realizar acciones sobre ellos. Para ello se han inspeccionado varios puntos y se ha realizado alguna acción.



(a) Vista de la UPV en el mapa.

(b) Inspección del POI de la ETSINF.

(c) Acción de llamada a la ETSINF.

Figura 5.4: Capturas del mapa 1.

El primer POI inspeccionado es una vez más el de la ETSINF, que aparece en la figura 5.4b. Al hacer en click en el diálogo aparecerán las acciones disponibles, y como se puede observar en la imagen 5.4c, se ha seleccionado la de llamar.

El otro punto inspeccionado ha sido el de la ETSII² (figura 5.5a. Y en este caso se ha seleccionado la opción de enviar un e-mail, con lo cual la aplicación ha abierto el gestor de correo seleccionado y ha iniciado la pantalla de redacción con el campo del destinatario ya relleno.

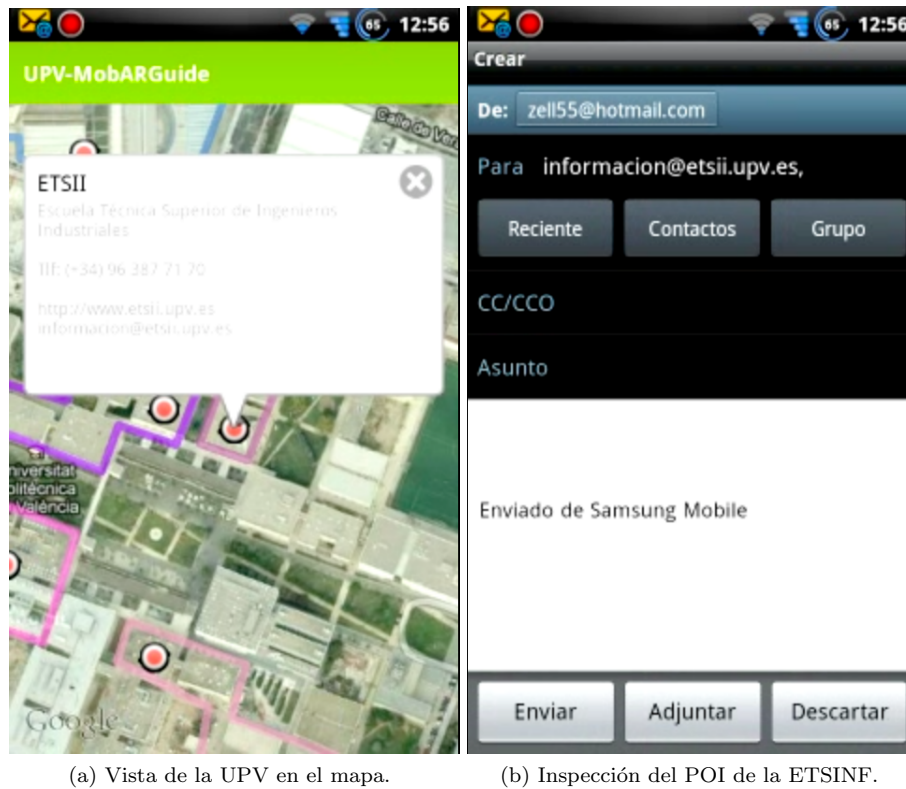


Figura 5.5: Capturas del mapa 2.

Al contrario que en la vista en realidad aumentada, aquí la posición de los marcadores no depende del hardware, pues no depende del punto de vista del usuario. Por tanto si las coordenadas son correctas los POI aparecerán bien posicionados. Lo que sí depende del HW es la posición del usuario, y una vez más se ha comprobado que la precisión de estos aparatos no siempre es buena. Las pruebas se han hecho con varios teléfonos, y en ambos casos puede llegar a haber un error de hasta 10 metros. Incluso estando en la misma posición, cada dispositivo puede indicar una cosa diferente, siempre dentro de ese rango de error. No es demasiado y puede considerarse una precisión aceptable para el uso doméstico. Otra vez, la aplicación es ajena a los elementos que proporcionan la geolocalización, por lo que se considera una prueba satisfactoria.

Resultado: Positivo.

²Escuela Técnica Superior de Ingenieros Industriales

5.1.4. Creación de POI

Esta es la parte crítica de las pruebas, pues en ella interviene la base de datos como agente activo, es decir, se trata de crear nueva información y que luego esté disponible para el resto de usuarios. Para llevar a cabo dicho test, se crearán nuevos POI con una imagen adjunta, se comprobará que son almacenados en el servidor y que son visibles por el resto de usuarios.

El primer paso es rellenar el formulario de creación de puntos que se puede ver en la figura 5.6a. Una vez finalizado, el POI estará disponible en la pantalla del perfil de usuario. En la imagen 5.6b se puede ver un perfil de usuario donde se han creado dos POI, uno sobre un concierto y otro sobre la venta de una bici. Además, también se ha comprobado que estén almacenados en el servidor, tanto los datos como la imagen.

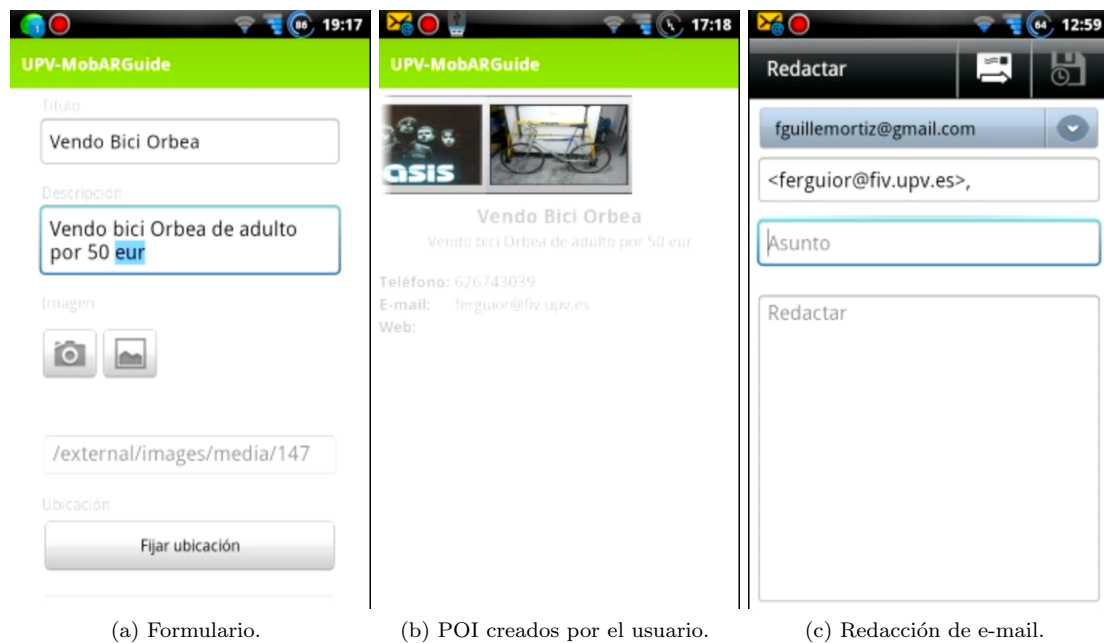
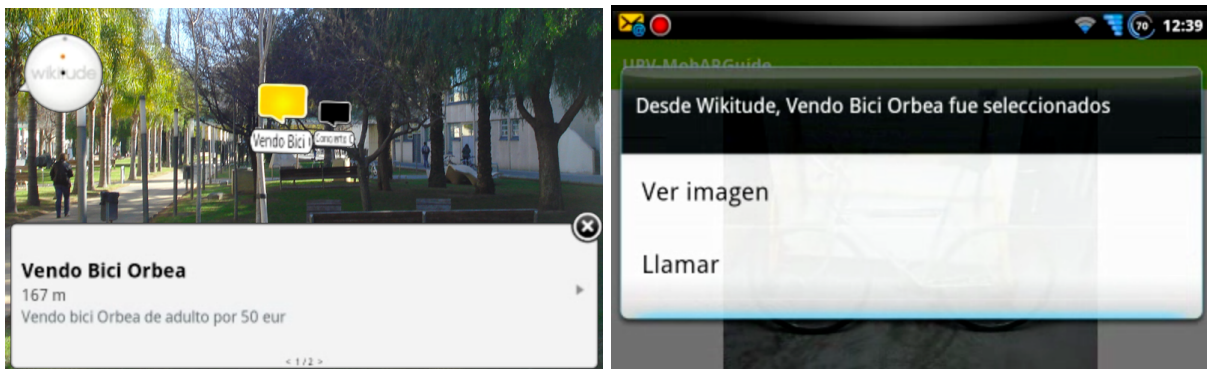


Figura 5.6: Capturas de la creación de POIs 1.

Ahora es necesario comprobar que los puntos creados son visibles en realidad aumentada y en el mapa, y no sólo por el usuario que los creó. Para ello hay que explorar la capa Comunidad UPV, que muestra los POI creados por usuarios registrados. Y efectivamente, se pueden visualizar correctamente como muestra la figura 5.7a.

Ya sólo resta comprobar que se pueden inspeccionar y que se pueden ejecutar acciones de contacto con el usuario que los creó. En las figuras 5.7b y 5.6c se muestra el intento de contactar al usuario que creó el POI por e-mail.

Todas las acciones descritas han finalizado satisfactoriamente, por lo que los resultados son positivos. Al igual que en las pruebas de las secciones previas sigue habiendo dependencia del hardware, por lo que



(a) Vista en RA del nuevo POI.

(b) Acciones sobre el nuevo POI.

Figura 5.7: Capturas de la creación de POIs 2.

un fallo en el mismo puede ocasionar malos resultados.

Resultado: Totalmente positivo.

5.2. Evaluación del sistema

Todo el proceso de evaluación ha ayudado a corregir errores en la aplicación, llegando al final a superar satisfactoriamente los casos de la sección anterior. Para mayor seguridad en la comprobación del buen funcionamiento se ha probado en varios dispositivos:

- Samsung Galaxy S
- LG Optimus Black

La razón de la elección de estos dispositivos es que eran los únicos disponibles, por tanto la evaluación está basada en los resultados que se han producido en ellos. Si hubiese habido otros dispositivos disponibles también se habrían testado. Y eso probablemente afirmarían aún más las siguientes peculiaridades encontradas cuando se ha probado la aplicación con ambos dispositivos a la vez y en el mismo lugar:

- La posición geográfica otorgada por cada dispositivo es diferente aún estando en el mismo lugar. Entre ambas puede haber de 5 a 10 metros de diferencia. No es una distancia crítica pero hay que tenerla en cuenta.
- La dirección proporcionada por la brújula de cada dispositivo también es ligeramente diferente, llegando a fluctuar hasta en 15° . Esto causa que el posicionamiento de POI en realidad aumentada no sea exactamente el mismo, ya que en uno dispositivo aparecerán desplazados hacia un lado. Tampoco es crítico pues la desviación es muy ligera, pero hay que tenerlo en cuenta.
- La inclinación proporcionada por los sensores de aceleración tampoco es la misma, causando que en un dispositivo los iconos se vean en la parte alta de la pantalla y en el otro en la parte baja.

Sigue sin ser algo crítico pues los POI hacen referencia a grandes espacios y no a cosas concretas donde haría falta más precisión.

Por ende, la dependencia del hardware afecta al funcionamiento ligeramente, aunque no impide cumplir los objetivos de la aplicación. Sí que hay que especificar entonces, que si el propósito fuese identificar objetos más pequeños en los que hiciese falta más precisión por parte del HW, entonces sí que consideraría un problema. Pero no es el caso, por tanto se puede considerar la evaluación como positiva.

Capítulo 6

Conclusiones y trabajos futuros

6.1. Vicisitudes

Esa sección pretende relatar todos los momentos destacables o los problemas encontrados durante el desarrollo del proyecto, tanto técnicos como personales.

En primer lugar, el desarrollo del proyecto se ha llevado compaginándolo con el trabajo, primero en Valencia, luego en Israel y finalmente en Valencia otra vez. Esto ha hecho que las horas diarias disponibles para la realización del mismo hayan sido escasas, forzándome a ocupar casi todo mi tiempo entre el trabajo y el proyecto. Esto fue especialmente duro durante mi estancia en Israel en verano, trabajando para mi actual compañía.

Desde los comienzos no ha sido fácil, pues mis conocimientos de Android se limitaban a la experiencia como usuario. Por tanto he tenido que aprender el desarrollo para este sistema casi desde cero, con todo lo que ello supone. Digo casi porque el desarrollo se hace en Java, que me es un lenguaje familiar y por tanto esa roca ya estaba sorteada. La consecuencia principal de este aprendizaje es el tiempo que pasó entre el comienzo del desarrollo y lo que podría considerarse como una fase productiva del mismo, es decir, un punto en el cual ya no hace mirar el manual continuamente.

Esto mismo sucedió con algunas de las herramientas utilizadas, como la API de Google Maps o el SDK de Wikitude. Y en la misma situación se puede encasillar el desarrollo del servidor con la base de datos, algo que también he tenido que aprender, aunque de eso se trata, de aprender. Además los conocimientos sobre HTTP que implicaba este último han sido especialmente útiles en mi trabajo actual sobre redes de telecomunicaciones.

En cuanto al desarrollo de la aplicación en sí, hay varios aspectos que han acarreado especial trabajo. Uno de ellos es el manejo de datos, su descarga desde el servidor y organización interna. Llevó bastante tiempo tener toda la estructura en funcionamiento. El otro, y a priori no lo parecía, fue el desarrollo con Google Maps, que debido a los requisitos que se pretendían tuvo más trabajo del esperado. Se concebía como una parte más fácil que otras a priori más difíciles que sin embargo luego no lo fueron tanto.

El resto del desarrollo se cumplió tiempos cercanos a los estimados, cerca de un año a tiempo parcial.

Esto sin contar la planificación e investigación previas, y la escritura de esta memoria.

6.2. Conclusiones

Citando la introducción de este documento, lo que se pretendía era realizar una aplicación móvil de realidad aumentada en la UPV que tuviese una componente social, que fuese participativa. Se ha desarrollado una aplicación que permite navegar por la universidad en realidad aumentada y que permite a los usuarios registrados compartir nueva información con el resto de la comunidad. Por tanto, se puede concluir que se han alcanzado los objetivos. La gestión de los datos ha probado ser efectiva y las estructuras de datos y algoritmos utilizados han logrado tiempos de proceso aceptables.

No obstante, se ha puesto de manifiesto en este documento la dependencia de la precisión del hardware para el buen funcionamiento de la aplicación. Se ha comprobado que los dispositivos actuales no están preparados para una realidad virtual en la que los objetos no sean edificios como en este caso, sino pequeños puntos. La precisión de la realidad aumentada basada en geolocalización a día de hoy es aceptable y desde luego suficiente para cumplir los objetivos de este proyecto, pero es algo que ha de mejorar con los años.

Por último, quiero destacar la complejidad del desarrollo de software para dispositivos móviles. A menudo se oye que este tipo de proyectos son cosa fácil o cosa de un rato. Y sin duda, si no se hace un buen desarrollo puede que así sea. Pero llevar a cabo un proyecto serio tiene una gran complejidad, que nada tiene que envidiar al desarrollo de otro tipo de software. Por tanto, es necesario desmitificarlo y eliminar el cliché de que el desarrollo para móviles es un desarrollo de segunda, pues no es así.

6.3. Trabajos futuros

El fin de este proyecto no es sacar una aplicación al mercado, sino hacer un prototipo de aplicación que con un poco de trabajo de integración en el ecosistema tecnológico de la UPV pueda salir al mercado. Partiendo de esa premisa, el primer trabajo futuro que se habría de realizar es la preparación necesaria para que la aplicación se haga pública. Esto implicaría o bien integrarla con la intranet de la universidad, para que sólo sus usuarios puedan acceder al modo social en el que pueden crear contenidos, o desarrollar algún otro procedimiento por el que un usuario se pueda registrar y por tanto acceder a los contenidos creados por el resto de la comunidad.

Una de las mejoras que podrían aportar mucho al entorno social sería dar la posibilidad de escribir mensajes en cada POI, de forma similar al muro de Facebook. Es decir, si hay un punto creado por un usuario sobre el que se desea comentar algo o añadir información, se podría escribir un mensaje sobre el mismo, que el resto de la comunidad podría visualizar cuando inspeccionase dicho POI. Esto implicaría también el desarrollo de políticas sobre censura, ya que puede haber contenidos con carácter ilícito, inapropiado o peyorativo.

Otro aspecto a desarrollar serían las políticas de privacidad, ya que en la aplicación actual todo es visto por todos. Pero puede que haya usuarios que no deseen compartir información con toda la comunidad sino sólo con ciertos grupos. Sería de especial utilidad para los centros, departamentos o institutos, ya podrían publicar información específica sólo para sus alumnos. Por ejemplo, si el departamento DSIC publica una beca de colaboración para estudiantes, pero ésta sólo está destinada a alumnos de Ingeniería en Informática, podría crear un POI geolocalizado en su edificio con la información pertinente para la solicitud y que sólo fuese visible por alumnos que cursasen esa titulación. Para ello habría que crear grupos de usuarios, en cierto modo semejante al sistema de grupos de Google+.

Por otra parte, el incremento del tráfico y del número de usuarios pasaría por la integración con otras redes sociales como Facebook o Twitter. Es una buena manera de potenciar el uso de la aplicación, ya que permite seguir utilizando los mismo contactos que se tiene en otras redes. Un ejemplo de este tipo de funcionamiento es Spotify, que mediante su integración con Facebook permite ver qué escuchan los contactos de Facebook. En este caso podría aplicarse de manera que los usuarios viesen qué nuevos POI han creado sus amigos, o qué nuevos comentarios han añadido.

Ambos campos, realidad aumentada y redes sociales, están en alza y probablemente su evolución ofrezca multitud de nuevas opciones. El margen de mejora por tanto es muy amplio, y quizá dentro de unos años los posibles trabajos futuros serían mucho más atractivos. Pero esto es el presente, y para llegar a esa mejoras futuras hay que trabajar hoy.

Bibliografía

- [1] Droid-ar. <http://code.google.com/p/droidar/>.
- [2] Layar official website. <http://www.layar.com/>.
- [3] MySQL AB. *MySQL 5.0 Reference Manual*. MySQL AB, 3 2008.
- [4] MySQL AB. *MySQL Query Browser*. MySQL AB, 3 2008.
- [5] Francisco J. Abad. Consejos para el desarrollo de tu proyecto final de carrera. <http://users.dsic.upv.es/~fjabad/>, 3 2010.
- [6] Lazar Agatonovic, Nikola Nikolic, Stefan Nikolic, and Predrag Radenkovic. Armed (augmented reality in medicine). Faculty of Science, University of Kragujevac.
- [7] Ronald T. Azuma. A survey of augmented reality. <http://www.cs.unc.edu/~azuma/ARpresence.pdf>, 1997.
- [8] Michael Beaudouin-Lafon. Beyond the workstation: Mediaspaces and augmented reality. In *People and Computers IX*. Laboratoire de Recherche en Informatique, Université de Paris-Sud, Cambridge University Press, 1994.
- [9] BMW. Proyecto bmw augmented reality. http://www.bmw.com/com/en/owners/service/augmented_reality_workshop_1.html.
- [10] G. Burdea and P. Coiffet. *Virtual reality technology*. Number v. 1 in Wiley-interscience publication. J. Wiley-Interscience, 2003.
- [11] T.P. Caudell and D.W. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *IEEE Hawaii International Conference on Systems Sciences*, pages 659–669, 1992.
- [12] comScore MobiLens. comscore reports november 2011 u.s. mobile subscriber market share. http://www.comscore.com/Press_Events/Press_Releases/2011/12/comScore_Reports_November_2011_U.S._Mobile_Subscriber_Market_Share, 11 2011.

-
- [13] N.I. Durlach, A.S. Mavor, National Research Council (U.S.). Committee on Virtual Reality Research, and Development. *Virtual reality: scientific and technological challenges*. National Academy Press, 1995.
- [14] Jeff Gilfelt. Android-mapviewballoons. <https://github.com/jgilfelt/android-mapviewballoons/>.
- [15] I. Gilfillan. *La biblia de MySQL*. La biblia de. Anaya Multimedia, 2003.
- [16] Rocío Vian Gimeno. Realidad aumentada. fundamentos y aplicaciones. Master's thesis, Universitat Politècnica de València, 9 2011.
- [17] Google. Página web de android para desarrolladores. <http://developer.android.com/>.
- [18] Ellen Gottesdiener. *The Software Requirements*. Memory Jogger™. GOAL/QPC, first edition, 2005.
- [19] David Gurecki. Ad-hoc social networking for the google android platform. <http://moss.csc.ncsu.edu/~mueller/rt/rt09/readings/projects/g3/>.
- [20] Haseman. Android-ar-kit. <https://github.com/haseman/Android-AR-Kit>.
- [21] Robert Hinn, Benedikt Redmer, and Gitta Domik. Ar-campus. http://www.cs.uni-paderborn.de/fileadmin/Informatik/AG-Domik/research/AR_Campus/domik-art02.pdf.
- [22] A.L. Janin, D.W. Mizell, and T.P. Caudell. Calibration of head-mounted displays for augmented reality applications. In *Virtual Reality Annual International Symposium, IEEE*, pages 246 – 255, 9 1993.
- [23] Simon Julier, Yohan Baillot, Marco Lanzagorta, Dennis Brown, and Lawrence Rosenblum. Bars: Battlefield augmented reality system. In *In NATO Symposium on Information Processing Techniques for Military Systems*, pages 9–11, 2000.
- [24] Michael Kalkusch, Thomas Lidy, Michael Knapp, Gerhard Reitmayr, Hannes Kaufmann, and Dieter Schmalstieg. Structured visual markers for indoor pathfinding.
- [25] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *ACM International Workshop on Augmented Reality*, pages 85–94, 1999.
- [26] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. In *IEICE Trans. Information Systems*, volume E77-D, pages 1321–1329, 1994.

-
- [27] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *ATR Communication Systems Research Laboratories*, 2351(Telemanipulator and Telepresence Technologies):282–292, 1994.
- [28] Mobilizy. Wikitude official website. <http://www.wikitude.com/>.
- [29] Mobilizy GmbH, Jakob-Haringer Str 5/IV, 5020 Salzburg, Austria. *Developing with Wikitude API*.
- [30] Johan Nilsson. Android-actionbar. <https://github.com/johannilsson/android-actionbar>.
- [31] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to L^AT_EX*, 4 2011.
- [32] Oracle. *MySQL Workbench*. Oracle, 4 2011.
- [33] Ahmad Rafi, Avijit Paul, Syazani Suhafi, Chor Choon Keong, Kamarulzaman Ab. Aziz, and Amar Zulkifli. Virtual merdeka. experiencing augmented reality for social engagement in a network environment. *Academia.edu*, 2011.
- [34] J. Rekimoto and K. Nagao. The world through the computer: Computer augmented interaction with real world environments. In *ACM Symposium on User Interface Software and Technology*, pages 29–36, 1995.
- [35] Cristina Portalés Ricart. *Entornos multimedia de realidad aumentada en el campo del arte*. PhD thesis, Universitat Politècnica de València, 5 2008.
- [36] L. Richardson and S. Ruby. *RESTful Web Services*. O’Reilly Series. O’Reilly, 2007.
- [37] Alexander Rubin. Geo (proximity) search with mysql. Technical report, MySQL AB, 2006.
- [38] R. Sears, C. van Ingen, and J. Gray. To blob or not to blob: Large object storage in a database or a filesystem. Technical report, Microsoft Research, Abril 2006.
- [39] W.R. Sherman and A.B. Craig. *Understanding virtual reality: interface, application, and design*, volume 2 of *Morgan Kaufmann series in computer graphics and geometric modeling*. Morgan Kaufmann, 2003.
- [40] Tobias Sielhorst, Tobias Obst, Rainer Burgkart, Robert Reiner, and Nassir Navab. An augmented reality delivery simulator for medical training. In *In International Workshop on Augmented Environments for Medical Imaging - MICCAI Satellite Workshop. 141*, 2004.
- [41] Ian Sommerville. *Software Engineering*. Addison Wesley, 2007.

- [42] Frank Steinicke, Gerd Bruder, Kai Rothaus, and Klaus H. Hinrichs. Poster: A virtual body for augmented virtuality by chroma-keying of egocentric videos. In *Proceedings of IEEE Symposium on 3D User Interfaces (3DUI) (Poster Presentation)*, pages 125–126. IEEE Press, 2009.
- [43] Elliot Stuff. Rockbreaker. <http://research.ict.csiro.au/mobile/research/labs/autonomous-systems/field-robotics/mining-robotics/rockbreaker>, 2009.
- [44] Ivan E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.
- [45] Ivan E. Sutherland. *A head-mounted three dimensional display*, volume 1866. ACM Press, 1968.
- [46] Don Tapscott and Anthony D. Williams. *Wikinomics: How Mass Collaboration Changes Everything*. 2006.
- [47] Toyota. Proyecto iq reality. http://www.toyota.co.uk/cgi-bin/toyota/bv/frame_start.jsp?id=iQ_reality.
- [48] West Virginia University. Have smart phone, will surf – wvu on the move with mobile. <http://wvutoday.wvu.edu/n/2010/4/28/have-smart-phone-will-surf-wvu-on-the-move-with-mobi>, 4 2010.
- [49] UPV. *Manual de Estilos*. Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia.
- [50] Piet van Oostrum. *Page Layout in LaTeX*. Dep. of Computer Science, Utrecht University, March 2004.