

Tesis del Máster de Ingeniería de
Computadores
Geolocalización de obras de arte en la
ciudad de Valencia
Desarrollo de un portal web y aplicación
Android

Autor: Alberto Domínguez López
Director: Enrique Hernández Orallo

Septiembre de 2012

Índice general

1. Introducción	6
1.1. Motivación	7
1.2. Objetivos	7
1.3. Alcance	7
2. Estado del arte	9
2.1. Programación de webs generadas dinámicamente	9
2.1.1. CGI	10
2.1.2. ASP	10
2.1.3. PHP	10
2.1.4. JSP	11
2.2. Programación de aplicaciones móviles	11
2.2.1. Windows Phone	11
2.2.2. iPhone	13
2.2.3. BlackBerry	13
2.2.4. Android	14
2.3. Sistemas de geoposicionamiento	15
2.3.1. Localización GSM	15
2.3.2. Localización WIFI	16
2.3.3. Localización GPS	16
2.3.3.1. Localización GPS asistida	17
2.4. Servidores de mapas	17
2.4.1. Google Maps	17
2.4.2. Bing Maps	18
2.4.3. OpenStreetMap	18
2.5. Paradigmas de programación	19
2.5.1. Modelo-Vista-Controlador	19
2.5.2. Modelo de datos	19

3. Arquitectura de la solución	21
3.1. Especificaciones funcionales	22
3.1.1. Tipos de usuario	23
3.1.2. Registro de usuarios, identificación y almacenamiento, cambio o reestablecimiento de las contraseñas	24
3.1.3. Estado de los usuarios	25
3.1.4. Insertar un nuevo autor	25
3.1.5. Insertar una nueva obra	26
3.1.6. Añadir una fotografía	27
3.1.7. Búsquedas	27
3.1.7.1. Búsqueda de autores	27
3.1.7.2. Búsqueda de obras	28
3.1.8. Foro de opinión	28
3.1.8.1. Uso de avatares	29
3.1.9. Mecanismos de moderación	29
3.1.10. Información propia	30
3.1.11. Subida de fotos geopositionadas	30
3.1.12. Visualización de las obras en un dispositivo móvil	30
3.2. Diseño de la aplicación	30
3.2.1. Base de datos	32
3.2.2. Organización del código	32
3.2.2.1. Usuarios	34
3.2.2.2. Contraseñas	35
3.2.2.3. Foro	35
3.2.2.4. Autores	37
3.2.2.5. Obras	37
3.2.2.6. Reportes	38
3.2.2.7. Conexión a la base de datos	38
3.3. Tecnología empleada	39
3.3.1. Lenguajes de programación	39
3.3.2. Dispositivos móviles	39
3.3.3. Envío de datos mediante XML	40
3.3.4. HTML 5	41

<i>ÍNDICE GENERAL</i>	3
4. Implementación	43
4.1. Puesta a punto	43
4.2. Esquema relacional	44
4.3. El envío de contraseñas a través de la red	47
4.3.1. Uso de HTTPS	47
4.3.2. Mecanismo de desafío-respuesta	48
4.3.3. Resumen de la contraseña	48
4.4. Elección de un método de geolocalización en la aplicación móvil	49
4.5. Elección de Google Maps como servidor de mapas	50
4.5.1. El uso del geocodificador inverso de google	51
4.6. Filtrado de elementos antes de usarlos en la base de datos	52
4.7. Compilación final de la aplicación móvil y posible subida a Google Play	53
5. Pruebas y evidencias de ejecución	55
5.1. Validación y testeo	55
5.2. Evidencias de ejecución	56
5.2.1. Portal web	56
5.2.2. Aplicación Android	67
6. Conclusiones	69
6.1. Experiencia en el desarrollo	69
6.2. Comparativa de los lenguajes de programación y de sistemas móviles	70
6.3. Mercado laboral	70
6.4. Posibilidad de generar ingresos mediante el proyecto	71
6.5. Datos y diseño extrapolables a otras aplicaciones	72
7. Bibliografía	73

Índice de figuras

2.1. Cronología de la introducción de las tecnologías de generación de páginas web de forma dinámica.	9
2.2. Evolución de la demanda de empleo según lenguaje de programación web. <i>www.indeed.com/jobtrends</i>	12
2.3. Distribución del mercado de sistemas operativos en smartphones durante el 2012. <i>www.phonearena.com/news/Android-iOS-wins-are-Symbian-BlackBerrys-losses-in-Q1-2012_id30532</i>	12
2.4. Porcentaje de aplicaciones gratuitas en el <i>Market</i> oficial por plataforma. <i>www.androidguys.com/2010/02/24/distimo-study-reveals-great-deal-android-market</i>	14
2.5. Aplicaciones disponibles en el <i>Market</i> oficial por plataforma. <i>www.androidguys.com/2010/02/24/distimo-study-reveals-great-deal-android-market</i>	15
2.6. Arquitectura Modelo-Vista-Controlador	19
3.1. Arquitectura del sistema	22
3.2. Diagrama de casos de uso del portal web	31
3.3. Diagrama de casos de uso de la aplicación móvil	32
3.4. Diagrama de Entidad - Relación de la base de datos de la aplicación	33
4.1. Modelo relacional de la base de datos	45
4.2. Ejemplo de geolocalización inversa con Google Maps	52
5.1. Página principal de la web	57
5.2. Información contextual en el mapa	58
5.3. Pantalla de registro	58

<i>ÍNDICE DE FIGURAS</i>	5
5.4. Configuración de la cuenta	59
5.5. Listado de temas en el foro	60
5.6. Vista de los mensajes	62
5.7. Añadir un reporte	63
5.8. Añadir un nuevo autor	63
5.9. Añadir una nueva obra	64
5.10. Añadir una foto de obra	64
5.11. Página de búsquedas	65
5.12. Listado de reportes	66
5.13. Moderación de mensajes	66
5.14. Administración de usuarios	67
5.15. Aplicación Android funcionando correctamente, funcionando sin acceso a internet y funcionando sin GPS	67
5.16. Subida de foto geoposicionada	68
5.17. Mapa de las obras en la aplicación Android	68

Capítulo 1

Introducción

El objetivo de este proyecto ha sido la creación de un portal Web y una aplicación móvil que catalogue de forma colaborativa información relevante sobre cualquier tipo de arte urbano que se pueda encontrar en la ciudad de Valencia, con un enfoque cultural y lúdico.

El propósito de la aplicación es unificar información, tanto de los propios usuarios como de fuentes libres de forma útil, accesible y rápida, para mostrarla localizada tanto en un ordenador de sobremesa como en un dispositivo móvil, aprovechando los mecanismos de localización de estos últimos.

Siendo Valencia una ciudad con una gran cantidad de arquitectura, escultura y arte urbano disperso en sus calles, resulta curioso que no haya aplicaciones que aprovechen las nuevas oportunidades que ofrecen los servicios de cartografía y geolocalización de internet para catalogar las obras de modo que se puedan admirar de una forma tan gráfica.

Para la creación del portal web se ha elegido PHP como lenguaje de programación para la obtención de código HTML generado dinámicamente y como gestor de base de datos se ha usado MySQL, mientras que el servidor web es un Apache corriendo bajo una distribución GNU/Linux basada en Debian. Además en la interfaz web se ha hecho uso de características exclusivas de HTML 5. Las razones de estas decisiones son varias. En primer lugar son libres y gratuitas, pudiéndose instalar en multitud de sistemas domésticos o empresariales. Además su curva de aprendizaje es bastante plana y es fácil desarrollar productos amplios y de calidad en poco tiempo.

La aplicación móvil está programada en Android. Es uno de los sistemas operativos móviles más difundidos hoy en día gracias a las funcionalidades que ofrece y a su adaptabilidad a una gran variedad de dispositivos móviles, tanto tablets como smartphones en cualquier rango de precios. Además, su tienda oficial de aplicaciones, Google Play nos ofrece una manera rápida y práctica de difundir nuestra aplicación de forma segura.

1.1. Motivación

Las razones que me han llevado a realizar este proyecto han sido varias. En primer lugar un interés personal en la realización de un proyecto PHP de una envergadura suficiente y en segundo lugar la programación en Android, dado que la realización de este proyecto me ha permitido iniciarme por primera vez en el mundo de las aplicaciones móviles y el geoposicionamiento.

1.2. Objetivos

Los objetivos de este proyecto han sido principalmente el desarrollo de un portal de internet cuyos datos son aportados por sus usuarios, y que ellos mismos se encargan de ampliar y moderar, en lugar de las páginas que he realizado hasta el momento, donde la totalidad de los datos ofrecidos dependían del webmaster.

Además la pequeña aplicación Android me ha resultado útil para comenzar en la programación de dispositivos móviles, en el campo de la geolocalización y a adentrarme en el SDK de Android.

Gracias al portal web y la aplicación se cubre un pequeño nicho de mercado hasta ahora desatendido, bastante útil tanto para habitantes de la ciudad que desconocen en buena medida las obras artísticas que pueden observar en su entorno como para turistas que de este modo podrán pasearse por la ciudad admirando las obras sin necesidad de cargar con voluminosas guías turísticas.

1.3. Alcance

El portal Web pretende ser una solución completa e integrada para todo aquel que desee conocer y catalogar el arte urbano en Valencia. Por ello, el portal tiene varias secciones dedicadas:

- En la vista del **mapa principal** se posicionarán sobre un mapa de la ciudad todas las obras catalogadas en la web y fotos subidas por los usuarios de la aplicación móvil. Al pulsar en cada una de ellas se abrirá un cuadro con la información relevante de la obra.
- La sección de **búsquedas** permite buscar en detalle entre las obras catalogadas y sus autores por multitud de criterios, ya sean nacionalidades, fechas, nombres...
La información resultante estará acompañada de las fotos que los usuarios han agregado.
- La introducción de **información escrita** relativa a las obras y los autores, desde datos objetivos como su nombre, edad o posición geográfica hasta descripciones subjetivas de los usuarios que suben la información.

- La introducción de **imágenes** que documenten de forma gráfica a los autores y las obras que se encuentran catalogadas en la web.
- El **foro de opinión** es el medio de comunicación para relacionarse con los demás usuarios de la web. Es un foro completo, con división temática y posibilidad de aumentar la expresividad de los mensajes mediante el formateado de texto, inclusión de imágenes, uso de avatares...
- Todos los datos, fotos y opiniones con las que contribuyen los usuarios pueden ser objeto de **moderación** por parte de los usuarios. Para ello, pulsando el icono correspondiente se abrirá una página web donde se explicará el motivo y cualquier administrador podrá ocultar o cambiar la información con un solo clic.
- El mecanismo de **registro y autenticación** permite a los usuarios anónimos de la web registrarse y loguearse, usando contraseñas cifradas. También se usa un mecanismo seguro para cambiar la contraseña o para restablecerla en caso de pérdida.
- La sección de **administración** permite a los administradores del sitio de forma gráfica –sin ningún tipo de interacción directa con la base de datos– cambiar o moderar cualquier tipo de información de la web y expulsar temporal o definitivamente a cualquier usuario.

La aplicación móvil es un buen complemento para acceder a la información de la web desde la calle. Ofrece dos servicios:

- El primero sube a la página una **foto geolocalizada** que se mostrará en la interfaz principal de la página web. Con un conjunto suficiente de fotos subidas por el usuario se podrían establecer mecanismos para puntuarlas por los demás usuarios y establecer concursos fotográficos.
- El segundo muestra las obras de arte en un **mapa interactivo** donde se muestra la posición del usuario, que se moverá a medida que cambie la ubicación del móvil. Además, en una pequeña ventana, se mostrará la obra de arte más cercana al usuario y se avisará con un sonido cada vez que tengamos nuevas obras a poca distancia de nuestra ubicación.

Capítulo 2

Estado del arte

En la actualidad existen multitud de tecnologías y plataformas que cumplen los requisitos requeridos para programar el portal web y la aplicación móvil de nuestro proyecto. Si analizamos con detenimiento los requisitos de la aplicación, podremos determinar con mayor fiabilidad cuales son las tecnologías que mejor se adaptan a dichos requisitos.

2.1. Programación de webs generadas dinámicamente

La programación de webs dinámicas generadas en el servidor tiene una larga historia detrás. Mientras que el primer servidor web se activó en 1990, el primer sitio web generado con contenido dinámico se creó sólo tres años más tarde (figura 2.1). Desde aquel entonces han surgido multitud de tecnologías en las que se añan un servidor web, un sistema de gestión de bases de datos y un lenguaje que interactúa con éstos para crear las páginas web que visualizará el usuario.

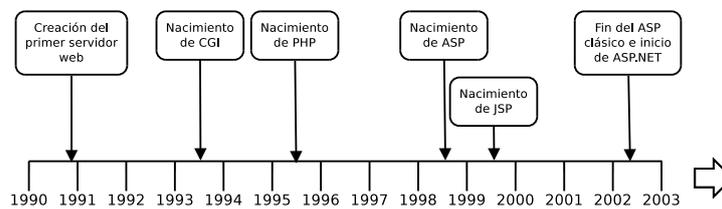


Figura 2.1: Cronología de la introducción de las tecnologías de generación de páginas web de forma dinámica.

2.1.1. CGI

Fue el primer mecanismo que permitía la generación de páginas web dinámicas más que un lenguaje propiamente dicho. Su nombre proviene de las siglas *Common Gateway Interface*. En CGI el servidor web pasa las solicitudes del cliente a un programa externo, que puede estar escrito en cualquier lenguaje que soporte el servidor. La salida de este programa son los datos que se envían al cliente web, en lugar de un archivo HTML estático.

Aunque fue el primer sistema creado, no se usa de forma habitual hoy en día. La dificultad para pasar los datos del servidor web al programa mediante parámetros en forma de cadenas de caracteres y las alternativas más cómodas y fáciles de usar que surgieron posteriormente hacen que este sistema sólo se use por compatibilidad con aplicaciones web antiguas o por aplicaciones muy específicas que requieren interactuar con la máquina a muy bajo nivel. Así se pueden crear ejecutables en C o ensamblador que se llamarán desde el servidor web sin ninguna capa intermedia. Su principal defecto, que afecta a su rendimiento en servidores web de alta demanda es la necesidad que tiene el sistema operativo de hacer un *fork* y lanzar una instancia del programa manejador por cada nueva petición web, en lugar de los sistemas más recientes donde se ejecutan los programas dentro de un módulo del propio servidor web.

2.1.2. ASP

ASP, siglas de *Active Server Pages* es una tecnología creada y patentada por Microsoft que se comercializa como un accesorio de su propio servidor web *Internet Information Services*, más conocido como IIS. Fue lanzada en 1998 pero en 2002 se migró a la tecnología .NET, de forma que muchas de las características que ofrecía el lenguaje cambiaron. Para crear las páginas se puede usar cualquier lenguaje soportado por .NET, como Visual Basic, C#, J# o Delphi.

Su principal desventaja es que estamos usando una tecnología propietaria que además no podemos obtener de forma gratuita. También estaremos obligados a usar el servidor web IIS, y por lo tanto un sistema operativo Windows.

Existen implementaciones alternativas como Sun ONE, actualmente discontinuada, o Arrowhead ASP, pero no ofrecen compatibilidad con la variedad de lenguajes que ofrece Microsoft, y no son compatibles con muchos módulos propietarios existentes en los que normalmente se apoyan algunas de las características de las web.

2.1.3. PHP

PHP es el acrónimo recursivo de *PHP Hypertext Pre-processor*, aunque en un principio significaba *Personal Home Page Tools*. A diferencia de las dos tecnologías anteriores, ésta se basa en la interpretación de código sin necesidad de compilar.

La primera versión fue creada en 1995, pero fue publicado oficialmente por primera vez en su tercera versión, en 1998. En un principio el código estaba escrito en Perl y usaba el mecanismo de CGI para ejecutar los scripts. A partir de la versión 5, PHP soporta el paradigma de orientación a objetos y el manejo de excepciones.

Sus principales ventajas son su sencillez y fácil aprendizaje, y al tener multitud de librerías orientadas prácticamente a cualquier propósito es uno de los lenguajes más usados –el séptimo en el ranking TIOBE–, además de ser libre, gratuito y ampliamente documentado. Como puede verse en la figura 2.2 actualmente es el lenguaje de programación web más demandado por las empresas. Sin embargo es un lenguaje bastante lento en su ejecución y gran consumidor de recursos, por lo tanto su elección para implementar proyectos que atiendan una gran cantidad de peticiones con hardware poco potente probablemente no sea el más adecuado.

2.1.4. JSP

Los JSP, acrónimo de *Java Server Pages* es la tecnología que ofrece Java para generar contenido dinámico en el servidor.

Los JSP realmente son una abstracción de los servlets de Java. De hecho, el código de los JSP son traducidos a servlets en tiempo de ejecución. Por ello la primera visualización de una página dinámica programada en JSP tarda más, ya que el resultado de esa traducción se guarda y solo se cambiará cuando el código del JSP sea alterado.

Una de sus principales ventajas es que los JSP usan Java, un lenguaje de propósito general, apto para crear clases que manejen la lógica de las aplicaciones. Esto permite separar en niveles las aplicaciones web, dejando en los archivos JSP solo el código encargado de generar el código HTML. Además gracias a la portabilidad de Java es posible ejecutar las aplicaciones sin cambios en cualquier plataforma en la que tengamos disponible el servidor de servlets Apache Tomcat, que puede ser cualquiera que disponga de máquina virtual de Java.

2.2. Programación de aplicaciones móviles

Tras la alianza de Microsoft con Nokia para instalar Windows Phone en sus teléfonos dejando Symbian solo en dispositivos de gama baja, actualmente tenemos un mercado de smartphones con los siguientes competidores.

2.2.1. Windows Phone

Es el más reciente de los sistemas operativos móviles, tras el abandono por parte de Microsoft de Windows Mobile. En un principio iba a ser una actualización de éste, pero se desechó la idea para afrontar el reto de crear un sistema operativo para dispositivos móviles totalmente compatible con el sistema

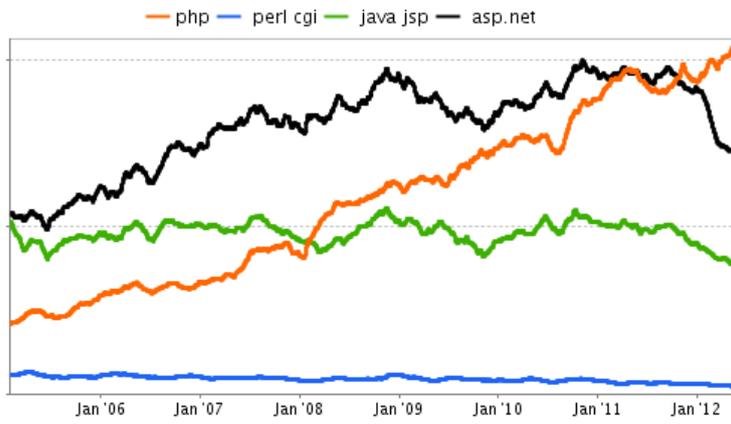


Figura 2.2: Evolución de la demanda de empleo según lenguaje de programación web.
www.indeed.com/jobtrends

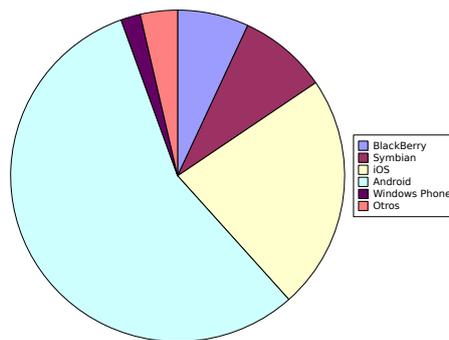


Figura 2.3: Distribución del mercado de sistemas operativos en smartphones durante el 2012.
www.phonearena.com/news/Android-iOS-wins-are-Symbian-BlackBerrys-losses-in-Q1-2012_id30532

operativo para ordenadores de sobremesa Windows 8, copiando sus funcionalidades y su interfaz de usuario. El sistema es rápido y robusto, con una interfaz limpia y fluida. Al estar diseñado desde cero no parte con los problemas inherentes a sus predecesores.

Este sistema operativo está siendo instalado en diferentes modelos de Nokia, HTC, Dell, LG, Fujitsu, Samsung, Acer y ZTE.

Para programar en él hay que emplear su SDK, basado en .NET y Silverlight, lo cual es una ventaja dado la amplia difusión que tiene .NET.

Su principal desventaja, sin embargo, es que ha llegado tarde, cuando el mercado ya se ha decantado claramente por Android e iOS. Es por ello que tiene una cuota de tan solo el 1,9 % de ventas de smartphones (ver figura 2.3). Además es un sistema operativo cerrado, como es habitual en los productos de Microsoft. De hecho para programar en él hay que usar el entorno de programación Visual Studio, cerrado y de pago. Si el teléfono no está desbloqueado –perdiéndose así toda garantía– es imposible instalar ninguna aplicación que no haya sido revisada y aprobada por Microsoft desde su tienda de aplicaciones.

2.2.2. iPhone

Los dispositivos iPhone son uno de los smartphones más populares y consiguen muy buen rendimiento en la ejecución de las aplicaciones. Pero este hecho es consecuencia de algunas políticas de Apple que se tornan en desventajas de cara a su compra y al desarrollo para esta plataforma.

Un dispositivo iPhone es un hardware totalmente cerrado que se vende únicamente como un modelo que se actualiza aproximadamente cada año. Así pues no tenemos libertad para elegir el modelo que más se adapte a nuestras necesidades o a nuestro bolsillo, pues solo podremos elegir entre el modelo actual o algún modelo descatalogado.

Las aplicaciones iOS se ejecutan directamente sobre el hardware del móvil, al contrario que en un dispositivo Android donde una máquina virtual controla un complejo sistema de permisos y capas de compatibilidad, restando potencia a las demás aplicaciones. Esto es así debido a la enorme cantidad de hardware que soporta Android.

Además, debido a que el código de una aplicación en iOS se ejecuta de forma más próxima al hardware, Apple debe comprobar que las aplicaciones que se suben a su tienda de aplicaciones cumplen ciertos requisitos, y en cualquier momento pueden ser borradas de la tienda. A pesar de estos impedimentos, es la tienda con mayor número de aplicaciones (ver figura 2.5).

2.2.3. BlackBerry

Los dispositivos de este tipo, pioneros en su día y orientados a usuarios empresariales con gran necesidad de tener acceso al correo de forma instantánea están en claro declive. La mejora continuada de las pantallas táctiles,

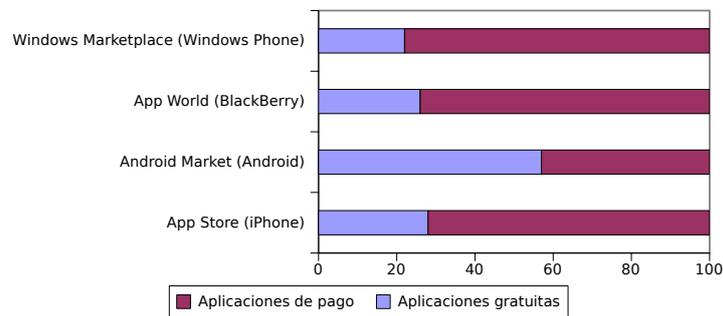


Figura 2.4: Porcentaje de aplicaciones gratuitas en el *Market* oficial por plataforma.

www.androidguys.com/2010/02/24/distimo-study-reveals-great-deal-android-market

como el paso de las pantallas resistivas a las capacitivas, el soporte multitáctil y las diferentes formas de interacción con la interfaz ha provocado que los teclados físicos, típicos de estos teléfonos empiecen a no ser necesarios, y los usuarios están optando por teléfonos Android o iPhone.

Su fabricante, RIM, ha iniciado un cambio en su estrategia, pasando a añadir pantallas táctiles a sus terminales e implementando un nuevo sistema operativo basado en QNX, un sistema operativo Unix de tiempo real.

Las aplicaciones disponibles en su *Market* son muy poco atractivas debido a que tienen que estar orientadas principalmente a dispositivos carentes de pantalla táctil. Además muchos de sus programadores están cambiando de plataforma de desarrollo viendo los importantes cambios que está programando la empresa en sus dispositivos, pensando que no merece la pena migrar a la nueva versión del sistema.

2.2.4. Android

Android es un sistema operativo para dispositivos móviles basado en Linux desarrollado por la *Open Handset Alliance*, liderada por Google. Inicialmente fue desarrollada por Android Inc, comprada por Google en 2005.

Este sistema operativo está desarrollado de forma abierta, al contrario que sus principales competidores, por lo que nada impide que la comunidad de desarrolladores estudie el código fuente y los modifique según sus necesidades. Estos sistemas modificados se suelen llamar *ROMs cocinadas*. Sin embargo, debido a problemas de licencia estas últimas no pueden tener las aplicaciones de Google como Google Maps o Youtube preinstaladas.

El hecho de que Android esté abierto a todas las plataformas también conlleva una serie de desventajas, dado que los diseñadores del sistema operativo no pueden exigir algunos requisitos mínimos en el hardware donde se ejecutará el sistema. Dada la gran variedad de dispositivos que implementan Android podremos encontrar smartphones con poca memoria y un procesador

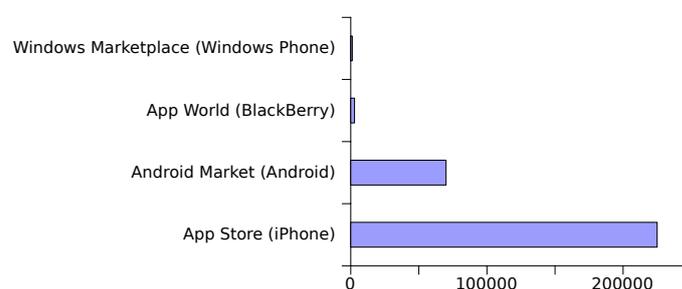


Figura 2.5: Aplicaciones disponibles en el *Market* oficial por plataforma. www.androidguys.com/2010/02/24/distimo-study-reveals-great-deal-android-market

lento que provoca que la interfaz sea muy poco fluida hasta móviles mucho más potentes que el modelo más reciente de iPhone.

También los desarrolladores de las aplicaciones deben tener en cuenta este hecho: mientras que un iPhone es un dispositivo cerrado cuyos desarrolladores conocen perfectamente la potencia y las características del hardware donde se ejecutará la aplicación, un dispositivo Android es totalmente abierto y no podemos aventurar qué tipo de pantalla tendrá, su memoria disponible o si dispondrá de cámara frontal.

Su tienda de aplicaciones está por detrás de la de Apple, pero se observa una tendencia muy a su favor. Un dato interesante es que el market de Google es el único en el que la cantidad de aplicaciones gratuitas superan a las de pago (ver figura 2.4).

2.3. Sistemas de geoposicionamiento

Una de las principales características de la aplicación móvil del proyecto es el uso de los sistemas de geoposicionamiento ofrecidos por Android para indicar la proximidad de las obras al usuario y para indicar el lugar donde se han tomado las fotos que se suben a la plataforma. Android pone a nuestra disposición varios métodos, que se podrán usar en caso de que el móvil donde se ejecute la aplicación disponga de los circuitos adecuados.

2.3.1. Localización GSM

La localización GSM nos ofrece una exactitud aproximada pero muy rápida de obtener. Para detectar nuestra ubicación el móvil comprueba las señales de las estaciones base de telefonía que se reciben y toma su identificación y la potencia con la que llegan. Google dispone de una base de datos con la posición de las estaciones base –en España son datos públicos– y mediante triangulación se puede calcular nuestra ubicación.

En una ciudad el método es bastante efectivo ofreciendo una precisión de menos de 50 metros, debido a la gran cantidad de antenas que se detectan en un espacio reducido. Sin embargo fuera de núcleos urbanos la precisión se degrada a centenares de metros. Además necesitamos una conexión activa a Internet para consultar la base de datos de las estaciones base. Si no disponemos de conexión a Internet no podremos detectar nuestra ubicación incluso en áreas urbanas.

2.3.2. Localización WIFI

Google inició en 2007 su proyecto *Street View*, en las que coches equipados con cámaras fotográficas en su techo fotografiaban las calles y carreteras de varios países. Además de almacenar las imágenes, los coches detectaban el SSID y la dirección MAC de cualquier punto de acceso WIFI que detectaba en su camino, guardando los datos junto a la posición exacta donde se detectaban.

La localización WIFI funciona de forma análoga a la localización GSM, pero debido a que las redes WIFI tienen un alcance de pocos metros, este tipo de localización tiene una precisión mucho más alta. Pero no es un método tan fiable como el anterior, debido a que cualquiera puede coger su punto de acceso inalámbrico y cambiarlo de ubicación.

Además ha sido un proyecto con varios problemas legales debido a que en determinadas legislaciones se reconoce las señales de identificación WIFI – que se difunden de forma libre y sin cifrar– como un dato personal¹.

2.3.3. Localización GPS

La gran parte de los smartphones que salen al mercado cuentan con un receptor GPS. Este sistema usa los datos enviados por 24 satélites estadounidenses para detectar su posición por triangulación. Este sistema tiene muchas ventajas. Su precisión es mucho mayor que los anteriores métodos, llegando a ser menor de dos metros cuando se tiene recepción de más de ocho satélites simultáneamente. Además no precisa de conexión a internet para funcionar y funciona tanto en lugares urbanos como rurales.

Sin embargo tiene dos problemas fundamentales:

- Su consumo de batería es muy elevado. Es por ello que al tomar los datos GPS en la aplicación, se puede configurar para que solo se tomen cuando pase un intervalo de tiempo predefinido desde la última muestra.
- Al iniciar por primera vez una aplicación que use el GPS, el móvil tardará varios minutos en detectar la ubicación por primera vez –tiempo conocido como *time to first fix*–. De hecho, si es la primera vez en 180

¹Para evitar que nuestras redes WIFI sean almacenadas por Google hay que añadir “_nomap” al final de nuestros SSID.

días que no se usa el receptor GPS, este tiempo puede llegar a ser de 15 minutos².

2.3.3.1. Localización GPS asistida

Para evitar que las aplicaciones tarden demasiado tiempo en detectar la señal GPS al activarse, en la configuración de los smartphones se puede activar la opción de “GPS rápido”. Para lograrlo el móvil recibe por internet la información del almanaque, la efeméride y el tiempo UTC con la mayor precisión posible mediante el protocolo NTP. Con estos datos podremos capturar algunos fragmentos de información y enviarlas al servidor de asistencia, que las comparará con las suyas propias, y recibiremos nuestra localización. Cuando el tiempo TTTF transcurre y podamos recibir tramas GPS completas, dejaremos de usar la red y funcionaremos igual que con un GPS tradicional.

Es por esto que usando el GPS asistido podemos obtener nuestra localización incluso en menos tiempo que el TTTF, a cambio de la necesidad de tener una conexión a internet.

2.4. Servidores de mapas

Cuando en nuestro proyecto necesitemos mostrar la información situada sobre un mapa, es muy útil emplear los servicios de cartografía disponibles en internet. En la actualidad las tres opciones más utilizadas son las siguientes:

2.4.1. Google Maps

Es probablemente el servicio más conocido y usado. Sus inicios se remontan a 2004, cuando Google compró el visor de mapas a la empresa *Where 2 Technologies*. El servicio nos ofrece información geográfica de varios tipos como mapas, ortofotos, fotos aéreas oblicuas a 45 grados, fotografías realizadas por usuarios, barridos fotográficos de las calles y carreteras o geocodificación de direcciones.

Además su API, disponible ya en su tercera versión, está basada en Javascript y es muy sencilla de usar. Google ofrece una documentación muy completa y varios tutoriales, pero dada su popularidad multitud de usuarios han escrito cursos y tutoriales propios. Dado que Google lidera el conglomerado de empresas que desarrolla Android, los componentes de Google Maps también están disponibles de forma predeterminada para añadir su funcionalidad en una aplicación móvil de Android. También se puede desarrollar para iPhone y Windows Phone, pero en este caso se deberá usar la versión en Javascript en vez de una propia del SDK del móvil.

²Los satélites GPS difunden, entre otras, dos informaciones indispensables: El almanaque y la efeméride. El primero se transmite cada 12,5 minutos y es válido por 180 días. El segundo se transmite cada 30 segundos y es válido durante 4 horas. La suma de estos dos valores, más el tiempo de adquisición de las señales de localización (TTTF) dan como resultado los 15 minutos.

Sin embargo hay que recordar que el servicio de Google Maps es privado y tiene varias limitaciones para usarlo de forma gratuita. Aunque los límites han aumentado debido a la competencia de otras compañías, sigue siendo un servicio caro³.

2.4.2. Bing Maps

Bing Maps es el nombre actual del servicio Windows Live Maps, disponible desde 2005. Las funcionalidades que ofrece son similares a las de Google Maps, pero no ha alcanzado su mismo nivel de popularidad. Su API está basada en Javascript y usa de forma extensiva las tecnologías AJAX. Para programar aplicaciones móviles, Microsoft ofrece su servicio de forma predeterminada al programar en Windows Phone, pero también está disponible para iPhone y Android.

Al igual que Google Maps es un servicio privado, si bien las condiciones de uso libre son diferentes, siendo gratis el uso si los mapas se muestran en páginas abiertas al público, sin contraseñas o registros para acceder y hay menos de 500.000 visualizaciones anuales. Los precios para cantidades mayores no son públicos, sino que Microsoft los negocia personalmente con las compañías interesadas.

2.4.3. OpenStreetMap

El servicio OpenStreetMap, disponible desde el 2004 es uno de los proyectos más novedosos debido a su principal virtud: los mapas no son ofrecidos por una empresa de cartografía y aerofotometría, sino que son los propios usuarios los que elaboran la información y la corrigen usando sus propios datos o datos provenientes de fuentes libres, ofreciendo de esta forma un mapa en continua evolución.

Para programar con OpenStreetMap hay que usar las librerías de Javascript OpenLayers, de las cuales hay gran cantidad de recursos didácticos. Sin embargo, un gran problema al usar este servicio consiste en que los mapas ofrecidos por OpenStreetMap lo son con el propósito de su edición y consulta, pero si se pretenden usar para crear aplicaciones propias no se deben consultar los datos online, sino que se ha de bajar la información para generar y ofrecer los tiles desde tu propio servidor. Esto dificulta enormemente el uso de esta herramienta para buena parte de los casos.

³Google permite visualizar un mapa 25.000 veces cada día por dominio web. Cada mil visitas adicionales constituye un cargo de 4 USD. Una empresa que haga un uso considerable de este servicio puede llegar a pagar cientos de miles de dólares.

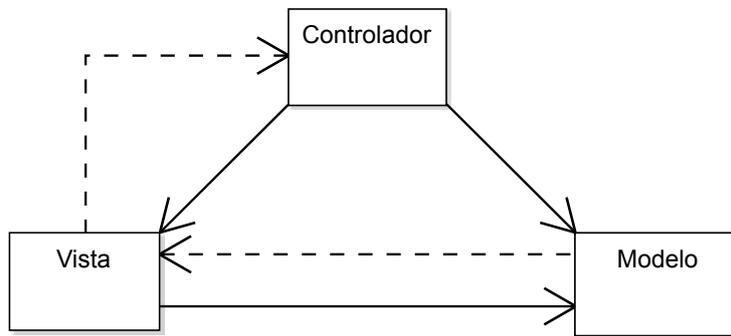


Figura 2.6: Arquitectura Modelo-Vista-Controlador

2.5. Paradigmas de programación

2.5.1. Modelo-Vista-Controlador

La arquitectura Modelo-Vista-Controlador es un patrón de diseño que define una separación clara entre los tres componentes básicos de nuestras aplicaciones. Como su nombre indica, MVC define las tres partes de una aplicación:

- Un **modelo** proporciona los datos internos y los métodos que ofrecen información al resto de la aplicación, pero no define ni la apariencia que se le muestra al usuario ni su comportamiento.
- Las **vistas** constituyen la interfaz de usuario. Una vista está compuesta por un documento HTML y sus elementos en el caso del portal web y una actividad en el caso de la aplicación Android.
- Un **controlador** suele estar vinculado a una vista. El controlador es responsable de recibir la entrada del usuario y reaccionar en consecuencia. Los controladores pueden acceder y actualizar una vista usando información del modelo, así como actualizar el modelo empleando el resultado de las interacciones del usuario en la vista. En definitiva, enlaza los otros dos componentes del MVC.

2.5.2. Modelo de datos

El portal web del proyecto requiere una interacción constante con un almacén de datos centralizado para lograr todos los requisitos que demandamos al sistema. Dado que para acceder se requiere una conexión a internet, es evidente que la única opción útil para obtener los datos es acceder a una base de datos remota.

Sin embargo para la aplicación móvil surge una duda. Si bien para subir fotos al servidor es necesario una conexión a internet, para posicionar en un plano los datos catalogados no es estrictamente necesario. Teniendo en cuenta que

puede ser una aplicación muy útil para turistas, y éstos probablemente no tengan acceso a internet móvil a causa de los altos precios de los datos en roaming, deberíamos tener en cuenta dos escenarios:

- Usar un **modelo local** de datos donde almacenaríamos todas las posiciones a mostrar en el mapa en una base de datos en nuestra propia aplicación. De forma periódica –cada mes, por ejemplo– se actualizaría el programa ofrecido a través de los canales de distribución con los datos actualizados de nuestra web.
- En cambio, el **modelo remoto** de datos necesitaría conectarse a internet cada vez que se inicie la aplicación móvil para comprobar que hayan nuevos datos en el servidor, y en su caso descargarlos como un fichero XML resultante de realizar las consultas pertinentes a la base de datos.

Capítulo 3

Arquitectura de la solución

En el siguiente capítulo describiremos los pasos que hemos seguido para diseñar un sistema, desde su concepción hasta el inicio de su codificación. Esta es la fase más importante del proyecto dado que cualquier cambio que queramos realizar o error que detectemos en la fase siguiente requerirá cambios tanto en la base de datos como en el código del controlador.

Para empezar, en la figura 3.1 se muestra el esquema de la arquitectura que implementaremos finalmente y a continuación comentaremos su forma de interactuar con los clientes

En el caso de usar un cliente web para acceder a Valengeoarte, el acceso a los recursos siguen el siguiente orden:

- (1) El usuario desde un navegador de internet entra a nuestra dirección y el servidor HTTP Apache envía el HTML de la página principal.
- (2) El código HTML incluye referencias al API de Google Maps que el navegador tendrá que descargar del servidor de Google.
- (3) Cuando la página web haya resuelto sus enlaces se ejecutará el código Javascript que contiene. Éste, de forma asíncrona, descargará de nuestro servidor dos ficheros XML con las obras y las fotos (ver sección 3.3.3), y mediante funciones de Google Maps se situarán en los puntos del mapa que correspondan.

Cuando se acceda a las funciones de Valengeoarte que se ofrecen desde la aplicación Android, se siguen más pasos pero buena parte de ellos son transparentes ya no solo al usuario, sino también al programador:

- (4) Al iniciarse la aplicación, el móvil detecta la identidad y potencia con la que se reciben las señales de las estaciones base de telefonía más próximas. Este paso es necesario para iniciar el GPS de modo asistido (ver sección 2.3.3.1).

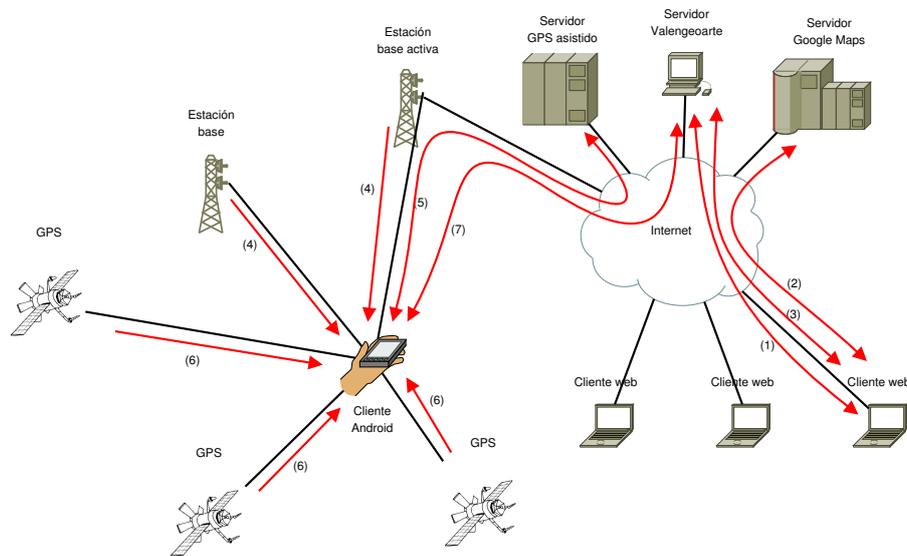


Figura 3.1: Arquitectura del sistema

- (5) Los datos son enviados de modo transparente al programador por el sistema Android a un servidor de Asistencia GPS. En él se identifican y localizan las estaciones base que ha detectado el móvil y se trianguliza la posición de forma aproximada teniendo en cuenta la potencia de la señal recibida. Tras su cálculo se envía al móvil la ubicación aproximada, el almanaque y la efeméride (ver sección 2.3.3).
- (6) Con los datos recibidos se puede acceder de rápidamente a la señal de los satélites GPS disponibles y conseguir una localización mucho más precisa en muy poco tiempo.
- (7) Cuando ya se ha geolocalizado el móvil, la aplicación permite conectar con nuestro servidor para subir una foto o para descargar un fichero XML con las obras, que se representarán en un mapa.

3.1. Especificaciones funcionales

Como se ha descrito en los objetivos del proyecto en la sección 1.2, la aplicación está enfocada a ofrecer información sobre el arte urbano al usuario de la forma más útil posible. Esto implica la necesidad de mostrar la información del modo más adecuado en cada momento. A través de un mapa para ver las obras de un vistazo, mediante una completa opción de búsqueda o en una aplicación móvil para poder acceder a la información en la calle. Además es necesario distinguir entre varios tipos de usuarios, pues necesitamos que no todos los usuarios tengan los mismos permisos para acceder o modificar la información almacenada. En la figura 3.2 podremos ver un esquema con los

casos de uso del portal web mientras que en la figura 3.3 estarán los casos de uso de la aplicación móvil.

3.1.1. Tipos de usuario

Los usuarios que visiten el portal Web están divididos en tres tipos, siendo cada uno de los grupos un supergrupo del anterior:

Usuarios anónimos: Son los usuarios que entran a la web por primera vez, o lo hacen sin identificarse. Este grupo de usuarios tiene permisos únicamente para obtener la siguiente información:

- Visualizar el mapa principal con las obras de arte geopositionadas y ver su información asociada.
- Realizar búsquedas sobre la base de datos de autores y obras.
- Entrar al foro de opinión pudiendo únicamente leer mensajes.
- Ver información estática de la Web como puede ser el apartado con información sobre el proyecto.
- Usar la aplicación móvil, tanto para subir fotos instantáneas geopositionadas como para visualizar el mapa.

Usuarios identificados: Son los usuarios que se han registrado de forma correcta en la web y se han identificado correctamente en el sistema. Este tipo de usuarios, además de los permisos anteriores, pueden realizar más acciones:

- Introducir nuevas obras, posicionarlas y catalogarlas en el sistema.
- Añadir información fotográfica relacionada con un autor u obra concreta.
- Crear nuevos temas y añadir nuevos mensajes en el foro de opinión.
- Poder notificar a los administradores discrepancias con respecto a la veracidad de la información aportada por otros autores –usualmente conocido como reportar– ya sea en obras, autores, mensajes del foro, fotografías instantáneas, o cualquier tipo de información de la aplicación.

Administrador: Es el usuario que al identificarse tiene todos los permisos posibles en el portal web. A los permisos anteriores hay que añadir:

- La capacidad de leer los reportes que crean el resto de usuarios.
- Cambiar cualquier tipo de información almacenada u ocultarla al resto de usuarios, ya sean obras, autores, mensajes o temas del foro, fotografías...
- Cerrar temas del foro, de modo que no admita más mensajes aunque se puedan seguir leyendo los existentes.
- Expulsar los usuarios de forma temporal o definitiva, o borrar su foto identificativa –avatar– con la que se representa su identidad en el foro.

3.1.2. Registro de usuarios, identificación y almacenamiento, cambio o reestablecimiento de las contraseñas

Debido a que tenemos que distinguir a los usuarios que entran a nuestro sistema, el método ideal para diferenciarlo es que deban registrarse e identificarse debidamente. Para lograr este objetivo se le piden dos datos al usuario: su correo electrónico y un nombre identificativo que se mostrará en los mensajes del foro. Los dos datos han de ser únicos en nuestra web.

La primera contraseña del usuario en la web se enviará por correo junto a un mensaje de bienvenida. Dicha contraseña estará formada por una combinación de ocho caracteres que pueden ser letras mayúsculas, minúsculas o dígitos. De esta forma solucionamos dos problemas:

1. Verificamos que la cuenta de correo electrónico que ha introducido el usuario es correcta, en uso y le pertenece.
2. Evitamos que los usuarios elijan una contraseña poco adecuada o criptográficamente débil.

En cualquier caso los usuarios pueden cambiar su contraseña en cualquier momento. Para realizar el cambio se deberá solicitar la opción e introducir la contraseña actual y la nueva. El único requisito que se le pide a las contraseñas es que tengan al menos cinco caracteres y no contengan espacios. Para evitar la transmisión de contraseñas sin cifrar por la red se han barajado varias opciones que se discuten en 4.3.

Cuando un usuario se registra no queda almacenado como un usuario activo, sino como un usuario pendiente. Si al cabo de veinticuatro horas de su registro no se ha identificado se borrará automáticamente del sistema. Este método tiene como objetivo evitar que un atacante provisto de una libreta de direcciones realice el alta de múltiples usuarios de forma fraudulenta, o que registre repetidamente a la misma persona para saturar su buzón de correo¹.

Si un usuario no recuerda su contraseña dispondrá una opción para restablecerla. Se le enviará a la cuenta de correo con la que se registró un mensaje que contendrá un código de quince caracteres que pueden ser letras mayúsculas, minúsculas o dígitos, que será válida únicamente durante los diez minutos siguientes a su generación. Tras el envío se mostrará un formulario donde se introducirá el código enviado y la nueva contraseña de acceso que elija el usuario. Sólo se podrá solicitar una petición de recuperación de contraseña al día por cuenta.

Cualquier atacante podría solicitar recuperar una contraseña de una cuenta ajena, sin embargo los códigos de recuperación son lo suficientemente robustos y tienen un tiempo de vida suficientemente corto para asegurar su

¹Se podría pensar en establecer un sistema de CAPTCHA propio o usar un servicio gratuito como reCAPTCHA, pero no lo he considerado necesario por ser un elemento ciertamente molesto y con las restricciones planteadas un atacante no tiene demasiada capacidad para dañar el sistema.

seguridad².

En cualquier momento un usuario puede cerrar la sesión de su usuario y pasar a ver la web como un usuario anónimo, o iniciar la sesión como otro usuario distinto. Para ello sólo tendrá que cerrar el navegador o pulsar en la opción de cerrar sesión.

3.1.3. Estado de los usuarios

Como se comentó en el apartado 1.3, un administrador puede cambiar según su criterio el estado de un usuario registrado en cualquier momento. Los estados son:

Usuario sin autenticar: Usuario que se ha registrado en la web pero no se ha identificado aún. Si tarda más de un día en hacerlo desde su registro, la cuenta se borrará.

Usuario activo: Usuario que puede hacer uso de todos los recursos de la web sin restricciones.

Usuario baneado: Un administrador puede banear un usuario y éste no podrá acceder a la web durante una semana. Si intenta iniciar una sesión se le informará con un aviso. El sistema comprobará periódicamente el tiempo transcurrido desde el *baneo* y cuando el plazo de tiempo se cumpla lo pasará automáticamente como un usuario activo de nuevo.

Usuario expulsado: El administrador, motivado por una falta grave puede expulsar definitivamente al usuario. Si intenta iniciar sesión un aviso le informará que ya no podrá hacerlo³.

En cualquier caso, la expulsión o *baneo* de un usuario no implica que sus aportaciones a la web se oculten o desaparezcan. Sin embargo no podrán seguir aportando información.

3.1.4. Insertar un nuevo autor

Para insertar un nuevo autor, un usuario identificado tendrá que seleccionar la opción correspondiente del menú y se mostrará un formulario en el que se pedirán los siguientes datos:

- Nombre del autor.

²La contraseña consta de una palabra de 15 caracteres de longitud con un alfabeto de 62 símbolos (26 letras mayúsculas, 26 minúsculas y 10 cifras) equivalente a 90 bits aproximadamente. A una media de 100 comprobaciones por segundo, en diez minutos no habríamos probado ni una trillonésima parte de las posibilidades.

³Nada impide a un usuario expulsado visualizar la web como un usuario sin registrar o registrarse de nuevo usando otro nombre y dirección de correo electrónico. Se puede confiar en que la comunidad de usuarios detectaría la falta. También se podría implementar un sistema de identificación de IP del usuario, pero su efectividad sería dudosa dado que la gran mayoría de las conexiones a Internet en España tienen asignadas IPs dinámicas.

- País de nacimiento (en el caso de ser conocido). Se podrá elegir de entre una lista de países que habrá insertado el administrador.
- Año de nacimiento (en el caso de ser conocido).
- Año de defunción (en el caso de ser conocido). También se puede indicar que sigue vivo.
- Una breve biografía del autor.

Antes de almacenar el autor en la base de datos tenemos que comprobar que los datos son correctos: El autor no debe existir anteriormente en la base de datos y el año de nacimiento debe ser anterior al de defunción.

3.1.5. Insertar una nueva obra

Para insertar una nueva obra, un usuario identificado seguirá unos pasos similares a los realizados para insertar un autor, sólo que esta vez hay que indicar más datos.

- Nombre de la obra.
- Nombre del autor (en el caso de ser conocido).
- Tipo de obra. Se podrá elegir entre una lista que habrá definido el administrador. Podrán ser esculturas, arquitectura, puentes, fuentes...
- Una breve descripción de la obra.
- El año de inauguración o creación de la obra (en el caso de ser conocido).
- La localización de la obra. Se podrá seleccionar en un mapa el punto exacto donde se encuentra la obra o introducir la latitud y la longitud de forma manual.

Los datos se validarán con las siguientes reglas:

- El autor debe existir en la base de datos.
- Un mismo autor no puede tener dos obras con el mismo nombre.
- Debido a que la web está pensada exclusivamente para catalogar el arte urbano de la ciudad de Valencia, el punto donde esté situado la obra debe estar dentro del término municipal del mismo, ya sea en la propia ciudad o en sus pedanías, algunas tan alejadas como el Saler, el Perellonet, Massarrojos o Vistabella.

Se pensó en un primer momento en añadir una condición para asegurar que la fecha de inauguración estuviese comprendida entre los años de vida del autor, pero se deshechó para permitir la posibilidad de añadir algún tipo de obra póstuma o inacabada.

3.1.6. Añadir una fotografía

Se pueden añadir fotos para describir la biografía de un autor o la descripción de una obra. Cualquier usuario registrado puede añadir fotos a estos datos sin necesidad de que la biografía o la descripción hayan sido subidas por él mismo. Uno de los puntos clave de las webs colaborativas es que cualquiera puede añadir y complementar información aportada por otra persona.

Para subir las imágenes a la web se seleccionará la imagen, que deberá estar codificada como PNG, GIF o JPEG y ocupará menos de 1 MB. En cualquier caso el servidor guardará solo fotos en formato JPEG con una resolución horizontal máxima de 800 píxeles, así que en la mayoría de los casos se deberá recodificar la imagen en el servidor. Estos requisitos buscan cumplir un objetivo:

- Como las imágenes a subir son fotografías, no diagramas o esquemas, el formato más adecuado es el JPEG pues es el que aporta una mejor relación entre calidad y espacio en este tipo de imágenes.
- Las imágenes a subir deben ocupar menos de 1 MB para evitar saturar el caudal de bajada del servidor en medida de lo posible.
- Su resolución horizontal ha de ser menor de 800 píxeles para evitar que el diseño web de la página se vea comprometido cuando cargue fotografías más anchas que el tamaño de la columna principal.

A la imagen se le debe acompañar un pequeño texto descriptivo a modo de pie de foto que servirá tanto como alternativa textual de la imagen como para describir la foto en las búsquedas.

3.1.7. Búsquedas

Las búsquedas parametrizadas en la página web se podrán realizar tanto para las obras como para los autores registrados. Cualquier usuario podrá entrar en la sección y podrá filtrar su búsqueda mediante los siguientes criterios:

3.1.7.1. Búsqueda de autores

Para filtrar un autor registrado en la web se puede usar cualquiera de los siguientes criterios de búsqueda:

- Buscar por el nombre del autor. Al empezar a introducir el nombre en el campo de texto aparecerán automáticamente sugerencias con nombres de autores disponibles en la web.
- Buscar por su país de origen en un desplegable.
- Buscar los autores que estuviesen vivos en un año determinado. Para cumplir este criterio hay que tener en cuenta que hay autores que pueden estar vivos en la actualidad y otros cuyo año de nacimiento o de defunción pueden ser desconocidos.

La lista de autores y sus fotos descriptivas aparecerán en pantallas de 10 resultados para evitar saturar el servidor en medida de lo posible.

3.1.7.2. Búsqueda de obras

Para la búsqueda de obras los criterios son más amplios:

- Se pueden filtrar todas las obras pertenecientes a un grupo temático señalándolo en un desplegable. Así se filtrarán todas las esculturas, fuentes, puentes...
- Buscar por el año de inauguración.
- Buscar por el título de la obra.
- Buscar por el autor de la obra.

Las búsquedas mediante título y autor se realizarán introduciendo los datos en una caja de texto que ofrecerá sugerencias a medida que se completen los datos.

Los resultados mostrarán además de su título, descripción y fotografías asociadas los datos referentes a su autor. Además un enlace situará la obra en un mapa abriendo una nueva ventana del navegador.

3.1.8. Foro de opinión

El foro de opinión tiene una apariencia y un uso similar a cualquier foro convencional que puede verse en internet. Al seleccionar la opción en el menú de la web se visualizará una lista de temas, ordenados por la fecha del último mensaje publicado. Cada tema en la lista tendrá un icono que los diferencie según el tema contenga o no mensajes nuevos o haya sido cerrado por el administrador. También se mostrará:

- El nombre del usuario que abrió el tema y en que momento lo hizo.
- El nombre del usuario que publicó el último mensaje en el tema.
- Acceso directo a cualquiera de las páginas en las que se divide la lista de mensajes, habiendo diez en cada una de ellas.

Cuando se entre en un tema concreto, aparece una lista con los mensajes en los que se muestra información relativa a su autor –su avatar, su estado y su nombre– y relativa al mensaje, como la fecha de publicación.

La vista del foro es distinta según el tipo de usuario que acceda:

- Un usuario sin registrar no podrá abrir temas nuevos ni añadir mensajes en los temas abiertos. Además no podrá comprobar si hay mensajes nuevos en los temas desde la última vez que entró.

- Un usuario registrado sí podrá hacerlo. Además los temas que contengan mensajes nuevos desde la última visita se mostrarán con un icono que los diferenciará.
- La vista del administrador tendrá una pequeña diferencia con la de un usuario identificado. Al ver los mensajes de cada tema, se dispondrá de un botón de moderación individual en cada mensaje. De este modo podrá ocultar comentarios que no cumplan las normas del portal al navegar por él, sin necesidad de recibir ningún aviso previo.

3.1.8.1. Uso de avatares

Todos los mensajes del foro van acompañados de una pequeña imagen a modo de avatar para identificar al usuario que ha publicado cada mensaje en un golpe de vista. Todos los usuarios en un primer momento tienen un avatar predefinido. No obstante pueden cambiarlo en cualquier momento en la opción de configuración. Se puede subir cualquier tipo de imagen siempre que ocupe menos de 1 MB, pero en el servidor se recodificará como una imagen PNG cuadrada de 100 píxeles cada lado.

Hay que tener precaución al elegir la imagen para el avatar y si es necesario modificarla antes enviarla, ya que si la imagen aportada no es cuadrada en la recodificación sufrirá un cambio de aspecto y se mostrará deformada.

3.1.9. Mecanismos de moderación

Cualquier tipo de información aportada por los usuarios que se visualiza en la página no pasa ningún proceso de filtrado antes de su publicación. Por ello es casi indispensable establecer un tipo de moderación distribuida cuya responsabilidad recaiga en un primer paso sobre los usuarios debidamente autenticados.

De este modo, cuando se visualicen obras, autores, fotografías o mensajes del foro se mostrará un pequeño icono para avisar al administrador de la existencia de un contenido que los usuarios creen inadecuado. Al pulsarlo se abrirá una página con un recuadro de texto donde el usuario debe explicar el motivo por el cual considera que el administrador debe cambiar el contenido del mensaje, ocultarlo o sancionar al autor correspondiente.

Cuando el administrador abra la sección del buzón de reportes, aparecerán ordenados por orden de llegada los avisos de los usuarios. En ellos constará:

- El usuario que ha generado el reporte.
- El usuario que ha generado el dato que se reporta.
- La fecha del reporte.
- El comentario del autor del reporte.
- Un enlace al dato reportado.

Si el administrador pulsa sobre el enlace se abrirá una nueva página con el dato reportado por el autor al que se le podrá cambiar cualquier detalle, o bien se podrá ocultar a los usuarios.

También se podrá pulsar sobre el nombre de los usuarios para poder sancionarlos, ya sea por un aviso falso o por no cumplir las normas de la web.

3.1.10. Información propia

La página web debe tener además una pequeña sección que ofrezca información sobre la página, sus autores y administrador y un modo de ponerse en contacto con ellos, ya sea mediante un enlace de correo electrónico o bien como un formulario web.

3.1.11. Subida de fotos geoposicionadas

Mediante la aplicación móvil y alguno de los sistemas de geolocalización que ofrecen los smartphones se debe poder subir al servidor una fotografía, realizada en ese mismo instante junto a los datos de latitud y longitud detectados. No se debe permitir subir una fotografía que no tenga datos de localización o que éstos no sean válidos.

En la página web se podrán visualizar las fotos de los usuarios en el mapa de la página principal y podrán ocultarse mediante un botón para dejar a la vista sólo las obras debidamente catalogadas.

3.1.12. Visualización de las obras en un dispositivo móvil

Otra de las características que debe implementar la aplicación móvil es la visualización de un mapa de la ciudad con los iconos de las obras dibujados. Además el mapa se debe centrar en la posición actual del usuario obtenida mediante el sistema de geolocalización del *smartphone*. Al pulsar encima de cada obra debe abrirse un pequeño cuadro de texto con el nombre de la obra, su autor y una breve descripción.

Además encima del mapa debe situarse un pequeño recuadro indicando la obra más cercana y los metros que nos separan de ella. Cada vez que cambie nuestra posición con respecto a las obras, se debe calcular cual es la más cercana y en caso de cambio, señalarlo mediante un aviso sonoro.

3.2. Diseño de la aplicación

En esta sección mostraremos las soluciones que hemos empleado para poder cumplir los requisitos expuestos en la sección anterior. Para ello tendremos que definir claramente la estructura de la base de datos subyacente a la aplicación y la organización del código que accede a los datos.

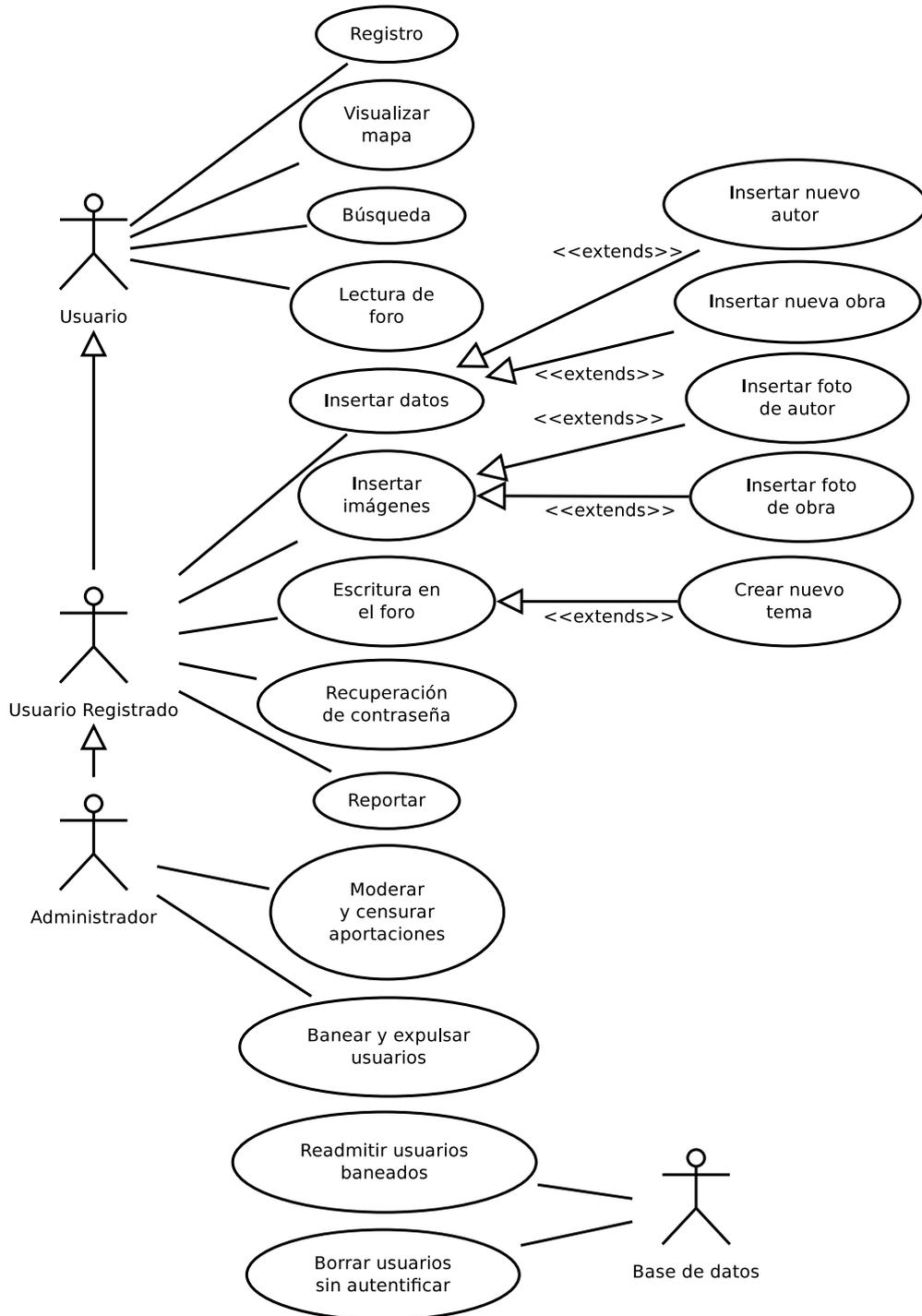


Figura 3.2: Diagrama de casos de uso del portal web

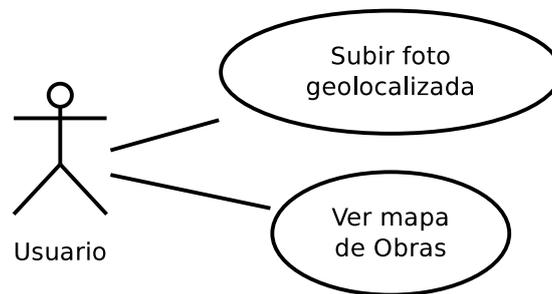


Figura 3.3: Diagrama de casos de uso de la aplicación móvil

3.2.1. Base de datos

Para diseñar nuestra aplicación hemos decidido que una única base de datos suministre la información a todas las secciones de la web y a la aplicación móvil. Esto nos proporciona grandes ventajas, como la coherencia de los datos entre las aplicaciones, mayor seguridad y facilidad de mantenimiento. De este modo no será necesario por ejemplo replicar los datos referidos a los usuarios si necesitan usar el foro o subir una foto.

Sin embargo, si el proyecto crece demasiado el uso de una base de datos centralizado provocará que las tareas necesarias para crear o restaurar copias de respaldo sean cada vez más lentas y costosas y las consultas pueden llegar a saturar su capacidad.

Tras analizar las necesidades de la aplicación, se han redactado una serie de entidades:

- Usuarios
- Autores
- Obras
- Fotos de autores
- Fotos de obras
- Fotos geoposicionadas
- Reportes

Que se relacionan entre sí siguiendo el diagrama de entidad-relación mostrado en la figura 3.4. Para simplificar el diagrama no se han añadido los atributos de las entidades, pero se pueden ver en el diagrama del modelo relacional en la figura 4.1.

3.2.2. Organización del código

Nuestro código en PHP será organizado en forma de clases, que serán instanciadas por las vistas. Estas clases acceden a la base de datos y nos devuelven

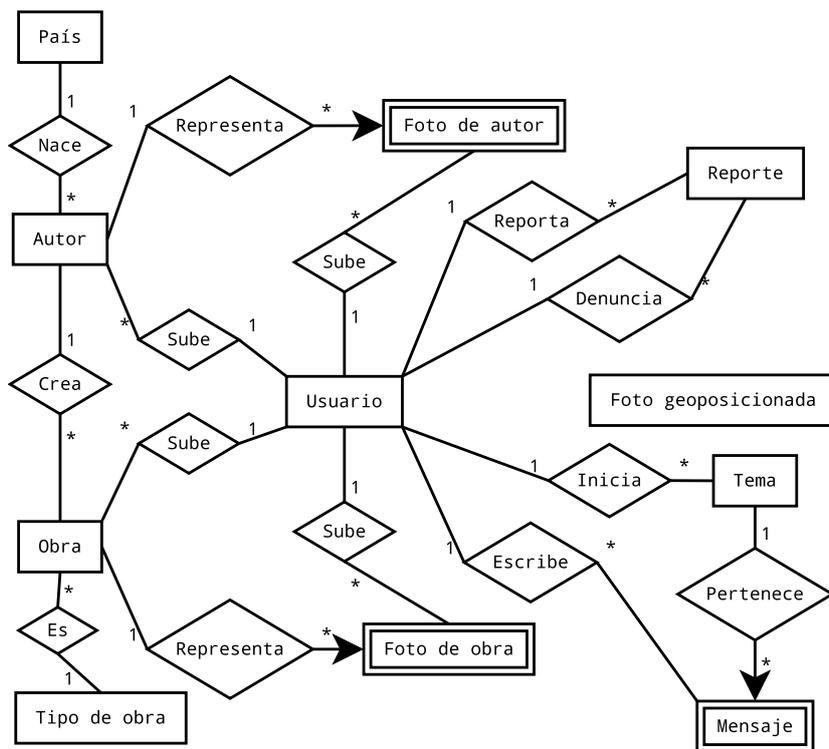


Figura 3.4: Diagrama de Entidad - Relación de la base de datos de la aplicación

los resultados de las consultas mediante valores o vectores. Dado que la aplicación no cuenta con la complejidad suficiente no he visto necesario el uso de las herramientas típicas de la orientación a objetos como la herencia o el polimorfismo. Sin embargo sí se han usado las herramientas que modifican la visibilidad de los métodos.

Los siguientes diagramas de clase muestran las vistas, las clases con sus métodos públicos y otras estructuras de datos asociadas.

3.2.2.1. Usuarios

Usuarios <i>Clase</i>	Usuario <i>Estructura</i>
<pre>+insertarUsuario(nombre:String,mail:String): (Boolean, String) +login(password:String,mail:String): (Boolean, String) +numUsuarios(): Int +listarUsuarios(pag:Int): [Usuario] +moderarUsuario(idUsuario:Int,borrarAvatar:Boolean, estado:Enum->(Act, Sin, Ban, Fir)): Boolean +buscarUsuarioPorId(idUsuario:Int): Usuario +nuevoAvatar(rutaAvatar:String): (Bool, String)</pre>	<pre>+idUsuario: Int +nombre: String +mail: String +rutaAvatar: String +password: String +estado: Enum->(Act, Sin, Ban, Fir) +fechaRegistro: Time +fechaUltimaVisita: Time</pre>

La clase Usuarios contiene el código encargado del registro, identificación y moderación de los usuarios, así como el establecimiento de avatares.

El método insertarUsuario devuelve una tupla de dos valores: el primero indica si la operación ha tenido éxito mientras que el segundo es un posible mensaje de error. Es una función que puede fallar en cualquiera de los pasos que realiza.

1. Al insertar un usuario con un nombre de usuario o un correo que ya exista en la base de datos.
2. Si la dirección de correo no es válida.
3. Si el servidor de correo no se encuentra disponible en ese momento.

En cualquier caso, como se indica en la sección 3.1.2 hasta que el usuario no se autentifique en el sistema por primera vez, la cuenta de usuario permanecerá inactiva.

El valor devuelto por moderarUsuario y login es TRUE en caso de que la operación finalice con éxito. En el caso de login un String informará del error y será mostrado en la vista. El método buscarUsuarioPorId devuelve el valor NULL en caso de no encontrar al usuario.

Cuando queremos mostrar en una página web una lista de datos muy larga es probable que tarde un tiempo excesivo al cargar y que sobrecarge nuestro servidor, así que en los listados deberemos aportar un parámetro *pag* en el que indicaremos la página del listado que queremos obtener. Cada página contendrá diez datos nuevos. Este sistema se repite de forma sistemática en todos los métodos que ofrecen un listado de datos.

Para moderar un usuario deberemos identificar el usuario y señalar si queremos que pueda continuar con su actual avatar o si bien queremos cambiar

su estado. No se permitirá ningún otro cambio, como el cambio de mail o de nombre. La vista está diseñada para impedir que un usuario que no sea administrador pueda ejecutarlo. Si el usuario al que se referencia con *idUsuario* no se encuentra en la base de datos, el método devolverá FALSE.

Para establecer un avatar diferente, un usuario registrado mediante la vista de configuración de cuentas puede ejecutar el método `nuevoAvatar`, simplemente indicando la ruta a la imagen. El valor devuelto indicará el éxito de la operación y el posible mensaje de error en su caso.

3.2.2.2. Contraseñas

Contraseñas
<i>Clase</i>
+CambiarPassword(idUsuario:Int,PasswordActual:String,nuevaPassword:String): (Boolean, String)
+enviarMailRecuperacion(mail:String): (Boolean, String)
+recuperarPassword(codigo:String,nuevaPassword:String): (Boolean, String)

A través de la clase `Contraseñas` tenemos acceso a los métodos para cambiar y recuperar la contraseña de los usuarios.

En los tres casos las tuplas que devuelven las funciones son un booleano indicando el éxito de la operación y un mensaje que se rellena en caso de fallo y se mostrará al usuario en la vista.

El uso de `enviarMailRecuperación` y `recuperarPassword` es el siguiente:

- Un usuario no identificado, mediante la vista `recuperarContraseña` indica su mail. En caso de que el mail esté registrado se le envía al usuario un mensaje con el código de recuperación usando la librería `PHPMailer` y la cuenta de Gmail asociada a la web.
- Cuando el usuario recibe el correo, mediante la misma vista se llamará al método `recuperarPassword` con el que si hay éxito se restablecerá la contraseña con un nuevo valor. Hay que recordar que las contraseñas que se pasen estarán siempre codificadas como SHA1 (ver la sección 4.3.3).

La vista está diseñada para que un usuario registrado solo pueda cambiar su propia contraseña, pero un administrador pueda cambiar la contraseña de cualquier usuario.

3.2.2.3. Foro

Foro <i>Clase</i>	Tema <i>Estructura</i>
<pre>+numTemas(): Int +listarTemas(pag:Int): [Tema] +nuevoTema(titulo:String): Void +buscarTemaPorId(idTema:Int): Tema +moderarTema(idTema:Int,estado:Enum->(Visible, Moderado, Cerrado),nuevoTema:Tema): Void +numMensajes(idTema:Int): Int +listarMensajes(pag:Int,idTema:Int): [Mensaje] +nuevoMensaje(idTema:Int,texto:String): Boolean +buscarMensajePorId(idMensaje:Int): Mensaje +moderarMensaje(idMensaje:Int,estado:Enum->(Visible, Moderado),nuevoMensaje:Mensaje): Void +actualizarUltimaVisita(): Void +ultimaVisita(): Time +filtrarHTML(texto:String): String +sustituirBBCodes(texto:String): String</pre>	<pre>+idTema: Int +idUsuarioCreador: Int +idUsuarioUltimo: Int +fechaCreacionTema: Time +fechaUltimoMensaje: Time +titulo: String +estado: enum->(Visible, Moderado, Cerrado)</pre>
	Mensaje <i>Estructura</i>
	<pre>+idMensaje: Int +idTema: Int +idUsuario: Int +texto: String +fechaDePosteo: Time +estado: Enum->(Visible, Moderado)</pre>

La clase Foro engloba todos los métodos empleados por las vistas de foro y mensajes. Antes de continuar hay que indicar que la vista que llama a éstas funciones conocen al usuario que navega por la web y por lo tanto no permitirá que un usuario sin privilegios ejecute métodos fuera de su alcance. Por ejemplo, un usuario sin autentificar no podrá insertar un mensaje nuevo y un usuario normal no podrá ejercer de moderador.

Además los métodos del controlador pueden conocer accediendo a variables de la vista que se muestran de forma pública el número de identificación del usuario, lo que permite identificar la persona con cuya cuenta crea nuevos mensajes y temas.

El método `actualizarUltimaVisita` se ejecuta cada vez que un usuario pide al servidor la página principal del foro. De este modo la vista puede comprobar, llamando al método `ultimaVisita` qué temas tienen mensajes que el usuario todavía no ha leído.

Como los listados se han de dividir en páginas con diez resultados en cada una tendremos que poder contar el total de resultados. Los métodos `numTemas` y `numMensajes` realizan esta labor, con la salvedad que si a `numMensajes` le indicamos un valor `idTema` inexistente obtendremos un valor NULL.

De forma análoga se comportan `buscarTemaPorId` y `buscarMensajePorId`. Si el identificador no se encuentra en la base de datos, en lugar de la estructura se devuelve un valor NULL. Los identificadores se añaden a la base de datos de forma correlativa, de forma que mediante un bucle se puede extraer toda la información necesaria.

Para insertar un nuevo tema o un nuevo mensaje se usan los métodos correspondientes `nuevoTema` y `nuevoMensaje`, que la vista solo permitirá usar si se tienen los permisos necesarios. El método `nuevoMensaje` puede fallar por los siguientes motivos: insertar un mensaje en un tema cerrado o moderado o hacerlo con un `idTema` inexistente. Como la vista controla estos casos y el error solo se puede dar si un usuario hostil modifica las cabeceras HTTP, no es necesario mostrar ningún error en dicho caso.

El método `moderarTema` y `moderarMensaje` funcionan de forma muy similar. Se indica el elemento que queremos moderar mediante su `id`, e indicamos el estado que queremos que tenga. Además, opcionalmente se le puede añadir un tercer parámetro para sobrescribir el mensaje. Esto es especialmente útil por parte del moderador para corregir faltas de ortografía o moderar partes de

un mensaje sin necesidad de ocultarlo en su totalidad. Se puede sobrescribir cualquier parámetro de los mensajes y temas excepto su identificador.

La clase también contiene métodos usados para filtrar los mensajes evitando que alguien escriba etiquetas HTML o convertir los BBCode en HTML válido.

3.2.2.4. Autores

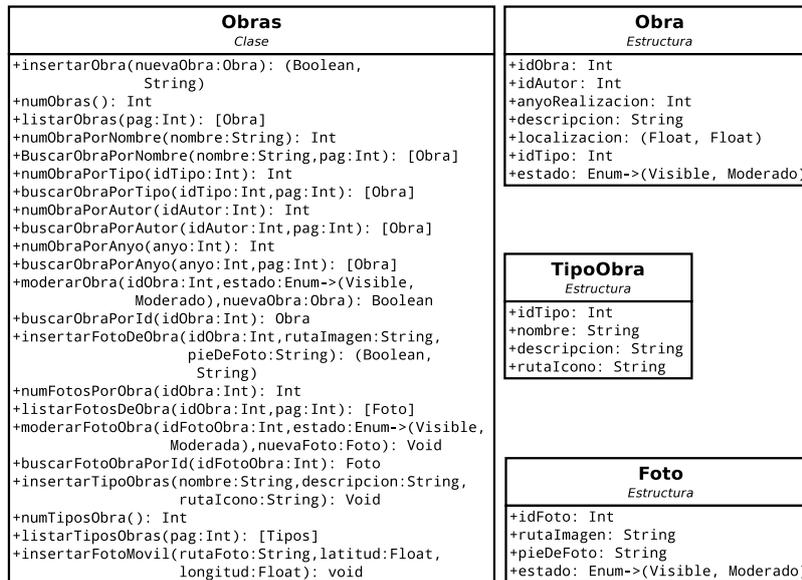
Autores <i>Clase</i>	Autor <i>Estructura</i>		
<pre> +insertarAutor(nuevoAutor: Autor): (Boolean, String) +numAutores(): Int +listarAutores(pag: Int): [Autor] +numAutorPorNombre(nombre: String): Int +buscarAutorPorNombre(nombre: String, pag: Int): [Autor] +numAutorPorPais(idPais: Int): Int +buscarAutorPorPais(idPais: Int, pag: Int): [Autor] +numAutorVivoEnAnyo(anyo: Int): Int +buscarAutorVivoEnAnyo(anyo: Int, pag: Int): [Autor] +moderarAutor(idAutor: Int, estado: Enum->(Visible, Moderado), nuevoAutor: Autor): Boolean +buscarAutorPorId(idAutor: Int): Autor +insertarFotoDeAutor(idAutor: Int, rutaImagen: String, pieDeFoto: String): (Boolean, String) +numFotosPorAutor(idAutor: Int): Int +listarFotosDeAutor(idAutor: Int, pag: Int): [Foto] +moderarFotoAutor(idFotoAutor: Int, estado: Enum->(Visible, Moderado), nuevaFoto: Foto): Void +buscarFotoAutorPorId(idFotoAutor: Int): Foto +insertarPais(nombre: String, codigo: String): void +numPaises(): Int +listarPaises(pag: Int): [Pais] </pre>	<pre> +idAutor: Int +nombre: String +anyoNacimiento: Int +anyoMuerte: Int +idPais: Int +biografia: String +estado: Enum->(Visible, Moderado) </pre> <table border="1" data-bbox="904 819 1050 927"> <thead> <tr> <th data-bbox="904 819 1050 866">Pais <i>Estructura</i></th> </tr> </thead> <tbody> <tr> <td data-bbox="904 866 1050 927"> <pre> +idPais: Int +nombre: String +codigo: String </pre> </td> </tr> </tbody> </table> <pre> +idFoto: Int +rutaImagen: String +pieDeFoto: String +estado: Enum->(Visible, Moderado) </pre>	Pais <i>Estructura</i>	<pre> +idPais: Int +nombre: String +codigo: String </pre>
Pais <i>Estructura</i>			
<pre> +idPais: Int +nombre: String +codigo: String </pre>			

La clase Autores además de implementar los métodos que se utilizan para listar, insertar y modificar autores, también lo hacen para los países y las fotos de los autores. Como en los demás casos, los métodos para insertar y modificar información requieren que en la vista correspondiente esté identificado un usuario registrado o un administrador respectivamente.

Para insertar un autor es necesario pasar como parámetro una estructura Autor al método insertarAutor con todos los datos correctos, excepto el identificador, cuyo valor se establecerá al realizar la inserción y el estado, que siempre es *Visible* al insertar un nuevo elemento. El método devolverá TRUE si se inserta con éxito. La cadena de texto contendrá los posibles errores, como un año de nacimiento posterior al de muerte o código de país inexistente.

Todos los demás métodos funcionan de manera análoga a los ya mostrados.

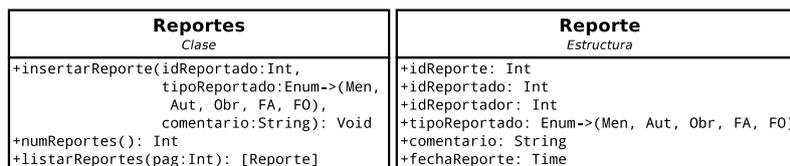
3.2.2.5. Obras



La clase Obras es muy similar a la clase Autores. En ella, además de los métodos para insertar, listar y moderar las obras, tenemos métodos para listar e insertar los tipos de obras y fotos de obras.

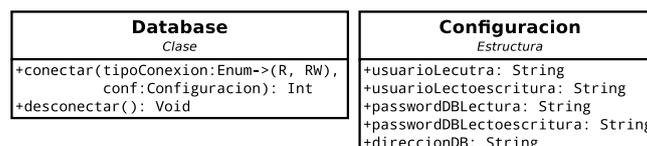
La novedad está en el método insertarFotoMovil, a la que se accederá mediante la aplicación móvil y guardará en la base de datos las fotos junto a su ubicación.

3.2.2.6. Reportes



La clase Reportes se usa en cualquier vista que contenga elementos introducidos por el usuario. Cualquier usuario registrado podrá insertar un reporte y mediante las vistas del administrador se podrán listar y actuar en consecuencia con los métodos de moderación.

3.2.2.7. Conexión a la base de datos



Esta clase se instancia en todas las clases anteriores para tener acceso a la base de datos. El valor devuelto se guarda en una variable de instancia privada que es usada por todos los métodos que requieren acceder a la base de datos. Cuando la clase se destruye automáticamente el destructor ejecuta el método desconectar, liberando la conexión.

3.3. Tecnología empleada

Una vez propuesta la estructura de la aplicación, vamos a ver cuales son las tecnologías necesarias para poder implementar las funcionalidades.

3.3.1. Lenguajes de programación

Después de discutir en la sección 2.1 las ventajas e inconvenientes de los lenguajes de programación de webs generadas dinámicamente, hemos decidido programar la web en PHP usando como base de datos MySQL. Es uno de los lenguajes más usados y el SGBD MySQL es robusto, libre, en continua evolución y es la base de datos por defecto de PHP. Además es el lenguaje con una mayor cantidad de recursos de ayuda y aprendizaje disponibles. Sin embargo no es fácil encontrar dominios de bajo coste que permitan el uso de PHP, MySQL y el envío de mail. Hay varias opciones disponibles:

- Contratar un servidor de pago.
- Usar un servidor gratuito, con la consecuencia de que automáticamente se incluirá publicidad en la web y que los límites de transferencia serán muy bajos.
- Usar la tecnología JSP y JDO en vez de PHP y albergar la página en Google App Engine.

Finalmente, la opción más práctica es la de instalar el servidor en nuestro sistema. De este modo tendremos un control total sobre el sistema, incluyendo el uso de disco duro, creación y restauración de copias de seguridad o la posibilidad de enviar correo. Sin embargo tendremos que cargar con el consumo del equipo informático y estaremos limitados por el ancho de banda de nuestro proveedor de Internet.

3.3.2. Dispositivos móviles

Para implementar la aplicación móvil hemos elegido la plataforma Android por varios motivos:

- En primer lugar porque se emplean tecnologías en las que ya teníamos cierto contacto con anterioridad, si bien hasta entonces eran aplicaciones muy simples.

- Es el único tipo de dispositivo que disponemos tanto para probar la aplicación como para establecer un grupo cerrado de usuarios que pruebe una beta de la aplicación. En caso de desarrollarse bajo Windows Phone o iPhone no sería posible.
- Y si decidimos publicar nuestra aplicación en el market no deberá pasar por las diferentes fases de comprobación y control de calidad que sí deben afrontar las aplicaciones de iPhone y Windows Phone.

Además, tal y como muestra la figura 2.3 si hacemos la aplicación para Android en una versión compatible lo suficientemente madura⁴, nos aseguramos la compatibilidad con más de la mitad de los dispositivos móviles.

3.3.3. Envío de datos mediante XML

Para dar formato y estructurar la información que se plasmará en los mapas de la web y de la aplicación móvil usaremos el formato XML. Esto nos da muchas ventajas con respecto a usar otras estructuras, como un texto o un archivo binario. La primera de ellas es que es un tipo de archivo fácil de crear al extraer los datos de la web. Y la segunda ventaja consiste en que tanto los navegadores como los dispositivos móviles traen de serie parsers para este tipo de ficheros.

Para ello enviaremos los iconos, mensajes y datos que se tienen que mostrar en los mapas siguiendo los dos esquemas siguientes. El primero envía la información de las obras registradas en la base de datos de la web.

```
<?xml version="1.0" encoding="UTF-8" ?>
<marcadores>
  <marcador lat="lat" lon="lon" icono="rutaIcono">
    <autor nombre="Nombre" id="id" />
    <obra nombre="Nombre" id="id" />
  </marcador>
</marcadores>
```

Este en cambio mostrará las fotografías aportadas por los usuarios de la aplicación móvil:

```
<?xml version="1.0" encoding="UTF-8" ?>
<marcadoresFotos>
  <marcadorFoto lat="lat" lon="lon" foto="rutaFoto" />
</marcadores>
```

⁴La fragmentación de mercado de Android es uno de sus principales inconvenientes a la hora de desarrollar aplicaciones, puesto que no son retrocompatibles. Por ejemplo, si empleamos elementos de la versión 2.2 en nuestra aplicación (como las notificaciones push) los usuarios de Android 1.6 no podrán instalarla. Por ello tenemos que declarar nuestra aplicación compatible con la menor versión de Android posible.

3.3.4. HTML 5

De cara a experimentar con las nuevas características que ofrece HTML5 en los formularios hemos decidido utilizarlos de forma exclusiva en el diseño de ciertas funciones del portal web. Debido al carácter experimental y académico del proyecto no vamos a preocuparnos por la posibilidad de que el usuario acceda al sitio con una versión no actualizada de su navegador de internet. En cualquier caso las funcionalidades que ofrecen los nuevos elementos HTML no se han suplido mediante JavaScript para aumentar la compatibilidad, lo cual sería impensable en un sitio web comercial.

En el cuadro 3.1 se muestran algunos de los elementos de HTML5 usados.

Elemento	Descripción	Disponibilidad
placeholder	Muestra una identificación en los campos de texto cuando la caja se encuentra vacía y no tiene el foco.	OK: IE, FF, Safari, Chrome, Opera
datalist	Es una colección de cadenas de texto que sirven para ayudar al usuario al introducir datos, pues se muestran como opciones de autocompletado del campo.	OK: IE, Opera, Chrome Aprox: FF Fail: Safari
email	Se muestra como un campo de texto normal, pero al pulsar sobre el botón de envío solo se validará si contiene una dirección de correo electrónico válida. El navegador sólo comprueba que en el campo haya una dirección de correo con formato correcto, pero no comprobará que exista.	OK: IE, FF, Chrome, Opera Fail: Safari
number	Se muestra un campo de texto acompañado de dos botones para aumentar o disminuir el número introducido. Se le puede indicar un valor máximo y mínimo para validar el dato y un intervalo <i>-step-</i> que se sumará o restará al valor mostrado al pulsar los botones.	OK: Safari, Chrome, Opera Aprox: IE Fail: FF
required	El formulario solo se podrá enviar si todos los campos marcados como requeridos tienen un dato válido en ellos.	OK: IE, FF, Chrome, Opera Fail: Safari
<header> <footer>	Los tags <i>header</i> y <i>footer</i> se comportan como un <i>div</i> normal, pero es mucho más expresivo de cara a diseñar los CSS de la web, o para ser leída por un sistema de voz sintetizada.	OK: IE, FF, Chrome, Opera, Safari

Cuadro 3.1: Elementos de HTML5 usados en el portal web

Capítulo 4

Implementación

En el siguiente capítulo mostraremos los pasos que hemos seguido para codificar todos los requisitos que hemos especificado en el capítulo anterior, haciendo especial hincapié en los pasos más complicados o problemáticos, señalando y justificando las soluciones que hemos empleado.

4.1. Puesta a punto

El primer paso para el desarrollo de un portal web en PHP y MySQL ha sido instalar el servidor Apache, PHP y MySQL en el ordenador. Gracias al comando `tasksel` disponible en Ubuntu es tan sencillo como ejecutarlo y seleccionar “Servidor LAMP”. También es muy útil instalar PHPMyAdmin para crear, consultar, hacer pruebas y copias de seguridad de la base de datos desde un entorno gráfico rápido y sencillo de usar.

Para escribir código en PHP he decidido usar Eclipse. Para ello, además de instalar el programa se debe instalar el plugin con las funcionalidades de detección de código PHP, HTML y javascript.

Para la aplicación Android deberemos bajar de la web de desarrolladores de google los programas auxiliares de Android y el plugin de Android para Eclipse.

Además como necesitaremos usar el componente `MapView` en la sección de visualización de obras, tendremos que autenticar la firma digital de nuestra aplicación en una página de Google dispuesta para dicho fin y conseguiremos de este modo un código de activación gratuito.

Por último, para el testeado de la aplicación móvil disponemos de dos opciones:

- Si usamos nuestro propio móvil deberemos activar la opción de depuración en el menú de configuración. El sistema operativo Ubuntu no reconocerá el móvil al conectarlo y tendremos que crear una regla en el sistema `udev` para que sea detectado. También fallarán los permisos necesarios para acceder al móvil en el caso que el programa ADB no

tenga los permisos de administrador. La opción más práctica es cambiar el propietario del programa a root y activar el bit setuid, aunque esto puede ser una fuente de problemas de seguridad, así que no es recomendable hacerlo en ordenadores de producción.

- También podremos usar el emulador de Android disponible en el paquete de software de Google. Al contrario que el emulador de iPhone, este programa no solo emula el software del teléfono sino su arquitectura hardware entera con el emulador QEMU. Esto provoca que su ejecución sea muy lenta, tanto al ejecutar los programas como al arrancar el emulador. Además su última versión tiene un bug en el programa que provoca que al emular la cámara aleatoriamente puede dejar de responder. Motivos por los cuales es preferible usar un móvil para depurar el código.

También es muy interesante modificar el archivo `php.ini` y activar la opción para mostrar los errores. Esto es muy útil en la fase de desarrollo ya que si no, si el código PHP contiene fallos simplemente la carga de la página web quedará interrumpida en ese punto. Tras el cambio se indicará el archivo y número de línea de los errores, el motivo del error y warnings de condiciones sospechosas de generar errores durante la ejecución. Naturalmente en fase de producción tendremos que volver a modificar el archivo y desactivar la opción, pues en caso de fallo de la página es importante no dar demasiadas pistas de la estructura interna de la web al público, pues podría dejar en evidencia algún punto débil de la misma y usar dicha información para lanzar una ataque contra ella.

4.2. Esquema relacional

Hemos creado una base de datos en MySQL implementando el modelo Entidad-Relación que hemos descrito en la figura 3.4.

Todas las relaciones establecidas en el esquema tienen garantizadas la integridad referencial ya que han sido creadas usando claves ajenas. Para ello hay que seguir los siguientes pasos, poniendo de ejemplo la relación entre país y autor:

- En primer lugar las claves ajenas han de estar declaradas como clave primaria.

```
CREATE TABLE IF NOT EXISTS Países (  
    Id int(10) unsigned NOT NULL AUTO_INCREMENT,  
    Nombre varchar(50) NOT NULL,  
   Codigo varchar(5) NOT NULL,  
    PRIMARY KEY (Id),  
    UNIQUE KEY Codigo (Codigo)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

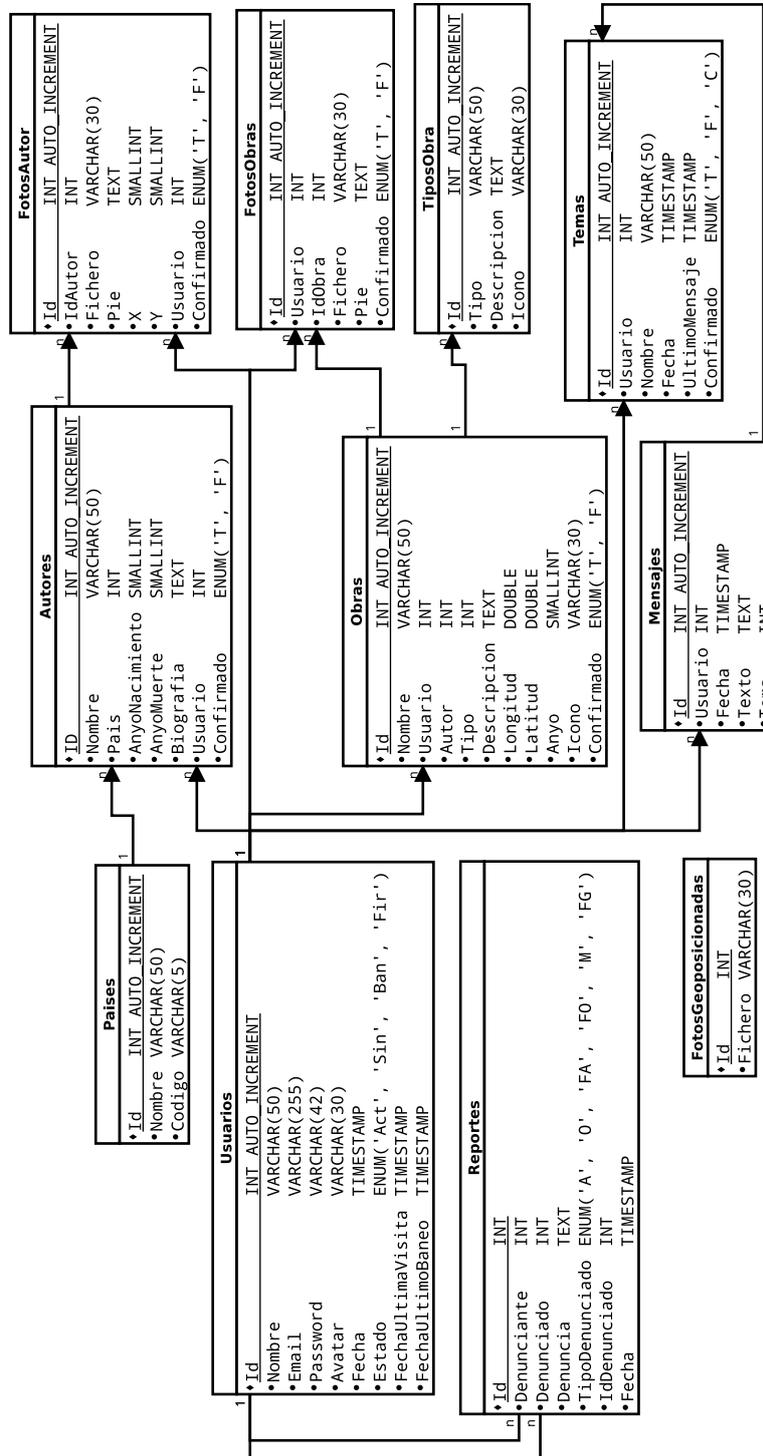


Figura 4.1: Modelo relacional de la base de datos

- Las claves ajenas han de estar declaradas como *Key*.

```
CREATE TABLE IF NOT EXISTS Autores (  
  Id int(10) unsigned NOT NULL AUTO_INCREMENT,  
  Nombre varchar(50) NOT NULL,  
  Pais int(10) unsigned NOT NULL,  
  AnyoNacimiento smallint(6) NOT NULL,  
  ...  
  PRIMARY KEY (Id),  
  UNIQUE KEY Nombre (Nombre),  
  KEY Pais (Pais)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

- Y finalmente añadimos la restricción de integridad.

```
ALTER TABLE Autores  
  ADD CONSTRAINT Autores_Paises  
  FOREIGN KEY (Pais) REFERENCES Paises (Id)  
  ON DELETE NO ACTION ON UPDATE CASCADE;
```

Tal y como están definidas, las restricciones hacen que sea imposible borrar datos de la web sin borrar todos los datos a los que se hacen referencia. De este modo si quisiésemos borrar un tema del foro deberíamos borrar todos sus mensajes. O peor aún: si queremos borrar a un usuario...

1. Tendremos que borrar todos los reportes, autores, temas, mensajes, obras o fotos que haya subido.
2. Si borramos los temas que el usuario haya iniciado también tendremos que borrar todos los mensajes de dichos temas, sean o no de dicho usuario.
3. Si borramos los autores que ha subido el usuario, deberemos borrar todas las obras de dicho autor, que pueden no haber sido subidas por el usuario que estamos borrando.
4. Y de forma análoga con las fotos de autores y las fotos de obras.

Es por ello que se ha decidido restringir el borrado. En cualquier caso el administrador puede cambiar los datos u ocultar la información que no se desee mostrar a los usuarios de la web.

Hay que tener en cuenta que estas restricciones de integridad solo se aplicarán si establecemos que las tablas se van a tratar con el motor InnoDB, que es el predeterminado de MySQL desde su versión 4.0 sustituyendo al motor MyISAM. Como podemos observar en el ejemplo anterior, las hemos definido así. El uso de InnoDB, además de ofrecernos la posibilidad de usar integridad referencial en nuestro modelo, es más robusto ya que se recupera más rápidamente tras un fallo y además controla sus propios accesos a disco y cachés en vez de delegar estas tareas al sistema operativo.

Sin embargo MyISAM no tiene la necesidad de acceder al disco duro por cada transacción –lo que en la práctica es un cuello de botella–, tiene un eficaz algoritmo de compresión de los datos de las tablas y no necesita ordenar los datos por su clave primaria, lo que le ha creado la fama de ser un motor más rápido.

En el esquema relacional de la figura 4.1 podemos observar variables que no son fácilmente identificables:

- La enumeración *Confirmado* es la que indica si el elemento se mostrará a los usuarios. Su valor lo cambia el administrador desde las diferentes secciones de configuración.
- La enumeración *Confirmado* en la tabla de temas tiene un estado adicional 'C', que indica que el tema está cerrado y no admitirá más mensajes.
- La enumeración *TipoDenunciado* de la tabla de reportes indica cual es el tipo de elemento que un usuario ha denunciado, ya sea un autor, una obra, la foto de un autor, la foto de una obra, un mensaje o una foto geoposicionada.
- El valor *IdDenunciado* de la tabla de reportes debería estar relacionado con el identificador del elemento denunciado en cualquiera de las tablas a las que hace referencia *TipoDenunciado*. Sin embargo esto no es posible ya que no podemos establecer este tipo de relación condicional entre varias tablas, de modo que deberemos cerciorarnos del buen funcionamiento del controlador que inserte los datos de reporte porque el SGBD no nos puede garantizar la coherencia de este dato¹.

4.3. El envío de contraseñas a través de la red

Como se comentó en la sección 3.1.2, el método que usemos para el envío de contraseñas y su almacenamiento en la base de datos tiene que ser fiable puesto que cualquier tipo de fallo puede suponer un grave riesgo que comprometerá la seguridad del sitio. En un primer momento se han planteado varias opciones.

4.3.1. Uso de HTTPS

Si configuramos nuestro servidor web para que acepte conexiones HTTPS, todo el tráfico entre los clientes y nuestro servidor se transferiría cifrado mediante un mecanismo de clave asimétrica, al menos durante el proceso de identificación. Si la clave es lo suficientemente robusta podremos garantizar la confidencialidad de las contraseñas. Sin embargo tenemos varias limitaciones:

¹Tenemos otra opción que es crear en la tabla un valor distinto para cada posible tipo de denuncia, sin embargo el SGBD sigue sin podernos garantizar que el tipo de denuncia coincida con el campo empleado sin escribir el código necesario en el controlador. Además es una opción que complica la tabla y aumenta el tamaño necesario para su almacenamiento.

- La generación de la firma de nuestra clave pública la realizan empresas que cobran una importante cantidad de dinero por dichos servicios², y el coste aumenta a medida que lo hace la robustez de la clave.
- Dichas empresas tienen un protocolo estricto de validación de peticiones. Requerirán información sobre nuestra organización y nuestros datos, por lo que no es un servicio orientado a proyectos académicos.
- El cifrado y descifrado que tiene lugar en el servidor necesita una cantidad considerable de recursos que pueden ser necesarios si hay una sobrecarga de peticiones, provocando una respuesta lenta.

4.3.2. Mecanismo de desafío-respuesta

Este método se puede implementar de varias formas, pero el más práctico sería el que a continuación se muestra:

- El servidor envía al cliente una cadena de texto aleatoria, el desafío.
- El cliente, mediante javascript o alguna otra técnica similar calcula un *hash* de la contraseña del usuario.
- Con los dos datos (desafío y *hash* de la contraseña) el cliente realiza alguna operación, como por ejemplo el XOR de las dos cadenas y se las envía al servidor.
- El servidor, que conoce la contraseña del usuario –o su *hash*– y la cadena aleatoria que ha generado, puede determinar que la respuesta recibida se ha generado con una contraseña válida o no.

Aunque por la red en ningún momento se ha transmitido ni la contraseña ni una huella de ésta, no podemos considerarlo un método seguro puesto que cualquier atacante puede capturar tanto el desafío como la respuesta y realizar ejercicios de criptoanálisis para descifrar la clave. De hecho, si usamos una función tan trivial como ésta la clave se descubriría sin necesidad de interceptar más de un inicio de sesión del usuario.

4.3.3. Resumen de la contraseña

Este método consiste en enviar un *hash* de la contraseña en vez de la contraseña real para autenticarse en el servidor. Este método no certifica la seguridad de la página puesto que cualquier atacante puede capturar el resumen de la contraseña de un usuario e identificarse como él de forma fraudulenta en el servidor. Sin embargo debido a la unidireccionalidad de las funciones *hash*

²Siempre nos queda la opción de generar nuestras propias claves, generar nuestro propio certificado CA y autofirmarlas. El sistema tendrá la misma seguridad, pero antes de entrar a la web el navegador avisará al usuario de un posible ataque, porque antes deberán aceptar nuestra clave de CA y añadirla a su repositorio. Sin embargo, los certificados de CA de las principales entidades de certificación de pago vienen instaladas de serie en el sistema operativo y en los móviles.

sí la imposibilita de que nadie conozca la contraseña auténtica, ni siquiera el administrador de la web.

Si tenemos en cuenta que según algunos estudios el 75 % de los usuarios usan la misma contraseña para su correo electrónico y para las redes sociales, parece una medida de seguridad básica que se ha tomado en cuenta para realizar la identificación. Debido a que el servidor no guarda las contraseñas no podremos reenviar una contraseña olvidada a un usuario, debiendo establecer un protocolo de restablecimiento como indicamos en la sección 3.1.2.

4.4. Elección de un método de geolocalización en la aplicación móvil

En la sección 2.3 indicábamos las diferentes formas de geolocalizar un dispositivo móvil de última generación. Si tenemos en cuenta los inconvenientes de la detección mediante WIFI y GSM, vamos a implementar la detección mediante GPS. El primer paso será dar permisos al programa para que pueda usar el GPS, por lo que estableceremos en el fichero `AndroidManifest.xml`

```
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION"
/>
```

Dado que queremos que sea imprescindible que el móvil tenga localización por GPS para poder ejecutar el programa, debemos agregar al mismo fichero una línea que obligue a que los móviles en los que se ejecute cumplan ciertos requisitos. De hecho si el móvil no dispone de GPS no se mostrará en la tienda de aplicaciones de Google:

```
<uses-feature
    android:name="android.hardware.location.gps"
/>
```

Que demos permisos de a la aplicación para que pueda usar el GPS y que indiquemos la necesidad de que el móvil disponga de GPS en el momento de instalar la aplicación no nos asegura que el usuario del móvil lo tenga activado cuando arranquemos la aplicación. Para cerciorarnos debemos comprobar su disponibilidad al iniciar el programa y mostrar un mensaje de error en su caso. Podremos hacerlo del siguiente modo:

```
boolean isGPSEnabled(){
    LocationManager lm;
    lm = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);
    return lm.isProviderEnabled
        (LocationManager.GPS_PROVIDER);
}
```

Uno de los grandes problemas del GPS, aparte de su falta de cobertura en el interior de edificios, es el consumo de batería que genera cuando aplicaciones que hacen uso de él están funcionando. El SDK de Android nos permite especificar cada cuanto queremos obtener una nueva lectura de la posición. Cuando arranquemos la aplicación solo necesitaremos una lectura para poder geolocalizar la foto que nos permite seleccionar el menú de opciones. Sin embargo si decidimos mostrar el mapa geoposicionado damos por hecho de que nos estaremos moviendo con el móvil por la calle. Necesitaremos cambiar nuestra ubicación en el mapa a medida que nos movamos por la calle. Una buena relación entre ahorro de batería y precisión de la posición puede ser el comprobar nuestra ubicación cada minuto si nos hemos desplazado al menos 100 metros. Para ello usaríamos el método `requestLocationUpdates` del siguiente modo:

```
LocationManager lm;
lm = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
lm.requestLocationUpdates(
    LocationManager.GPS_PROVIDER,
    60000L, /* Tiempo de muestreo en ms */
    100.0f, /* Distancia en metros */
    this /* Clase que implementa LocationListener */
);
```

4.5. Elección de Google Maps como servidor de mapas

En la sección 2.4 mostrábamos las características de los servidores de mapas que ofrecen sus servicios por internet. Debido a su popularidad, fácil aprendizaje, disponibilidad en Android y a no ser necesario servir los datos mediante nuestros propios servidores, he decidido usar Google Maps para implementar las vistas de mapas.

Aunque en la versión tercera del API de Google Maps, a diferencia de la versión segunda no es necesario registrarse y conseguir una llave para acceder al servicio, sigue siendo recomendable para controlar mediante la consola de usos de Google el consumo de datos y el límite restante que permite seguir usando el servicio de forma gratuita. Para registrarse hay que acceder con tu cuenta de Google a la página <https://code.google.com/apis/console> y activar la API referida a Google Maps v3. A continuación acceder a la sección API Access y crear una nueva clave asignada a la dirección de Internet donde se acceda al mapa. En nuestro caso será `valengeoarte.no-ip.org`.

Cargar un mapa en la página web es muy sencillo. Para ello tendremos que crear un div en la página donde deseemos que se vea el mapa. Darle el tamaño adecuado y cargar el mapa mediante javascript.

1. Insertamos en la página web el div:

```
<div id="div_mapa" style="width:150px; height:300px;" />
```

2. Cargamos con Javascript el código que maneja el mapa:

```
<script type="text/javascript"
  src="https://maps.google.com/maps/api/js?
  sensor=false&key=NUESTRO_CÓDIGO">
</script>
```

3. Cargamos en el div el mapa y lo centramos en Valencia a un nivel de zoom adecuado para que se vea entera la ciudad. El código tendrá que ejecutarse cada vez que se inicie la página, así que lo ideal será establecerlo como onload en la etiqueta body de la página:

```
function alCargarWeb() {
  var myOptions = {
    zoom: 12,                /* Nivel de zoom adecuado */
    center: new google.maps.LatLng(39.46973, -0.37633),
                              /* Coordenadas de Valencia */
    mapTypeId: google.maps.MapTypeId.ROADMAP
  /* Visualizar el mapa como calles, no como ortofotos*/
  };
  var map = new google.maps.Map(
    document.getElementById("div_mapa"),
    myOptions);
}
```

4. Si necesitamos cargar más datos deberemos hacerlo mediante tecnología AJAX, por lo que mediante javascript se leerá un documento XML de forma asíncrona a la carga de la página y se cargarán los datos en el mapa.

4.5.1. El uso del geocodificador inverso de google

Uno de los requisitos a la hora de subir a la web fotos y obras geoposicionadas, como se ha descrito en la sección 3.1.5 es que estén en el término municipal de la ciudad de Valencia –dado que no entra en los objetivos de nuestra web catalogar fotos y obras de otros lugares–. Tanto si se cataloga una obra desde nuestra web como si se sube una foto desde la aplicación móvil, Google Maps nos ofrece un geocodificador inverso donde indicando la latitud y la longitud nos devuelve los datos referidos a dicha localización. Se puede ver un ejemplo programado en javascript en la figura 4.2.

Los datos que nos ofrece el servicio son la calle, el código postal, la ciudad, la provincia y el estado. Simplemente comprobando que la ciudad de la localización fuese Valencia tendríamos solucionado el problema. Sin embargo el geolocalizador siempre intenta de forma aproximada buscar una calle o carretera próxima, fallando en caso contrario. Así si seleccionamos un punto sin urbanizar del término municipal de Valencia cercano a una carretera de



Figura 4.2: Ejemplo de geolocalización inversa con Google Maps

otro municipio, el geocodificador fallará. Además tiene un rango de búsqueda, por lo que también fallará si seleccionamos un espacio libre grande, como puede ser un punto en mitad de la Albufera.

Por todos los problemas detectados, además del tiempo adicional necesario para esperar el resultado y la complejidad resultante, se ha empleado finalmente un sistema más simple aunque rudimentario. Se han fijado límites a los cuales se podrá agregar información geográfica a la web. Para ello hemos decidido elegir dos meridianos y dos paralelos terrestres que engloben todas las pedanías valencianas entre los cuales tendrán que posicionarse los elementos.

- Al norte, el paralelo 39,5627, coincidente con el punto más septentrional de la pedanía de Massarrojos.
- Al sur, el paralelo 39,278, en el punto más meridional de la pedanía de El Perellonet.
- Al este, el meridiano -0,2867, en el punto más oriental de la pedanía de Rafalell.
- Al oeste, el meridiano -0,4326, tangente al punto más al occidental de la pedanía de Massarrojos.

4.6. Filtrado de elementos antes de usarlos en la base de datos

Una de las forma más fáciles de comprometer la seguridad de una web es la inyección SQL. En ella se intenta comprometer la integridad de nuestra base de datos usando los propios métodos de acceso que hemos generado. Por ejemplo, si un usuario intenta loguearse en nuestra web, usará la vista de iniciar sesión e introducirá su usuario y contraseña.

Las credenciales llegarán al controlador, y usando el método *login* mostrado en la sección 3.2.2.1 se ejecutará una orden tal que así:

```
login('mail@Usermalvado', 'PassSHA1Coded');
```

Dentro del código de ese método, las credenciales se comprueban en la base de datos mediante órdenes SQL de la siguiente forma:

```
SELECT Id FROM Usuarios WHERE Mail='mail@Usermalvado'  
AND Pass='PassSHA1Coded';
```

En el caso de coincidir el usuario y la contraseña se devuelve el número de usuario. Sin embargo, imaginemos que el atacante, con conocimientos suficientes modifica el POST HTTP y envía como contraseña la cadena '||1=1. El código SQL resultante es:

```
SELECT Id FROM Usuarios WHERE Mail='mail@Usermalvado'  
AND Pass='' || 1 = '1';
```

Cuyo resultado es siempre correcto para cualquier usuario registrado de la web. Una forma práctica de resolverlo es *escapando* los caracteres que pueden dar problemas, como las comillas. De esta forma MySQL en vez de detectar un símbolo del lenguaje detectará un elemento más en una cadena de texto. Usando para ello la función de PHP `real_escape_string` el código a ejecutar sería:

```
SELECT Id FROM Usuarios WHERE Mail='mail@Usermalvado'  
AND Pass='\'' || 1 = \'1';
```

Que evidentemente fallaría.

Otro método algo menos peligroso pero igualmente molesto es la inyección de código HTML. Imaginemos que en una descripción de una obra alguien usa etiquetas de HTML como `<h1>` o `<marquee>`. Al listar el contenido mediante una búsqueda, saldría el texto con un formato desagradable, en vez de usar los BBcodes que nosotros permitimos.

Para eliminar este riesgo deberemos filtrar las cadenas de texto antes de almacenarlos en la base de datos, y sustituir los símbolos que abren y cierran las etiquetas HTML. “<” se sustituirá como “<” y “>” como “>”. Despojando de su sentido semántico a estos dos caracteres cualquier intento de este tipo de inyección resultará infructuosa.

4.7. Compilación final de la aplicación móvil y posible subida a Google Play

Los dispositivos Android necesitan que todas las aplicaciones instaladas hayan sido firmadas digitalmente con un certificado cuya clave privada la tenga

el desarrollador. De esta manera se puede conocer el autor de forma segura, y establecer permisos de confianza entre las aplicaciones. Conviene aclarar que las aplicaciones las firma el propio desarrollador con un certificado autofirmado. No es necesaria la intervención de ninguna entidad de certificación.

Antes de firmar la aplicación hay que comprobar que esté disponible la aplicación Keytool ya que es la herramienta necesaria para crear nuestro par de claves.

1. Obtendremos nuestra clave privada autofirmada. Debe tener una validez posterior al 22 de Octubre de 2033. Un ejemplo sería:

```
$ keytool -genkey -v -keystore valengeoarte.keystore  
-alias valengeoarte -keyalg RSA -keysize 2048  
-validity 10000
```

2. Con la clave generada volvemos a pedir una nueva llave de activación de Google Maps y la añadiremos al proyecto en el layout del MapActivity.
3. Compilamos, firmamos y alineamos el fichero APK con la opción de exportar de Eclipse, en File > Export.

Para subir el APK resultante a Google Play –el market de Android– deberemos pagar 25\$ para abrirnos una cuenta en el sitio. Deberemos aportar el fichero APK, dos capturas de pantalla de la aplicación, un icono de alta resolución, una lista con los lenguajes ofrecidos por la aplicación, un título y una descripción breve.

Además tendremos que asignarle una categoría, que en el caso de nuestra aplicación, la más conveniente sería Travel & local.

Capítulo 5

Pruebas y evidencias de ejecución

Antes de poner a disposición de los usuarios el portal web y la aplicación móvil hace falta asegurarse que las dos funcionan de forma correcta y proporcionarán una experiencia de usuario satisfactoria.

Para ello es necesario establecer unos criterios claros y estructurar los contenidos de la web y la aplicación de acuerdo a dichos objetivos. Si se han empleado mecanismos y herramientas adecuadas, tras su diseño e implementación deberíamos conseguir los resultados esperados.

5.1. Validación y testeo

Para comprobar que nuestro código genera páginas web estándar que podrán visualizarse sin problemas en cualquier navegador moderno deberemos usar herramientas adecuadas. Además se ha de comprobar que el código cumple las expectativas que hemos expresado al redactar los requisitos. Hemos seguido los siguientes pasos:

- En primer lugar se ha validado el código HTML y CSS de la web mediante los validadores oficiales del W3C, en la dirección validator.w3.org y jigsaw.w3.org/css-validator respectivamente. Con esto nos aseguramos que mientras el navegador cumpla los estándares, la página debería visualizarse de la misma forma en cualquier navegador. Caso aparte es si el navegador es compatible con los elementos propios de HTML5 que la página usa en los formularios.
- El código PHP se ha probado activando el nivel más alto de detección de warnings y comprobando que todas las condiciones que se pueden dar en la web funcionan sin ningún error o advertencia.
- Más adelante se ha permitido el acceso a la web a un grupo de personas de confianza en una suerte de versión beta donde se han detectado la

gran mayoría de fallos y se han propuesto mejoras, aunque solo se han ejecutados las que únicamente implicasen cambiar elementos de la vista. En ningún caso se han modificado o añadido requisitos o elementos de diseño.

En el caso de la aplicación móvil el testeo ha consistido en instalar el programa en varios *smartphones*¹ de distintas capacidades y comprobar si la herramienta se comporta y muestra de la forma correcta. Se ha tomado en consideración los siguientes casos:

- Que la aplicación se cargue correctamente incluso en el móvil con menor cantidad de memoria disponible para la ejecución de programas.
- La aplicación se visualice correctamente en los diferentes tipos de pantalla y densidad de píxeles.
- Se detecta la señal del GPS de forma correcta y se desactiva cuando la aplicación pasa a segundo plano.
- Comprueba al inicio de forma correcta su conectividad a internet.

Debido a que todos los *smartphones* con los que se ha dispuesto para las pruebas disponían de cámara de fotos y GPS, no se ha podido comprobar si los premisos de la aplicación impiden su instalación en ellos.

5.2. Evidencias de ejecución

En esta sección insertaremos las capturas de pantalla necesarias para mostrar la apariencia y el comportamiento que ofrece nuestra web. Para ello seguiremos una pequeña interacción de un nuevo usuario con el sistema.

También mostraremos el uso de nuestra aplicación móvil a la hora de realizar fotos y mostrar las obras.

5.2.1. Portal web

Página principal

El usuario, tras entrar a la web, verá la página principal mostrada en la figura 5.1. Todas las vistas están divididas en cabecera, cuerpo y pie.

- En la **cabecera** se muestra el icono y el título de aplicación, el bloque de identificación del usuario y el menú, adaptado a los permisos del usuario actual. Como en este caso la web la visita un usuario anónimo, solo se muestran las opciones para listar datos.

¹Los móviles con los que se ha probado la aplicación son los cuatro siguientes: Huawei U8650, Samsung Galaxy SII, Samsung Galaxy S y ZTE Skate.



Figura 5.1: Página principal de la web

- El **cuerpo** de la página principal muestra un mapa, centrado en la ciudad, con todas las obras y fotos efectuadas con la aplicación móvil que se encuentran registradas en nuestra base de datos. A continuación aparece una leyenda que se irá actualizando a medida que el administrador inserte más categorías.
- El **pie** de la página muestra el nombre del autor, el año y enlaces tanto a la UPV como al DISCA.

En la figura 5.2 podemos ver la información que se muestra al pulsar sobre el mapa en cualquier obra o fotografía. Podemos ver todas las características de las obras pulsando los enlaces.

Registro e inicio de sesión

El usuario que se quiera registrar pulsará el enlace *Regístrate* e introducirá sus datos en los campos de texto que aparecen en la figura 5.3. Un placeholder como los que se pueden observar en el cuadro de identificación nos ayudarán a introducir los datos. Tras pulsar el botón de enviar nos llegará un correo con el siguiente texto:

Hola AlbertoDominguez, gracias por darte de alta en Valen-geoarte. Te recordamos que para ingresar en la web debes poner la contraseña IVDmEEx y el mail aldolo@mail.com



Figura 5.2: Información contextual en el mapa

Figura 5.3: Pantalla de registro

La cuenta se activará cuando entremos por primera vez con nuestras credenciales. Usaremos el cuadro de identificación de la cabecera.

Configuración de la cuenta

Cuando entramos a nuestra cuenta podemos ver que en el menú han aparecido las opciones que se pueden ver en la figura 5.4. Lo primero que se debería hacer es cambiar la contraseña y subir un avatar para mostrar en el foro. Para ello pulsamos en la opción de *configuración*, donde podemos realizar las dos acciones.

Para el cambio de contraseña se ha de poner la contraseña actual y la nueva contraseña dos veces. Mediante Javascript se comprobará a medida que se escribe que las contraseñas coincidan, mostrando un aspa roja cuando sean distintas o una marca verde cuando coincidan.

Para el cambio de avatar se seleccionará una imagen en JPG, PNG o GIF en tu ordenador que se subirá a la web. Hay que ser prudente porque el ratio de aspecto de la imagen puede cambiar, así que lo mejor es subir imágenes cuadradas. Cuando pulsemos el botón de enviar cambiará el avatar genérico por el que nosotros hayamos subido.

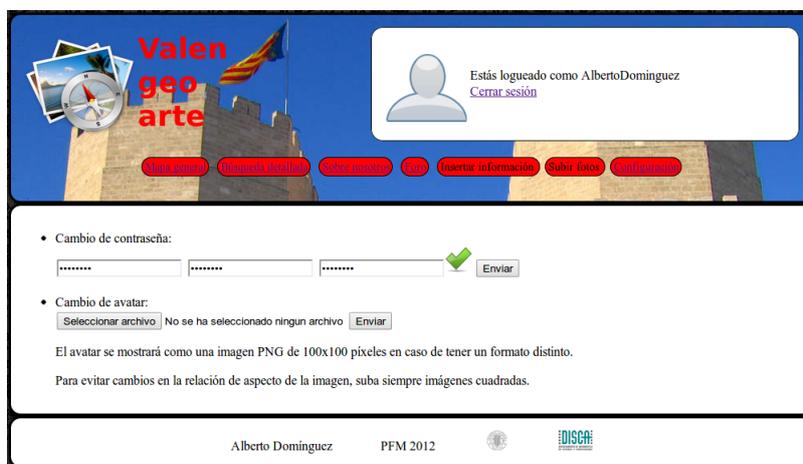


Figura 5.4: Configuración de la cuenta

Foro

Si estamos identificados podremos entrar al foro con los permisos suficientes para escribir mensajes nuevos y añadir temas. La primera imagen que veremos es la se muestra en la figura 5.5.

En ella podemos ver nuestro nuevo avatar en la caja de identificación y un enlace para cerrar nuestra sesión.

En el cuerpo de la página hay una lista de temas disponibles en el foro, ordenados por la fecha del último mensaje escrito. Para cada tema se muestra el autor del tema y del último mensaje junto a la fecha de creación y última aportación al mismo.

Los iconos que aparecen indican con un candado si el tema está cerrado o si tiene mensajes nuevos desde la última vez que entramos al foro con un icono rojo.

Si el foro tiene más de diez temas se mostrará una lista de páginas para acceder al tema deseado. De forma similar, en cada tema se puede acceder a una página concreta. (Ver la sección 3.2.2.3 para leer más sobre la paginación de los listados).

La parte inferior del cuerpo de la página contiene un formulario para poder iniciar un tema nuevo. En la figura hemos escrito un tema de ejemplo en el que mostramos el uso de los BBCode. Con ellos podemos modificar el formato del texto del mensaje. Los iconos son bastante descriptivos: se pueden poner textos en negrita, cursiva, subrayado, se pueden añadir imágenes, enlaces o citas de otros usuarios.

Reportes

Al rato de escribir nuestro mensaje otro usuario ha escrito una respuesta, que podemos ver en la figura 5.6. Podemos seguir añadiendo mensajes con el for-

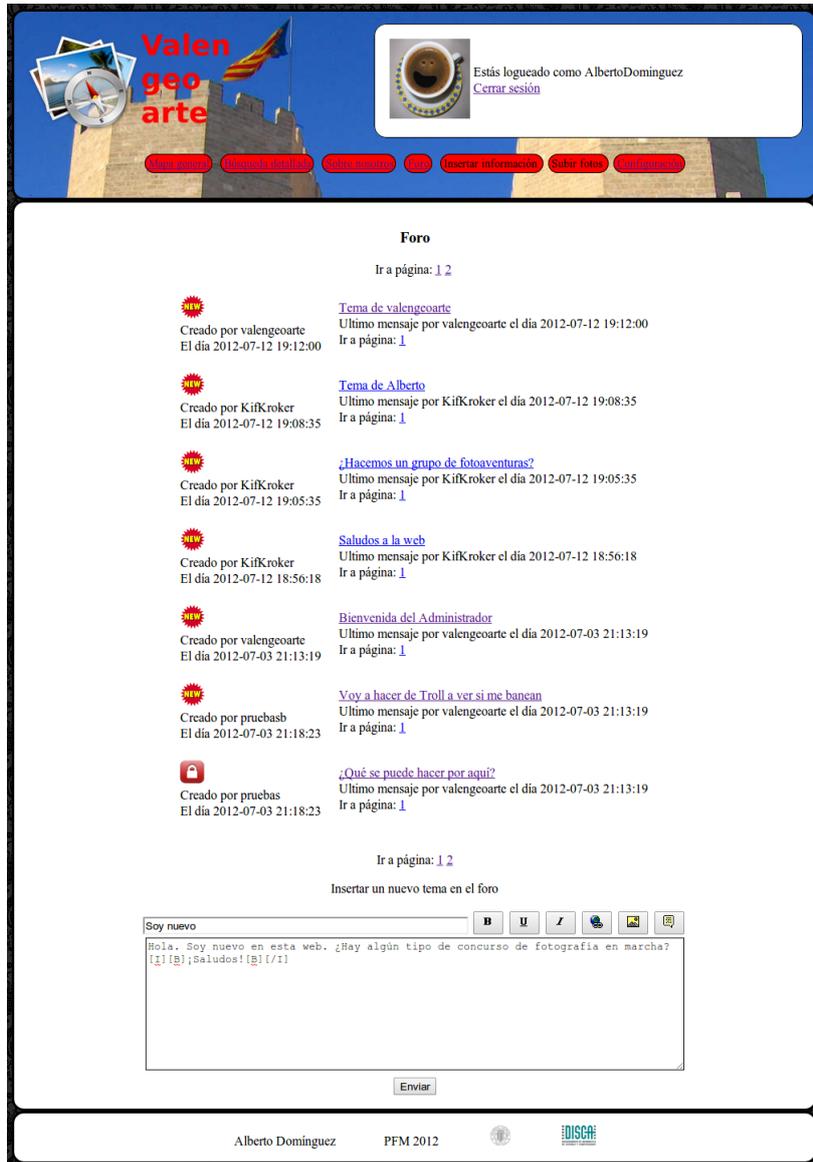


Figura 5.5: Listado de temas en el foro

mulario al final de la web.

Como podemos observar, debajo del avatar de cada usuario podemos ver un icono con un signo de admiración. Pulsándolo se abrirá una nueva página para reportar cualquier tipo de aportación inadecuada como veremos más adelante.

En la figura 5.7 vemos la página de reportes. En ella debemos escribir en el cuadro de texto los motivos por los que consideras que la aportación no es válida.

Añadir un nuevo autor

Como nuevos usuarios que somos, queremos añadir información a la web. Vamos a insertar a un nuevo autor. En la figura 5.8 podemos el cuerpo de la web con los datos que son necesarios.

Para introducir los años se puede usar el tipo de entrada *number*, de HTML5, que permite introducir una cifra positiva hasta el año actual. Como se puede apreciar, si seleccionamos la opción *desconocido*, o *vive*, el campo numérico queda desactivado. El texto biográfico se puede formatear con los BBCodees disponibles.

Cabe reseñar que se pueden añadir países al desplegable que indica la nacionalidad desde una de las vistas de configuración del administrador.

Añadir una nueva obra

Para añadir una nueva obra se debe completar un formulario similar mostrado en la figura 5.9. Las peculiaridades de éste son que el campo donde escribimos el autor se autocompleta a medida que lo escribimos con los datos de los autores registrados.

Además, para geolocalizar la obra podemos poner las coordenadas a mano, o de un modo más intuitivo, pulsar el lugar deseado en el mapa.

Añadir una foto

Después de añadir la obra podemos añadirle fotos. Hay que recordar que se pueden agregar cuantas fotos se quiera tanto a los autores como a las obras.

Para ello se pulsa la opción correspondiente del menú y se selecciona el archivo, tal y como se muestra en la figura 5.10. Además hay que escribir un breve texto que se mostrará a modo de pie de foto y por supuesto, nombrar la obra a la que corresponde la imagen.

Búsquedas

En la sección de búsquedas podemos obtener los datos en detalle de cualquiera de los autores y obras de la web. Podemos buscarlos mediante varios criterios de búsqueda. En la imagen 5.11 se pueden apreciar los detalles.

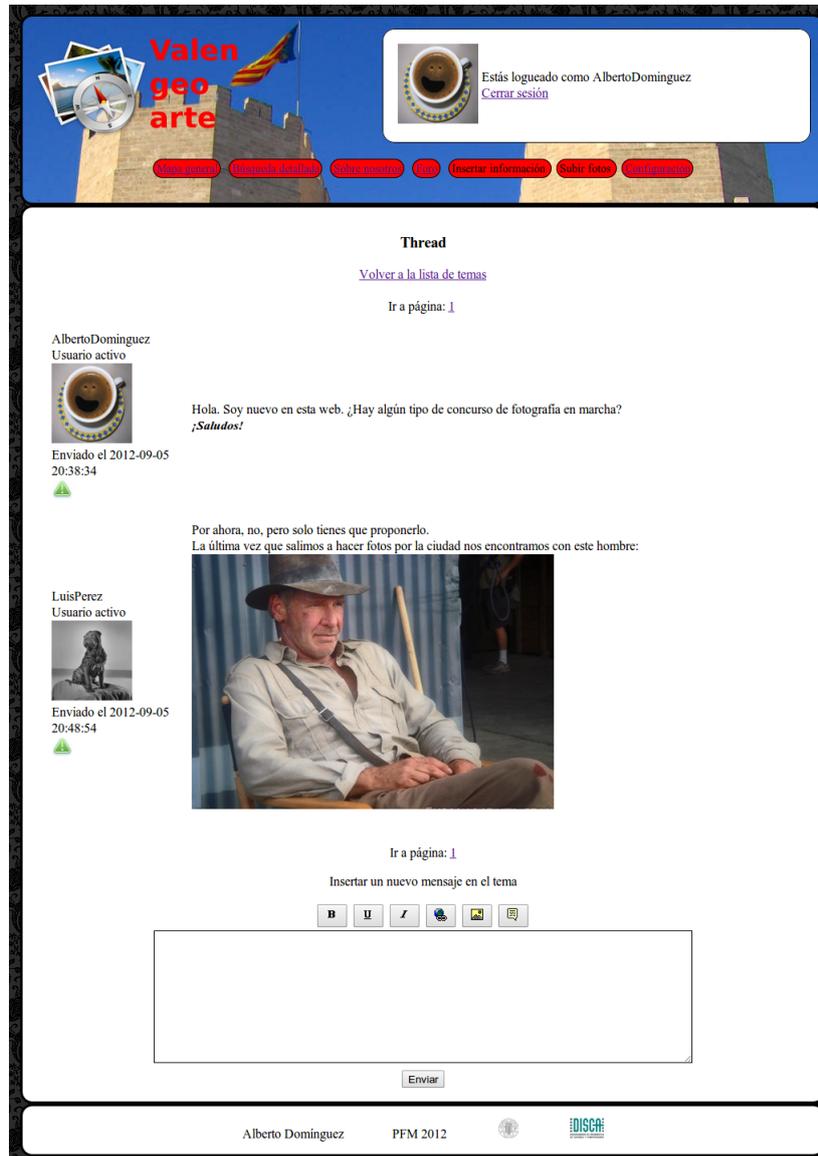


Figura 5.6: Vista de los mensajes

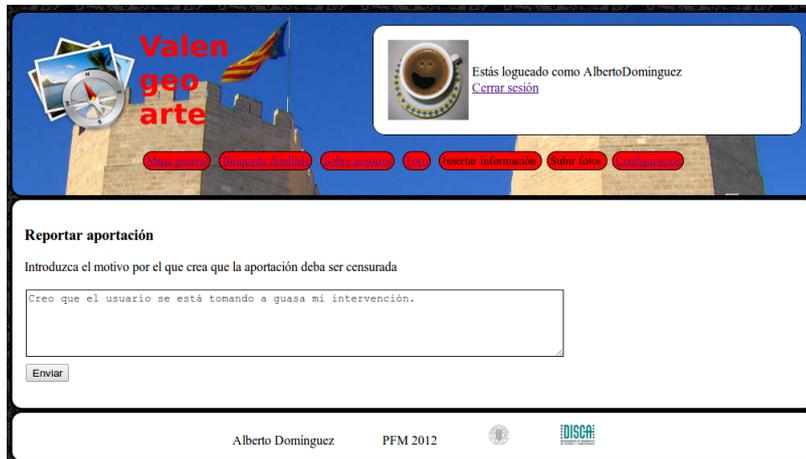


Figura 5.7: Añadir un reporte

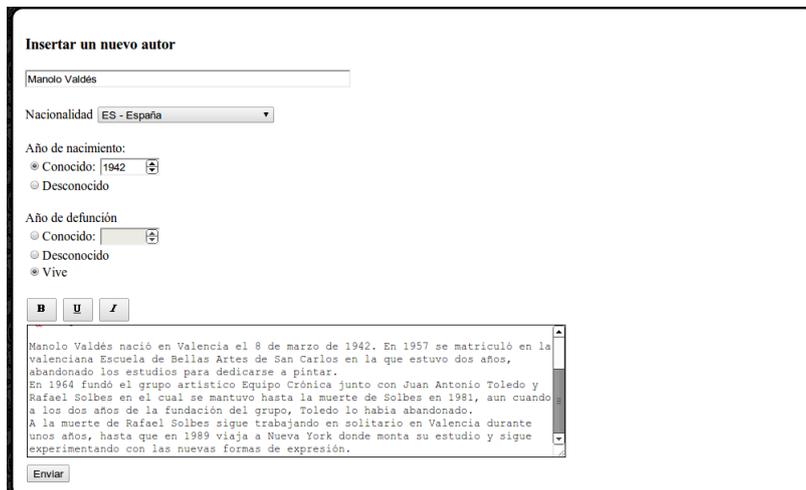


Figura 5.8: Añadir un nuevo autor

Insertar una nueva obra

Tipo de Obra:

Los tipos de obras a elegir son:

- *Escultura*: Esculturas y conjuntos escultóricos que podemos encontrar en las calles de Valencia.
- *Arquitectura*: Hitos arquitectónicos valencianos.
- *Puentes*: Relación de los puentes que atraviesan el antiguo cauce del Turia.
- *Fuentes*: Relación de algunas de las fuentes monumentales de la ciudad.
- *Museos*: Algunos de los museos valencianos más sorprendentes.

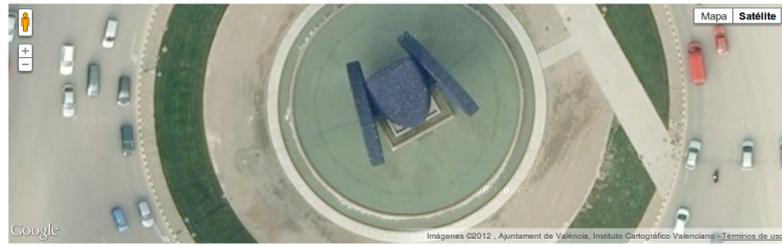
Inspirada en la Dana de Elche, se trata de un monumento de mas de 20 metros de altura, realizada en piezas de cerámica vidriada de color azul, con el objetivo de que reflejen los cambios cromáticos de la luz según las horas del día. La obra fue regalada por el escultor a la ciudad de Valencia y fue inaugurada en el año 2.007. Manuel Valdés ha sido declarado [1]hijo predilecto de la ciudad/[1]. En la realización de la escultura colaboró el artista fallero Manolo Martí, ya que el interior de la dama está formada por plataformas metálicas de trabajo.

Año de creación de la obra:

Conocido:

Desconocido

Selecciona un punto en el mapa para geolocalizar la obra



Mapa

Google Imágenes ©2012. Ajuntament de València, Institut Cartogràfic Valencià. [Términos de uso](#)

Figura 5.9: Añadir una nueva obra

Insertar fotos de obras

La foto deberá ocupar menos de 1 MB para ser subida. Si ocupa más no se subirá. Además si se sube en un formato distinto al JPG o con una resolución mayor de 800 píxeles de ancho se recodificará.

Figura 5.10: Añadir una foto de obra

Búsqueda

- Búsqueda de autores
 - Nombre del autor
 - Desconocido
 - Estaba vivo en el año: Año
- Búsqueda de Obras
 - Nombre del autor
 - Nombre de la obra
 - Inaugurada en el año Año
 - La obra pertenece al siguiente grupo: Escultura

Resultados

Ir a página: [1](#)

- Obra: Dama Ibérica (2007) 
 - Autor: Manolo Valdés, España (1942-Act.)
 - Descripción de la obra :
 Inspirada en la Dama de Elche, se trata de un monumento de mas de 20 metros de altura, realizada en piezas de cerámica vidriada de color azul, con el objetivo de que reflejen los cambios cromáticos de la luz según las horas del día. La obra fue regalada por el escultor a la ciudad de Valencia y fue inaugurada en el año 2.007. Manuel Valdés ha sido declarado *hijo predilecto de la ciudad*. En la realización de la escultura colaboró el artista fallero Manolo Martí, ya que el interior de la dama está formada por plataformas metálicas de trabajo.



Foto de la Dama Ibérica tomada desde la boca de metro de Beniferri

Ir a página: [1](#)

Figura 5.11: Página de búsquedas

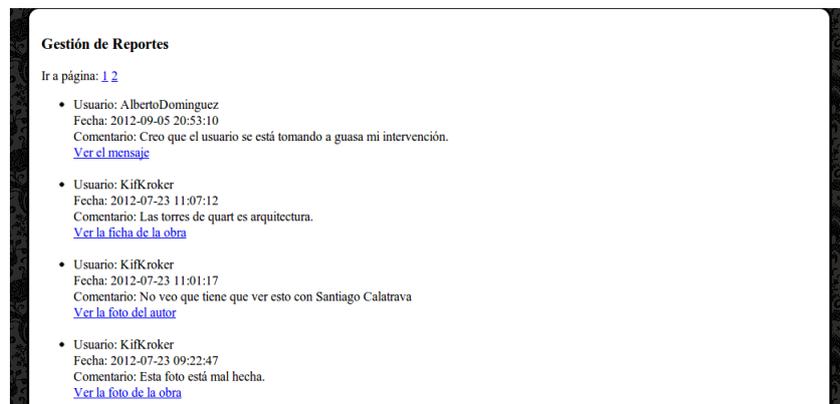


Figura 5.12: Listado de reportes



Figura 5.13: Moderación de mensajes

Opciones de administrador

Si nos identificamos como administrador en el sitio web, en el menú aparecerá una lista con opciones de configuración. Desde ellas se puede cambiar cualquier aspecto relacionado con las aportaciones de los usuarios sin necesidad de entrar a la base de datos.

En la figura 5.12 podemos ver la lista de reportes que nos han enviado los usuarios. Tras cada mensaje se puede acceder con un enlace a la página de administración correspondiente, ya sea mensaje, dato o foto para cambiarlo o ocultarlo a los usuarios.

Si pulsamos en el primer reporte nos aparecerá la página de administración de mensajes, mostrada en la figura 5.13, en la que podemos cambiar el texto, o moderarlo. En este caso, en el foro no aparecerá el texto sino la frase *“Este mensaje ha sido moderado por el administrador”*.

Otra de los varios recursos del administrador es el baneo, la expulsión y el borrado de avatares. Se accede a través de la página mostrada en la figura 5.14.

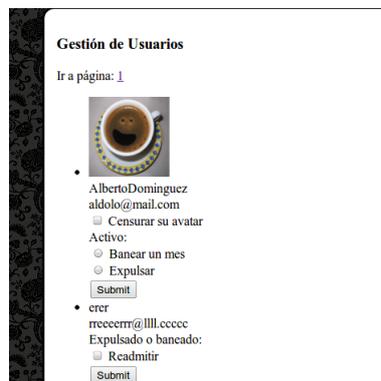


Figura 5.14: Administración de usuarios

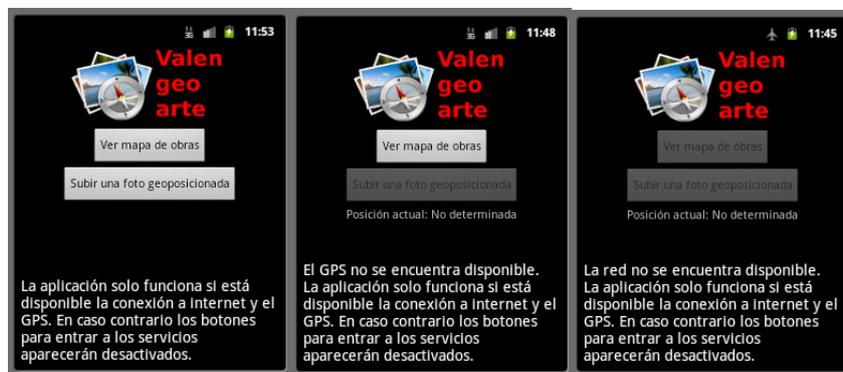


Figura 5.15: Aplicación Android funcionando correctamente, funcionando sin acceso a internet y funcionando sin GPS

5.2.2. Aplicación Android

Las opciones que dispone la aplicación de Android son las siguientes:

Pantalla principal

La pantalla principal sirve de bienvenida a la aplicación, como menú de opciones y como visualizador de errores. En caso de ejecutar la aplicación sin ningún tipo de acceso a internet, o sin GPS, la aplicación lo detecta e inhabilita las dos opciones en el primer caso, o sólo la subida de fotos cuando no se pueda acceder al GPS. En la figura 5.15 se muestran los tres casos.

Hacer foto

Pulsando el botón “Subir una foto geoposicionada” se abre mediante un *intent* el programa interno de la cámara que permite hacer una foto. Al pulsar aceptar se sube de forma automática al servidor. La confirmación se muestra



Figura 5.16: Subida de foto geoposicionada



Figura 5.17: Mapa de las obras en la aplicación Android

mediante un mensaje corto sobreimpresionado. Se muestra la función en la figura 5.16.

Mostrar mapa

Finalmente si pulsamos el botón del mapa, se abrirá un nuevo intent donde se muestra un MapView como en la figura x, con las obras indicadas con los mismos iconos que en la versión web. Al pulsar sobre un icono se abre un pequeño diálogo indicando el nombre y el autor de la obra.

Además el mapa se centrará en nuestra ubicación a medida que nos vayamos desplazando y cuando nos acerquemos a menos de 50 metros de una obra, el móvil emitirá un aviso sonoro.

Capítulo 6

Conclusiones

En este trabajo de la tesis del master se ha implementado un portal web con gestión de usuarios y herramientas colaborativas y sociales. Además de una aplicación móvil de apoyo que emplea técnicas de localización, posicionamiento y acceso a datos remotos.

El propósito de la aplicación es proporcionar al usuario toda la información disponible con respecto al arte urbano en la ciudad de Valencia de forma rápida, sencilla e intuitiva, aplicando criterios de búsqueda y filtrado para encontrar los datos precisos sin necesidad de indagar entre todas las obras catalogadas. Una vez concluido el desarrollo del portal y la aplicación podemos afirmar que los objetivos previstos inicialmente se han cumplido de un modo satisfactorio.

El desarrollo de este proyecto me ha permitido además entender cuáles son los procesos que hay que seguir para poner a disposición del público una aplicación móvil y hacer accesible una web en un entorno potencialmente hostil y propicio a los ataques como es Internet.

6.1. Experiencia en el desarrollo

Una de las frustraciones más comunes en los desarrolladores al inicio del proceso de aprendizaje es ver que los resultados no ofrecen una aplicación práctica real hasta que se consigue dominar la programación lo suficiente. El paso de programar un hola mundo hasta llegar a aplicaciones medianamente útiles es laborioso e implica dedicación y constancia. Sin embargo considero que el lenguaje PHP tiene en su baza una curva de aprendizaje muy suave, una muy buena documentación y un gran número de usuarios, por lo tanto la gran mayoría de dificultades encontradas han podido ser resueltas con la ayuda de programadores que han tenido el mismo problema anteriormente.

La aplicación móvil en Android pese a ser en apariencia más sencilla ha tenido muchos momentos de indecisión, desde la fase de diseño y análisis hasta

el momento de realizar las pruebas. Su mecanismo de permisos de uso es algo intrincado para un novato, y conservar los sensores al dejar un Intent y entrar en otro ha supuesto varios quebraderos de cabeza.

Personalmente, me encuentro satisfecho con el trabajo desarrollado y considero que tengo los fundamentos suficientes para la realización de proyectos de mayor envergadura tanto en PHP como en Android. Asimismo, tras acabar el proyecto, uno se da cuenta que el esfuerzo que se requiere para el aprendizaje no es excesivo y las limitaciones vienen dadas más por falta de imaginación durante la fase de análisis que en la fase de diseño o implementación.

6.2. Comparativa de los lenguajes de programación y de sistemas móviles

Para la realización del proyecto se ha tenido que analizar los diferentes lenguajes y plataformas que permiten programar páginas web generadas dinámicamente, al igual que varios sistemas operativos para *smartphones*. La comparativa me ha permitido conocer el mercado que soporta cada uno de los sistemas, su difusión o el número de programadores de cada uno, obteniendo una perspectiva certera, aparte de las consabidas estrategias de marketing con las que las empresas desarrolladoras nos venden sus sistemas.

Como conclusión me quedo con que en la actualidad el mejor sistema operativo, tanto por su facilidad de aprendizaje como por su versatilidad y variedad de dispositivos es Android, aunque el estricto control al que somete Apple a su tienda de aplicaciones, a mi modo de ver, hace que ésta tenga una mayor calidad a expensas de la libertad de desarrolladores y posibles usuarios.

Sin embargo como sistema de programación web, considero que la mejor solución hoy en día está comprendida entre JSP y PHP, dependiendo del alcance y la complejidad del sistema que se quiera programar. Aunque PHP es mucho más fácil de usar y es el más adecuado para proyectos pequeños y medianos, considero que Java ofrece innegables ventajas para grandes proyectos, además de ser un sistema en apariencia más escalable, robusto y rápido.

6.3. Mercado laboral

La percepción que tengo respecto al mercado laboral en torno al desarrollo de dispositivos móviles es mucho más positiva que con respecto al desarrollo web, que alcanzó su techo años atrás. Consultando las ofertas de trabajo en diferentes portales web, uno se da cuenta que obviando las sempiternas ofertas de consultoras, el grueso de ellas es para programadores experimentados en Android y iPhone.

6.4. Posibilidad de generar ingresos mediante el proyecto

La posibilidad de generar ingresos directos por el uso del portal web –mediante cuota de registro o renovación– está totalmente descartado, ya que el cobro por el uso de servicios web a usuarios finales dejó de tener sentido hace tiempo. Además no es concebible cobrar a usuarios teniendo en cuenta que son ellos los que deben usar su tiempo y ganas en aportar los datos necesarios en la web. Sin embargo se pueden barajar otros tipos de ingreso:

- Tendría más sentido la inclusión de banners publicitarios en lugares reseñados de la web si ésta llega a un cierto tráfico mensual. En caso contrario la inclusión de publicidad perjudicará la experiencia de navegación del usuario y los ingresos generados serán ínfimos. En cualquier caso, debido a la proliferación, con gran éxito, de extensiones del navegador que bloquean los banners publicitarios nos hace pensar que este tipo de publicidad deberá reinventarse en breve.
- También podría usarse la dirección de correo de los usuarios para el envío de publicidad, pidiendo la conformidad expresa del usuario al realizar el registro. En este caso el éxito es aún menor puesto que los usuarios tienden a usar cuentas de correo dedicadas a registrarse en sitios web en exclusiva por este mismo motivo. Además los filtros de spam de los principales servidores de correo son cada vez más precisos.

Los ingresos generados por la aplicación móvil en cambio generan otro tipo de escenarios.

- En la página principal de la aplicación se puede incluir un banner publicitario mediante herramientas proporcionadas por Google. Para su visualización –y cobro por nuestra parte– se necesita que la aplicación esté conectada a Internet, pero es un requisito que la propia aplicación comprueba. Hay métodos para evitar la visualización de este tipo de banners, pero al contrario que los filtros de los navegadores web, éstos apenas han tenido difusión.
- También se puede subir la aplicación al mercado de Google con un precio, pero por muy bajo que sea éste, es una traba que gran parte de los usuarios no aceptará. Además, como en el caso de la web, no parece moralmente correcto hacer pagar a un usuario por una aplicación cuyos datos los ha proporcionado él mismo.

Nos queda otra opción, que podría ser el patrocinio cultural del proyecto o la compra por otra empresa, ya sea de viajes, cultural, etc.

6.5. Datos y diseño extrapolables a otras aplicaciones

Como los datos de la aplicación son accesibles mediante mensajes XML es posible crear cualquier otra aplicación que las use. Podremos proporcionar al usuario nuevos tipos de interfaces y utilidades¹. Por ejemplo, aplicaciones de trivial sobre el conocimiento de la ciudad o generación de rutas turísticas entre monumentos.

Más interesante parece, sin embargo, el empleo de la aplicación con fines totalmente distintos con unos pocos cambios. Hay que tener en cuenta que bajo la página web y la aplicación subyace un sistema de catalogación y geoposicionado que cambiando muy poca información de la interfaz y localizaciones iniciales de los mapas, puede servir para catalogar cualquier tipo de elementos en cualquier ciudad del mundo.

¹Uno de los mejores ejemplos es la página biciv.com, de un compañero de la escuela. Tomando los datos del sistema de bicicletas de alquiler de Valenbisi ha creado un sistema mucho más completo, útil y usable que el oficial.

Capítulo 7

Bibliografía

Svennerberg, Gabriel *Beginning Google Maps API 3*. Ed. Apress, 2010. 310p.

Purvis, Martin *Beginning Google Maps applications with PHP and Ajax: from novice to professional*. Ed. Apress, 2006. 358p.

Doyle, Matt *PHP práctico*. Ed. Anaya multimedia, 2010. 848p.

Álvarez, Alonso *Manual imprescindible HTML 5*. Ed. Anaya multimedia, 2012. 479p.

Meng Lee, Wei *Beginning Android application development*. Ed. Wiley, 2011.

Documentación de Android <http://developer.android.com>

Documentación de PHP <http://php.net/docs.php>

Manual de PHP <http://php.net/manual/es/index.php>

Stackoverflow <http://stackoverflow.com>

Javascript tutorial <http://www.w3schools.com/js/default.asp>

Android Authority <http://www.androidauthority.com>

Wikipedia <http://en.wikipedia.org>