



Proyecto Fin de Carrera

**GEO-REMINDEME: Notificaciones basadas
en geo-localización para ANDROID**
**GEO-REMINDEME: Notifications based in
geo-location for ANDROID**

Para acceder al Título de
INGENIERO EN INFORMÁTICA

Autor: Eduardo Calzado Roura
Director: Marta Zorrilla Pantaleón

Junio - 2013

Dedicatorias

Dedicado a mi familia, que supo inculcarme constancia, capacidad de sacrificio, madurez, tenacidad y exigencia. Gracias a ellos he conseguido la meta académica que más persistencia requiere y que más trabajo ha costado. La mayor de vuestras virtudes ha sido dejarme escoger el camino permitiendo que me equivocara, para así recibir las enseñanzas que da la vida.

Dedicado a Javier Guitián, una de las personas de las que más enseñanzas académicas he aprendido y que con sus clases maestras, consiguió alimentar la confianza necesaria que necesitaba en los peores momentos de la carrera. Me hiciste ver que era capaz de esto, y mucho más.

Dedicado a todos aquellos que pretenden hacer cada vez más difícil la enseñanza pública, discriminando a jóvenes sin recursos que tienen derecho a tener las mismas oportunidades que aquellos que por suerte los tienen. Gracias a vosotros he comprendido lo mucho que cuestan las cosas, y el esfuerzo que implica trabajar para tener el simple derecho de estudiar.

Agradecimientos

Agradecido a todos aquellos profesores que tienen vocación y que mucho más lejos los límites de su trabajo, mostraron intención de inculcar sus conocimientos sobre nosotros.

Agradecido a Marta Zorrilla y Diego García que me han guiado hasta aquí, siempre intentando sacar el mejor partido de este proyecto y proporcionándome los medios que requería en cada momento. Gracias por vuestra ayuda.

Agradecido a Daniel Quevedo Ortiz, compañero de carrera y amigo que me ha ayudado enormemente en la consecución de este proyecto. Sus amables consejos, me han ayudado siempre a conseguir el máximo partido en cada una de las partes de éste.

Agradecido a ti, que me proporcionaste el cariño y la estabilidad necesaria en un año complicado. Tus palabras fueron la fuerza que me impulsó a embarcarme en este proyecto. Gracias por escucharme cada día y poner tu mejor sonrisa.

Agradecido a los esfuerzos económicos del ministerio de educación puesto que han conseguido que un joven con recursos bajos haya conseguido acabar su única meta, que era la de ser licenciado.

Resumen

Las aplicaciones móviles han impulsado el comercio de los Smartphones y han dado sentido a su uso. Es difícil imaginar el gran éxito de estos dispositivos entre los usuarios sin el atractivo de las apps, que han sabido romper el mercado y encontrar más aceptación de la que cabría imaginar en un principio. Tanto es así, que se ha hecho necesario tener al alcance de la mano en cualquier momento y desde cualquier lugar, acceso a cualquier tipo de información.

Con este vertiginoso avance de la tecnología y la aparición de los nuevos dispositivos móviles, la importancia de las aplicaciones basadas en geo-localización está creciendo exponencialmente. La tecnología actual ofrece un campo de posibilidades y la infraestructura necesaria para el desarrollo de nuevas aplicaciones que satisfacerían, e incluso optimizarían, las necesidades del usuario a las que están enfocados dichos dispositivos que requieren estar conectados en todo momento.

El hecho más desalentador es que existen actualmente al menos una decena de sistemas embebidos en Smartphones y Tablets, lo cual desde el punto de vista del desarrollador, hace complicada cualquier intención de abordar el inicio de una aplicación multiplataforma que tenga la aceptación suficiente para obtener un amplio número de usuarios activos.

El almacenamiento de la información es otro de los problemas del sector de desarrollo móvil. El almacenamiento web está experimentando un fuerte crecimiento en el sector y el modo en el que está evolucionando la infraestructura de la nube está dando lugar, como consecuencia, a una mayor demanda de productos de almacenamiento. Esto llevado a las plataformas móviles otorgaría innumerables ventajas dadas las limitaciones de espacio en estos dispositivos.

Por todo lo anteriormente descrito, este proyecto pretende aunar varias tecnologías con el fin de mostrar la capacidad de desarrollo que estas proponen y sus ventajas puestas al servicio común. Por ello, se desarrolla una aplicación móvil multiplataforma basada en geo-localización mediante tecnologías web, y a su vez, se desarrolla un sitio web con el mismo propósito para proporcionar su descarga y registro entre otros. La información residirá en un servidor web permitiendo el acceso a la información, tanto a la aplicación móvil como al sitio web.

Palabras clave: Geo-localización, app, multiplataforma, tecnología web, nube.

Abstract

Mobile applications have boosted the trade of Smartphones and have justified their use. It is hard to imagine the success of these devices among users without the lure of apps, that they have known how to break the market and find an incredible acceptance. Now is necessary to have in your hands at any time and from anywhere, access at any type of information.

With the rapid advance of technology and the emergence of the new mobile devices, the importance of geo-location applications is growing exponentially. Today's technology offers a range of possibilities and the necessary infrastructure to develop new applications that would satisfy, and even would optimize, the requirements that user needs to be connected at all times.

The most discouraging fact is that currently, at least, there are a dozen operative systems embedded in smartphones and tablets, which from the point of view of the developer, makes difficult any purpose to begin the development of multiplatform application that has enough acceptances to get a large number of active users.

The storage of information is another problem of the mobile development sector. The web storage is experiencing a strong increase and the way it's evolving the cloud's infrastructure is inducing, therefore, to an increase of the demand for storage products. It, in the mobile platforms, would provide many advantages regarding the limitations of space in these devices.

For the above described, this project aims to combine several technologies in order to show the ability of the development proposed and to join their advantages. Therefore, it develops a mobile application platform based on geo-location using web technologies, and in turn, develops a website with the same purpose to provide some actions like download or registration. The information will reside on a web server that can be accessed from the mobile application and website.

Key words: Geo location, app, multiplatform, web technologies, cloud.

Índice

Tabla de ilustraciones.....	7
Capítulo 1. Introducción	
1.1 Motivación y Antecedentes	9
1.2 Objetivos.....	12
1.3 Metodologías de desarrollo software	14
Capítulo 2. Especificación de Requisitos	
2.1 Captura y análisis de requisitos	17
2.2 Requisitos Funcionales	17
2.2.1 Requisitos de Invitado	18
2.2.2 Requisitos de Usuario Registrado	19
2.2.3 Requisitos de Sistema	21
2.3 Requisitos No Funcionales	23
2.3.1 Requisitos de Producto.....	23
2.3.2 Requisitos de Proceso	23
2.3.2.1 Captura de requisitos, análisis y diseño	24
2.3.2.2 Software y tecnologías en la construcción de la aplicación móvil.....	24
2.3.2.3 Lenguaje de desarrollo en la implementación de la aplicación móvil.....	25
2.3.2.4 Técnica de desarrollo de la aplicación móvil	27
2.3.2.5 Software y tecnologías en la construcción de la aplicación web	27
2.3.2.6 Gestor de bases de datos, introducción a la nube y al contenido web dinámico	28
2.3.2.7 Software y técnicas utilizadas para el desarrollo gráfico de componentes web y móvil	30
Capítulo 3. Diseño e Implementación	
3.1 Visión general del sistema	32
3.2 Diseño de la arquitectura del sistema	33
3.2.1 Estilo y patrón arquitectónico	33
3.2.2 Arquitectura lógica del sistema	34
3.2.3 Distribución de los componentes lógicos	35
3.3 Arquitectura, diseño e implementación de la aplicación móvil	37
3.3.1 Implementación JS API.....	39

3.3.2 Implementación nativa.....	41
3.3.3 Acción y reacción ante eventos nativos	41
3.4 Diseño e implementación de la capa de persistencia.....	43
3.4.1 Repositorio de datos	43
3.4.2 API o Web Layer intermedia.....	44
3.5 Secuencia de eventos	46
3.5.1 Registro en la aplicación móvil	46
3.5.2 Inicio de sesión y recepción de tareas en aplicación móvil	46
3.5.3 Inicio y envío de tareas en la aplicación móvil	48
3.5.4 Búsqueda de direcciones mediante la API de Google Maps	49
Capítulo 4. Interfaz Gráfica	
4.1 Interfaz gráfica de la aplicación móvil.....	51
4.2 Interfaz gráfica de la aplicación web	61
Capítulo 5. Pruebas	
5.1 Evaluación y Pruebas	63
5.2 Software y tecnologías utilizadas para las pruebas en el sistema	64
5.3 Control de versiones	65
Capítulo 6. Problemas e impedimentos encontrados	
6.1 Uso de la API Google Maps v3	66
6.2 Simulador SDK Eclipse	66
6.3 Encriptación y contraseñas en el CMS Wordpress	67
6.4 Dominio y servidor web	67
6.5 Comunicación remota Cross-Domain	68
6.6 Diseño responsable	68
6.7 Dispositivos móviles.....	69
Capítulo 7. Conclusiones	70
Capítulo 8. Trabajos futuros	73
Capítulo 9. Bibliografía	75

Tabla de Ilustraciones

Capítulo 1.

Figura 1-1. Estudio sobre el uso de Smartphone en el mercado español	9
Figura 1-2. Estudio sobre el tipo uso de Smartphone en el mercado español	9
Figura 1-3. Porcentaje de aumento de ingresos en App. Fuente: Flurry Analytics	10
Figura 1-4. Implementación básica de desarrollo de PhoneGap	10
Figura 1-5. Cloud Computing	11
Figura 1-6. Gráfico orientativo del sistema	12
Figura 1-7. Esquema de la metodología Extreme Programming	15
Figura 1-8. Fases de la metodología DPC	16

Capítulo 2.

Figura 2-1. Casos de uso usuario Invitado	18
Figura 2-3. Casos de uso usuario registrado en la aplicación web.	19
Figura 2-2. Casos de uso usuario registrado en la aplicación móvil.....	20
Figura 2-4. PhoneGap Build. Abstracción del uso del framework	25
Figura 2-5. Proceso de desarrollo de la aplicación móvil.....	27
Figura 2-6. Popularidad de Wordpress en el sector CMS.....	28
Figura 2-7. Maquetación HTML.....	30

Capítulo 3.

Figura 3-1. Diseño de arquitectura a alto nivel	32
Figura 3-2. Diagrama del patrón arquitectónico "Tres capas"	34
Figura 3-3. Aproximación lógica del sistema.....	35
Figura 3-4. Patrón de distribución lógica de los componentes	36
Figura 3-5. Distribución lógica de los componentes	36
Figura 3-6. Tipos de Arquitecturas de desarrollo en aplicaciones móviles	37
Figura 3-7. Arquitectura en el desarrollo de la aplicación móvil	38
Figura 3-8. Diagrama de clases JS API	39
Figura 3-9. Arquitectura en el desarrollo de la aplicación móvil.	40
Figura 3-10. Reacción ante eventos nativos	41
Figura 3-11. Diagrama de relación de tablas	43
Figura 3-12. Abstracción de implementación de la Web Layer	44
Figura 3-13. Diagrama de secuencia de petición de registro	45
Figura 3-14. Diagrama de secuencia de inicio de sesión y recepción de tareas.....	47

Figura 3-15. Diagrama de secuencia de cierre de sesión y envío de tareas.....	48
Figura 3-16. Búsqueda de direcciones mediante la API Google Maps	49

Capítulo 4.

Figura 4-1. Versión Android y Dispositivo de implementación	51
Figura 4-2. Pantalla de Bienvenido y condiciones de contrato	51
Figura 4-3. Pantalla de inicio de sesión.....	52
Figura 4-4. Pantalla de Registro de nuevo usuario	52
Figura 4-5. Pantalla principal de la aplicación.	53
Figura 4-6. Ejemplo de compartición y GPS desconectado.....	54
Figura 4-7. Pantalla de Perfil.....	55
Figura 4-8. Pantalla de Navegación/Mapa	55
Figura 4-9. Leyenda de marcadores en la pantalla Mapa.....	56
Figura 4-10. Pantalla de Navegación/Mapa – Street View.....	56
Figura 4-11. Pantalla de Tareas y Visualización de nota.....	57
Figura 4-12. Pantalla de Tarea, formulario.	58
Figura 4-13. Pantalla de contactos, cámara, audio, y selección de foto en álbum.....	59
Figura 4-14. Pantalla de búsqueda de dirección.....	60
Figura 4-15. Status Notification Bar	60
Figura 4-16. Pantalla principal de la página web	61
Figura 4-17. Pantalla Geo-RemindMe	62
Figura 4-18. Pantalla Geo-RemindMe – Tareas y Formulario	62

Capítulo 5.

Figura 5-1. Control de versiones y cambios de la aplicación móvil	65
---	----

Capítulo 6.

Figura 6-1. Resoluciones de pantalla en dispositivos móviles.	69
--	----

Capítulo 1. Introducción.

1.1 Motivación y antecedentes

Hoy en día todo gira en el desarrollo de sistemas móviles. Como muestra la Figura 1-1, Un 59,6% de la población dispone ya de móviles inteligentes con acceso a internet. Esto provoca que a medida que aumentan los usuarios, crezcan exponencialmente las expectativas que proporciona el hecho de tener un Smartphone, lugar donde juega un papel importante el desarrollo de apps.



Figura 1-1. Estudio de *Empirica*, sobre el uso de Smartphones en el mercado español. [34]

Por si esto no fuera suficiente para plantearse el desarrollo de aplicaciones móviles como factor a tener en cuenta, cabe destacar que el 16% de usuarios que compran un Smartphone lo hacen para disfrutar de una gama de aplicaciones variada, siendo este el tercer motivo de uso de un Smartphone. Véase Figura 1-2.

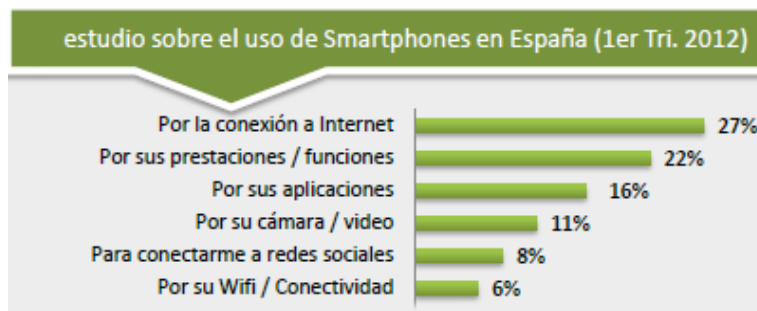


Figura 1-2. Estudio de *Empirica*, sobre el tipo uso de Smartphones en el mercado español. [34]

Estas estadísticas, mostradas en la Figura 1-3, marcan la pauta de interés del mercado actual. En la mano del desarrollador, y esta vez en mi caso, se llega a la conclusión que es una oportunidad de negocio emergente con un gran impacto. Si a esto le sumo que en España se han aumentado en este último año los ingresos por aplicación, se toma la decisión de plantear y desarrollar el presente proyecto fin de carrera.



Figura 1-3. Porcentaje de aumento de ingresos en apps. Fuente: Flurry Analytics.

El único problema que se plantea es que framework utilizar para que la aplicación llegue al mayor número de usuarios posibles. La dificultad radica en que el mercado actual se muestra variable en cuanto al uso de sistemas nativos fluctuando entre Android e IOs. Por ello, la solución adoptada fue utilizar un framework, que permitiera desarrollar una app mediante una tecnología que posibilite implementar una aplicación portable y multiplataforma. Véase Figura 1-4.



Figura 1-4. Implementación básica de desarrollo de PhoneGap.

Como no podía ser de otra forma, aplicación y móvil son sinónimos de movilidad, y actualmente el Cloud Computing es la tecnología en la que se basan todas las grandes empresas como Apple (iCloud), Google (GoogleDrive) o Microsoft (SkyDrive).

Las últimas innovaciones en este campo provocan que las aplicaciones comerciales sean ubicuas y cuenten con mayor capacidad de colaboración, pudiéndose acceder a información independientemente del dispositivo físico desde el cual se acceda, como se muestra en la Figura 1-5.



Figura 1-5. Cloud Computing

También cabe destacar el rol que está tomando Google en el ámbito de las aplicaciones móviles con su sistema operativo Android. Actualmente, Google se halla desarrollando intensamente APIs con fines de uso público en su plataforma móvil por excelencia. Esto hace que todas las aplicaciones basadas en geo-localización escojan Android por ser Google Maps el mejor preparado para el efecto.

Este proyecto se enmarca en el desarrollo de aplicaciones móviles, y en concreto, en la creación e implementación de una aplicación basada en geo-localización mediante la herramienta PhoneGap, que es el framework multiplataforma más reciente y completo. A su vez, las interesantes ventajas del Cloud Storage provoca la implementación de un sitio web y unos servicios que permitan el acceso desde ambas aplicaciones al uso de la información alojada en el servidor de almacenamiento.

1.2 Objetivos

El objetivo principal de este proyecto es crear e implementar la infraestructura necesaria para el desarrollo de apps basadas en geo-localización multiplataforma, probar su implementación en el sistema móvil Android mediante el framework PhoneGap y desviar su almacenamiento a un servidor que gestiona una base de datos. La Figura 1-6 presenta un gráfico orientativo de lo que se pretende.



Figura 1-6. Gráfico orientativo del sistema

Este objetivo, se desglosa en los siguientes sub-objetivos:

- Estudio del framework de desarrollo PhoneGap. Herramienta de desarrollo de aplicaciones móviles multiplataforma.
- Estudio de la API Google Maps v3. API que permite implementar su sistema de mapas y geo-localización en entornos basados en tecnologías web.
- Estudio de la Cloud Storage e implementación en una base de datos remota. Esta será funcional para el dominio web y mediante una API personalizada garantizará la accesibilidad a la aplicación móvil.
- Especificación de requisitos de la aplicación móvil como caso de uso. Permitirá a través de la ubicación del usuario, realizar notificaciones relativas a su proximidad respecto a un evento no prioritario definido anteriormente. Es importante destacar que no se trata de una aplicación

tipo calendario puesto que su objetivo principal será el de optimizar el tiempo en cuanto necesidades o tareas del usuario basadas en proximidad. Un ejemplo de uso sería: Recoger el paquete de la oficina de correos de aquí a 7 días. No es una tarea prioritaria, por tanto la app optimizará el tiempo invertido en atender recados cuando se esté lo suficientemente cerca como para completar la acción con el menor trastorno posible.

- Desarrollo, pruebas y puesta en marcha de la aplicación en varios dispositivos móviles. Se atenderá a las necesidades de varios dispositivos móviles con diferentes versiones de Android y resoluciones de pantalla para garantizar su diseño responsable e implementación regresiva en plataformas antiguas.
- Desarrollo de sitio web para gestionar la información y garantizar la usabilidad de la aplicación móvil.
- Despliegue de aplicación web en el dominio GeoApps-project.com y la aplicación móvil Geo-RemindMe en Google Play, el mercado de aplicaciones oficial de Android.

1.3 Metodologías usadas

El desarrollo de aplicaciones móviles sufre prácticamente los mismos problemas que la gran mayoría de desarrollos de software. Aunque hay que tener en cuenta sus principales peculiaridades como son la corta duración de sus desarrollos, la gran competencia del sector que obliga a una constante innovación, los cambios frecuentes en la plataforma de desarrollo, las innovaciones a nivel hardware así como la simplicidad de algunas aplicaciones.

La mayor parte de los proyectos de desarrollo de software para aplicaciones móviles se llevan a cabo por equipos pequeños de desarrolladores que habitualmente utilizan una metodología de desarrollo de software ágil, como Scrum, Crystal o XP por los siguientes motivos:

- La alta volatilidad del entorno provoca que constantemente el equipo de desarrollo se deba adaptar a nuevos terminales, cambios en la plataforma o en el entorno de desarrollo. Un ritmo cambiante que requiere una alta respuesta al cambio más que al seguimiento de un plan concreto.
- Los equipos de desarrollo móvil suelen ser integrados por pocas personas. No más de ocho o diez desarrolladores entorno a una misma aplicación o, incluso, un único desarrollador como es en este caso.
- Los periodos de desarrollo son cortos entorno de un mes a seis meses.
- El propósito principal es una realimentación rápida cuyo objetivo sea realizar varias actualizaciones de una aplicación según se van aumentando y/o perfeccionando funcionalidades.

El motivo principal del uso de estas metodologías es la demanda de los usuarios, en los distintos markets, sobre aplicaciones que aprecien la frecuente mejora de la aplicación para mantenerla viva y actualizada.

Particularmente, se ha utilizado una metodología XP (Extreme Programming) para el desarrollo de la aplicación móvil. Esta permite que los cambios de requisitos sobre la marcha sean un aspecto natural, inevitable e incluso deseable en el desarrollo del proyecto. Proporciona la capacidad de adaptación a los cambios de requisitos en cualquier punto de la vida del proyecto demostrando que es una mejor aproximación y más realista, que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

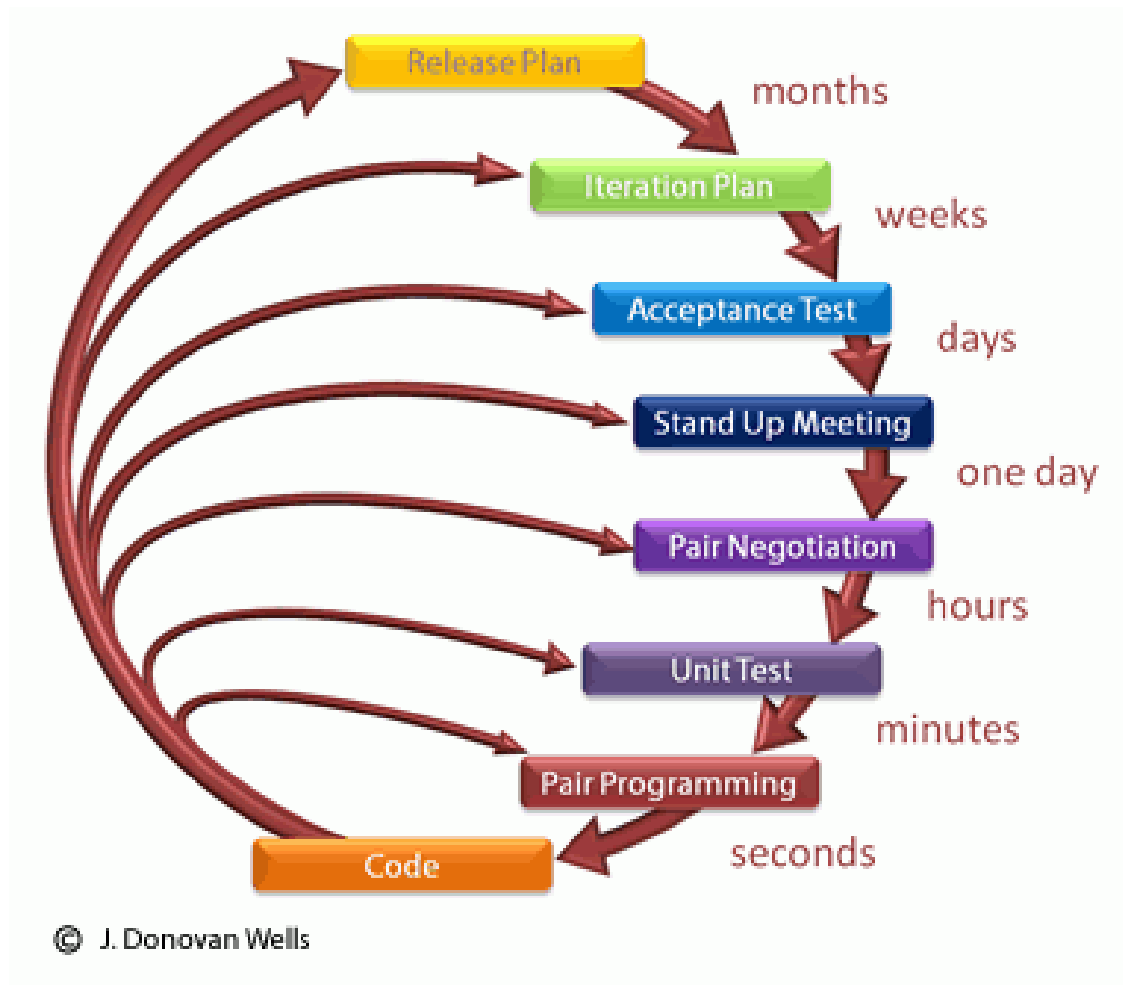


Figura 1-7. Esquema de la metodología Extreme Programming

Las características que definen esta metodología son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Frecuente interacción del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidades adicionales si es necesario.

Por su parte, el desarrollo de la plataforma web está basada en un desarrollo DPC (Desarrollo por capas). Este permitirá evaluar de forma independiente cada una de las fases que conforman el proceso de desarrollo del sitio web.

Las fases de esta metodología son:

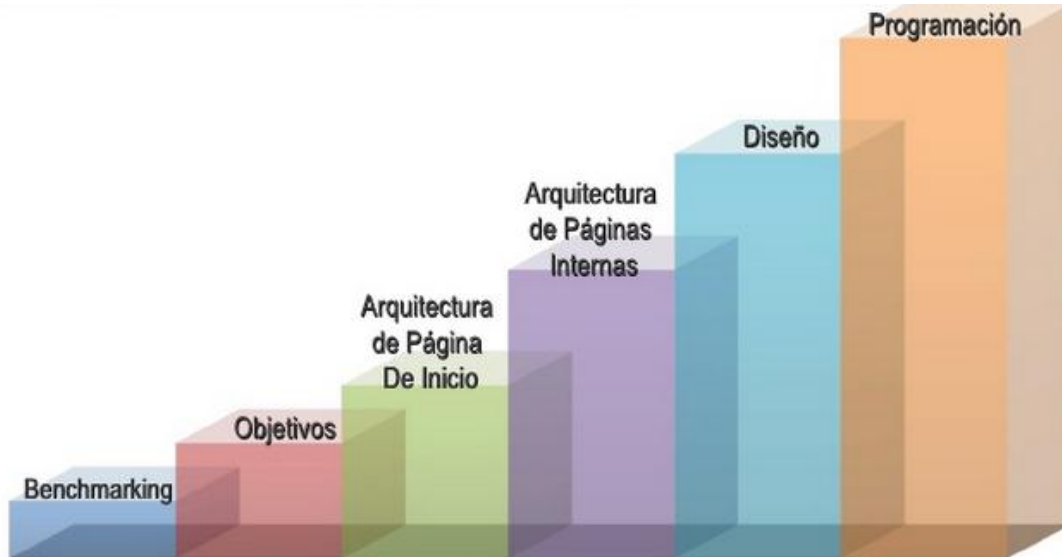


Figura 1-8. Fases de la metodología DPC.

- **Benchmarking:** Asegurar la incorporación de elementos y funcionalidades destacados en sitios web de la categoría de productos del cliente, para ofrecer el sitio web más competitivo.
- **Objetivos:** Se establece el contenido, funcionalidades y jerarquización de prioridades de comunicación, así como garantizar una clara interpretación de lo que se quiere como resultado final.
- **Arquitectura página de Inicio:** Distribuye el contenido prioritario en la página de inicio, reflejando lo acordado en el documento de objetivos.
- **Arquitectura de páginas internas:** Desarrollar todas las páginas y asegurar una navegabilidad intuitiva del sitio web para generar una experiencia positiva en cada visita del usuario.
- **Diseño:** Apoyar y resaltar la comunicación visual y gráfica asegurando una línea gráfica consistente.
- **Programación:** Entregar el producto final de forma que sea 100% autoadministrable por parte del cliente.

Capítulo 2. Especificación de Requisitos

2.1 Captura y análisis de requisitos

El primer paso para desarrollar cualquiera de las dos aplicaciones, al igual que en cualquier proyecto software, es identificar a los actores del sistema y capturar los requisitos. Estos pueden ser funcionales y no funcionales:

- Los **funcionales** describen la funcionalidad y/o servicios del software, entre los que se distinguen:
 - De usuario. Descripciones de alto nivel de lo que el sistema debe hacer.
 - De sistema. Descripción de los servicios con más detalle.
- Los **no funcionales** establecen restricciones sobre el producto desarrollado, el proceso de desarrollo o consideraciones relacionadas con factores externos:
 - De producto. Comportamiento del sistema.
 - De proceso. Indican estándares, tecnologías, entornos de desarrollo entre otros.
 - Externos. Relacionados con factores externos del sistema.

2.2 Requisitos funcionales

En primer lugar, identificaremos los actores que van a interactuar en nuestro sistema, que se compone de dos aplicaciones, la aplicación web y la aplicación móvil. Para ambas tendremos los mismos actores, aunque con diferentes funciones en algunos casos. Son los siguientes:

- Usuario no registrado: Usuario que no tiene acceso a ninguna de las funciones de la aplicación y solo verá información en el sistema web. Será tratado en los diagramas como "Invitado" y estará diferenciado según el sistema que use, web o móvil.
- Usuario registrado: Usuario ya inscrito en la base de datos, que efectuó su registro previamente y que por medio de su login tiene acceso a cualquier característica y componente tanto en la web como en la aplicación. Será tratado en los diagramas como "Usuario".

- Usuario de mantenimiento: Se trata de técnicos informáticos que gestionarán el uso de la base de datos, mantendrán el sistema web ante los cambios necesarios que se produzcan en la aplicación móvil, atenderán las necesidades de los usuarios y controlarán todos aquellos riesgos de seguridad que observen en cada petición al servidor que almacena la base de datos. No se aborda en este proyecto inicialmente pero en el futuro se plantea crear una interfaz que sirva para su mantenimiento en el sistema web.

2.2.1 Requisitos de Invitado

La Figura 2-1 define con casos de uso lo que este tipo de usuario requiere de la herramienta:

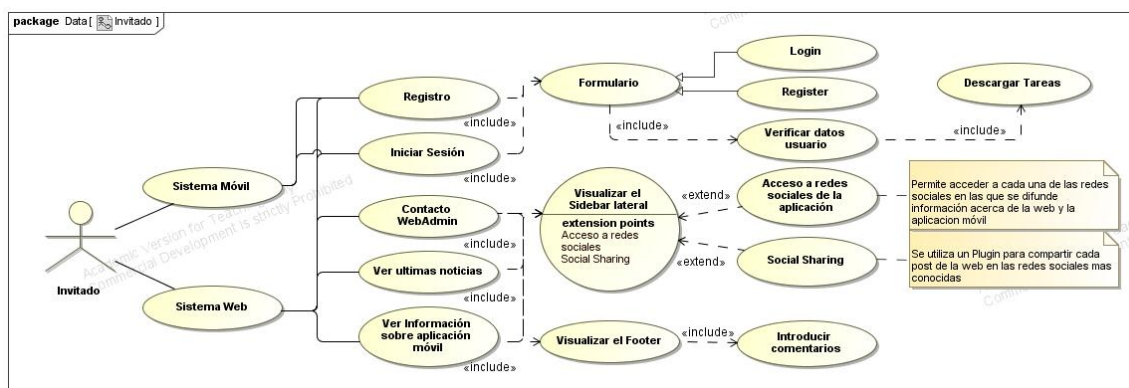


Figura 2-1. Casos de uso usuario Invitado

A la vista de los casos de uso, los requisitos textuales de la aplicación web son los siguientes:

- Interfaz gráfica de acceso a la aplicación web *GeoApps-Project*.
- Registro de usuario e inicio de sesión.
- Página de bienvenida, con diferentes secciones de acceso.
- Acceso a página de noticias y a la página informativa de la aplicación.
- Visualización de las últimas noticias.
- Posibilidad de compartición en redes sociales.
- Posibilidad de rellenar formularios de comentarios relativos a cada noticia.
- Posibilidad de rellenar formularios de comentarios relativos a cada aplicación.

A la vista de los casos de uso, los requisitos textuales de la aplicación móvil son los siguientes:

- Interfaz gráfica de acceso a la aplicación móvil *GeoApps-Project*.
- Registro de usuario e inicio de sesión.

2.2.2 Requisitos de Usuario Registrado

Los requisitos textuales de la aplicación web son los siguientes:

- Posibilidad de acceso a todas las funciones que tiene un usuario invitado.
- Visualizar las tareas alojadas en la base de datos.
- Añadir una tarea.
- Eliminar una tarea.

La Figura 2-2 define con casos de uso lo que este tipo de usuario requiere de la herramienta web:

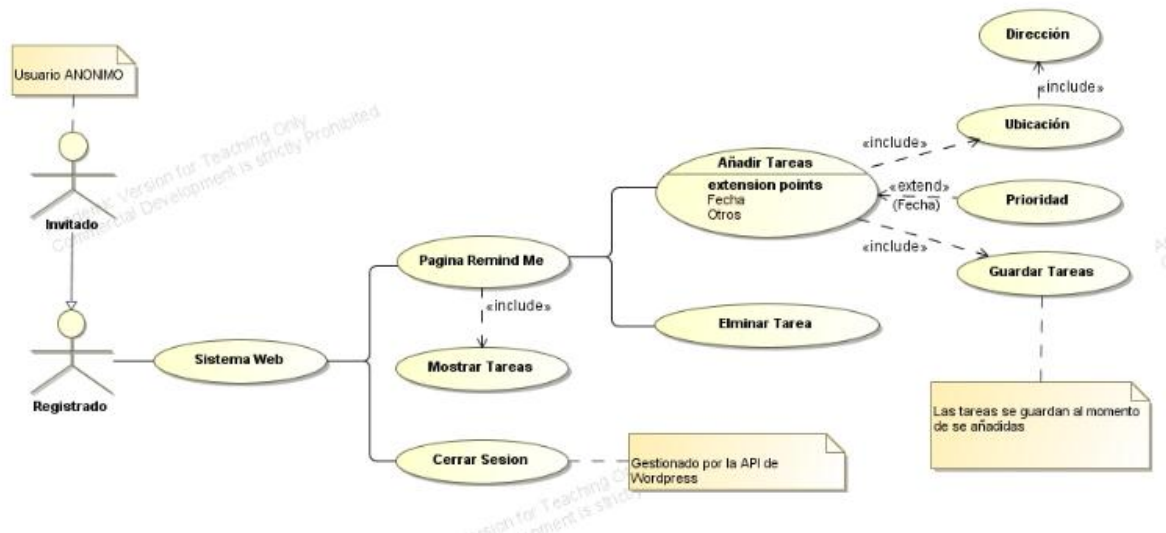


Figura 2-2. Casos de uso usuario registrado en la aplicación web

Por otra parte, los requisitos textuales de la aplicación móvil se especifican a continuación:

- Registro de usuario e inicio de sesión.
- Interfaz gráfica de uso de la aplicación móvil *GeoApps-Project*.
- Visualizar las tareas expiradas y no expiradas.
- Visualizar de forma independiente cada tarea, observando toda la información relativa a ésta (notas, nombre, foto en su caso, contactos añadidos) además de un mapa para intuir o determinar su ubicación con un simple vistazo.
- Seguir, en modo navegación, nuestra localización y las tareas representadas gráficamente con marcadores en diferentes colores.

Geo localización en Android

- Ver, en modo navegación, la información relativa a la tarea pulsando sobre el marcador, como por ejemplo la distancia, el nombre de la tarea.
- Crear nuevas tareas, siempre definiendo una ubicación para ésta. Otras características también son necesarias como la fecha, el título...
- Borra tareas.
- Ajustar la prioridad de la tarea para definir así su importancia a la hora de decidir una fecha de finalización.
- Agregar fotos mediante la cámara o galería fotográfica a la tarea.
- Agregar un contacto al campo de notas de la tarea.
- Agregar una ubicación, la cual podrá ser definida por la búsqueda de la posición actual mediante GPS y/o red, o mediante la inserción de la dirección manualmente.
- Cambiar el ajuste de notificaciones, tales como activar sonido, vibración y definir la distancia mínima a tarea con motivo de iniciar una notificación.
- Ver información relativa al usuario y su registro. Nombre, fecha de registro, nickname y correo electrónico entre otros.
- Notificar aquella tarea cercana, dentro del rango definido por el usuario, al usuario mientras esté activa la aplicación.
- Cerrar sesión y guardar todos los cambios realizados durante el uso de la aplicación.

La Figura 2-3 define con casos de uso lo que este tipo de usuario requiere de la herramienta móvil:

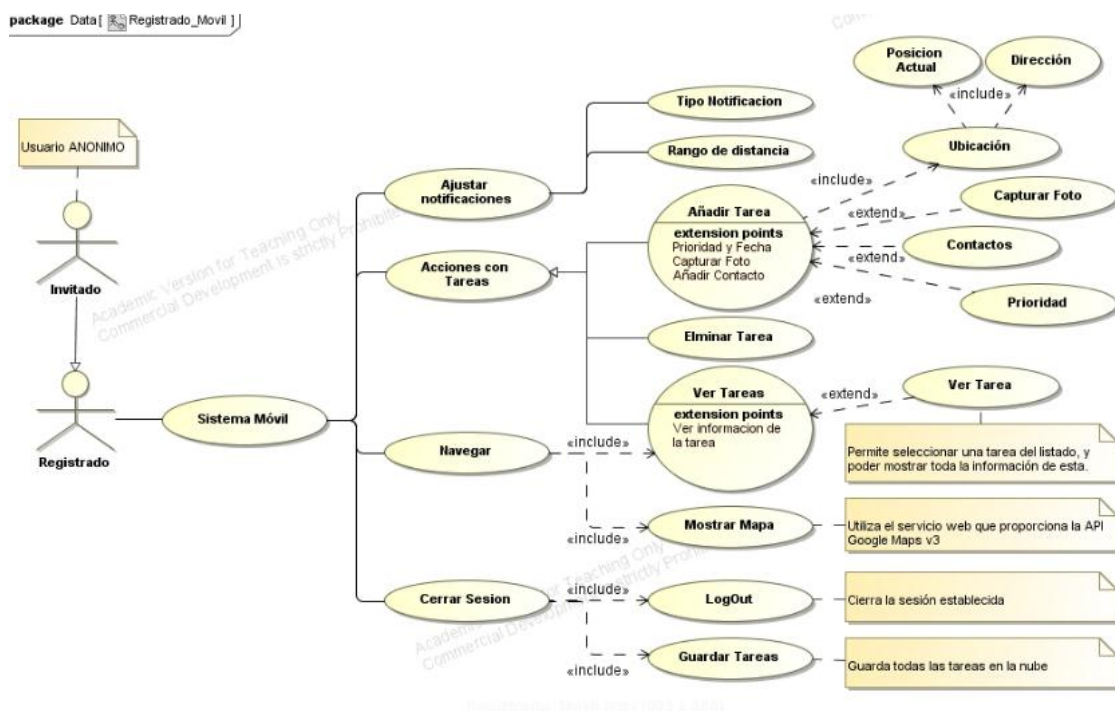


Figura 2-3. Casos de uso usuario registrado en la aplicación móvil

2.2.3 Requisitos de Sistema

Se ofrecen a continuación las funcionalidades que el sistema debe ofrecer al usuario de la aplicación móvil:

- **Confiable.** La aplicación será funcional en background y avisará al usuario en caso de error o notificación cercana.
- **Adaptable.** La base de datos interna del dispositivo físico funcionará como intermediario con el fin de optimizar la velocidad de uso de la aplicación.
- **Óptima:**
 - Se reducirá el número de transacciones con la base de datos, con el objetivo de no sobrecargar la base de datos con peticiones.
 - Notificará con una alerta a través de la barra de notificaciones y en el interior de la aplicación.
- **Eficiente:**
 - Las consultas, serán ligeras para minimizar el impacto en el uso de datos.
 - Se comprobará si existen tareas cercanas para, en este caso, notificar por medio del sistema definido por el dispositivo móvil.
- **Portable.** Su desarrollo permitirá usarse en cualquier plataforma disponible en lo relativo a dispositivos móviles.
- **Preciso.** Deberá obtener la localización por GPS proporcionando que su margen de error posicional sea mínimo. Así mismo, deberá cumplir con el efecto para el que se desarrolló, optimizar nuestro tiempo en dependencia de las tareas a realizar según su ubicación.
- **Interoperable:**
 - Deberá ser capaz de gestionar la conexión con todos los servicios web disponibles, que permiten una completa operabilidad.
 - Intermediará entre las necesidades del usuario y gestionar el uso de los servicios web que la API de Google Maps v3 proporciona.
- **Escalable.** La aplicación permitirá ser evolutiva. Ningún cambio será traumático, facilitando la progresión que conlleva mantenerse actualizado en un mundo que cambia día tras día.

Se ofrecen a continuación las funcionalidades que el sistema debe ofrecer al usuario en ambas aplicaciones:

- Segura.
 - Sesiones: En el acceso de varios usuarios simultáneamente, se requerirá que el sistema controle las sesiones de forma que no interfieran unas en otras y que cada una trabaje con sus datos.
 - Concurrencia: Las ejecuciones concurrentes que acceden a datos comunes se harán de forma sincronizada.
 - Encriptación: Los datos enviados estarán cifrados.
- Atractiva. El diseño de la aplicación debe proporcionar una interfaz intuitiva y agradable a la vista del usuario.
- Mantenibilidad: La modularidad y la documentación del código que de la aplicación proporciona una más que asumible capacidad de modificación en caso que sea necesario.
 - Asegurará un mantenimiento sencillo, facilitará la escalabilidad y aumentará la seguridad.
- Accesibilidad. Deberá ofrecer acceso a otras aplicaciones futuras basadas en geo localización con motivo reusar el código y minimizar el tiempo de desarrollo.
- Disponibilidad. La obtención de los datos alojados en el servidor de datos deberán estar disponibles cuando el usuario así lo requiera.
- Funcionalidad. Las características para las que fue descrita la aplicación obligarán a que ésta cumpla con sus objetivos de desarrollo.
- Consistente. Toda transacción deberá garantizar la consistencia de los datos, y mantener la base de datos en un estado conocido, controlando cualquier tipo de uso inapropiado del sistema.
- Control de errores. Evitará que la herramienta se bloquee por un mal uso de esta, advirtiéndole de los errores que se han producido y aportando soluciones. Alguno de los errores pueden ser:
 - Fallo en el inicio de sesión, registro o cierre de sesión.
 - Fallo en el envío de tareas al servidor de datos o su recepción.
 - Fallo de posicionamiento o ubicación.
 - Fallos internos del aplicativo, fallos software en caso del dispositivo móvil o HTML en caso de la web.

2.3 Requisitos no funcionales

2.3.1 Requisitos de producto

Las aplicaciones web y móvil de Geoapps-project.com están orientadas a ofrecer una accesibilidad que permita que los servicios que ofrecen, puedan ejecutarse en dispositivos con configuraciones, funcionalidades y sistemas diferentes, haciendo posible la utopía de la multiplataforma y la unificación de lenguajes de programación. Este requisito es la parte central de la estructura del proyecto, con el fin de invitar a desarrollar aplicaciones móviles multiplataforma accesibles desde cualquier dispositivo o pc.

La confiabilidad en el almacenamiento remoto, está enfocada en el marco "Cloud Computing" y debe ser analizada positivamente. Las últimas tendencias se inclinan por esta tecnología que está en auge ya que permite, entre otras cosas, acceder a la información cuando y desde donde se quiera.

Otro requisito a tener en cuenta es la fiabilidad del sistema de gestión de la base de datos. Si una aplicación es accesible simultáneamente y los datos son propietarios del usuario, es decir modificables, las transacciones deberán ser consistentes.

Como toda aplicación que se precie en los markets de aplicaciones conocidos, la sencillez es la nota predominante. Es por esto que la aplicación móvil desarrollada deben centrarse en ser sencilla e intuitiva, proporcionando una sensación de usabilidad para todo tipo de usuarios.

La escalabilidad es otro factor clave en el desarrollo de aplicaciones móviles. La tasa de actualización de las aplicaciones más importantes a día de hoy es de periodos muy cortos, hablamos de entre uno y tres meses. No desarrollar una aplicación actualizable, provocaría realizar un gasto económico innecesario y de tiempo sustancial.

2.3.2 Requisitos de proceso

Se definen a continuación las herramientas, métodos y tecnologías que se van a usar para el diseño e implementación de las aplicaciones móvil y web. La captura de requisitos se hizo atendiendo a los requisitos funcionales y de producto, realizando un análisis exhaustivo de las herramientas y tecnologías de mercado.

2.3.2.1 Software y tecnologías para la captura de requisitos, análisis y diseño

El lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML servirá en este proyecto como herramienta para modelar los procesos de negocio y funciones de sistema, el modelo de clases, el esquema de base de datos y los componentes de software reusables. Se utilizará la versión 2.2 que incluye 14 tipos de diagramas diferentes, aunque no todos se usarán en este proyecto.

Acerca de las herramientas disponibles para el desarrollo de UML, existen multitud en la actualidad, con diferencias funcionales entre ellas. En este caso, se ha optado por utilizar la herramienta Magic Draw UML Enterprise en su versión 17.0. Entre sus ventajas cabe destacar por encima de los demás su compatibilidad con el IDE de desarrollo, Eclipse, y con la máquina virtual java, el cual es el lenguaje de programación en que se basa la herramienta que posteriormente se explicará para compilar las aplicaciones en modo multiplataforma.

Estas características y su uso habitual en las asignaturas de Ingeniería de Software I y II cursadas durante la carrera, han hecho decidirme por utilizar esta solución.

2.3.2.2 Software y tecnologías utilizadas en la construcción de la aplicación móvil

La aplicación móvil desarrollada "GeoRemind-Me" está implementada en Android, pero se ha desarrollado en otro entorno diferente al habitual.

Primero, explicar que Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles que permite programar en un framework Java aplicaciones sobre una máquina virtual Dalvik, en un marco de desarrollo OpenSource. Su elevada utilización y su entorno de programación libre, condujo a explorar nuevas soluciones para mejorar el desarrollo de programación de las actuales, encontrando así un nuevo framework que permite crear aplicaciones multiplataforma bajo tecnologías de desarrollo Web.

La solución encontrada al respecto es PhoneGap, en su versión 2.4.0. Se define como un framework para el desarrollo de aplicaciones nativas en sistemas operativos móviles, haciendo uso de tecnologías web como HTML5, CSS3 y JavaScript. PhoneGap permite desarrollar aplicaciones para los siguientes sistemas operativos: Android, iOS, Windows Phone, BlackBerry OS, Web OS, Symbian y Bada como se puede observar en

la Figura 2-4. Inicialmente, fue desarrollado por Nitobi, pero posteriormente Adobe anunció oficialmente su adquisición pasando así al control del gigante del software, el gran interesado en la evolución HTML5.

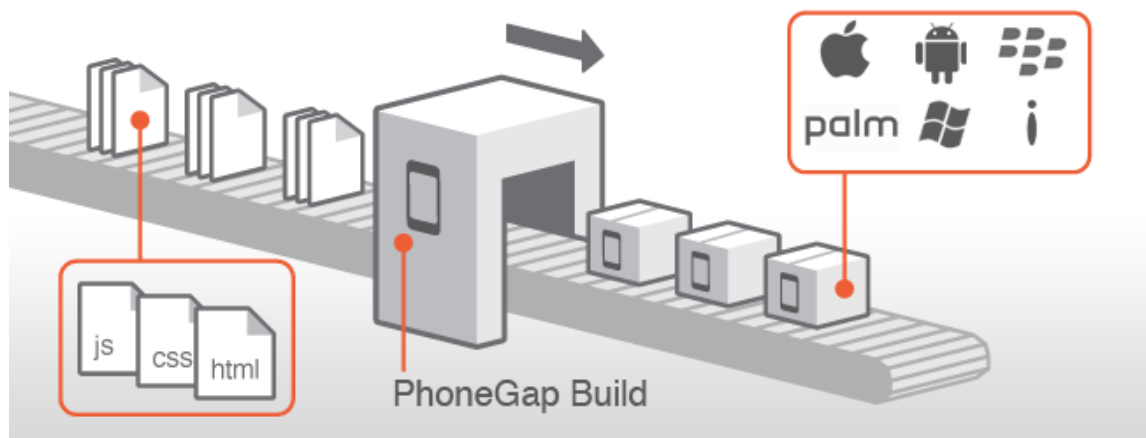


Figura 2-4. PhoneGap Build. Abstracción de uso del framework.

La decisión en la elección de la tecnología a implantar fue relativamente sencilla debido a la atronadora cantidad de virtudes que aporta esta con respecto a las habituales y ya conocidas. Entre esas virtudes destacan: el precio (gratis), su capacidad multiplataforma, su estilo de programación moderno puesto que usa HTML5, su rapidez de compartición, la cómoda maquetaación en diferentes sistemas y la sencillez de implementación en los IDEs más habituales de programación.

El IDE por defecto para desarrollar aplicaciones Android y por extensión Java es Eclipse. NetBeans es otro entorno factible para ello, pero Eclipse proporciona una interfaz mucho más desarrollada que se ajusta a las necesidades que requiere implementar una aplicación Android. Además de esto, el plugin de desarrollo ADT y el SDK Manager de Android están dispuestos para ser utilizados con Eclipse. No obstante, existen también adaptaciones para NetBeans funcionales pero no son aún tan fiables como los del entorno utilizado en este proyecto.

2.3.2.3 Lenguaje de desarrollo utilizado en la implementación de la aplicación móvil

Del apartado anterior, comprendemos que PhoneGap es un framework que permite crear aplicaciones móviles mediante tecnologías web. Esto obliga a buscar soluciones al respecto en cuanto a maquetaación, uso de funciones básicas, implementación y almacenamiento en el mercado.

La tecnología web con mejor rendimiento, más popular, y con una mayor evolución, implementación y desarrollo es el conjunto de tres lenguajes web que se suman para dar el resultado óptimo. Estas son HTML5 + CSS3 + JavaScript. Una web que presente estas tres tecnologías hará de ella una web más rica, completa, interactiva, atractiva e interesante que las que se desarrollan hoy día debido a que ofrece posibilidades infinitas con gráficos e imágenes vectoriales, fotos, videos y otros elementos. Para una mejor comprensión, se describen brevemente estas tecnologías:

- HTML5. La última versión de HTML, el cual constituye el lenguaje básico en que se construyen los sitios web. Extrapolado a un ejemplo, en una casa sería su estructura desde los cimientos al tejado. La última versión de este lenguaje incorpora soluciones tipo Flash para dotar de movimiento a las webs dinámicas.
- CSS3. También la última versión de CSS, definido como el lenguaje que se utiliza para dar forma a los sitios web. Extrapolado al mismo ejemplo anterior, en este caso el CSS constituiría la pintura, los muebles y la finalización estética de la casa.
- JavaScript. Lenguaje de programación que permite al cliente hacer cambios funcionales en la web y dotarle de dinamismo. También es utilizado para generar funciones específicas en la web, a modo de funciones java. Siguiendo con el mismo ejemplo, este lenguaje sería la disposición de los muebles en la casa.

Ahora bien, queda explicar por qué usar esta tecnología para desarrollar una aplicación Android, o más bien, multiplataforma. El hecho es que Adobe, con PhoneGap, pretende unificar los lenguajes de desarrollo para dispositivos móviles ofreciendo la ventaja multiplataforma con motivo de impulso de esta tecnología web, en la que están interesadas las compañías más potentes de la red.

En definitiva, la aplicación móvil se desarrolló como si de una página web se tratará con algunos matices que paso a comentar. El cuerpo de la aplicación, está desarrollado como una página web y con eso quiero decir su diseño y relativa programación interna. A parte, para acceder al sistema nativo de cada dispositivo móvil y sacar su máximo rendimiento utilizando una aplicación lo más similar a sus competidores nativos, existen APIs y Plugins de PhoneGap que permiten acceder a hardware del dispositivo y realizar acciones como si de una aplicación nativa se tratase. Esto permitirá crear aplicaciones generales multiplataforma, que posteriormente podrán ser adaptadas con mucha facilidad a cada sistema móvil.

2.3.2.4 Técnica de desarrollo de la aplicación móvil

La aplicación se desarrolla en un entorno híbrido entre aplicación nativa del sistema Android y aplicación web. Esta técnica de desarrollo puede resultar confusa, y por se agrega la Figura 2-5 que complementa la siguiente explicación.

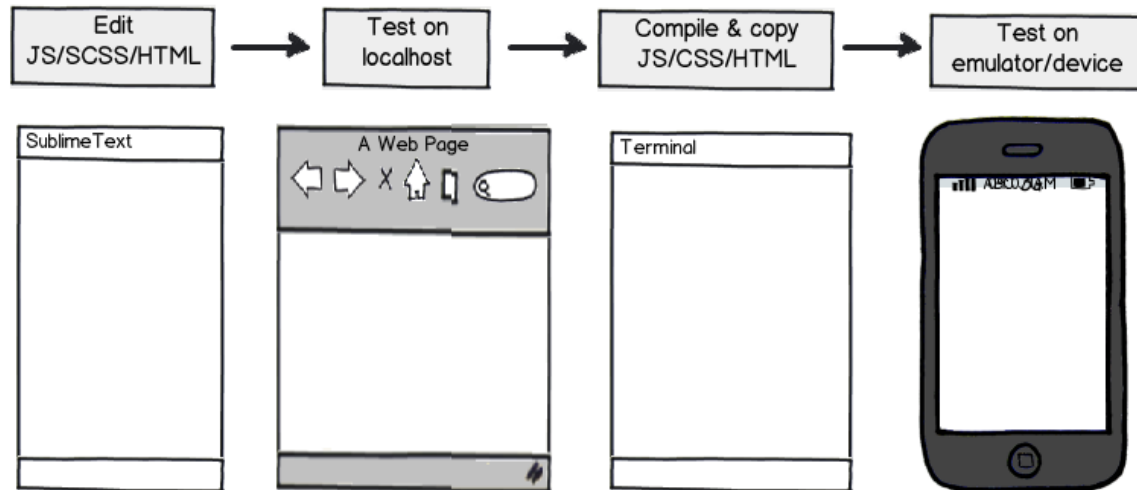


Figura 2-5. Proceso de desarrollo de la aplicación móvil.

Los dos pasos iniciales son simples y se ciñen a la técnica habitual de desarrollo en entornos web. Edición de estructura HTML, creación de hoja de estilos CSS, uso del lenguaje de script JavaScript y test en navegadores web montados sobre un servidor en localhost. Los dos siguientes se integran ya en el proceso de adición de uso hardware cuyo entorno de desarrollo implica el uso de la herramienta Eclipse con su simulador Android SDK, para probar funcionalidades únicas en cada uno de los sistemas móviles.

2.3.2.5 Software y tecnologías utilizadas en la construcción de la aplicación web

La aplicación web está desarrollada bajo un gestor de contenidos. Un gestor de contenidos (CMS) es la aplicación que permite publicar y editar el contenido de la web desde un sencillo panel de control. Cabe destacar que el diseño de la web es independiente del gestor que utilices, siendo el conjunto de las tres tecnologías descritas en el apartado anterior las que se ocupan de ello (HTML5+CSS3+JS). Por tanto, la función de un CMS es la ofrecer flexibilidad y facilidad a la hora de implementar funcionalidades personalizadas. En lo referente a soluciones o frameworks que implementen este tipo de gestores podemos encontrar tres clásicos: Wordpress, Joomla! y Drupal. Salvo excepciones, el factor más importante en la elección de casi

cualquier tecnología es su popularidad, sobre todo si es de desarrollo libre. Esto es garantía de fiabilidad y profesionalidad en el desarrollo del CMS.

Es por esto que el CMS utilizado en el proyecto es Wordpress. Es usado por el 64,96% de desarrolladores web que utilizan CMS en sus aplicaciones web, lo cual indica su notoriedad en el entorno, véase Figura 2-6. Su programación es modular y brillante para el poco código utilizado. Además, su extrema sencillez y la documentación existente al respecto hacen de este gestor el más fácil de entender. Pero aún existe una ventaja más que lo hace ser la elección de este proyecto, y es que es el mejor en lo que a posicionamiento en buscadores (SEO) se refiere sin menospreciar, por supuesto, su facilidad de gestión.

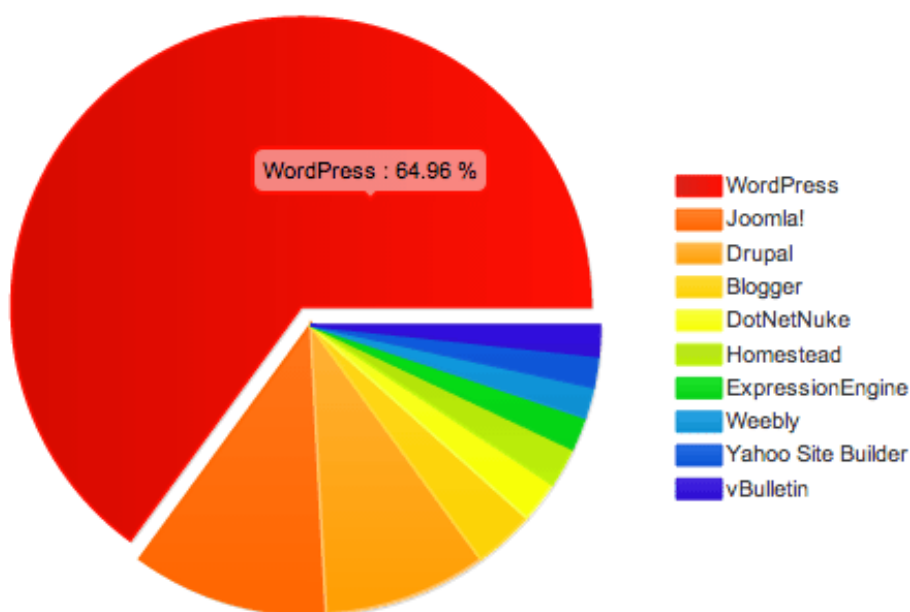


Figura 2-6. Popularidad de Wordpress en el sector CMS.

2.3.2.6 Gestor de bases de datos, introducción a la nube y al contenido web dinámico

Uno de los puntos más importantes de este proyecto es donde y de qué forma se almacenan los datos describiendo a su vez cómo interactúan estos con las aplicaciones desarrolladas.

En primer lugar hay que contestar de qué forma se almacenaran esos datos. En esto interviene la elección de un sistema gestor de bases de datos (SGBD) que es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos, además de proporcionar herramientas para añadir, borrar, modificar, analizar, mantener la seguridad de los datos, administrar el acceso a usuarios etc. Las herramientas más conocidas disponibles que podemos

utilizar para implementar el gestor son ORACLE, SQL Server, Postgre SQL y MySQL. En este proyecto se ha utilizado MySQL debido a que es GPL (General Public Lincence) y no tiene costo de licencia. Además de ser portable y ligero, está íntimamente relacionado con la plataforma de gestión phpMyAdmin que implementan el 90% de servidores web económicos del mercado.

En segundo lugar, hay que definir dónde se guarda la información. Uno de los impedimentos de los dispositivos móviles es su reducido tamaño en disco para almacenar información y datos relativos a las aplicaciones cuando el espacio usado suele ser utilizado para almacenar tipos de datos multimedia. Además de esto, centralizar la información en internet permite poder acceder a esa información desde cualquier dispositivo, siendo este el principal objetivo. Actualmente, la corriente del Cloud Computing y el Cloud Storage está teniendo un impulso manifiesto y se ha propuesto en el desarrollo de este proyecto. Como indica esta tecnología, el objetivo es ofrecer a los usuarios la capacidad de disco necesaria para su almacenamiento virtualmente ilimitado y ofrecerle servicios web para acceder a la información en la red. Para esto se ha desarrollado una API propia mediante código PHP, que permite a la aplicación móvil acceder al servidor donde están almacenados los datos, y mediante estas funciones propias de la API, poder acceder a la información requerida por la aplicación y devolver unos resultados esperados. Así ambas aplicaciones accederán a los datos de internet para recoger los datos, modificarlos en su caso y volverlos a guardar al finalizar el uso de la aplicación, proporcionándonos seguridad, autoría de los datos y posibles recuperaciones antes desastres y backups automáticos.

Por tercer y último lugar, es necesario saber cómo trabajar con peticiones a base de datos en tiempo real. El primer problema que encontramos es el hecho de que HTML es una estructura estática e inflexible. Para ello es necesario utilizar una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en la aplicación móvil o navegador mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas. Sabiendo esto, la tecnología que más se ajusta a nuestras necesidades es AJAX, que está basado en JavaScript siendo este uno de los lenguajes mencionados en el desarrollo del diseño de las aplicaciones móvil y web. AJAX es una combinación de 4 tecnologías ya existentes:

- XHTML o HTML.
- DOM. Lenguaje de scripting para mostrar e interactuar dinámicamente con la información presentada.

- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web.
- JSON y HTML pre formateado, es el formato usado generalmente para la transferencia de datos solicitados al servidor.

Con la mezcla de estas tres tecnologías, podremos acceder dinámicamente a datos de la bases de datos remota desde cualquiera de las dos aplicaciones desarrolladas.

2.3.2.7 Software y técnicas utilizadas para el desarrollo gráfico de componentes web y móvil

El diseño de una herramienta, es una de las características más importantes que definirán gran parte de la funcionalidad, usabilidad y atractivo en las últimas fases de vida del producto. El diseño se divide en dos partes:

- Maquetación y diseño estructural. Es el marco en el que desarrollaremos posteriormente nuestra aplicación. Se corresponde con la elección espacial de los contenidos y con la separación completa del contenido del archivo HTML de las definiciones del aspecto CSS. Véase un ejemplo en la Figura 2-7.

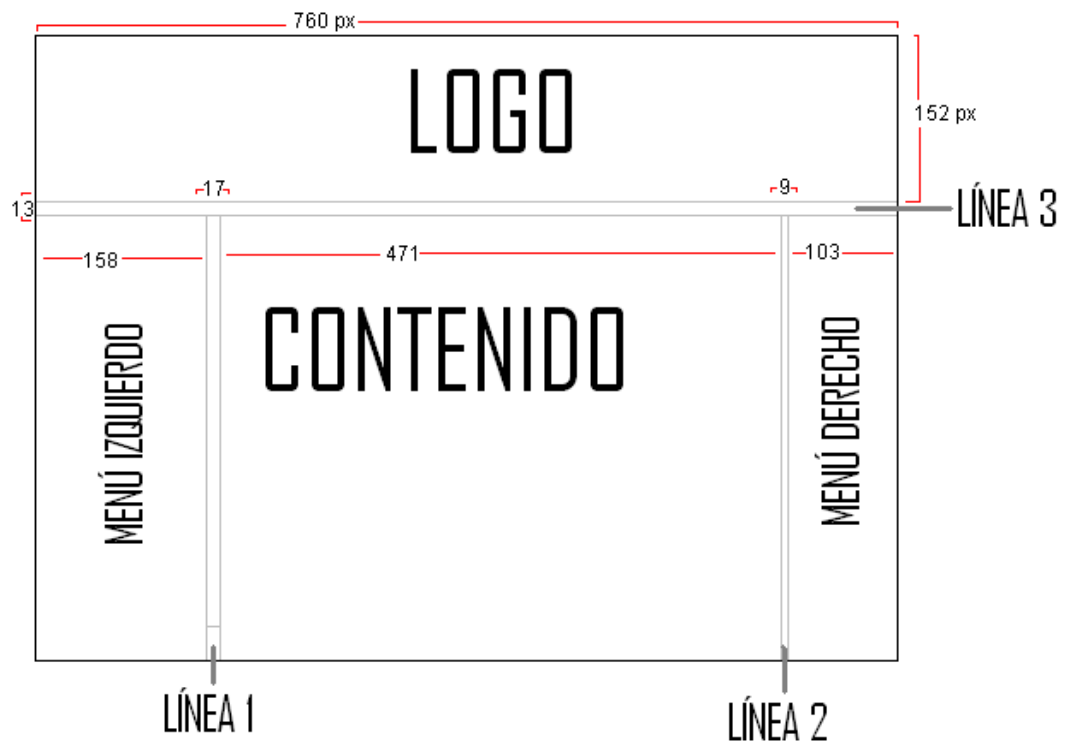


Figura 2-7. Maquetación HTML

- Diseño gráfico de contenidos. El diseño gráfico utiliza signos, imágenes y fotografías para estructurar y organizar los elementos con objetivo de dar fuerza y claridad mediante las técnicas, recursos y gustos que marquen el momento. Esta área de la informática, cuida los siguientes aspectos:
 - Imagen corporativa: Logos, marcas.
 - Comunicación visual: Señales, pictogramas.
 - Tipografía: Fuentes y tipos de letras.
 - Diseño publicitario: Carteles, folletos, catálogos.
 - Diseño web: páginas web.
 - Diseño multimedia: imagen 3D, animaciones.

Más del 50% de cada producto que desarrolla es evaluado por los usuarios en relación a su entorno visual y no por su funcionalidad, generando una primera impresión al usuario, pocas veces modificable. De esto se ocupa de forma general un desarrollador gráfico, pero su cooperación a veces supone un gasto no asumible en el diseño del proyecto, lo que obliga al informático a tener nociones y así implementar un producto aceptable en este sentido. El diseño es entonces una disciplina fundamental puesto que hoy no se concibe nada sin un diseño, sin un esfuerzo por pensar creativamente en la comodidad y estética de lo que se va a usar.

El software utilizado para el diseño gráfico de todos los componentes son los siguientes:

- Adobe Photoshop CS4. Herramienta en forma de taller de pintura y fotografía que trabaja sobre un "lienzo" y que está destinado a la edición, retoque fotográfico y pintura a base de imágenes de mapa de bits
- Adobe Illustrator CS4. Herramienta de creación y manipulación vectorial en forma de taller de arte que trabaja sobre un tablero de dibujo, conocido como "mesa de trabajo" y está destinado a la creación artística de dibujo y pintura para ilustración

Existen tecnologías que facilitan la implementación del diseño agilizando el trabajo del desarrollador. En este proyecto se ha utilizado el framework JQuery que está basado en JavaScript y que aporta una serie de funciones o códigos para realizar tareas habituales y que nos simplificará la programación en este lenguaje. Más detalladamente, nos ofrece una infraestructura con la que tenemos mucha mayor facilidad para crear aplicaciones complejas del lado del cliente, como por ejemplo, en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax etc.

Capítulo 3. Diseño e Implementación

En este capítulo se describe el diseño y la implementación de cada módulo que compone la aplicación. Se comenzará con el diseño a alto nivel y a continuación se explicarán los detalles más relevantes de la implementación. A lo largo del capítulo se mostrarán ilustraciones que ayuden a entender mejor el funcionamiento interno del sistema implementado.

3.1 Visión general del sistema

La Figura 3-1 muestra una visión general de la estrategia de desarrollo tomada en este proyecto, que posteriormente se analizará paso por paso. Para entender el concepto general del sistema, podemos observar dos elementos que actúan como aplicativo en el lado del cliente. Estos son el dispositivo móvil y el sistema web. Por un lado, el dispositivo móvil realiza peticiones a una API o Web Layer alojada en el servidor que a su vez conecta con el sistema de almacenamiento de la base de datos. Por otra lado, y de forma análoga, tenemos el sistema web que mediante un gestor de contenidos, que actúa como intermediario, podrá acceder al mismo sistema de almacenamiento al que accede la aplicación móvil. De esta manera, se permitirá a los usuarios poder acceder desde cualquier plataforma a los datos requeridos.



Figura 3-1. Diseño de arquitectura a alto nivel

3.2 Diseño de la arquitectura del sistema

El diseño general de la arquitectura requirió el análisis exhaustivo de los requisitos de la aplicación, fijando la atención no solo en la programación necesaria para que esta funcionara, si no también teniendo muy en cuenta la funcionalidad y el perfil de los usuarios potenciales.

En este proyecto se desarrollan dos aplicaciones, el sitio web y la aplicación móvil. A continuación se detallará cada uno de los diseños utilizados para desarrollar ambas aplicaciones.

3.2.1 Estilo y patrón Arquitectónico

El patrón de "Arquitectura en tres capas" es el patrón arquitectónico escogido en este proyecto. El motivo principal de esta elección es que las distintas unidades lógicas presentes en la aplicación se separen y posean estructuras bien planteadas. La figura 3-2 se corresponde al patrón "Tres capas" utilizado.

- La capa de presentación o Web Client layer. Se encarga de proveer una interfaz entre el sistema y el usuario. Básicamente, se responsabiliza de que se le comunique información al usuario por parte del sistema y viceversa, manteniendo una comunicación exclusiva con la capa de negocio que se describirá a continuación.
- La capa de negocio o Middle layer. Es la capa que contiene los procesos a realizar con la información recibida desde la capa de presentación así como gestionar las respuestas que se han de enviar a la capa de presentación. Así mismo interacciona con la capa de datos para recuperar o almacenar los datos necesarios para cada proceso.
- La capa de datos o Database Layer. Esta es la responsable de acceder al gestor de bases de datos y modificar, insertar, borrar o recuperar los datos solicitados por la capa de negocio.

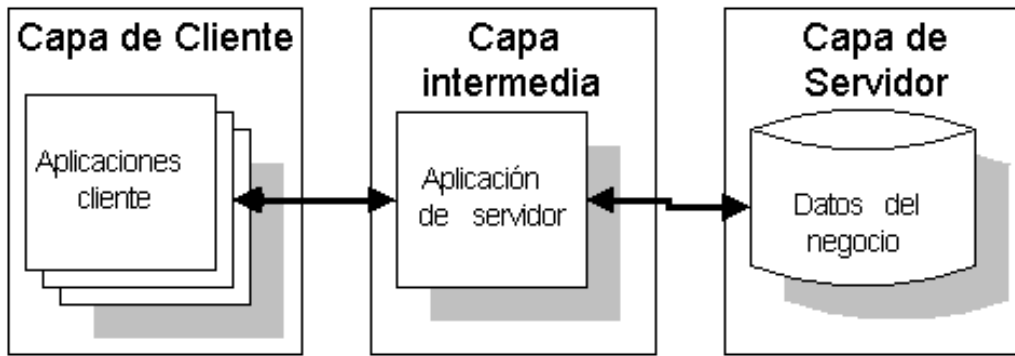


Figura 3-2. Diagrama del patrón arquitectónico “Tres capas”

Las ventajas de este diseño arquitectónico son:

- Separación clara de la interfaz de usuario de la lógica de la aplicación. Esta separación permite tener diferentes presentaciones accediendo a la misma lógica.
- La redefinición del almacenamiento de información no tiene influencia sobre la presentación.
- Los objetos definidos en la capa de negocio pueden brindar servicios (lógica de la aplicación) a otras aplicaciones.

3.2.2 Arquitectura lógica del sistema

La Figura 3-3 muestra la arquitectura lógica del sistema que se conforma de la siguiente forma:

- Capa de Presentación. Es el responsable de otorgar una interfaz gráfica, diseño de formularios, y métodos de validación de la interfaz. En este caso, mediante HTML, CSS3, JS y jQuery se desarrolla la interfaz gráfica de la aplicación móvil y el sistema web.
- Capa de negocio. Prepara y gestiona operaciones complejas que marcan las pautas lógicas del sistema. En la aplicación móvil, el modulo AJAX/JS se encargará de la gestión del funcionamiento de la aplicación y de las peticiones al servidor web.
- Capa de datos. En ella se diseña el mapeo objeto-relacional que permite persistir los datos. En el caso de la aplicación móvil, se ha diseñado una API o Web-Layer que permite su acceso a modo de servicio web. En caso del sistema web, el CMS será quien lo gestione.



Figura 3-3. Aproximación lógica del sistema

3.2.3 Distribución de los componentes lógicos

Es importante destacar que los límites entre las capas son lógicos, por lo que es posible ejecutar las tres capas en la misma máquina. Lo importante es que el sistema está claramente estructurado y que hay una buena planificación de los límites entre las diferentes capas.

La distribución lógica que marca el proyecto es la que sigue la pauta del patrón de distribución orientada a la web. Esta es un caso particular de la distribución del componente de la capa de presentación ya que la particularidad de este caso es la utilización de tecnología web en la capa de presentación. Se distribuye el componente en dos subcomponentes, uno reside en el cliente, y el otro en el servidor.

- El componente a nivel del cliente es un cliente web tradicional (web browser).
- El componente a nivel del servidor consiste en un servidor web capaz de generar contenido dinámico, junto a la implementación de este contenido.

Esta disposición orientada a la web, se puede ver reflejada en la Figura 3-4 que se muestra a continuación.

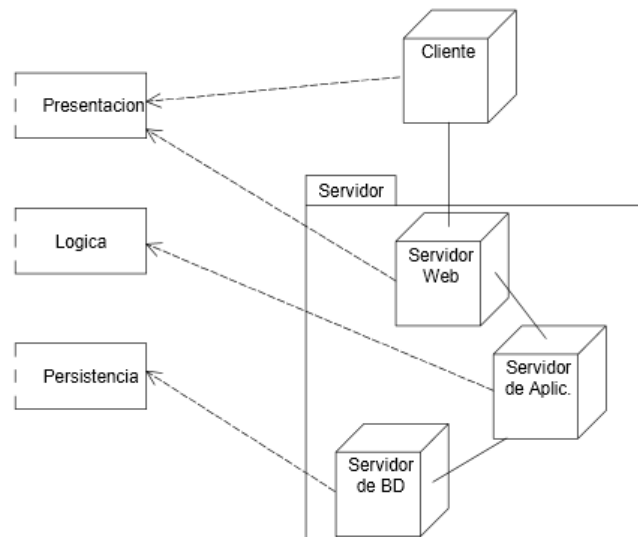


Figura 3-4. Patrón de distribución lógica de los componentes

Por tanto según el patrón anteriormente explicado se considera que existen los siguientes componentes lógicos:

- Servidor. Componente que comprende el servidor web, la base de datos, el CMS Wordpress y la API desarrollada para la comunicación entre la aplicación móvil y la base de datos.
- Aplicación. Componente dedicado a la aplicación móvil
- Web. Componente que comprende el sitio web.

La siguiente Figura 3-5 se corresponde a la distribución de los componentes lógicos del sistema.

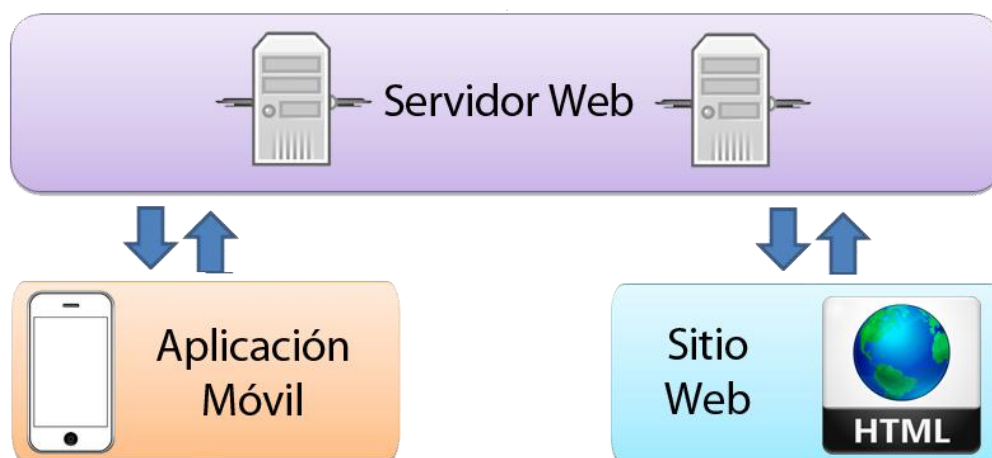


Figura 3-5. Distribución lógica de los componentes

3.3 Arquitectura, diseño e implementación de la aplicación móvil

El diseño de la arquitectura de la aplicación es una de las primeras investigaciones previas al desarrollo de la aplicación. Un resumen de las existentes, como muestra la Figura 3-6, y sus características son las siguientes:

- Acceso de Navegador. Escritos en HTML, JS y CSS. Desarrollo muy rápido y barato, pero lejos de la potencia que otorga un sistema nativo.
- App Híbridas-Web. Escritos en HTML con librerías de ejecución de Worklight para ejecutarse en una Shell nativa.
- App Híbrida-Mezcla. El usuario añade al código web, código nativo con necesidad única y la experiencia de usuario. Es la utilizada en este proyecto.
- App Nativa. Específica de cada plataforma. Requiere conocimiento experto, caras por naturaleza y con ciclos largos de desarrollo. Por otra parte suelen otorgar una experiencia rica al usuario.

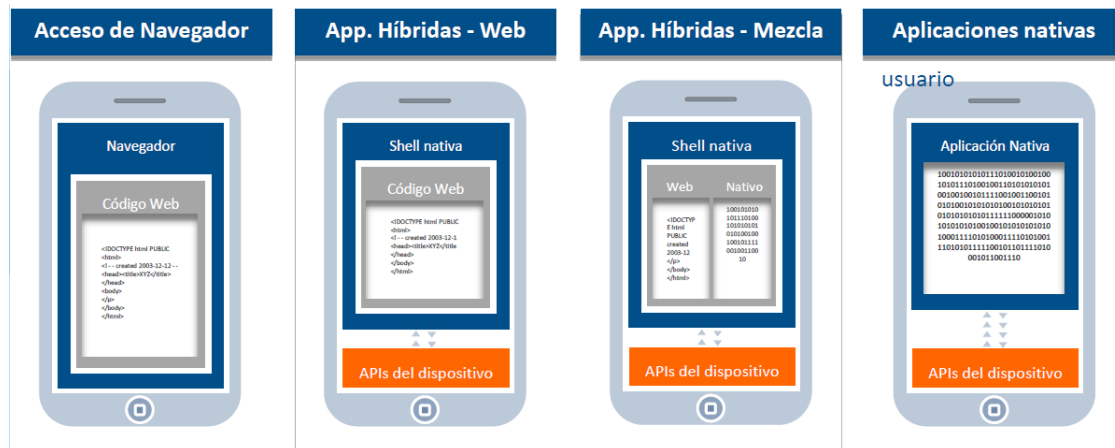


Figura 3-6. Tipos de Arquitecturas de desarrollo en aplicaciones móviles.

Las aplicaciones híbridas están construidas de tal manera que son capaces de ejecutar código nativo y de interpretar contenido web, de forma que el usuario ve conjuntamente información de los dos tipos. Además, las páginas HTML pueden provenir de cualquier servidor web o residir en el propio dispositivo. De esta forma disponemos de lo mejor de cada arquitectura. Por ejemplo, una aplicación híbrida permite, como en el caso de nuestra aplicación móvil, presentar en pantalla una página HTML5 sobre la que se muestra la posición actual proporcionada por el GPS. El

coste de desarrollo es similar al de las aplicaciones web y nos permite preparar las aplicaciones para trabajar sin necesidad de estar conectadas al servidor, es decir offline.

Las estrategias que se usan en aplicaciones híbridas es básicamente la que todos los proveedores de estos entornos han seguido para diseñar el entorno de ejecución.

En la capa inferior, véase Figura 3-7, se encuentra un Contenedor de Dispositivo (DC o Device Container) escrito con el SDK (aplicación nativa). Éste proporciona a las capas superiores acceso a las funciones del dispositivo (cámara, acelerómetro, etc.) y una interfaz común (DII o Device Integration interfaces) para poder abstraer esta capa y hacer a las demás independientes del dispositivo. El creador del framework deberá construir este contenedor para cada sistema operativo al que desee dar soporte.

Sobre el DC encontramos el Web Container, que en esencia se trata de un Navegador Web con capacidades para renderizar HTML y ejecutar JavaScript. Aquí el acceso a los servicios que dé la cámara, GPS, agenda, etc. los encontramos disponibles a través de funciones JS o lenguaje de marcas, dependiendo de las tecnologías ofrecidas por del framework.

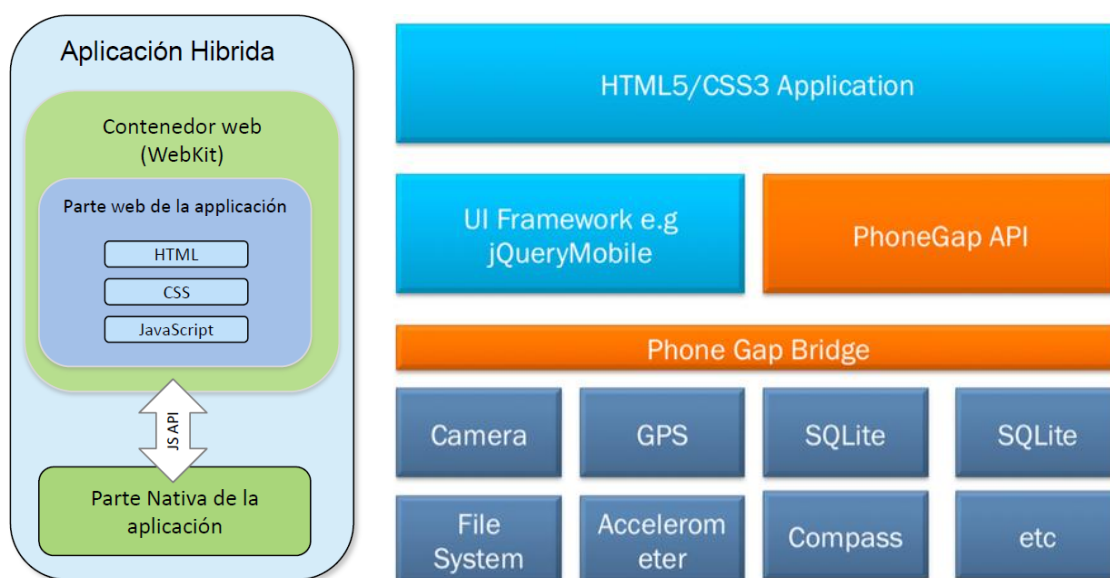


Figura 3-7. Arquitectura en el desarrollo de la aplicación móvil.

3.3.1 Implementación JS API

Como se muestra en la anterior Figura 3-7, el entorno de desarrollo PhoneGap proporciona dos componentes sobre los que implementar nuestra aplicación. En este apartado se explicará la implementación del componente PhoneGap API que centra su desarrollo en el entorno portable basado en tecnologías web.

Es importante precisar previamente a observar la Figura 3-8, que existen complicaciones en el desarrollo del modelado de un diagrama de clases debido a que esta implementado en un lenguaje no orientado a objetos. Por un lado, JavaScript no es un lenguaje orientado a objetos sino que es, en realidad, un lenguaje de programación basado en herencia de prototipos. Este está constituido por una sintaxis similar al lenguaje Java del que hereda parte de sus funcionalidades, pero que solo actúa como fachada que hace parecer a este como un lenguaje orientado a objetos. Por otro lado, un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos mostrando el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Con lo descrito anteriormente, es complicado llegar a la conclusión de que JavaScript pueda modelarse mediante UML. Aun así, existe una razón por la cual JavaScript intenta esconder su verdadera naturaleza detrás de una fachada clásica de POO, y es la complejidad que representa la comprensión de un nuevo paradigma particular a este lenguaje. La idea de disfrazar el uso de JS bajo la apariencia típica del clásico POO no es nueva, puesto que de forma convencional se enseña a los nuevos programadores todos los conceptos clásicos de Orientación a Objetos y modelado de Software mediante UML, dejando de lado a otras posibles soluciones.

Es por esto, que en post de una comprensión rápida y práctica del diseño que se ha implementado posteriormente en la implementación y porque ya se ha utilizado en otras ocasiones, se ha decidido modelar mediante UML. Queda aclarado por tanto que el diseño ofrecido en el diagrama de clases de la Figura 3-8 no define de forma estricta la implementación real, pero que otorga muchas de las ventajas que ofrece UML, para obtener una comprensión rápida y directa de lo que es su diseño conceptual.

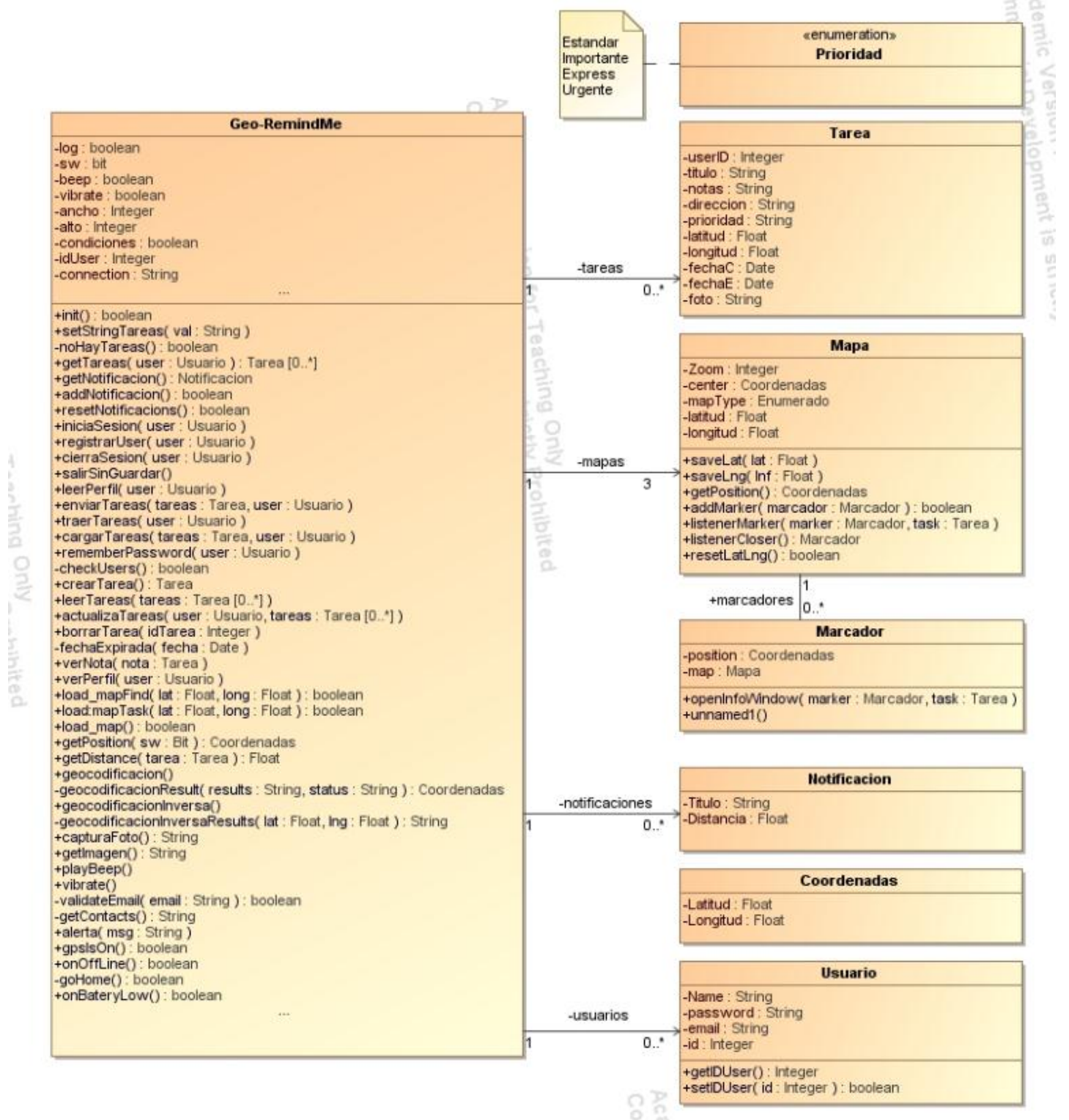


Figura 3-8. Diagrama de clases JS API

3.3.2 Implementación Nativa

En este apartado se explicará la implementación del componente nativo que centra su desarrollo en el sistema móvil Android. Posteriormente se muestra en la Figura 3-9 el diagrama de clases JAVA desarrollado en este apartado para explicar su funcionamiento. Este diagrama de clases, se corresponde con todos aquellos ficheros Java que se utilizan como plataforma de comunicación entre el la pasarela web y la parte nativa.

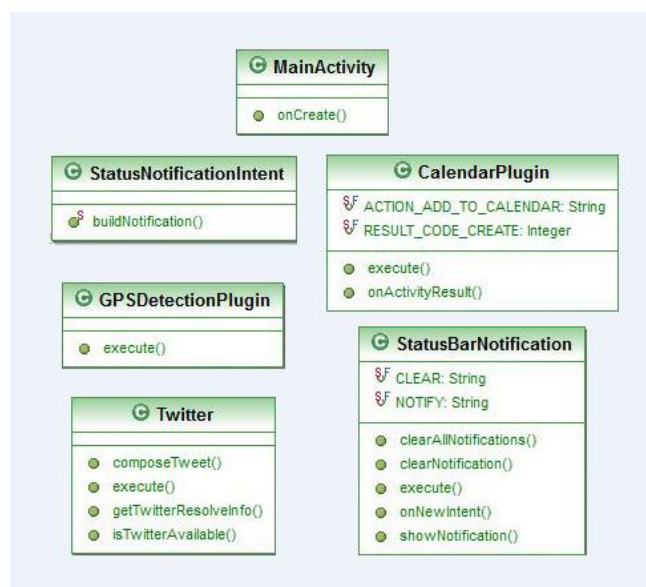


Figura 3-9. Arquitectura en el desarrollo de la aplicación móvil.

3.3.3 Acción y reacción ante eventos nativos

La primera duda que surge en el desarrollo de esta aplicación es el proceso necesario para hacer efectiva la comunicación entre un entorno web y eventos nativos del propio sistema móvil. Esa duda razonable es una de las principales utilidades que presenta el framework utilizado en este proyecto, PhoneGap. Este framework, presenta una serie de plugins o addons que permiten, según el sistema móvil objeto de desarrollo, acceder a eventos nativos.

De esta manera, cada vez que nuestra aplicación móvil reacciona ante un evento que requiere el uso de componentes hardware, como por ejemplo: el uso de la cámara, reconocimiento de voz, aplicaciones como Twitter o notificaciones de sistema en la barra de tareas; se produce un proceso de acomodación necesario que se resume en la necesidad de desencadenar eventos nativos a través de ordenes web.

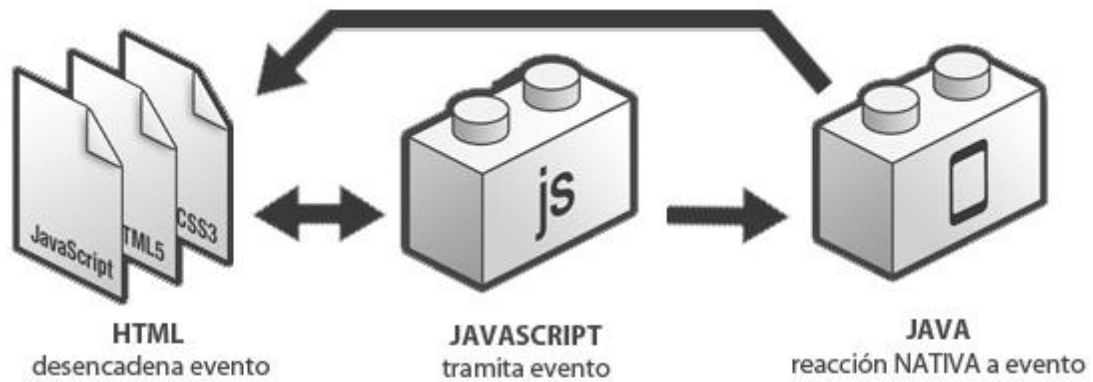


Figura 3-10. Reacción ante eventos nativos

Una explicación más explícita es la que trata de mostrar la Figura 3-10, y que se pasa a tratar a continuación:

- **HTML.** Se refiere al entorno de desarrollo web, el cual ha sido explicado previamente. Esta es la parte portable de nuestra aplicación y suele constituir el 90% del total de desarrollo. Es aquí donde se desencadenan eventos de usuario, como la pulsación de un botón, que genera la tramitación del evento de la cual se encarga el código JS de la aplicación.
- **JS.** Proporciona la lógica de negocio de la aplicación móvil. Es capaz de dirimir el significado de la información proveniente del evento HTML desencadenado y desarrollar un proceso con un objetivo definido. Este, debe diferenciar entre dos tipos de procesos:
 - **Web.** Aquellos que validan datos de usuario, utilizan operaciones aritmético-lógicas que no requieren servicios nativos. Cuando estos se producen generará directamente la respuesta al evento.
 - **Nativos.** Aquellos que requieren del uso de hardware del dispositivo físico o el acceso a datos del dispositivo internos. Si el proceso es de este tipo, se requerirá el uso de clases nativas.
- **JAVA.** Módulo de desarrollo en el lenguaje de programación específico de cada sistema móvil. En el caso del proyecto este módulo está desarrollado en JAVA debido a que es el lenguaje natural de Android, en el que están implementada la aplicación móvil. Esto da acceso a cualquier tipo de operación nativa simulando una respuesta de sistema en una aplicación de desarrollo web.

3.4 Diseño e implementación de la capa de persistencia

3.4.1 Repositorio de datos

En esta capa se encuentran almacenados los metadatos propios de la aplicación web y móvil así como los datos relativos al almacenamiento, control y gestión de operaciones que proporciona el CMS usado en la aplicación web, Wordpress.

Una base de datos constituye la parte más importante de un Sistema de Gestión de Contenidos. Y en ocasiones esa base de datos puede corromperse. Wordpress utiliza su base de datos para almacenar y recuperar información cuando sea necesario.

En total, 11 tablas correspondientes a Wordpress^[36] en versiones superiores a 2.3.x más 1 tabla relacionada con la aplicación móvil, son utilizadas en este proyecto que se pueden observar en la Figura 3-11 y que son las siguientes:

- wp_comments. Ambas tablas contienen los datos relacionados a los comentarios. wp_commentmeta se refiere a los metadatos acerca de cada comentario, estos datos son los del usuario que publicó el mensaje. En el panel de control de Wordpress, la sección en cuestión es: Comentarios.
- wp_links. Información relacionada con las links ingresados al Blogroll Manager de Wordpress. (Enlaces)
- wp_options. Tabla que almacena la configuración del usuario wp_options. (Opciones)
- wp_postmeta. Los metadatos de cada entrada en el blog se almacenan aquí. (Metadatos de entradas)
- wp_post. Tabla que almacena las entradas (posts) que son guardados en esta tabla. (Post)
- wp_terms. Tabla que recoge a que categoría se define la entrada definida.
- wp_term_relationship. Tabla que recoge asociaciones de entradas con categorías y etiquetas de la tabla wp_terms.
- wp_term_taxonomy. Contiene descripciones de la taxonomía (categoría, link o etiqueta) para los datos mantenidos en la tabla wp_terms.

Geo localización en Android

- wp_usermeta. Cada usuario tiene sus propios metadatos, que se guardan en esta tabla. (Usuarios)
- wp_users. Contiene la lista de usuarios. (Usuarios)
- app_remindme_tareas. Aquí se almacena toda la información relevante a cada tarea almacenada por el usuario. (Aplicación móvil)

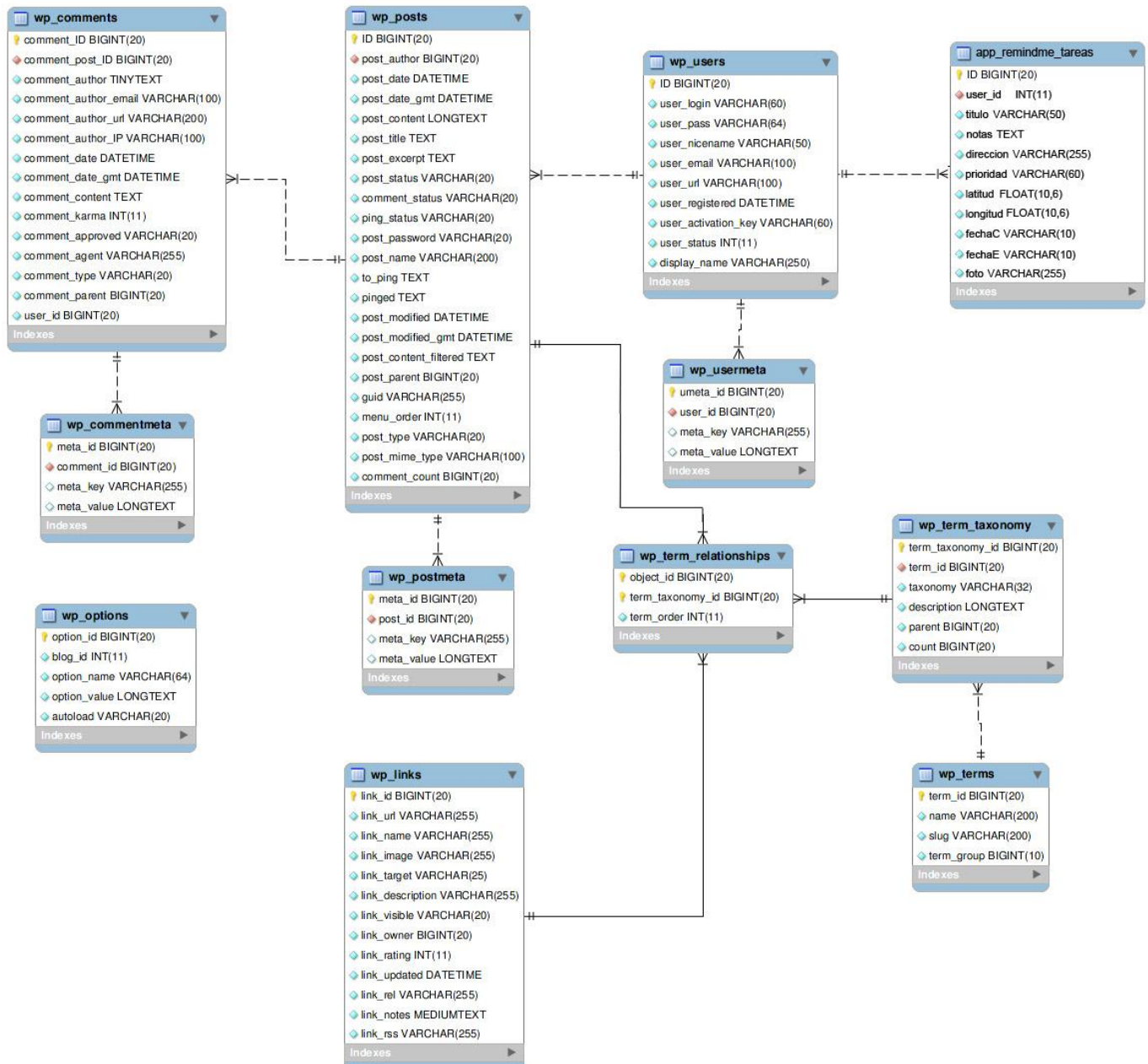


Figura 3-11. Diagrama de relación de tablas

3.4.2 API o Web Layer

Este componente se ha definido como una pasarela o interfaz de comunicación entre la aplicación móvil y el repositorio de datos de tal forma que ofrezca modularidad y seguridad al sistema. Su función es la misma que la de un servicio web, que es la de ofrecer mecanismos que permiten ejecutar funciones remotas permitiendo interoperabilidad, ubicuidad, rendimiento y soporte de mantenimiento eficiente.

Aun así, definirlo como un servicio web es un error, debido a que la definición de Web Service integra una serie de estándares de implementación que no se han llevado a cabo. Es por esto que esta capa intermedia presente en la capa de persistencia se define como API o Web Layer, la cual proporciona la capacidad a la aplicación de: cerrar/abrir sesión en la aplicación, conectar/desconectar de la base de datos, tomar datos de perfil, registrar usuario y recibir/guardar tareas como se muestra en la Figura 3-12.

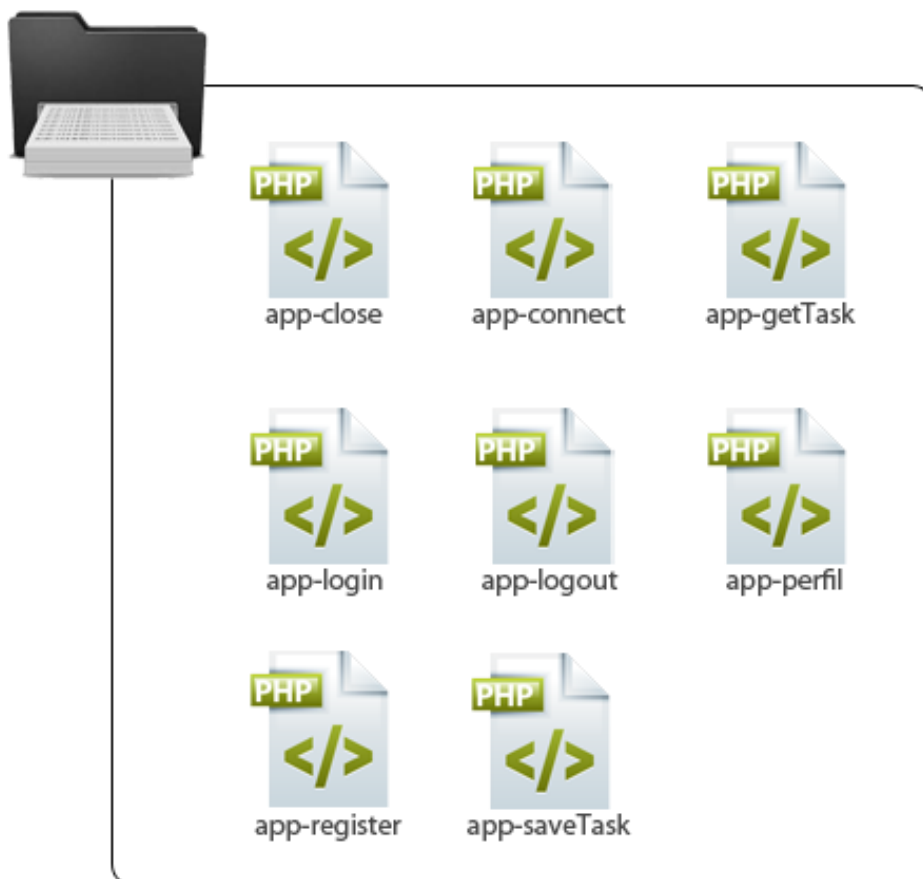


Figura 3-12. Abstracción de implementación de la Web Layer

3.5 Secuencia de eventos

En este apartado se describen las secuencias más importantes, a muy alto nivel, de las llamadas entre las diferentes capas y módulos.

3.5.1 Registro en aplicación móvil

La figura 3-13 muestra como es la secuencia de eventos en el registro de un usuario en la aplicación web.

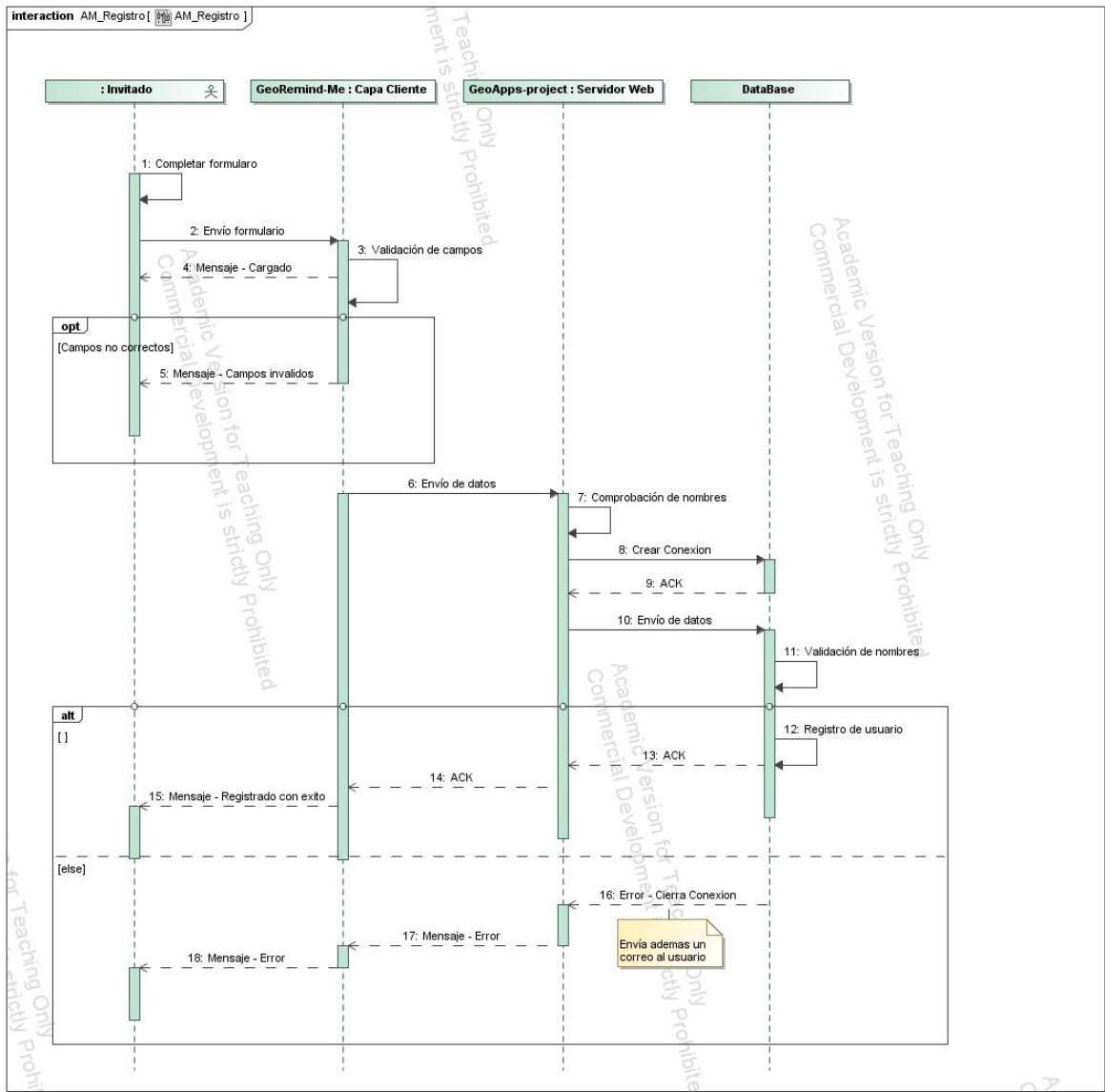


Figura 3-13. Diagrama de secuencia de petición de registro

Como se puede observar en la imagen, un usuario anónimo completa el formulario de registro para acceder a la aplicación móvil. En caso de que el formulario no esté completado debidamente, como campos por rellenar o correo según patrón incorrecto, la aplicación notificará mediante una alerta al usuario. Si la validación del formulario es correcta, la aplicación enviará los datos al Servidor web, donde se realizará la comprobación de nombres. Si existiera duplicidad en alguno de los campos enviados se notificará al usuario. En caso contrario, se procederá al registro. En caso de registrarse, se notificará al usuario mediante un mensaje y un correo al email de registro.

3.5.2 Inicio de sesión y recepción de tareas en aplicación móvil

La figura 3-14 muestra como es la secuencia de eventos en el inicio de sesión de un usuario en la aplicación web.

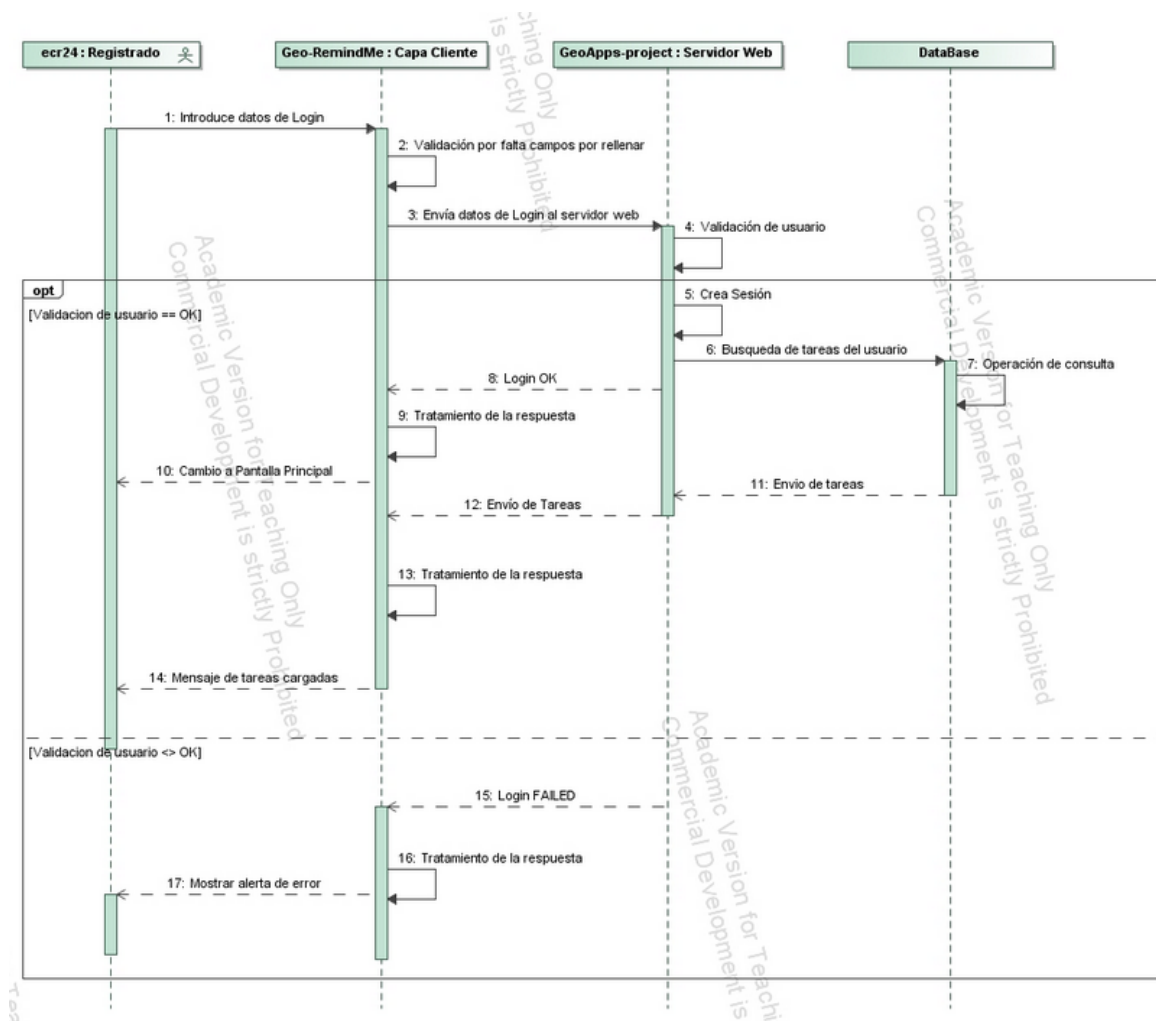


Figura 3-14. Diagrama de secuencia de petición de inicio de sesión y recepción de tareas

Como en el apartado anterior, lo primero que comprueba la aplicación es la validación de campos del formulario. En caso que la respuesta sea correcta, se procede a validar ese usuario y contraseña en el servidor web. Si se produce conexión con la base de datos, esta comprobará si tiene tareas definidas el usuario que inicia sesión. En caso afirmativo, la aplicación cargará los datos recibidos por el servidor, en caso contrario, se notificará una alerta con el error sucedido.

3.5.3 Cierre de sesión y envío de tareas en aplicación móvil

La figura 3-15 muestra como es la secuencia de eventos en el cierre de sesión de un usuario en la aplicación web.

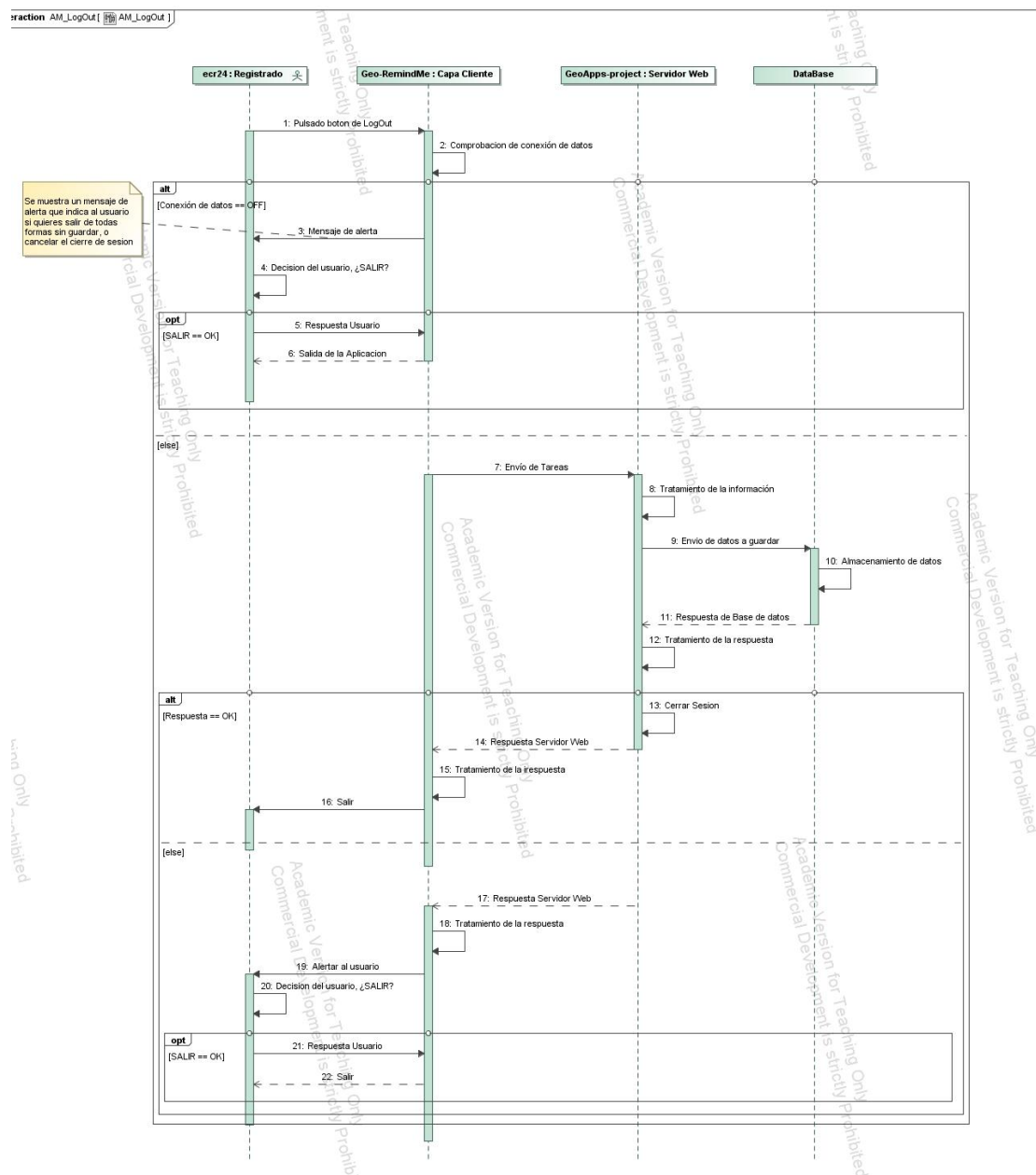


Figura 3-15. Diagrama de secuencia de petición de cierre de sesión y envío de tareas

Se trata de hacer la misma operación que antes, pero a la inversa. Es decir, si el usuario pulsa sobre el botón cerrar sesión y existe una conexión de datos activa se procederá al envío de las tareas del usuario al servidor web. Una vez en el servidor, se tratará la información para enviársela a la base de datos y almacenar las tareas. Si se produce un error durante el proceso, el usuario será notificado de ello, en caso contrario se mostrará la pantalla principal.

3.5.4 Búsqueda de direcciones mediante la API Google Maps

La figura 3-16 muestra como es la secuencia de eventos la búsqueda de direcciones a través de la API de Google Maps.

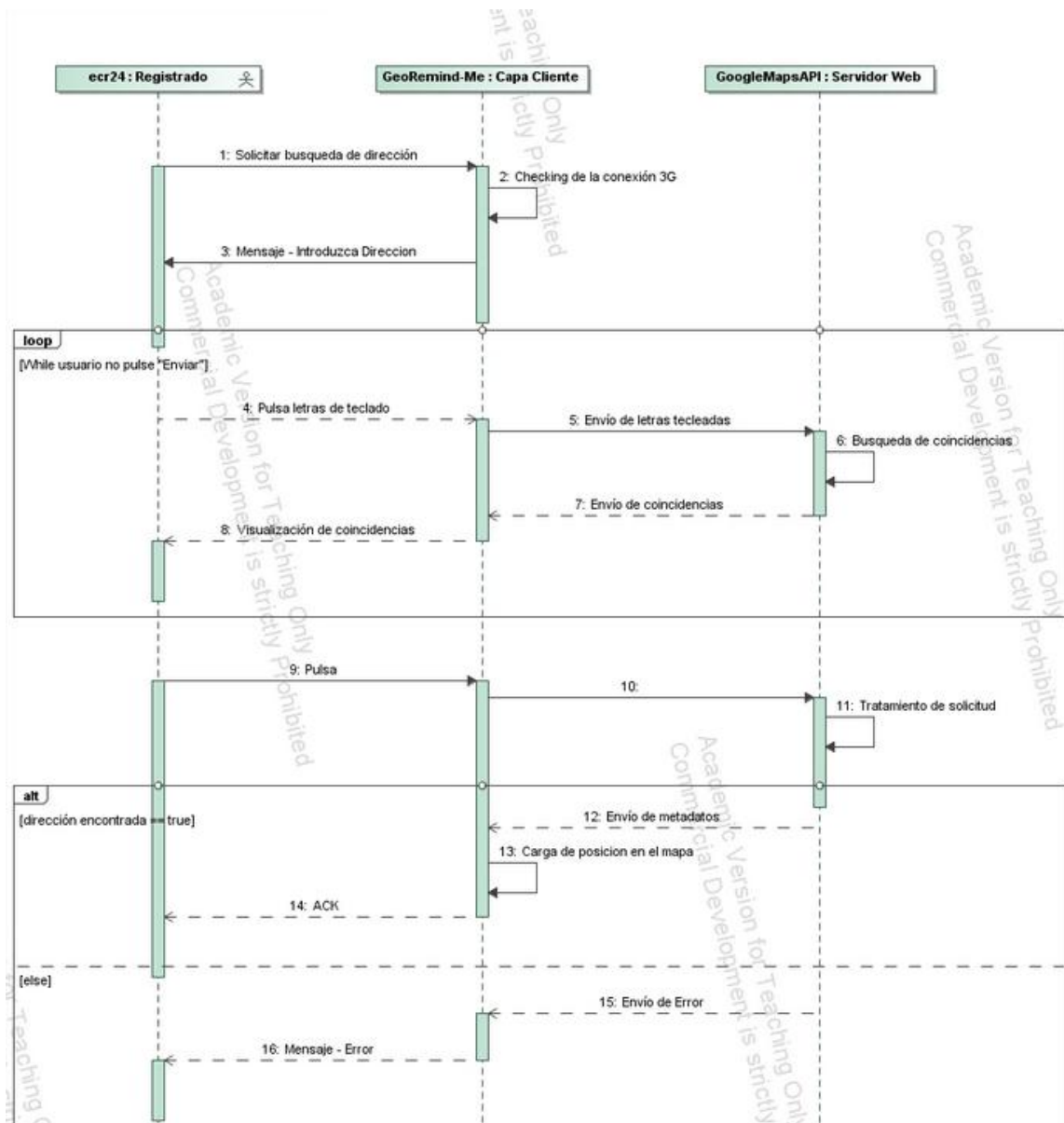


Figura 3-16. Búsqueda de direcciones mediante la API Google Maps

La API de Google Maps sirve entre otras cosas, tal y como se ve en la figura 3-16, para proporcionar coincidencias dependiendo de la inserción del usuario a modo de autocompletado. En el diagrama se observa, que el usuario solicita insertar una dirección para definir la ubicación de la tarea. Si existe conexión, mientras el usuario teclea la dirección, la aplicación envía al servidor las letras insertadas. Posteriormente, se tratan en el servidor y se envían a la API de Google Maps que responderá con ciertas propuestas de dirección. Si el usuario selecciona una dirección proporcionada por esta API, se cargará un mapa con el marcador en la dirección correspondiente. En caso contrario se alertará al usuario que la dirección insertada no existe en la base de datos.

Capítulo 4. Interfaz

4.1 Interfaz gráfica de la aplicación

Los resultados de la interfaz gráfica y su implementación final están desarrollados en el sistema operativo Android. De esta manera, la aplicación después de ser descargada como fichero comprimido “.apk” y ejecutable en Android, puede ser instalada para su utilización. Las imágenes utilizadas a continuación son sobre el dispositivo Huawei 3G Ascend con la versión Android 2.3.6. Una vez instalada, podremos iniciarla pulsando sobre el botón de la aplicación con nombre RemindMe.



Figura4-1. Versión Android y Dispositivo de implementación

En primer lugar, podremos observar la pantalla de bienvenida, que se muestra mientras se carga por completo la aplicación. Tras esto, aparecerán los términos de contrato para poner en situación al usuario que va a iniciar la aplicación. Figura 4-2.

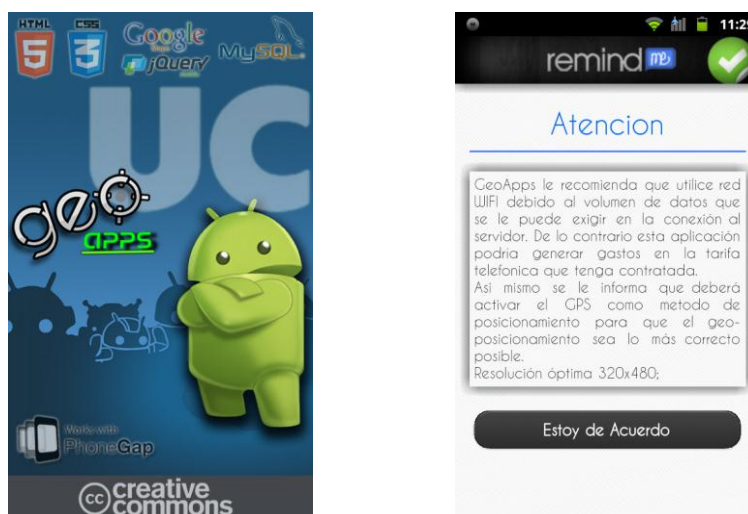


Figura4-2. Pantalla de Bienvenido y condiciones de contrato

Tras esto, aparecerá la pantalla de inicio de sesión. Es necesario para utilizar la aplicación estar registrado en la BD, pues será donde se almacenen los datos. Existe una pestaña con nombre “Recuérdame” que autocompletará los campos del último usuario que ha accedido a la aplicación. Véase Figura 4-3.

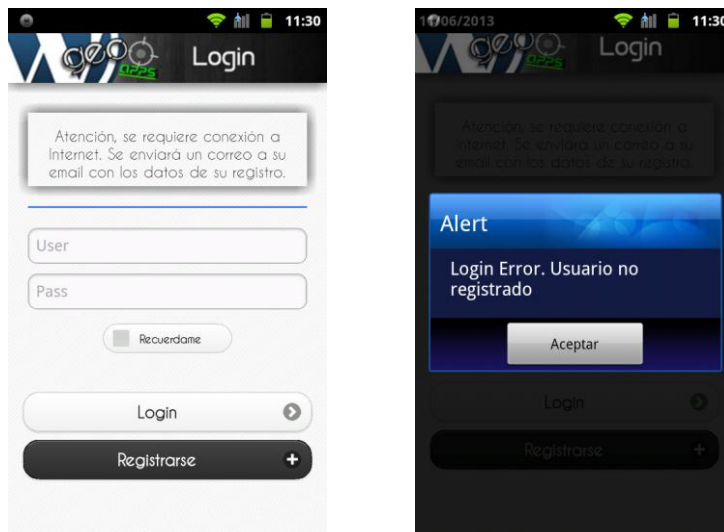


Figura4-3. Pantalla de Inicio de Sesión

En caso que el inicio de sesión produzca algún tipo de error, ya sea un fallo en la transacción, en el envío de datos, o en la autenticación del usuario, se notificará mediante el sistema de notificación emergente del propio sistema nativo, en este caso Android. En caso de no estar registrado en la BD, se podrá registrar a través de otro formulario en la propia aplicación, pulsando el botón “Registrarse”. De la misma manera que la pantalla anterior, el formulario está protegido contra cualquier mal uso de este o fallo en la comunicación. Véase Figura 4-4.

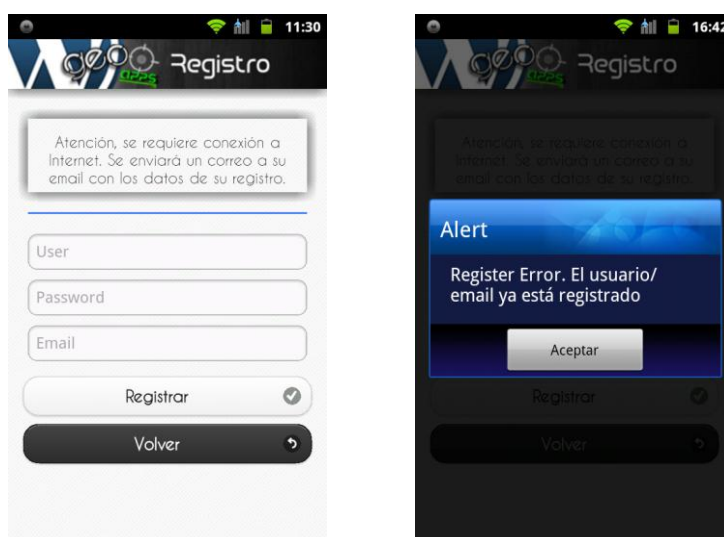


Figura4-4. Pantalla de Registro de nuevo usuario

En caso que el inicio de sesión haya sido correcto, podremos acceder al panel central de aplicación, donde podemos advertir los siguientes elementos:

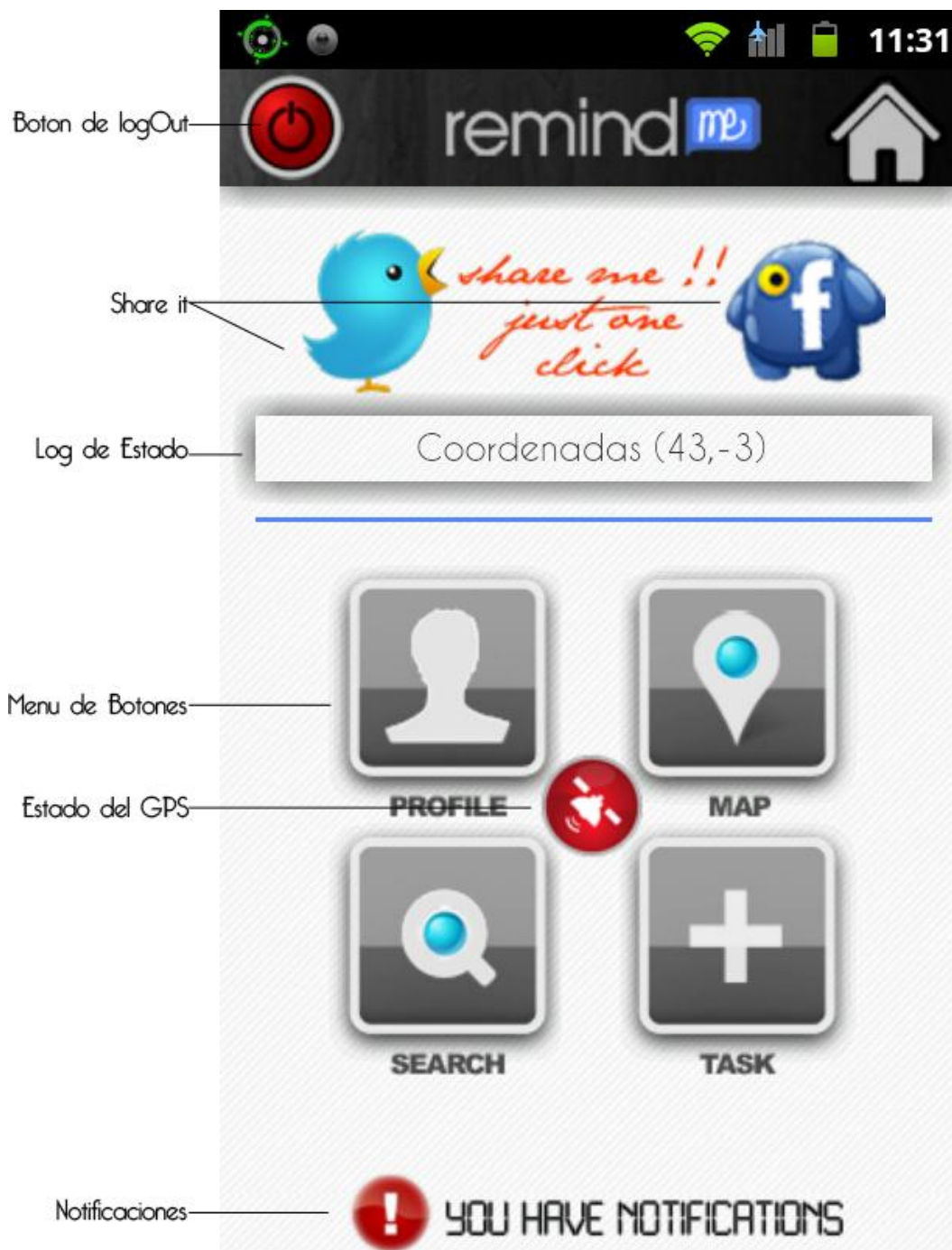


Figura4-5. Pantalla principal de la aplicación.

- Botón Logout. Sirve para guardar los cambios en la aplicación y enviarlos a la BD para su almacenamiento. También tramitará el cierre de sesión.
- Share it. Permitirá compartir la aplicación en las redes sociales más importantes: Facebook y Twitter. Véase Figura 4-6.

- Log de Estado. Si se produce algún error en la carga de las tareas o no se encuentra la posición GPS aparecerá un mensaje informativo.
- Menú de Botones. Navegador de la aplicación que ofrece la posibilidad de acceder a las cuatro partes principales de la aplicación.
- Estado del GPS. Marcará de forma visual si la aplicación ha detectado tu posición actual en el mapa. Existen dos estados: Roja y Gris. El color rojo indica que el GPS está posicionado, el gris que se está esperando respuesta del GPS.
- Notificaciones. Todas aquellas tareas que estén dentro del rango de distancia fijado para la notificación. Si pulsamos sobre ella aparecerá un pop-up que mostrará la tarea y la distancia a la que se encuentra.

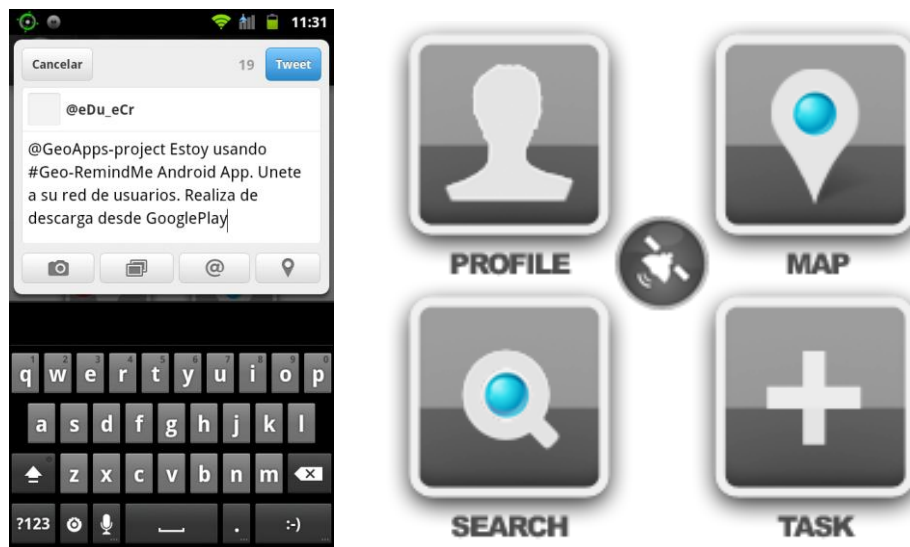


Figura4-6. Ejemplo de compartición y GPS desconectado.

Tras obtener la posición GPS, podremos acceder a cualquiera de las opciones descritas en Menú de botones. En primer lugar, pulsaremos sobre profile (perfil) donde podemos observar los siguientes elementos:

- Sección de información de usuario. Forma parte de la mitad superior de la pantalla. Se muestra información de registro del usuario tal que su nombre, fecha de registro en la aplicación y email registrado. Véase Figura 4-7.
- Sección de Notificaciones. Donde el usuario tiene la opción de añadir un sonido beep o vibración pulsando sobre el checkbox correspondiente. También se podrá definir el rango de distancia hasta la notificación.

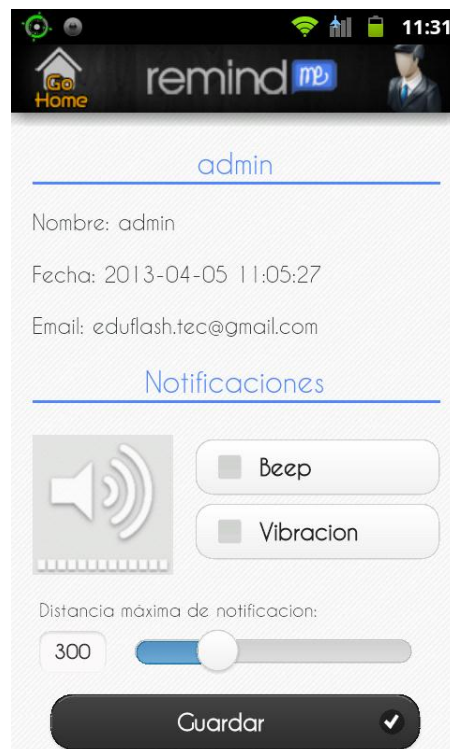


Figura5-7. Pantalla de Perfil

La siguiente pantalla a visualizar es map (mapa). En esta pantalla podemos observar el mapa a tamaño máximo de pantalla. La interfaz usada es la de Google Maps v3, que se puede observar en la Figura 4-8. Fue escogida por su versatilidad, calidad de servicio e interfaz.

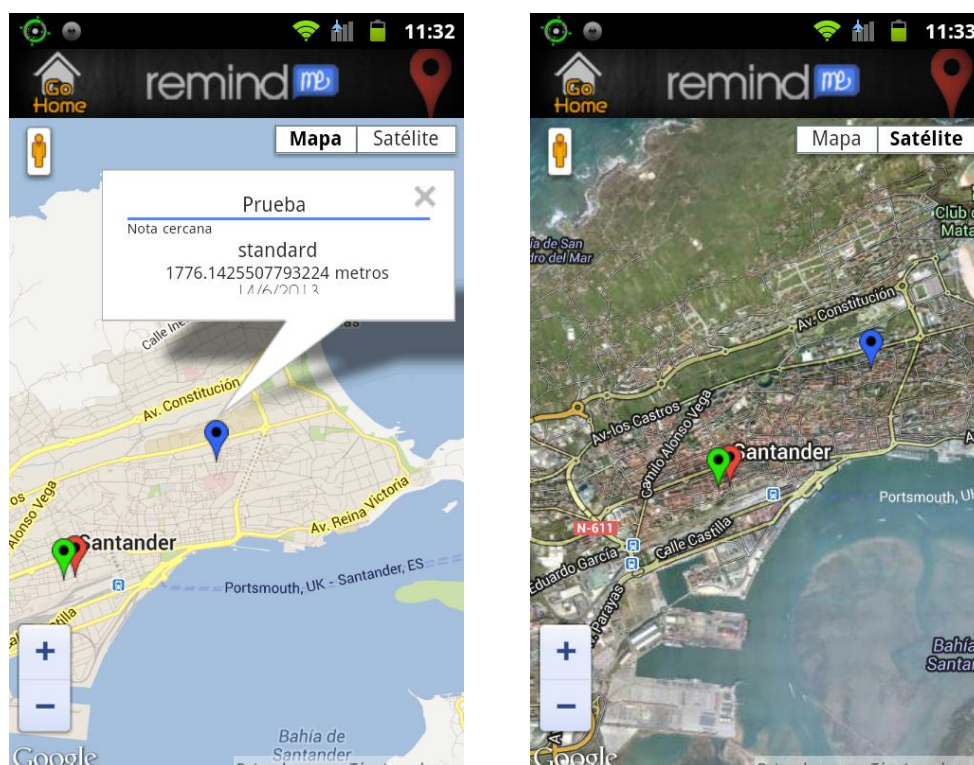


Figura4-8. Pantalla de Navegación/Mapa

En la Figura 4-8, podemos observar que se utiliza la interfaz de Google Maps v3, puesto que encaja perfectamente con el funcionamiento de nuestra herramienta porque es sólo HTML, CSS y JavaScript trabajando junto. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un <div> en la página HTML. Las características a tener en cuenta son las siguientes:

- Cada tarea aparecerá dependiendo de la cercanía con un color diferente como se muestra en la Figura 4-9. Son los siguientes:
 - Rojo – Posición actual.
 - Azul – Tarea NO cercana.
 - Verde – Tarea cercana
 - Gris – Tarea con fecha Expirada.

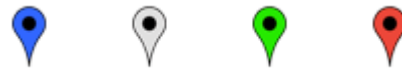


Figura4-9. Leyenda de marcadores en la pantalla Mapa

- Se podrá pulsar sobre cada marcador para observar que tarea está asignada y mostrar un dialogo que indique la información necesaria, como título, notas, prioridad y metros a los que se encuentra.
- Capacidad de mover el mapa arrastrando con los dedos, hacer zoom pinchando sobre los botones de la parte inferior izquierda o pellizcando la pantalla.



Figura4-10. Pantalla de Navegación/Mapa – Street View.

- Dos tipos de capas: Mapa o Satélite.
 - Mapa – Mapa callejero.
 - Satélite – Mapa callejero con imágenes satélite.
- Street View. Con el icono de la parte superior izquierda con forma de hombre, se puede acceder a esta utilidad arrastrándolo a la calle que queramos hacer zoom. Con esto, podremos ver algo parecido en la Figura 4-10.

Geo localización en Android

Tras pulsar el botón "BackButton" de Android, o el botón de la parte superior izquierda "GoHome", volvemos al menú principal, esta vez para entrar en la pantalla TASKS.

La pantalla tasks (tareas) mostrará todas aquellas tareas cargadas a través de la BD. Pueden existir tantas tareas como el dispositivo sea capaz de almacenar. Las tareas se actualizan de forma automática cada vez que se entra en esta pantalla.

Se mostrarán las tareas en modo lista de botones, pre-visualizando el nombre de la tarea y una parte del campo notas. Podremos a su vez diferenciar todas aquellas tareas expiradas, cuyo plazo de realización terminó observando un tachón sobre ellas. La ordenación de las tareas depende de la fecha de creación.

En caso de querer agregar una tarea nueva a la lista, se puede pulsar sobre el botón "Add Task", lo cual nos llevará a la pantalla de la siguiente explicación. Podremos también pulsar sobre la tarea de la lista que deseemos y ver su información de forma extendida, véase en la figura 4-11.

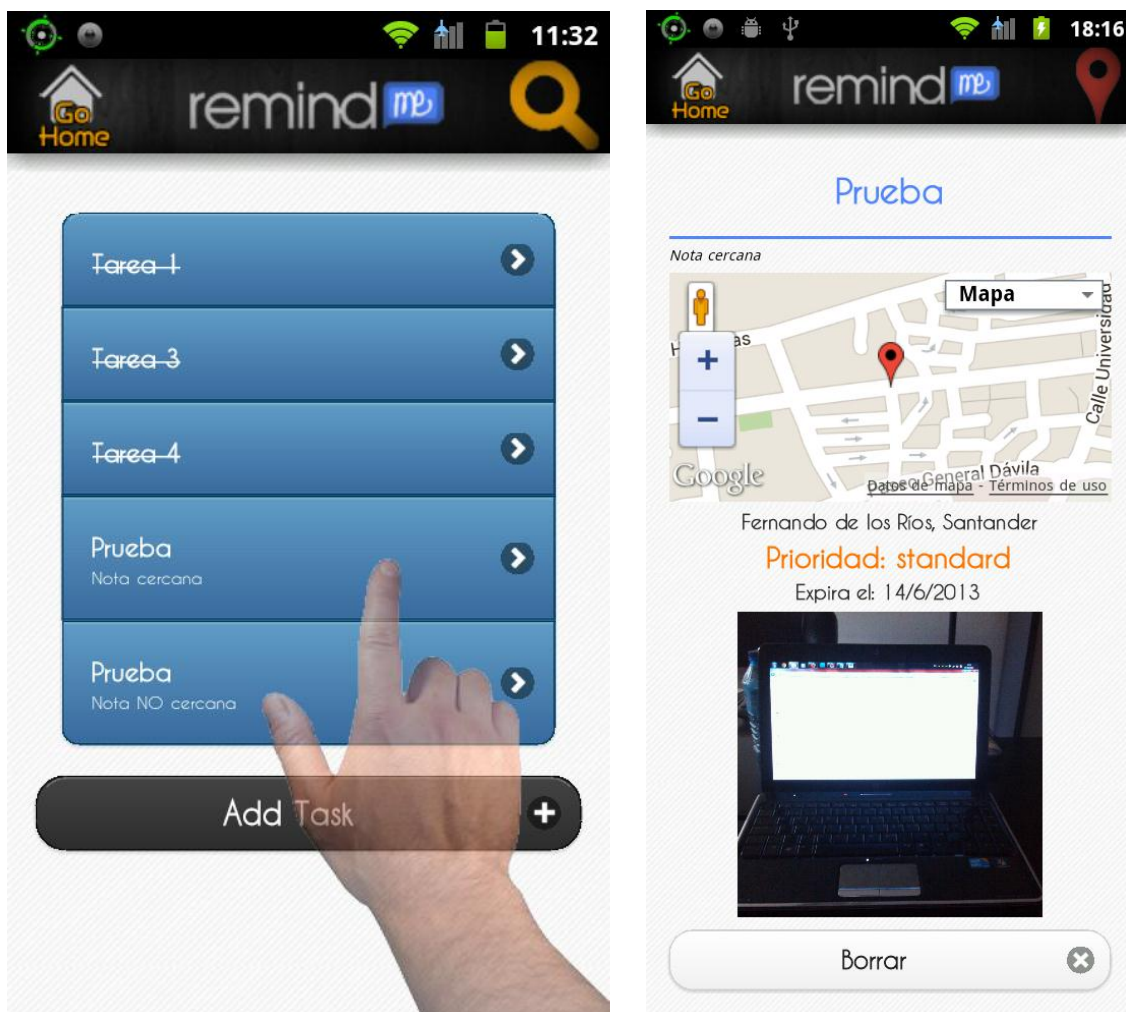


Figura4-11. Pantalla de Tareas y Visualización de nota.

Como se puede observar en la Figura 4-11 al pulsar sobre una tarea de la lista, avanzamos a otra pantalla que nos muestra toda la información de la tarea como es: Nombre, notas, localización en el mapa, dirección almacenada, prioridad y fecha en la que expira y en su caso foto. El botón "Borrar" permite borrar la tarea y eliminarla de la lista.

La última pantalla que queda por navegar es task (tarea). Básicamente es un formulario utilizado para agregar tareas como se ve en la Figura 4-12 que es observa a continuación. De esta manera, pasamos a explicar cada parte de la aplicación y su función:

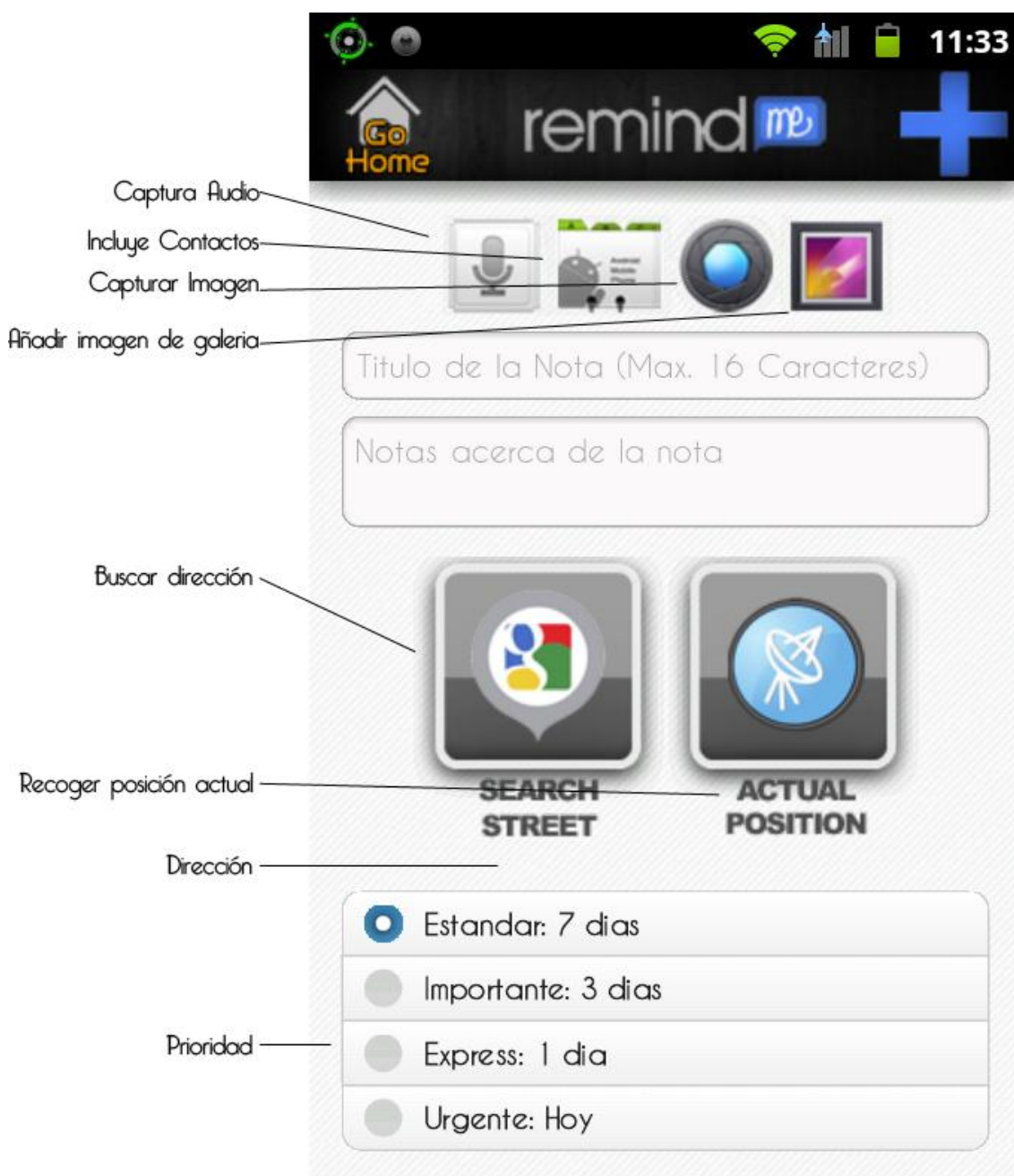


Figura4-12. Pantalla de Tarea, formulario.

Geo localización en Android

- Barra superior. Es aquí donde se encuentran todos los complementos multimedia que podemos añadir a la nota, véase Figura 4-13. Se podrá:
 - Capturar Audio. No implementado finalmente.
 - Incluir contactos a la tarea. Al pulsar sobre un contacto, este se incluirá automáticamente en el campo notas de la tarea a crear.
 - Capturar una Imagen.
 - Añadir una imagen de la galería de fotos.

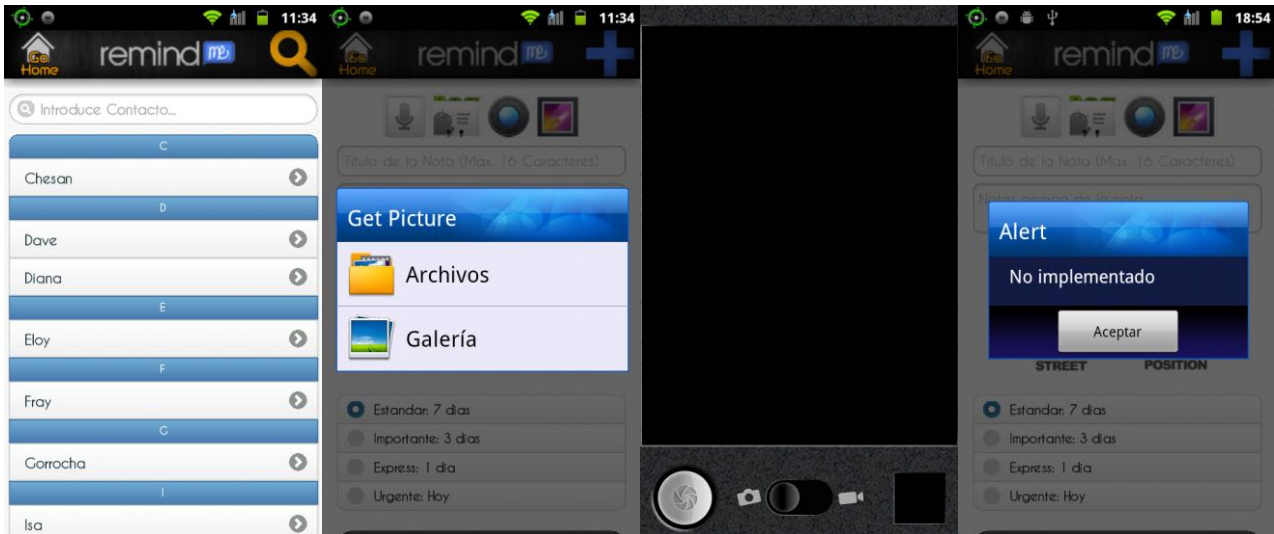


Figura4-13. Pantalla de contactos, cámara, audio, y selección de foto en álbum

- Notas acerca de la nota. Una breve descripción en caso de ser necesaria.
- Almacenamiento de la dirección. Existen dos formas de añadir una ubicación a la tarea.
 - Buscar dirección. Nos llevará a otra pantalla, que nos permitirá introducir una calle y nos proporcionará direcciones interpretables por la aplicación lo más aproximadas. Tras definir la ubicación aparecerá un mapa con un marcador que se posicionará en la ubicación propuesta. Si es correcta, se podrá pulsar sobre guardar y almacenar esa ubicación. Véase Figura 4-14.
 - Posición actual. Hace una llamada al GPS para determinar la posición actual, que mediante geo-codificación inversa determinará la dirección en la que nos encontramos.
- El campo dirección está oculto hasta la inserción de una ubicación. En el momento de insertarse aparecerá la dirección solicitada.
- Prioridad. Este Checkbox determina la fecha en la que expira la nota.

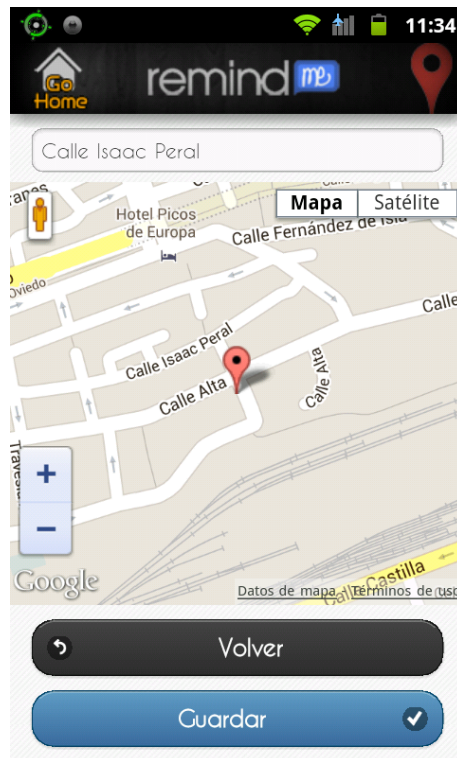


Figura4-14. Pantalla de búsqueda de dirección

En el momento que se produzca una notificación por cercanía a una tarea no expirada, aparte de la notificación interna de la aplicación, se mostrará en la barra de superior el símbolo de la aplicación avisando de esta. Si deslizamos la barra de superior veremos la notificación como en la Figura 4-15.

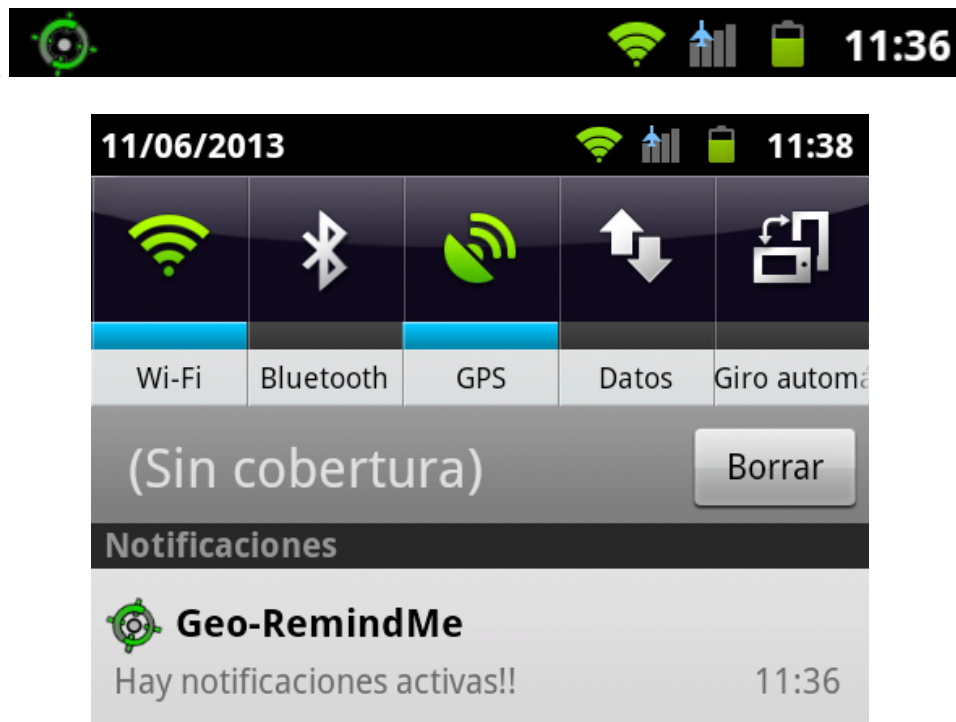


Figura4-15. Status Notification Bar

4.2 Resultado de la Interfaz gráfica de la aplicación web

La aplicación web está implementada y subida a un dominio que se corresponde con la dirección <http://www.geoapps-project.com>, y es accesible sin necesidad de registro. Por ello, se realizará una descripción somera de su interfaz.

Al acceder a la aplicación web, se observan cuatro secciones, como se observa en la Figura 4-16. Estas se corresponden con una página de contacto e información sobre el sistema, dos páginas sobre ideas futuras de desarrollo, y por último la página web correspondiente a la aplicación móvil desarrollada en este proyecto.

En la parte inferior aparecerá el último post o noticia generada en la web de forma que sea lo más accesible posible. Para verla en toda su extensión, se puede pinchar en "leer más" y acceder a la página de noticias.



Figura4-16. Pantalla principal de la página web

En la página de contacto, se puede encontrar información relativa al motivo de creación de la web, mis datos y demás. Nada relevante e indispensable de conocer no siendo objeto de este proyecto.

La sección que desempeña un papel importante en el sistema se corresponde a la que tiene el mismo nombre que la aplicación móvil. Si pulsamos sobre ella podremos destacar el panel de tareas que aparece, si previamente hemos iniciado sesión. Podremos observar exactamente las mismas tareas que desde la aplicación móvil, una muestra de ello es la Figura 4-17.

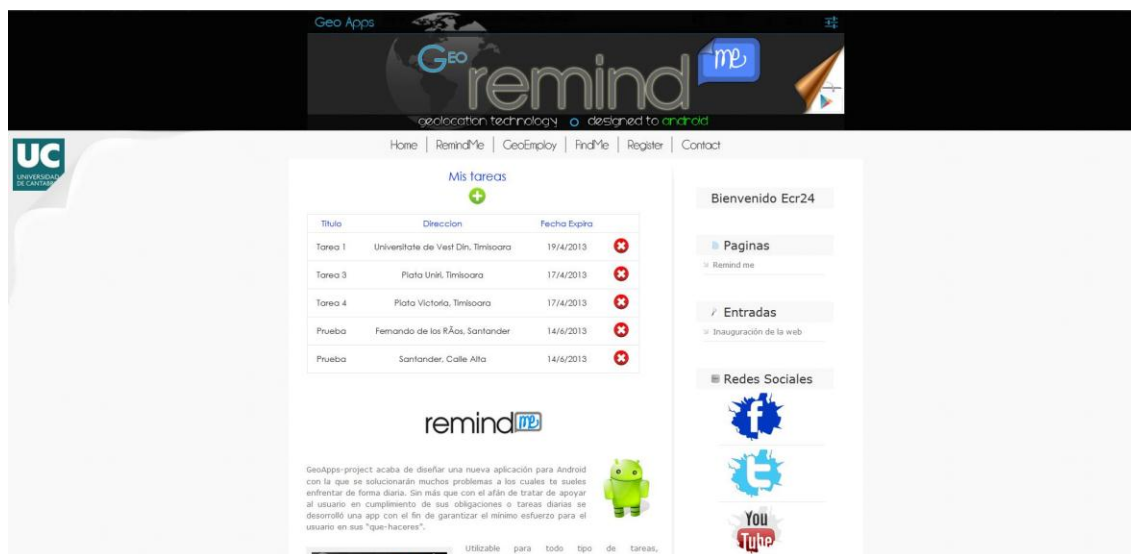


Figura4-17. Pantalla Geo-RemindMe

Si observamos detenidamente el panel de tareas, se observan varios botones que otorgan una funcionalidad extra a esta página web. El botón rojo, permitirá eliminar la tarea que corresponda. Por su parte, el botón verde generará un pop-up con un formulario para añadir una tarea a la lista. Véase Figura 4-18.



Figura4-18. Pantalla Geo-RemindMe – Tareas y Formulario

Capítulo 5. Pruebas

5.1 Evaluación y pruebas

Testear una aplicación es un proceso muy efectivo para encontrar fallos, pero no para demostrar la ausencia de ellos. Este es habitualmente uno de los dolores de cabeza más comunes para los desarrolladores de apps.

Todos sabemos que hay miles de modelos de móvil y Tablets diferentes. En el mundo Apple, solamente hay unos pocos, pero en el mundo Android hay literalmente miles de dispositivos con características diferentes, sin contar el resto de plataformas para los que está desarrollado esta web, pero el uso de un entorno de desarrollo de terceros (PhoneGap) fragmenta la pelea por universalizar el funcionamiento de una app en todos los dispositivos posibles.

En el caso de la web, existen herramientas y utilidades SEO (Search Online Optimization) online, para hacer pruebas, que proporcionan todo tipo de información sobre una página, un dominio, su optimización SEO, su repercusión social etc.

Aun así, se han probado los flujos de funcionamiento habituales, puesto que testear todas las posibles combinaciones sería innecesario si formalizamos un reporte de bug en nuestra aplicación. Como consecuencia de esto, se utiliza una estrategia de testing inspirada en box-approach que se divide en dos partes:

- Caja Negra: Desde fuera la funcionalidad solo nos importa lo que hace, no como lo hace.
- Caja Blanca: Dentro de la funcionalidad, nos importan los detalles y como se llevan todas las operaciones a cabo.

El testing, básicamente trata de cubrir todos los casos de ejecución durante el ciclo de vida del software, es por ello que debemos prestar atención a estos tipos de test adicionales, vitales en el desarrollo de una aplicación móvil y que se han realizado a la hora de desarrollar este proyecto.

- Interfaz. Validación de botones, inputs, flujos de navegación, orientación del dispositivo, es decir, validar cada pantalla de forma integral.
- Usabilidad. Comprobación de navegabilidad, redundancias, y control de realimentación (feedback) en interacciones con la app.

- Perfeccionamiento. Control de conexiones de Red, gestión de envío de datos a servicios web utilizados, relación de aspecto de imágenes, optimizar código redundante, cumplimentación de las exigencias de usuario y uso de recursos y su liberación en el sistema cuando quedan en desuso.
- Seguridad. Los envíos son codificados y/o autenticados con soporte multiusuario.
- Servicios. Las apps actúan como clientes por tanto es necesario validar que si se cae un servicio se notifiquen los correspondientes errores de servicio.
- Recursos de bajo nivel. Se comprueba que la app borre los archivos temporales, gestione la base de datos local, y si usa demasiada memoria.
- Compatibilidad, multi-dispositivos y entornos. Validación en diferentes versiones de SO y modelos acordados.

5.2 Software y Tecnologías utilizadas para las pruebas en el sistema.

Durante el desarrollo de la aplicación y cada una en su ciclo de vida correspondiente, hemos utilizado las siguientes herramientas que han ayudado a testear la aplicación en la medida anteriormente descrita.

- Ripple Emulator. Complemento de Chrome que permite cargar una página web en un emulador de un dispositivo móvil. Entre las opciones que ofrece este emulador se encuentran: definir la posición para visualizar la página en sentido vertical u horizontal, configuración de los datos de geo localización y acelerómetro.
- Firebug. Complemento de Firefox, creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar, editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e *inline*.
- Simulador SDK Android. Emula el sistema operativo pero su finalidad es más de análisis y estudio del comportamiento de una aplicación en un entorno nativo. Así, con este software se pueden emular todo tipo de dispositivos y ajustar un sinfín de opciones y configuraciones.

5.3 Control de versiones

La configuración software es la manera en que diversos elementos se combinan para construir un producto software. Se han de combinar todos los elementos que intervienen en el desarrollo los cuales son:

- Documentos del desarrollo
- Código fuente
- Programas, datos y resultado de las pruebas
- Manuales de usuario
- Documentos de mantenimiento, informes de problemas y cambios
- Prototipos intermedios
- Normas particulares del proyecto

Es por esto, que dado que los elementos software van evolucionando a lo largo del desarrollo se requiere:

- Control de versiones. Almacenar de forma organizada las sucesivas versiones de cada elemento de la configuración. Véase Figura 4-1 la gestión de la configuración resultante de este PFC, mantenido con subversión.

Versión	Cambios
1.0	Maquetación HTML.
2.0	Introducción de diseños CSS.
3.0	Introducción de lenguaje JavaScript.
4.0	Diseño XHTML y funciones AJAX.
5.0	Implementación de jQueryMobile.
6.0	Diseño beta de aplicación en inicio de sesión y registro.
7.0	Incluye conexión con base de datos remota.
8.0	Incluye conexión con API de Google Maps
9.0	Alertas y ajustes de notificación beep y vibración.
10.0	Incluye detector de GPS encendido.
11.0	Nuevo diseño en página principal, Twitter y aviso de notificación.
12.0	Notificaciones pop-up y control de flujo de uso.
13.0	Diseño de tareas actualizado.
14.0	Inclusión de fotos e imágenes en tareas.
15.0	Inclusión de contactos en tareas y navegación en mapa
16.0	Cambio de diseño en búsqueda de direcciones.
17.0	Diseño mejorado y recordar contraseña
18.0	Loaders de espera (Iconos de carga).
19.0	Diseño de notificación y diseño responsable.
20.0	Condiciones de usuario.
21.0	Primera beta de la aplicación consistente.

Figura5-1. Control de versiones y cambios de la aplicación móvil.

Capítulo 6. Problemas e Impedimentos

6.1 Uso de API de Google Maps

Google Maps, como ya se ha explicado, provee una API para desarrolladores de aplicaciones Android que facilita enormemente el uso de mapas en dichas aplicaciones. La versión de la API con la que empecé el proyecto fue la 2, la cual fue lanzada a finales de 2012.

A mitad del transcurso del proyecto, los desarrolladores de Google iniciaron la proclamación de su nueva API v3.0. Todo esto no sería un problema, si a partir de Febrero de 2013 con su lanzamiento de la tan proclamada API v3 no hubieran declarado obsoleta a partir de Mayo de 2013 la versión 2.

Esto me produjo un trastorno considerable debido a que, aunque los servicios web sean mejores y sus objetivos básicamente los mismos, ciertos métodos, funciones y parámetros pasaron a ser obsoletos, modificados, o no funcionaban de forma correcta. Esto provocó que por segunda vez tuviera que hacer un estudio de la API nueva, sobre la que la información era muy escasa, modificando el código desarrollado en los casos que fue necesario.

6.2 Simulador SDK Eclipse

El simulador SDK de Eclipse, aunque sea el más completo del momento y capaz de simular en gran medida el sistema Android, ha sido realmente frustrante durante todo el proceso de desarrollo de la aplicación móvil.

Los continuos errores de compilación y la lentitud de ejecución, fueron los principales problemas de que decidiera olvidarme de él, y probar directamente, en cada iteración o fase de pruebas, en el dispositivo móvil ya que otros no me ofrecían la interactividad hardware que buscaba con un dispositivo físico.

Cabe esperar que Android, si desea aumentar su calidad de apps muy inferior al de IOs, desarrolle un simulador más ligero, fiable y completo.

6.3 Encriptación de contraseñas en el CMS Wordpress

Desde la versión 2.5 de Wordpress, la encriptación de las contraseñas está basada en "Phpass". Este es una herramienta de cifrado de contraseñas de dominio público para usos en aplicaciones PHP. La manera de manejar esta herramienta es algo confusa y tiene un alto nivel de desinformación en contraposición a su anterior y débil compañera, el algoritmo Md5. Bien es cierto que la nueva solución que plantea Wordpress para potenciar la seguridad en su entorno es incomparablemente más potente, pero esto ha supuesto un escollo en el desarrollo de este proyecto.

El problema surgido ha sido el siguiente. La base de datos, está controlada por un gestor de contenidos que trata a los usuarios de una forma tan ordenada y segura como compleja y poco difundida. El problema es que no se contó en un principio con el inicio de sesión y registro de usuarios fuera del entorno Wordpress, que posteriormente se tuvo que desarrollar para conectar a los usuarios de la aplicación móvil con los datos almacenados en la BD y restringidos para usuarios registrados.

Este hecho, propició que se investigara la clase maestra que proporciona Wordpress para tratar su encriptación interna. La información encontrada fue escasa, pero finalmente se consiguió desencriptar y encriptar mediante este nuevo método, haciendo así posible la interconexión entre Wordpress, la BD y mi aplicación móvil y así finalmente, aprovechar las virtudes que me ofrecían cada una de ellas.

6.4 Dominio y servidor web

Cabe comentar, que el desarrollo del proyecto en sus inicios estuvo implementado en un servidor local donde realicé el 90% de pruebas, mientras por otro lado investigaba sobre la viabilidad del proyecto que quería desarrollar. El problema surgió cuando no me conformé con que funcionara en servidor local, sino que quería implementar el producto comercialmente.

Al pretender cargar mi aplicación web en un servidor, me encontré el primer problema, y es que la Universidad de Cantabria no me ofrecía un dominio de hosting en el que poder implementar mi proyecto y demostrar su funcionalidad. Me involucré tanto en ello, que decidí comprar un dominio compartido a la empresa de hosting www.redcoruna.es y poder demostrar su correcto funcionamiento, teniendo que gastar un dinero que pienso no debería ser necesario.

6.5 Comunicación remota Cross Domain

Para comunicar la aplicación móvil con la base de datos remota, hay ciertas restricciones basadas en el Cross Domain Calls. Este es un mecanismo de seguridad en navegadores actuales. Evitan que un script (XMLHttpRequest de AJAX) de una página web puedan acceder a un servidor web diferente del que residen.

La solución fue agregar, a esa interfaz de comunicación, una política de dominio que permitiera el acceso a cualquier host externo que haga peticiones. Esta medida quiebra en cierta medida la seguridad del dominio, ya que cualquier usuario con ciertos conocimientos puede realizar peticiones con objetivos malintencionados.

Por esto, se toma la decisión de fortalecer la seguridad ante tal contratiempo, codificando los datos enviados y recibidos entre app y BD mediante encriptación JSON. Esta solución es habitualmente usada y recomendada en conexiones pasarela mediante la tecnología AJAX Calls, una de las que implementa la aplicación móvil.

6.6 Diseño responsable

El diseño responsable en el diseño web, o más conocido como “*Responsive Design*” es uno de los mayores quebraderos de cabeza de los desarrolladores y diseñadores web. Este hecho implica que tu web o aplicación móvil, si está desarrollada con tecnologías web, sepa ajustarse a la resolución y/o tamaño de pantalla de tu dispositivo desde el cual la visualizas. Este problema, no lo es tanto en el campo de los sitios web, porque las resoluciones utilizadas son menos y están más definidas en los ordenadores actuales que respecto a los móviles. Por ejemplo el iPhone 320x480, los Nokia N-Series varían entre 240x320, 176x208, 352x416 y Blackberry entre 160x160 y 324x352. En la Figura 6-1 se pueden observar las resoluciones más habituales.

La solución adoptada en este proyecto fue la de adaptar el diseño de la aplicación en consonancia a la pantalla, es decir, en el momento de la ejecución inicial de la aplicación detectar el tamaño de pantalla en pixeles de nuestro dispositivo. Posteriormente, el diseño del CSS, que contiene los estilos, está programado para que varíe entorno a este parámetro de resolución que se divide ancho y alto. De esta forma, todos los objetos que integran la aplicación se auto-ajustarán de forma dinámica a la pantalla móvil.

Resolution	Devices
320×240	➤ Blackberry Devices: Curve 8530, Pearl Flip ➤ Android Devices: Motorola Charm, Sony Ericsson Xperia X10 Mini, others ➤ Symbian OS Devices: Nokia E63, others
320×480	➤ Apple OS Devices: iPhone, iPod ➤ Android devices: HTC Dream, HTC Hero, Droid Pro, i7500 Galaxy, Samsung Moment, others...
480×360	Blackberry Devices: Torch, Storm, Bold
360×640	Symbian OS Devices: Nokia N8, Nokia C6-01, others
480×800	➤ Android Devices: Liquid A1, HTC Desire, Nexus One, i9000 Galaxy S, others ➤ Maemo (Linux) Devices: Nokia 900, others ➤ Windows Mobile 6 Devices: Sharp S01SH ➤ Windows 7 Phone Devices: Venue Pro, Samsung Omnia 7, HTC 7 Pro, others
768×1024	iPad
640×960	iPhone 4
1280×800	➤ Android Devices: Motorola Xoom, Samsung Galaxy Tab 10.1 ➤ Windows OS Devices: Asus Eee Pad EP121 ➤ Apple OS Devices: Axiotron Modbook

Figura 6-1. Resoluciones de pantalla en dispositivos móviles.

6.7 Dispositivos móviles

En el desarrollo y prueba de aplicaciones móviles, es importante contar con dispositivos móviles de diferentes características en los que probar tu aplicación, su funcionamiento y completar así su implementación.

En el caso de este proyecto, aunque la aplicación móvil está desarrollada al 80% para cualquier tipo de dispositivo, no se ha podido realizar una implementación en otro tipo de sistema operativo por falta de estos dispositivos.

Capítulo 7. Conclusiones

El objetivo principal de este proyecto fin de carrera era crear e implementar la infraestructura necesaria para el desarrollo de aplicaciones móviles basadas en geo-localización multiplataforma, y probar su implementación en el sistema móvil Android mediante el framework PhoneGap desviando su almacenamiento a una base de datos alojada en un servidor web. Para su consecución, se han realizado todas las fases del ciclo de vida software, esto es, la captura de requisitos, análisis, diseño e implementación, evaluación y pruebas tanto del sitio web www.GeoApps-project.com como de la aplicación móvil Geo-RemindMe, cumpliendo con los objetivos propuestos inicialmente.

En su diseño se ha tenido en cuenta que el perfil de usuarios es variado, por lo que se ha realizado un gran esfuerzo para que la interfaz sea intuitiva y no requiera conocimientos adicionales para su uso. Además se ha desarrollado con un framework multiplataforma de modo que la aplicación pueda implementarse en un gran número de dispositivos.

La geo-localización es la tecnología en la que se apoya esta aplicación. Mediante la API de Google Maps v3 y las características hardware de la mayoría de los dispositivos móviles actuales, se ha creado una aplicación móvil capaz de localizar la posición del usuario, crear tareas basadas según su ubicación, definirlas en el mapa y mostrar notificaciones dependiendo de la proximidad, prioridad y fecha al conjunto de tareas definidas. En definitiva, la aplicación tiene por objeto optimizar el tiempo del usuario respecto a tareas no prioritarias, definidas en el espacio, que permitan completarlas por medio de notificaciones con el mínimo consumo de tiempo.

El sitio web responde a las necesidades básicas con las que fue especificado. Ofrece un diseño similar al de la aplicación móvil y tiene la función de publicitar el sistema implementado y, acceder a los mismos datos a los que se muestran desde la aplicación móvil para garantizar totalmente su uso desde cualquier dispositivo sea o no móvil.

Generalmente, el almacenamiento interno disponible en estos dispositivos móviles es reducido y caro, siendo habitualmente usado con motivos multimedia. Además, no se garantiza la seguridad ante pérdidas o daños, debido a que el móvil se puede perder o romper en cualquier momento puesto que su uso es continuo y es cada vez es más importante poder acceder a los datos desde cualquier plataforma. Es por esto que, con objetivo de garantizar la disponibilidad de la información desde

cualquier dispositivo, el almacenamiento de los datos de la aplicación residen en un servidor web que proporciona un almacenamiento remoto.

El motor de base de datos MySQL, usado para el almacenamiento de los datos ofrece seguridad, tiempo de computación y la eficiencia necesaria para un funcionamiento esperado de la herramienta.

Las diferentes tecnologías usadas son las siguientes:

- UML 2.2. Lenguaje de modelado, usado en las fases de captura de requisitos, análisis y diseño de este proyecto, en el entorno visual Magic Draw UML 16.8
- HTML. Lenguaje web en el que están basadas ambas aplicaciones.
- JavaScript. Lenguaje de programación interpretado, usado para desarrollar todas las funciones de la aplicación móvil.
- CSS. Código de control de estilos usado en ambas aplicaciones.
- Java. Lenguaje de programación usado en el entorno de desarrollo Eclipse para desarrollar la aplicación Android.
- JSON. Formato ligero de intercambio de datos, usado en el envío y recepción de datos con la API que dispone Google Maps v3.
- SQL. Lenguaje de consulta estructurado, usado en torno a la base de datos implementada en el proyecto, con MySQL como gestor de base de datos.
- PHP. Lenguaje utilizado en el desarrollo web. Es, además, utilizado en la API que conecta la aplicación móvil a la base de datos, y forma parte del desarrollo del sitio web.
- MD5. Algoritmo de reducción criptográfica usado en la comunicación entre la aplicación móvil y el sistema de almacenamiento de datos.
- PHPASS. Algoritmo de encriptación y seguridad en que está basado el gestor de contenidos utilizado, Wordpress.
- XML. Utilizado por la interfaz Android y algunos archivos de configuración de la herramienta PhoneGap.
- Interfaz DOM. Usado en la traducción de mensajes XML.
- AJAX. Que incluye HTML, XHTML, DHTML, XML con DOM, el objeto XMLHttpRequest, CSS y JavaScript, como tecnología RIA para implementar la Interfaz web y móvil en el entorno de programación y desarrollo Eclipse.

El sistema implementado en este proyecto ofrece una funcionalidad innovadora, que responde a todas aquellas preguntas que surgen al adentrarse en el

entorno del desarrollo de las aplicaciones móviles. El mercado, respondiendo a las necesidades del usuario, es exigente. Esto provoca que las aplicaciones sean cada vez más completas sin ser ya suficiente que sean innovadoras, atractivas y funcionales y es por esto que, tras una investigación sobre las aplicaciones y destacando sus mejores virtudes, se han unido todas las tecnologías posibles para proporcionar al usuario el mayor número de ventajas posible.

El aprendizaje y uso de estas tecnologías, así como la búsqueda de información de investigación en campos relacionados con las aplicaciones móviles, sitios web, encriptación, seguridad y bases de datos han servido para aplicar y afianzar los conocimientos adquiridos durante la carrera así como para extenderlos y aprender sobre muchas otras tecnologías y conceptos nunca vistos en las asignaturas de la carrera.

Capítulo 8. Trabajos futuros

Existen numerosas tecnologías en continuo desarrollo en el entorno de desarrollo móvil. Este proyecto ha abordado el desarrollo híbrido multiplataforma en búsqueda de un equilibrio adecuado entre reaprovechamiento y especificidad. Así mismo se ha diseñado de forma modular de tal forma que sea más sencilla la incorporación de los cambios tecnológicos que se produzcan en un mercado tan ágil, dinámico y volátil, como este.

Los datos estadísticos de Gartner [35] sostienen que en pocos años, el 50% de las aplicaciones móviles serán híbridas, y recomienda activamente que las empresas piensen en plataformas que, como PhoneGap, permitan desarrollar aplicaciones híbridas, para dar respuesta a la siempre creciente diversidad de dispositivos.

El desarrollo de aplicaciones móviles va por delante de la experiencia en PC. Actualmente se busca una solución móvil antes que ninguna otra, y es que los Smartphones y Tablets se están haciendo con el control total del mercado. Aun así, el mercado móvil tiene que evolucionar mucho y está lejos de alcanzar la madurez.

Por otro lado la nube, comienza su andadura y parte como favorito. Su progreso y aceptación comenzó con la gran acometida de las grandes empresas como Apple, Google y Microsoft que desarrollaron soluciones entre sus usuarios como si de una batalla se tratase. Difícil era imaginarse que no llegara al entorno de las aplicaciones móviles, pero no que llegara tan rápido. iCloud por parte de Apple, casi siempre pionera, ha allanado el camino a las demás y por supuesto, se ha puesto de moda provocando un boom en el mercado tecnológico.

Si a todo esto se suma la funcionalidad avanzada de las aplicaciones móviles, sinceramente no se ve límite en el horizonte. Google, acaba de anunciar la llegada de Google Engine Maps, que permite a las empresas utilizar la infraestructura en la nube de Google para crear capas sobre un mapa de Google, y compartir su mapa personalizado con empleados, clientes o con el público en general. Por tanto, una empresa puede, con un clic, compartir internamente sus mapas o publicarlos en la web, para así hacer sencillo el acceso a los datos desde cualquier lugar.

En el contexto de este proyecto, se contempla desarrollar dos aplicaciones móviles: Geo-EmployMe y Geo-FindMe. Se implementarán haciendo uso de todas las tecnologías de este informe y añadiendo alguna más como el servicio de notificación push. Geo-EmployMe pretende realizar una aplicación web de similares características a la conocida "infojobs" que promueva la difusión de ofertas de usuario entre todos

sus usuarios. Estos usuarios además podrían descargarse una aplicación móvil, por supuesto multiplataforma, que actuaría en cierta medida como la ya desarrollada pero con algunos avances importantes. La aplicación móvil avisaría a través de notificaciones y controlaría tu ubicación de aquellas ofertas de trabajo disponibles sobre el terreno y, de acuerdo al curriculum previamente ingresado del usuario. A esa oferta se la podría atender de dos maneras, una mediante el envío de un mensaje de texto o correo electrónico a través de la aplicación y dos, permitiendo la navegación en tiempo real hasta la empresa que ofrece esa oferta de empleo.

Por otro lado, Geo-FindMe pretende ofrecer la posibilidad de tener un botón de pánico en el móvil que avise a unos contactos definidos. Esta aplicación permitirá gestionar amigos, los cuales recibirán notificaciones en caso que pulses este botón. Esa notificación estará compuesta de la posición aproximada del usuario con la emergencia, el nombre del usuario y una serie de opciones de respuesta como responder con llamada, llamar a los servicios de emergencia correspondientes al lugar en el que se produce la emergencia, o enviar una alerta a otros usuarios más cercanos.

Con estos ejemplos, se muestra la capacidad manifiesta que ofrece la tecnología y herramientas que se ofrecen en el ámbito del desarrollo móvil. No existe cuantía para medir su impacto en la sociedad ni techo para seguir evolucionando en el mercado actual.

Bibliografía

1. Pilgrim, Mark. (2010). *HTML5 Up and Running*. O'Really Google Press.
2. Aubry, Christophe. (2012). *HTML5 + CSS3. Revolucione el diseño de sus sitios web*. Editorial ENI.
3. Gauchat, J. (2012) *El gran libro de HTML5, CSS3 y JavaScript*. Marcombo ediciones técnicas.
4. Perry, B. W. (2006). *AJAX. Los mejores trucos*. Anaya Multimedia.
5. Deitel, H. M., & Deitel, P. J. (2008). *Ajax, Rich Internet Applications y desarrollo Web para programadores*. Anaya Multimedia.
6. Smith, D. & Negrino, T. (2009). *Javascript and Ajax for the web: Visual QuickStart Guide*. PeachPit Press Pearson Education.
7. Heilman, C. (2006). *Beginning JavaScript with DOM Scripting and AJAX: From Novice to Professional*. Aprress.
8. Pérochon, S.(2012). *Android: Las bases para un buen inicio en el desarrollo para Smartphones y Tablet*. Ediciones ENI.
9. Ostrander, J. (2012, July) *Android UI Fundamentals: Develop & Design*. PeachPit Press Pearson Education.
10. M. Wargo, J. (2012). *PhoneGap Essentials: Building Cross-platform Mobile Apps*. Pearson Education.
11. Holdener, A. (2011). *HTML5 Geolocation*. O'Reilly Media.
12. Neil, T. (2012). *Mobile Design Pattern Gallery: Ui Patterns for mobile applications*. O'Really Media.
13. Lecomte, S. & Boulanger, T. *XML práctico: Bases esenciales, conceptos y casos practicos*. Editorial ENI.
14. Philips, J. & Davis, M. (2007). *Learning PHP & MySQL: Step-by-step Guide to creating database-driven WebSites*. O'Really.
15. Cobo,A., Gomez, P., Perez, D. & Rocha, R. (2005) *PHP y MySQL: Tecnologías para el desarrollo de las aplicaciones web*. Ediciones Diaz Santos.
16. Erickson, J., & Siau, K. (2008). Web Services, Service-Oriented Computing, and Service-Oriented Architecture: Separating Hype from Reality. *Journal of DataBase Management*, 42-54.

17. Lal, R. & Chava, L. (2012). *Developing web widget with HTML, CSS, JSON and AJAX: A complete guide to web widget*.
18. Svennerberg, G. (2013). *Beginning Google Maps API 3*. Apress.
19. Esposito, D. & Saltarello, A. (2009). *Microsoft .NET: Architecting Applications for the Enterprise*. Microsoft-O'Really Media.
20. Sun Microsystems (2009, June). *Introduction to Cloud Computing Architecture*. White Paper, 1st Edition.
21. Grossman, R.L., Gu, Y., Sabala, M., & Zhang, W. (2008). *Compute and storage clouds using wide area high performance networks*. Elsevier B.V.
22. Xu, M., Gao, D., Deng, C., Luo, Z & Sun, S. (2009) Cloud computing boosts business intelligence of telecommunication industry. *Cloud Computing: First International Conference* (pp. 224-231). Beijing: Sptinger- Verlag Berlin Heidelberg.
23. Zheng, P. & Ni, L. (2006). *SmartPhone and Next Generation Mobile Computing*. Morgan Kaufmann.
24. Reinheimer, P. (2000) *Web APIs with PHP: Ebay, Coogle, Paypal, Amazon, FedEx plus Web Feeds*. Wiley Publishing.
25. Gurugé, A. (2004). *Web Services Theory and Practice*. Elsevier Digital press.
26. Córdoba, E., González, C. & Córdoba, C. (2006). *Photoshop CS5: Curso avanzado*. Ra-Ma editorial.
27. Jim Melton, A. (2000). *Undestanding SQL and Java Together: A Guide to SQLJ, JDBC and related technologies*. Academic press.
28. Popescu, A. (2012). *Geolocation API Specification*. W3C. Retrieved from: <http://dev.w3.org/geo/api/spec-source.html>
29. Microsoft. (n.d.) *MSDN SQL Server*. Retrieved from <http://msdn.microsoft.com/en-us/library/bb545450.aspx>
30. NetBeans. (n.d.) *NetBeans IDE*. Retrieved from <http://netbeans.org/>
31. PhoneGap. (n.d.) *PhoneGap Documentation*. Retrieved from: <http://docs.phoneGap.com/en/2.8.0/index.html>
32. JQueryMobile. (n.d.) *JQuery Mobile API Documentation*. Retrieved from: <http://api.jquerymobile.com>
33. Android SDK. (n.d.) *API Guides*. Retrieved from: <http://developer.android.com/guide/topics/ui/index.html>

34. Empirica Research. (n.d.). *Segundo estudio sobre el uso de smartphones en España*. Retrieved from: www.slideshare.net/Empirica/empirica-segundo-estudio-sobre-el-uso-de-smartphones-en-espa
35. Computer World. (n.d.). Gartner: *El 50% de las aplicaciones móviles serán híbridas en 2016*. Retrieved from: <http://www.computerworld.es/movilidad/gartner-el-50-de-las-aplicaciones-moviles-seran-hibridas-en-2016>
36. Wordpress. (n.d.). *Codex es:DataBase Description*. Retrieved from: http://codex.wordpress.org/es:Database_Description