



UNIVERSITAT DE BARCELONA



Treball fi de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques

WEB-MÓVIL

David Rodríguez Picón

Director: Simone Balocco

Realizado en: Departament de Matemàtica Aplicada i Anàlisi. UB

Barcelona, 20 de junio de 2014

Índice

Abstract	6
Resumen	7
1.Introducción	8
1.1.Definición de objetivos	8
1.2.Motivación	9
2.Tecnología utilizada	10
2.1.Servidor XAMPP Apache + MySQL + PHP + Perl	10
2.2.Eclipse	10
2.3.Node.js	11
2.4.Android SDK	11
2.5.ADT-Plug-In	11
2.6.Frameworks multiplataforma	12
2.6.1.Titanium Appcelerator	12
2.6.2.Strawberry	12
2.6.3.Apache Cordova/PhoneGap	13
2.6.3.1.Diferencias entre Apache Cordova y PhoneGap	13
2.6.3.2.Licencia	14
2.6.3.3.Plataformas soportadas	15
2.6.3.4.PhoneGap Build	15
2.7.Emulador y dispositivos físicos utilizados	16
2.7.1.Emulador Android	16
2.7.2.PhoneGap Emulator (Ripple)	17
2.7.3.Dispositivos Físicos	18
3.Estudio de mercado	20
3.1.Análisis de la competencia	20
3.2.Conclusión del análisis	23
4.Base de datos	26
4.1.Estructura	26
4.1.1.Cliente	27
4.1.2.Servidor	27
4.2.Vista relacional	28

4.2.1.Tablas	28
4.2.2.Atributos	29
4.2.3.Ficheros PHP	31
4.2.4.Almacenamiento de imágenes	31
5.Planificación	34
5.1.Primer etapa	34
5.1.1.Declaración de objetivos	35
5.1.2.Tablas de asignación temporal de función	35
5.2.Segunda etapa	37
5.2.1.Declaración de objetivos	37
5.3.Diagramas de Gantt	38
6.Desarrollo	44
6.1.Casos de uso	44
6.1.1.Explicación de los casos de uso	45
6.1.1.1.Sign Up	45
6.1.1.2.Log In	45
6.1.1.3.Profile	46
6.1.1.3.1.Personal Information	46
6.1.1.3.2.Manage your flats	47
6.1.1.4.Find	47
6.1.1.4.1.Find flat	47
6.1.1.4.2.Search in backpack	48
6.1.1.5.Offer	48
6.1.1.5.1.Add flat	48
6.1.1.5.2.Add me to backpack	49
6.1.1.5.3.Remove me from the backpack	50
6.1.1.6.Friends	50
6.1.1.6.1.Add friend	50
6.1.1.6.2.Friend requests	51
6.1.1.6.3.Your friends	51
6.2.Diseño gráfico de la aplicación	52
6.3.Multiplataforma	54

6.3.1.Config.xml	55
6.3.1.1.Propiedades esenciales	55
6.3.1.2.Selección de la plataforma de compilación	56
6.3.2.Android	57
6.3.3.iOS	57
6.3.3.1.Requerimientos	57
6.3.4.Windows Phone 8	58
6.3.5.Symbian y WebOs	58
6.3.6.BlackBerry10	58
7.Conclusiones y trabajo futuro	60
7.1.Errores y posibles mejoras	60
7.1.1.Errores a mejorar	61
7.1.2.Posibles mejoras	61
7.1.3.Trabajo futuro	62
8.Bibliografía	64
9.Anexo	65
9.1.Preparar el entorno	65
9.1.1.Configurar XAMPP	65
9.1.2.Crear un proyecto PhoneGap con Eclipse	67
9.2.DynDNS	69
9.3.Manual de usuario	69
9.4.Instalar el SDK de iOS y Apache Cordova	82
9.4.1.Crear nuevo proyecto	82
9.4.2."Testing" en Emulador	83

Abstract

It is proposed to build a service around PhoneGap/ Apache Cordova free license framework that can show all the features it can do in a software level and the easiness creating an app or a service by using HTML + CSS + PHP and AJAX, and provide it like a cross-platform application without making any code changes. To show this, we have built a flat-sharing application called PhoneFlat designed primarily for college students that are looking for a temporary residence in the desired city or town, although it can be used by other kind of users too. This service will show the different resources that PhoneGap/Apache Cordova use from smartphone devices like: camera, internal storage or GPS-Location and the use of them to complete the goals. Once finished the implementation, will be shown how this framework generates all the executable files for multiple operating systems on the market such as Android, iOS or Windows Phone, using a build system in the cloud called PhoneGap Build, distributed by Adobe Systems. It has been considered to use English language thinking to attract maximum number of users.

Resumen

Se propone construir un servicio alrededor del framework de licencia libre PhoneGap/Apache Cordova, que pueda mostrar los recursos a nivel de software que puede abarcar y la facilidad de poder crear una aplicación o servicio mediante lenguajes de programación web, y poder ofrecer-lo en multiplataforma sin la necesidad de modificar código o de codificarlo nuevamente para conseguirlo. Para mostrarlo, se ha construido un servicio de compartimiento de pisos, pensado principalmente para estudiantes de universidad o de otros estudios, que busquen una residencia temporal en la ciudad o pueblo deseado. Este servicio mostrará los diferentes recursos que aprovecha PhoneGap/Apache Cordova de los smartphones, tales como la cámara, almacenamiento o la geolocalización y hará uso de ellos para conseguir los objetivos propuestos. Una vez terminada la implementación, se mostrará como este framework genera de manera automática un ejecutable para varios sistemas operativos presentes en el mercado tales como Android, iOS o Windows Phone 8, mediante un sistema de compilación en la nube llamado PhoneGap Build, distribuido por Adobe Systems. Se ha considerado adecuado ofrecer una interfaz totalmente en inglés, pensando en atraer al máximo número de usuarios.

1.- Introducción

Actualmente, hay mucha oferta en cuanto a apps para dispositivos smartphones se refiere, ya que el mercado cada vez evoluciona de manera más rápida y los dispositivos portátiles ofrecen cada vez mayores servicios necesarios o imprescindibles en nuestro día a día. Es por ello, que las empresas otorgan facilidades a los desarrolladores para motivarles a desarrollar para su plataforma o Sistema Operativo.

El problema de los servicios que se quieren trasladar a los dispositivos móviles, es que precisan de diferentes lenguajes de programación para poder ser ejecutados en las diferentes plataformas del mercado, y eso conlleva a proceder a varios desarrollos independientes para un mismo fin.

1.1.- Definición de objetivos

El objetivo principal del proyecto es mostrar la capacidad de generar un servicio multiplataforma mediante un mismo código, haciendo uso de un framework PhoneGap/Apache Cordova.

Se desea mostrar el potencial que posee este framework y sus servicios en la nube para la compilación de diferentes tipos de archivos ejecutables, así como las configuraciones necesarias para poder desarrollar código mediante esta tecnología.

Para ello, se ha montado un servicio alrededor de PhoneGap/Apache Cordova que sea útil para cualquier usuario que desee buscar cualquier tipo de residencia temporal, ya sea por estudios o trabajo, y pueda ponerse en contacto de manera rápida y sencilla, con cualquier persona que este dispuesta a compartir piso desde su dispositivo smartphone o tablet.

1.2.- Motivación

La motivación nace al descubrir que hay otras maneras de poder desarrollar aplicaciones para ejecutar ya no solo en smartphones Android, sino en diferentes sistemas operativos actuales, haciendo uso de lenguajes de programación web. Hasta poco antes del comienzo de la realización del proyecto, no me atraían los lenguajes de programación web, ya que nunca pensé en querer desarrollar páginas web o servicios online, pero poco antes de la elección del proyecto, descubrí las grandes ventajas que ofrecía PhoneGap mediante uso de lenguajes de programación web, aplicables a dispositivos móviles y soportando la multiplataforma, algo que me atrajo mucho a investigar sobre ello.

Sobre el desarrollo de PhoneFlat, la motivación nace al observar que en muchas universidades existen tabloneros donde personas se anuncian con el fin de encontrar residencia, o poder compartirla con alguien que esté interesado.

El problema surge que al ser un anuncio colgado en un tablón físico de una universidad, hace que todo el proceso de contacto sea muy lento y se asume que el anuncio no será visto por una gran cantidad de gente, además del hecho de tener que exponer al público tu número de contacto o datos personales, ya que no todo el mundo se siente cómodo al proporcionarlos de manera directa y sin previo contacto con las personas involucradas.

2.- Tecnología utilizada

2.1.- Servidor XAMPP Apache + MySQL + PHP + Perl

PhoneFlat se ha desarrollado bajo el servicio de XAMPP (Apache + MySQL + PHP + Perl). Se ha optado por esta tecnología, ya que es necesario proporcionar al servicio de hosting donde almacenar la información y realizar consultas SQL.

La elección de XAMPP contra sus otras versiones compiladas en otros Sistemas Operativos (WAMP - Windows, LAMP - Linux, MAMP – Mac OS), se relaciona con toda la experiencia obtenida desarrollando otros proyectos didácticos para la “Universidad de Barcelona”, y además ofrece las funcionalidades de almacenamiento y consulta de datos demandados para el servicio que se quiere desarrollar, haciendo uso de cualquier sistema operativo ya que XAMPP puede ser instalado y ejecutado en cualquier Sistema Operativo.

El manual de usuario para realizar las instalaciones oportunas se encuentran en el anexo del documento.

2.2.- Eclipse

Eclipse es un entorno integrado de desarrollo (IDE), compuesto por un conjunto de herramientas de programación de código abierto multiplataforma, utilizado típicamente para el desarrollo de software. Este IDE cuenta con un editor de código, un compilador y un depurador.



Figura 1

En nuestro caso, hemos utilizado este entorno ya que nos ofrece la posibilidad de poder codificar nuestro código HTML haciendo uso de los recursos que nos ofrece Apache Cordova y PhoneGap. A continuación se muestra como configurar el entorno para poder empezar a trabajar.

Enlace de descarga: <http://www.eclipse.org/downloads/>

2.3.- NODE.JS

Node.js es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos. Soporta protocolos TCP, DNS y HTTP.

Tendremos que hacer uso de este entorno JavaScript , ya cuenta con un gestor de paquetes que nos facilitará los siguientes pasos.



Figura 2

2.4.- Android SDK

Es el Plugin oficial de Android, que ofrece las herramientas necesarias para programar una aplicación Android desde el entorno Eclipse. En el SDK podemos encontrar el emulador que también ha sido utilizado para realizar pruebas.

Lo podemos descargar desde el siguiente enlace:

Android SDK: <http://developer.android.com/sdk/index.html>

2.5.- ADT Plug-in

Android Development Tools (ADT) es un plug-in para Eclipse que ha sido diseñado para ofrecer al usuario un potente entorno integrado donde construir y compilar aplicaciones Android. Será necesario para poder crear cualquier aplicación Android mediante Eclipse.

ADT Plugin: <http://developer.android.com/tools/sdk/eclipse-adt.html>

2.6.- Frameworks multiplataforma

Actualmente se disponen de varios frameworks o entornos para el desarrollo de código orientado a la multiplataforma. Con tantas opciones, es preciso escoger que plataforma es la adecuada para desarrollar un código, que sea fácil de exportar al mayor número de dispositivos, sin tener que realizar variaciones.

El proyecto propuesto, marcaba las pautas para poder descubrir el potencial de desarrollo que ofrece PhoneGap/Apache Cordova, pero no es la única tecnología disponible a día de hoy.

2.6.1- Titanium Appcelerator

Titanium Appcelerator es una plataforma para desarrollar aplicaciones móviles y de escritorio utilizando tecnologías web.

Está desarrollado por Appcelerator Inc. y fue lanzado en diciembre del 2008. En junio de 2009 se añadió soporte para el desarrollo de aplicaciones móviles para Android e iPhone.



Figura 3

En abril de 2010 se añadió soporte para el desarrollo de aplicaciones para Ipad.

Se distingue de otros frameworks en que genera aplicaciones nativas en lugar de aplicaciones que se ejecutan dentro de un navegador.

2.6.2.- Strawberry

Para el desarrollo de aplicaciones en Strawberry se utilizan dos lenguajes de programación: C++ para el desarrollo de toda la lógica del programa y HTML/CSS para definir el diseño. De esta manera se separa completamente el diseño de la programación y facilita la separación de roles: diseño y



Figura 4

desarrollo.

Strawberry utiliza la aceleración por hardware OpenGL y está programado íntegramente en C++. Dispone de su propio renderizado lo que ayuda a que las aplicaciones se visualicen igual en todas las plataformas y dispositivos.

2.6.3.- Apache Cordova/PhoneGap

PhoneGap es un framework para el desarrollo de aplicaciones móviles multiplataforma. Gracias a PhoneGap podemos crear, mediante un mismo código fuente, una aplicación que luzca y se comporte igual tanto en Android, como en IOS, como en Windows Phone como en Symbian y BlackBerry10. Para ello este framework hace uso de tecnologías web como son HTML5, CSS3 y javascript. El código genérico es compilado obteniéndose los ejecutables para cada uno de los sistemas móviles soportados. PhoneGap utiliza API que permiten el acceso a elementos de los dispositivos en los que se ejecutará, tales como el acelerómetro o la cámara.

PhoneGap es gratuito, libre, y multiplataforma. Fue creado por la empresa Nitobi. En noviembre de 2011 Adobe anunciaba la compra de Nitobi y de esta forma el control sobre PhoneGap pasaba a manos de Adobe. El código de PhoneGap fue donado a la fundación Apache, manteniéndose así el corazón del proyecto como libre. Así nació Apache Cordova, pero aun así PhoneGap sigue existiendo, y puede ser considerada como una distribución de Apache Cordova.

Las aplicaciones generadas por este servicio no son realmente aplicaciones nativas al dispositivo, ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema, pero no se tratan tampoco de aplicaciones web, ya que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema nativo.

2.6.3.1.- Diferencias entre Apache Cordova y PhoneGap

Adobe pretende que PhoneGap sea una versión "enriquecida" de Cordova. Se puede considerar a Cordova como el núcleo de PhoneGap aunque actualmente ambos servicios frameworks proporcionan los mismos recursos al desarrollador.

Así pues, se considera que PhoneGap es el nombre comercial con el que sale a la luz Apache Cordova tras ser comprada por Adobe.

2.6.3.2.- Licencia

Actualmente tanto PhoneGap como Cordova son gratuitos, libres y multiplataforma.

PhoneGap concretamente, se distribuye bajo licencia Apache 2.0. Como cualquier otra de las licencias de software libre, la licencia Apache permite al usuario del software la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo, y distribuir versiones modificadas de ese software.



Figura 5

La Licencia Apache no exige que las obras derivadas (versiones modificadas) del software se distribuyan usando la misma licencia, ni siquiera que se tengan que distribuir como software libre/open source. La Licencia Apache sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache.

Estos dos servicios se pueden encontrar mediante los siguientes links de descarga:

Apache Cordova: <https://www.apache.org/dist/cordova/>

PhoneGap: <http://PhoneGap.com/install/>

2.6.3.3.- Plataformas soportadas

	Amazon- fireos	Android	blackberry10	Firefox OS	Ios	Ubuntu	WP7 (Windows Phone 7)	WP8 (Windows Teléfono 8)	win8 (Windows 8)	Tizen
Cordova CLI	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓ Windows	✓	✗
Incrustado WebView	✓ (ver detalles)	✓ (ver detalles)	✗	✗	✓ (ver detalles)	✓	✗	✗	✗	✗
Plug-in Interfaz	✓ (ver detalles)	✓ (ver detalles)	✓ (ver detalles)	✗	✓ (ver detalles)	✓	✓ (ver detalles)	✓ (ver detalles)	✓	✗
Plataforma API										
Acelerómetro	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cámara	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Captura	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
Brújula	✓	✓	✓	✗	✓ (3GS +)	✓	✓	✓	✓	✓
Conexión	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Contactos	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Dispositivo	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Eventos	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Archivo	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Geolocalización	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Globalización	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗
InAppBrowser	✓	✓	✓	✗	✓	✓	✓	✓	(partial) utiliza el iframe	✗
Los medios de comunicación	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Notificación	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
SplashScreen	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Almacenamiento de información	✓	✓	✓	✗	✓	✓	localStorage & indexedDB	localStorage & indexedDB	localStorage & indexedDB	✓

Figura 6

2.6.3.4.- PhoneGap Build

Herramienta construida para ser utilizada en la nube, dando la posibilidad de compilar una aplicación web móvil desarrollada para una plataforma concreta mediante PhoneGap, en otras a las que da soporte.

Hasta la versión 2.9 de Cordova/PhoneGap, PhoneGap Build ofrecía soporte para: Android, IOs, Windows Phone 8, Blackberry5/6/7, Symbian. Actualmente y tras actualizar sus versiones a la 3.0, PhoneGap Build dejó de dar soporte a varias plataformas móviles que están cayendo en la decadencia, y se centra básicamente en tres de ellas, abarcando

igualmente un amplio sector del mercado móvil:

- IOS
- Android
- Windows Phone 8

Se ha elegido desarrollar PhoneFlat en PhoneGap, ya que permite al usuario familiarizarse más con los hábitos de programación web y ofrece grandes posibilidades explotando los recursos de los dispositivos móviles:

- Acelerómetro
- Brújula
- Geolocalización
- Multimedia
- Cámara
- Almacenamiento
- Archivos
- Contactos

2.7.- Emuladores y dispositivos físicos utilizados

Para poder desarrollar una aplicación o servicio de manera eficiente, es necesario probar cada paso que se evoluciona en el proyecto, pero no siempre se dispone de un dispositivo capaz de ejecutar el código desarrollado. Es por eso que se debe hacer uso de emuladores que nos permitan ver el comportamiento del front-end (interficie) y hacer test sin necesidad de hacer un deployment en el dispositivo.

2.7.1.- Emulador Android

Android ADT Plugin nos proporciona una herramienta donde crear un simulador para dispositivos Android. Está integrado en Eclipse IDE una vez se ha instalado, y nos ofrece la posibilidad de probar los archivos .apk generados por el entorno de programación.

Para el desarrollo de PhoneFlat, se ha utilizado un emulador con la siguiente configuración:

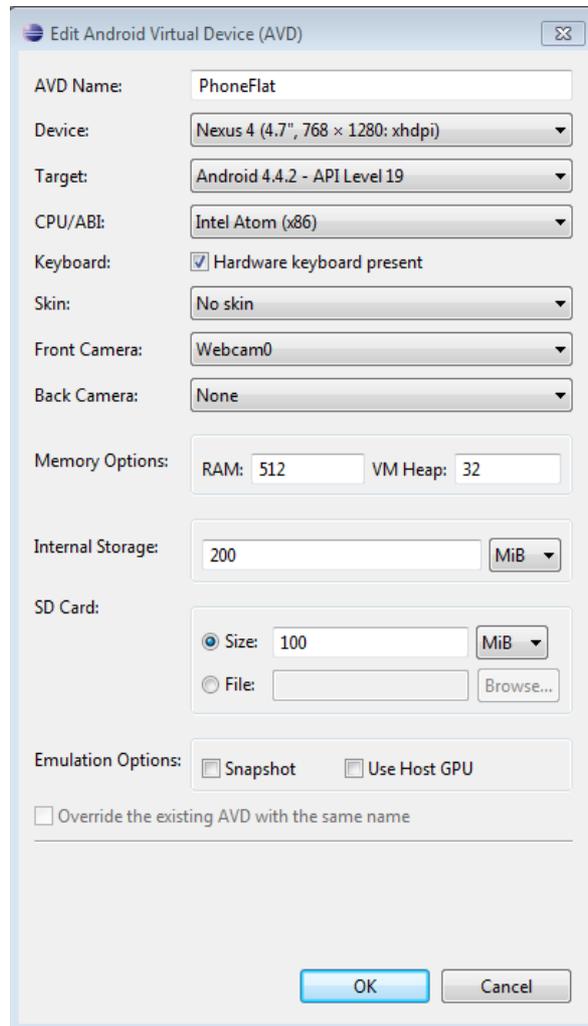


Figura 7

Para realizar las pruebas con la cámara, se ha dispuesto de una cámara web con conexión USB, incorporada al emulador para poder realizar las capturas de imagen en tiempo real.

2.7.2.- PhoneGap Emulator (Ripple)

El emulador de PhoneGap permite probar las aplicaciones desarrolladas desde el Escritorio.

Permite varias plataformas y diferentes versiones de estas:

Plataformas:

- Apache Cordova/PhoneGap
- BlackBerry10 WebWorks
- WebWorks
- WebWorks – Tablet OS
- Mobile Web

Ya que actualmente PhoneGap/Apache Cordova no da soporte a BlackBerry10, se considera como plataforma independiente en términos de emulación para Ripple, aunque permite su emulación debido a que hasta la versión 2.9 de PhoneGap/Apache Cordova si contaba con ese soporte por parte del framework y puede ser probada para las versiones 1.0.0 y 2.0.0.

2.7.3.- Dispositivos físicos

Para realizar las pruebas finales, se ha dispuesto de un smartphone, para poder corroborar que los datos de entrada/salida y visualización de PhoneFlat, se corresponden a los del emulador en un entorno real.

Características del dispositivo:

Dimensiones:

139 x 71 x 7,9 mm

5,47 x 2,79 x 0,31 pulgadas

Display:

TFT de 5 pulgadas

16 millones de colores
1920 x 1080 píxeles

Interior:

Google Android 4.4
Procesador de cuatro núcleos



Figura 8

3.- Estudio de mercado

Antes de embarcarse en una aplicación o proyecto, debemos observar el mercado que nos envuelve para ver que posibilidades nos ofrece, y si ya nos está facilitando las herramientas que nosotros queremos desarrollar.

Será lo más común, que ya existan otros servicios muy parecidos al que nosotros queremos desarrollar, y es ahí, donde debemos analizar qué es lo que podemos mejorar de los servicios actuales, que hagan de nuestra aplicación esté por encima de las demás.

3.1.- Análisis de la competencia

Fijémonos primero en que aplicaciones ofrecen un servicio similar:

yaencontre.com

Aplicación con una interfaz sencilla y clara, que ofrece buena navegación a través de sus menús, pero que dispone de demasiada información respecto a sus ofertas. No se trata solo de una aplicación de compartimento de pisos. Al acceder al menú del tipo del tipo de inmueble que deseamos buscar, nos aparecen viviendas, terrenos, oficinas,naves..



Figura 9

En otro tipo de servicio sería interesante contar con todas estas posibilidades y ofertas, pero se aleja de lo que PhoneFlat busca, que es la rapidez, sencillez y eliminación de información y datos sobrantes, para que nuestros usuarios no se compliquen a la hora de buscar.

Otro aspecto que suele repetirse en todas las aplicaciones de esta temática, es la

necesidad de ofrecer tus datos personales para tener que contactar con los usuarios ofertantes. Más tarde explicaremos como PhoneFlat consigue evitar esto.

Finalmente, encontramos que yaencontre.com, dispone de nuevas versiones lanzadas el 29/05/2014 para Android e IOs, a parte de su versión en página web.

Pisos.com

Aplicación con una interfaz sencilla también, con buena navegación pero muy parecida a yaencontre.com. Dispone de buen sistema de geolocalización para encontrar las ofertas pero dispone también, de gran número de ofertas proporcionadas por inmobiliarias que distraen a los usuarios de su verdadero objetivo. Esto hace que las búsquedas sean mas lentas y torpes, ya que vas a tener que lidiar con muchos anuncios que no ofrecen ni lo que estas intentando buscar.



Figura 10

Para poder contactar con los ofertantes, el sistema ofrece públicamente el número de teléfono que ha proporcionado el usuario. También ofrece la posibilidad de enviarle una solicitud de contacto al correo, cómo en la mayoría de servicios, que hace todo mucho más lento.

Pisos.com ofrece su servicio en página web, Android e IOs, al igual que la anterior que hemos visto.

Segundamano.es

Análisis obligado a esta aplicación, que sin duda cuenta con una cantidad de usuarios enorme, pero no se acerca para nada a nuestros objetivos, debido a la gran cantidad de ofertas que ofrece, de diferentes tipos de temática.

Aún así, cuenta con un apartado interesante donde ofrece anuncios para pisos compartidos. El servicio obliga a poner tu email y tu número, que serán enseñados públicamente para que la gente pueda contactar contigo vía correo, SMS o mediante una llamada.

Soporta Android, Web y IOs.

Bolsa Pisos

Interfaz algo sencilla con colores oscuros, que ofrece la posibilidad de comprar y alquilar todo tipo de residencias, pero no ofrece la posibilidad a los usuarios de añadir sus propias ofertas de compartimento o alquiler de viviendas.

Pues sabiendo esto, vemos que se aleja mucho del planteamiento que ofrece PhoneFlat, y no la podríamos considerar competencia directa, aunque podría estar dirigida a un pequeño sector de nuestros usuarios.

Bolsa pisos, sólo ofrece sus servicios en la plataforma Android y página web, desaprovechando las oportunidades que ofrece la multiplataforma.



Figura 11

Acabando el análisis de unas cuantas aplicaciones que pueden estar relacionadas, se han omitido muchas de ellas que solo ofrecen sus servicios en página web, ya que no es el sector en el que se va a mover PhoneFlat, y es más real poder analizar aquellas que se mueven en sus mismas plataformas.

Ahora vamos a ver que es lo que ofrece PhoneFlat, aunque se hará un análisis más exhaustivo más adelante en este mismo documento.

PhoneFlat

PhoneFlat dispone de una interfaz sencilla, fácil para el usuario y rápida para encontrar las ofertas que buscas. Es una aplicación donde interaccionarás con otros usuarios que buscan piso para compartir, o compañeros para su piso actual.

El servicio ofrece un sistema de registro por usuario y contraseña, que serán los únicos datos que deberás ofrecer ya que dispone de un sistema de mensajería instantánea, que permite contactar con otros usuarios sin la necesidad de ofrecer tu número personal, ni tu nombre y apellidos, y que posibilita la opción de empezar a hablar de manera inmediata, sin tener que rellenar formularios ni solicitudes que tardan más tiempo.

PhoneFlat está disponible para las plataformas Android, IOs, Windows Phone 8 y Symbian.

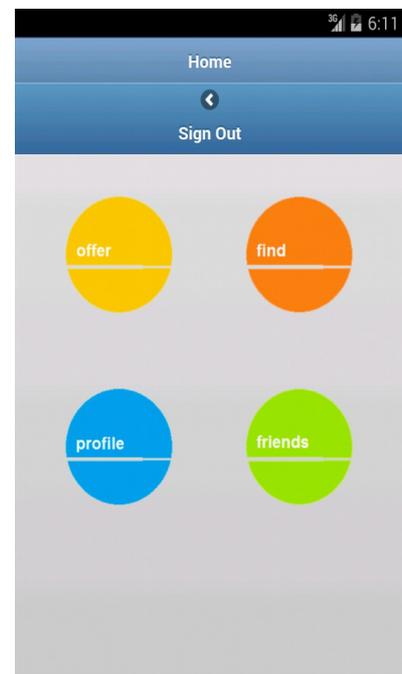


Figura 12

3.2.- Conclusión del análisis

Sencillez: Todas las aplicaciones analizadas, cuentan con una interfaz sencilla que proporciona sencillez al usuario para navegar. Algunas de ellas sobrecargan el sistema con muchos filtros o ofertas que no son de interés para los usuarios que PhoneFlat quiere

atraer.

Plataformas

Todas las plataformas que se consideran posibles competidores, disponen de sus versiones en multiplataformas al igual que PhoneFlat, aunque esta última dispone de versiones tanto en Windows Phone 8 cómo en BlackBerry10, ampliando el rango de usuarios.

Búsqueda

Las búsquedas son de carácter muy parecido en todas las apps vistas, aunque algunas como pisos.com, disponen de búsqueda por geolocalización GPS, funcionalidad que PhoneFlat tenía casi incorporada, pero que hubo que rechazarla por falta de tiempo. Pero PhoneFlat incluye algo que los demás no hacen, y es una búsqueda de personas por el sistema de “mochila”, que más adelante se explicará en que consiste.

Privacidad

Otro punto a favor de PhoneFlat. Mientras que todos los servicios solicitan tu información personal para añadirte a sus bases de datos, PhoneFlat requiere únicamente un nick de usuario y una contraseña para empezar a buscar ofertas y a proponerlas. Eso garantiza que los usuarios puedan sentirse más confiados a utilizar el servicio, ya que tienen la seguridad de que nadie desconocido va a saber su número personal.

Método de contacto

PhoneFlat vuelve a estar un paso por delante de sus competidores, al ofrecer un sistema de mensajería instantánea, que permita abstenerse de enviar formularios de contacto, y agilice mucho más las cosas.

PhoneFlat pues, se posiciona muy bien en cuanto a necesidades requieren los usuarios, y ofrece algo que no se ve en el mercado, una aplicación exclusiva de compartimento de pisos, donde poder contactar de manera independiente por mensajería instantánea y sin ofrecer datos personales.

4.- Base de datos

4.1.- Estructura

PhoneFlat requiere de conectividad hacia elementos externos a la aplicación alojados en la nube, así que se precisa poder almacenar toda la información en una base de datos accesible de manera remota.

Un pequeño esquema de los recursos que utiliza la aplicación para lograrlo se ilustra en la siguiente figura:

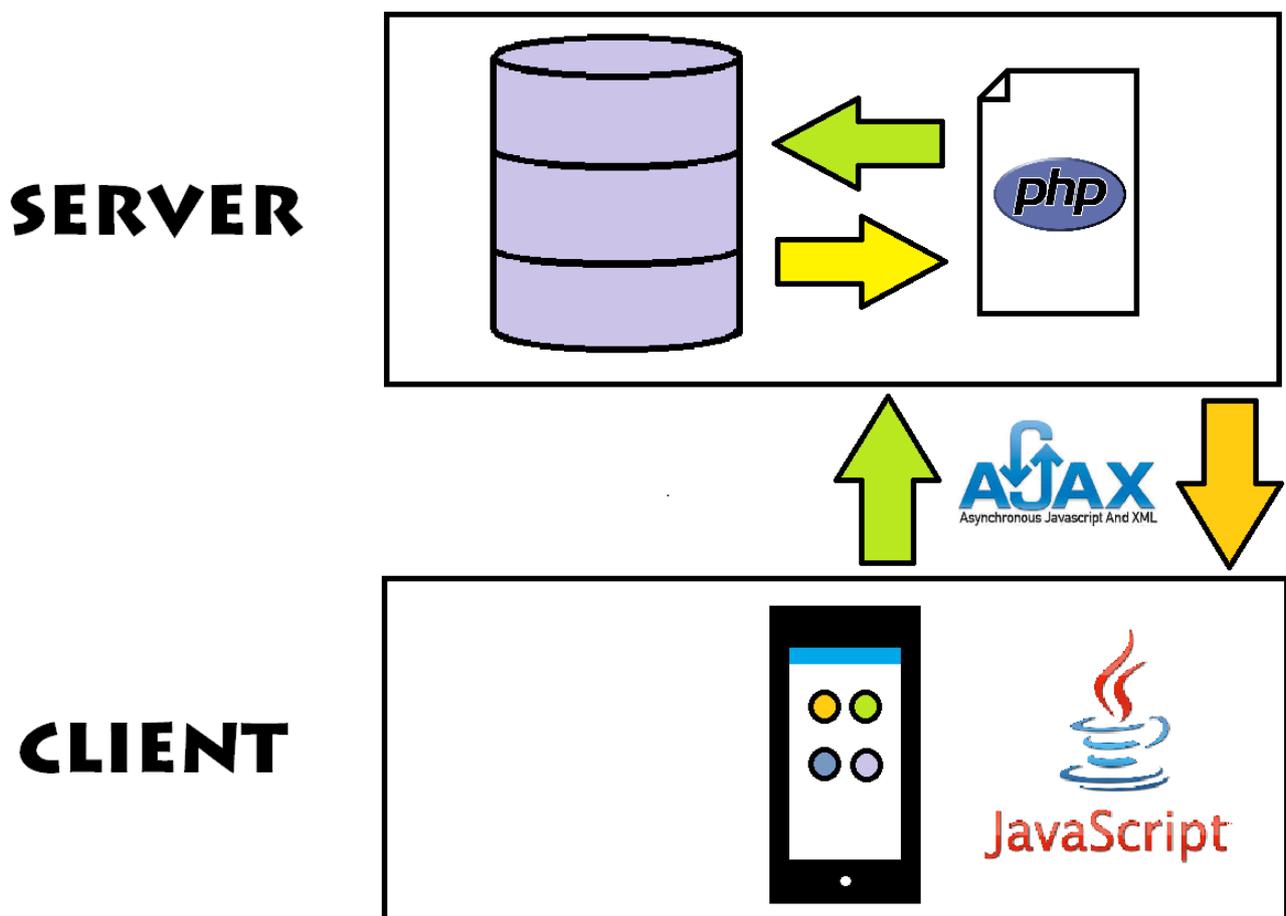


Figura 13

Disponemos de dos estructuras claras funcionando en nuestro servicio:

- Cliente
- Servidor

4.1.1.- Cliente

La parte del cliente son todos aquellos recursos a los que puede acceder cada usuario de manera local.

Todo lo que el usuario posee es una aplicación instalada en su smartphone, que utiliza como vía para conectarse a los servidores y obtener toda aquella información que precise.

Esta conexión es posible gracias a javascript, que es un lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de AJAX, que incorpora el código de PhoneFlat y que nos ayudará a conectarnos al servidor haciendo uso de PHP. Esto hará que se puedan mostrar los resultados de manera dinámica sin hacer uso del refresco manual.

4.1.2.- Servidor

En la parte del servidor, encontramos lo siguiente:

- Base de datos
- Ficheros PHP

A continuación se explicarán las tablas que forman la base de datos y su función dentro del servicio.

También como donde se almacenan los ficheros PHP y las imágenes subidas por los usuarios.

4.2.- Vista relacional

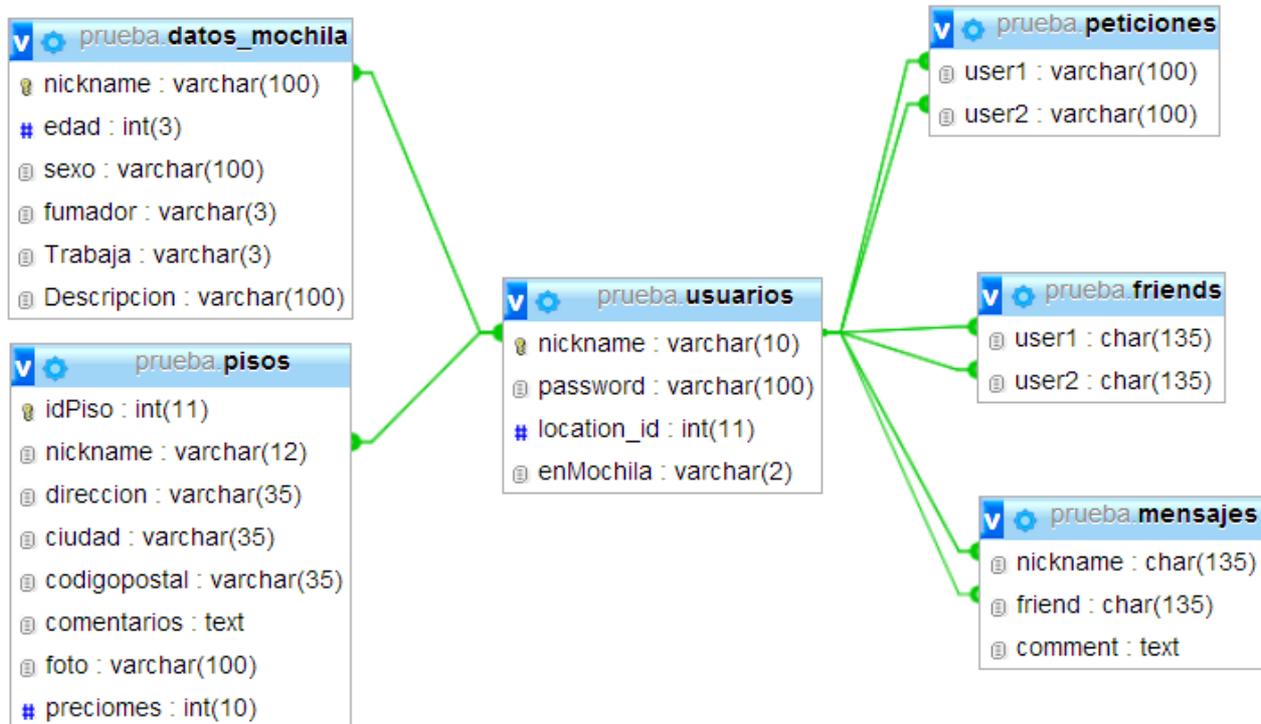


Figura 14

Cómo se observa, el atributo nickname de la tabla usuarios está presente en todas las tablas, y se verá modificado si se altera su condición.

4.2.1.- Tablas

- Usuarios: Contiene el nombre los nombres de usuario y contraseña de todo aquel que utilice la aplicación.
- Pisos: Contiene toda la información de cada anuncio de piso, asignándolo a un usuario en concreto.
- Peticiones: Contiene todas las peticiones de amistad que se envíen entre usuario.

- Friends: Contiene todas las relaciones de amistad que se han aceptado, y que por lo tanto vinculan como amigos a dos usuarios.
- Mensajes: Contiene todos los mensajes que se envían entre usuarios.
- Datos_Mochila: Contiene la información pública de cada usuario, lista para ser mostrada en caso de ser buscado.

4.2.2.- Atributos

Usuarios

<u>ATRIBUTO</u>	<u>TIPO_VALOR</u>	<u>INFORMACIÓN QUE ALMACENA</u>
nickname	varchar(10)	Nombre de usuario del cliente
password	varchar(10)	Contraseña del cliente
enMochila	varchar(2)	Si se encuentra en la mochila*

Pisos

<u>ATRIBUTO</u>	<u>TIPO_VALOR</u>	<u>INFORMACIÓN QUE ALMACENA</u>
idPiso	int(11)	Identificador del nuevo anuncio
propietario	varchar(12)	Nombre de usuario del ofertante
direccion	varchar(35)	Dirección del piso
ciudad	varchar(35)	Ciudad dónde se localiza el piso
codigopostal	varchar(35)	Código postal del piso
comentarios	text	Información del piso
preciomes	int(4)	Precio del alquiler a pagar

Peticiones

<u>ATRIBUTO</u>	<u>TIPO_VALOR</u>	<u>INFORMACIÓN QUE ALMACENA</u>
user1	varchar(10)	Usuario que realiza la petición
user2	varchar(10)	Usuario que recibe la petición

Amigos

<u>ATRIBUTO</u>	<u>TIPO_VALOR</u>	<u>INFORMACIÓN QUE ALMACENA</u>
user1	varchar(10)	Nombre de usuario
user2	varchar(10)	Nombre de usuario

Mensajes

<u>ATRIBUTO</u>	<u>TIPO_VALOR</u>	<u>INFORMACIÓN QUE ALMACENA</u>
user1	varchar(10)	Usuario del emisor del mensaje
friend	varchar(10)	Usuario del receptor del mensaje
comment	text	Texto que envía user1 a friend

Datos_mochila

<u>ATRIBUTO</u>	<u>TIPO_VALOR</u>	<u>INFORMACIÓN QUE ALMACENA</u>
nick	varchar(10)	Nombre de usuario
edad	int(3)	Edad del usuario
sexo	varchar(100)	Especifica su género sexual
fumador	varchar(2)	Especifica si es una persona fumadora
Trabaja	varchar(2)	Especifica si tiene trabajo actualmente
Descripción	varchar(100)	Descripción personal del usuario

4.2.3.- Ficheros PHP

Los ficheros PHP se encuentran en el ordenador personal en el que se ha instalado XAMPP, dentro de la misma carpeta `xampp/htdocs`, asignando posteriormente otra carpeta en este directorio, a la que nombrar con el nombre del proyecto deseado.

Todos los ficheros PHP realizan su conexión a la base de datos mediante este código:

```
$con=mysql_connect("localhost","root","opsivir")  
or die("Error: ".mysql_error());  
$name = 'prueba';  
$db = mysql_select_db($name,$con)  
or die("Error: ".mysql_error());
```

Figura 15

Como se puede apreciar en la primera línea de código, xampp hace uso de usuario y contraseña para poder acceder a la base de datos, asegurando la integridad de los datos para que no sean vulnerados tan fácilmente.

En la variable `'$name'` se observa que el nombre de la base de datos utilizada responde a 'prueba', ya que cuando se empezó a desarrollar el proyecto se le asignó ese nombre para realizar pruebas y finalmente se ha mantenido con ese nombre, ya que no hay otras bases de datos que puedan confundirse.

4.2.4.- Almacenamiento de imágenes

Para poder almacenar las imágenes, se ha optado por crear un directorio independiente donde se ubican los ficheros PHP llamado `uploads`, que almacenará cada imagen que decida subir cada usuario.

En este directorio, se generan tantos subdirectorios (de manera automática) como usuarios se registren a la aplicación, guardando en cada uno de ellos la imagen de perfil propia de cada usuario. Cada vez que se actualice la foto de perfil, se borrará la antigua, almacenando tan solo una fotografía por perfil. En el caso de que no se suba ninguna imagen de perfil, PhoneFlat cargará la imagen visible en la figura X defecto:



Figura 16

Para las fotos de los pisos se aplicará el mismo método. Se creará un subdirectorío dentro de “uploads” con el nombre del id del piso añadido, y dentro se almacenarán las fotografías que vaya subiendo el propietario del piso.

Esta sería una visualización del directorio de los ficheros:

Aplicación Web-Móvil PhoneGap/Apache Cordova

 uploads	15/06/2014 21:17	Carpeta de archivos	
 addFriend.php	01/06/2014 0:04	Archivo PHP	2 KB
 addIntoBagpack.php	06/06/2014 20:54	Archivo PHP	2 KB
 borrarFoto.php	07/06/2014 22:50	Archivo PHP	1 KB
 buscarPisobyFilter.php	15/06/2014 21:31	Archivo PHP	3 KB
 friends.php	09/06/2014 13:05	Archivo PHP	2 KB
 guardar.php	15/06/2014 21:45	Archivo PHP	1 KB
 login.php	13/06/2014 20:05	Archivo PHP	2 KB
 mostrarImagenes.php	08/06/2014 13:41	Archivo PHP	2 KB
 mostrarImagenPerfil.php	16/06/2014 20:12	Archivo PHP	1 KB
 mostrarImagenPiso.php	21/04/2014 13:50	Archivo PHP	1 KB
 mostrarPhotos.php	28/11/2013 5:12	Archivo PHP	1 KB
 peticionesAmistad.php	30/05/2014 16:22	Archivo PHP	2 KB
 recibirCiudadMiPiso.php	05/05/2014 21:56	Archivo PHP	1 KB
 recibirComentariosMiPiso.php	05/05/2014 21:57	Archivo PHP	1 KB
 recibirCPMiPiso.php	05/05/2014 21:57	Archivo PHP	1 KB
 recibirDeMochila.php	09/06/2014 17:13	Archivo PHP	3 KB
 recibirDescripcion.php	22/04/2014 9:33	Archivo PHP	1 KB
 recibirDireccionMiPiso.php	05/05/2014 21:57	Archivo PHP	1 KB
 recibirEdad.php	22/04/2014 9:33	Archivo PHP	1 KB
 recibirFumador.php	22/04/2014 9:33	Archivo PHP	1 KB
 recibirInfoMiPiso.php	05/05/2014 21:57	Archivo PHP	1 KB
 recibirLocalizacion.php	10/02/2014 16:05	Archivo PHP	1 KB
 recibirMensajes.php	15/06/2014 21:47	Archivo PHP	1 KB
 recibirMisPisos.php	12/06/2014 18:25	Archivo PHP	2 KB
 recibirOwnerPiso.php	15/06/2014 21:34	Archivo PHP	1 KB
 recibirPeticiones.php	08/06/2014 18:09	Archivo PHP	2 KB
 recibirSexo.php	22/04/2014 9:34	Archivo PHP	1 KB
 recibirTrabaja.php	22/04/2014 9:34	Archivo PHP	1 KB
 rellenarDesplegables.php	09/02/2014 1:23	Archivo PHP	1 KB
 removeFriend.php	09/06/2014 23:03	Archivo PHP	1 KB
 subi.php	29/10/2013 11:43	Archivo PHP	1 KB
 subirFotoPerfil.php	16/06/2014 20:09	Archivo PHP	3 KB
 subirFotoPiso.php	12/05/2014 19:16	Archivo PHP	1 KB
 subirInfo.php	31/05/2014 18:46	Archivo PHP	2 KB
 subirPiso.php	15/06/2014 20:32	Archivo PHP	2 KB
 subirPisoModificado.php	15/06/2014 20:35	Archivo PHP	1 KB
 subirUser.php	13/06/2014 20:02	Archivo PHP	2 KB
 testImages.php	10/02/2014 16:38	Archivo PHP	1 KB
 testing.php	06/04/2014 21:31	Archivo PHP	1 KB
 testLogin.php	07/02/2014 22:56	Archivo PHP	2 KB

Figura 17

5.- Planificación

Para realizar el proyecto, primero se tuvo que planificar cuanto se pretendía tardar en realizar cada tarea del mismo, haciendo una aproximación válida para poder tener todos los aspectos del proyecto a tiempo para su entrega.

Mediante la realización del mismo, fueron surgiendo algunas funcionalidades que no se consideraban en un inicio y se fueron eliminando otras que no se creían útiles ni necesarias de implementar.

Debido a problemas con la familiarización del entorno y la variación de ideas en torno al servicio que construir alrededor, se tuvo que posponer la entrega del proyecto al segundo semestre.

En un principio, se pretendía presentar las posibilidades que puede ofrecer PhoneGap/Cordova en cuanto a desarrollo multiplataforma, y se implementó un servicio que funcionara como repositorio multimedia, incorporando sistemas de geolocalización y servicio de mensajería instantánea, pero se consideró que se debía construir un servicio más grande, capacitado de mostrar de las posibilidades de este framework.

Finalmente, se decidió montar un servicio de compartimiento de pisos, utilizando la experiencia obtenida en los meses previos y utilizando parte del servicio ya codificado.

5.1.- Primera etapa

Esta primera etapa se comprende entre Setiembre y Enero, el primer semestre del curso lectivo 2013/2014. Esta abarca todo el proceso de familiarización con el entorno y asignación de objetivos previstos para el proyecto.

Debido a que no se tenía una idea clara sobre que mostrar acerca de PhoneGap/Cordova, se consideró oportuno posponer la defensa hasta tener claro que servicio se quería construir alrededor de esta distribución de Apache.

5.1.1- Declaración de objetivos

Los objetivos marcados pretendían poder recuperar datos de otros dispositivos smartphones en tiempo real, utilizando Apache Cordova, o en este caso una distribución suya llamada PhoneGap, de licencia libre.

Una vez definida que herramienta utilizar, se investigó acerca de sus capacidades y limitaciones para ver que podía ofrecernos y como utilizarlo de la manera adecuada.

El tiempo de aprendizaje autónomo en esta primera etapa fue crucial para plantear todo el proyecto y su planificación, ya que se alargó demasiado y afectó a las etapas de desarrollo y documentación.

5.1.2.- Tablas de asignación temporal de funcionalidades

Para esta primera etapa, pensada para ser la única del proyecto, se consideraron oportunas las siguientes fechas de entrega y de desarrollo para cada una de las partes, considerando que cada semana tiene 5 días de trabajo y que en cada día se dedican aproximadamente 6 horas, aunque en la práctica se dedicaron unas tres horas diarias y es por eso que el proyecto se alargó hasta ocupar dos semestres lectivos completos .

Tabla de planificación teórica (Etapa1):

Tareas	Días aproximados
Análisis de requisitos	5
Esbozo de la solución	5
Aprendizaje autónomo	15
Implementación	30
Test con usuarios	5
Corrección de errores	5
Documentación	15
TOTAL	85

Figura 18

Tabla de planificación práctica (Etapa 1):

Tareas	Días aproximados
Análisis de requisitos	3
Esbozo de la solución	5
Aprendizaje autónomo	20
Implementación	30
Test con usuarios	0
Corrección de errores	15
Documentación	0
TOTAL	73

Figura 19

Como se puede apreciar en la tabla anterior, las horas de aprendizaje autónomo se vieron incrementadas en 5 días, ya que hubo problemas con la familiarización del entorno y la configuración e instalación de diferentes softwares, y para ello se tuvieron que dedicar horas no previstas.

Debido a que el proyecto acogió todo su desarrollo en esta primera etapa, no se realizaron test con usuarios ya que no se consideró necesario, debido a la cantidad de errores que se detectaban de funcionalidad básica.

También se diferencia un gran cambio de horas en la corrección de errores, ya que surgieron más de los previstos, y su correcta corrección llevó mas tiempo del esperado, alterando considerablemente los tiempos previstos para las tareas restantes.

Finalmente vemos que no se dedican horas a la documentación, ya que antes de empezar su etapa de redacción, se decidió posponer la defensa y se emplearon las horas previstas a poder seguir desarrollando código y corrigiendo errores.

5.2.- Segunda etapa

La segunda etapa se comprende entre Enero y Junio, segundo semestre del curso lectivo 2013/2014 y define los nuevos objetivos que toma el proyecto, el nuevo rumbo a seguir y todo el desarrollo de PhoneFlat y su documentación.

5.2.1- Declaración de objetivos

Para esta segunda etapa de desarrollo, se tuvieron que re-definir los objetivos aplicando una planificación muy parecida a la de la primera explicada con anterioridad.

Tabla de planificación teórica (Etapa2):

Tareas	Días aproximados
Definición de nuevos objetivos	10
Esbozo de la solución	5
Aprendizaje autónomo	5
Implementación	30
Test con usuarios	7
Corrección de errores	10
Documentación	15
TOTAL	82

Figura 20

Tabla de planificación práctica (Etapa 2):

Tareas	Días aproximados
Definición de nuevos objetivos	7
Esbozo de la solución	5
Aprendizaje autónomo	7
Implementación	40
Test con usuarios	2
Corrección de errores	15
Documentación	15
TOTAL	91

Figura 21

Cómo se puede observar, esta vez se implementaron más horas de las previstas al contrario de la primera etapa, que se esperaban unas 510 horas (85 días), y se utilizaron aproximadamente unas 348 horas (58 días). Debido a la falta de horas de la primera etapa y a la falta de objetivos claros al empezar la segunda, hizo que el desarrollo de esta segunda etapa fuera más lento que el anterior y se tuvieron que dedicar más horas.

Analizando la segunda tabla de planificación de la segunda etapa, se puede apreciar que los “*timings*” han seguido un desarrollo bastante parecido, que se ve alterado nuevamente por la corrección de errores, ya que en Julio de 2014, PhoneGap/Cordova deja de dar soporte a varias plataformas y eso hizo que se tuvieron muchos más problemas de los esperados en cuanto a realizar el port del servicio a otras plataformas.

5.3.- Diagramas de Gantt

A continuación se muestran los diagramas de Gantt correspondientes a las planificaciones vistas con anterioridad en las tablas, para poder ver como se se planificó y ejecutó el desarrollo de manera más visual, cada una de las dos etapas del proceso. En los diagramas se aprecia el hecho de haber trabajado durante fines de semana.

Los diagramas aparecerán en 4 páginas seguidas respetando el siguiente orden:

- Diagrama teórico (Etapa 1)
- Diagrama práctico (Etapa 1)
- Diagrama teórico (Etapa 2)
- Diagrama práctico (Etapa 2)

Para cada uno de los diagramas se deberá tener en consideración la siguiente leyenda:

-  Muestra la planificación correcta según la previsión.
-  Establece que la ejecución de la tarea excede de los término previstos.

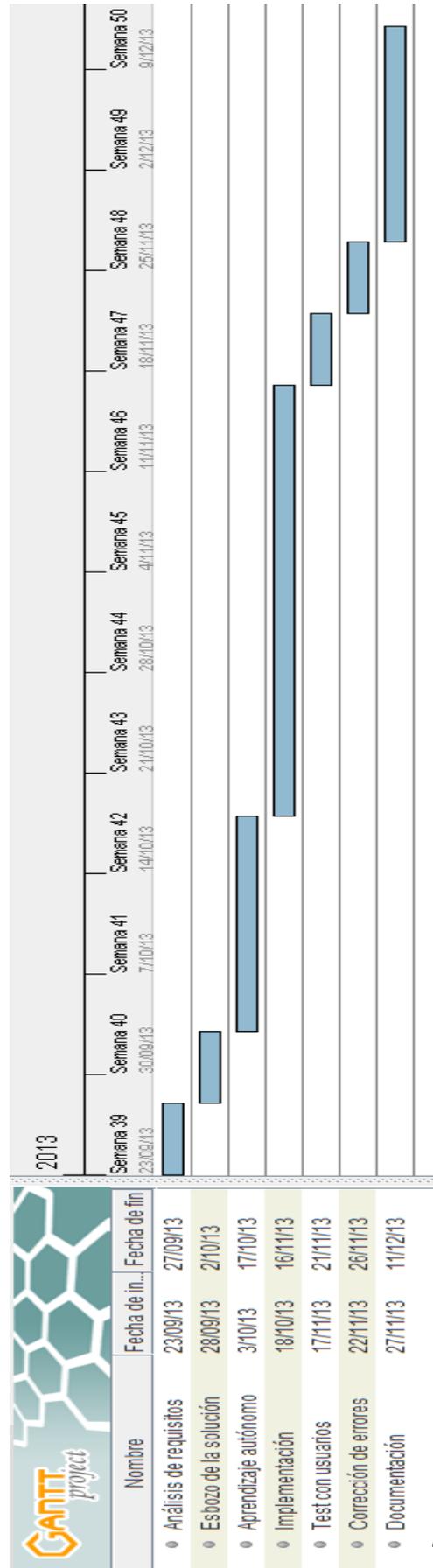


Figura 22

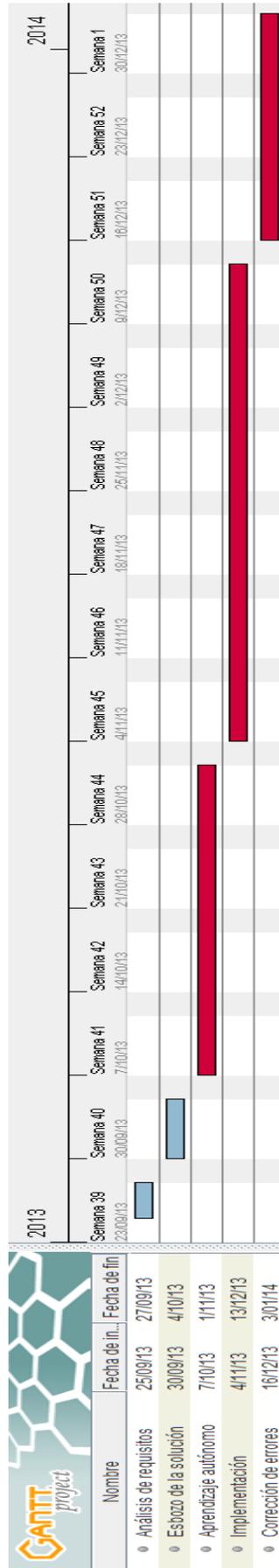


Figura 23

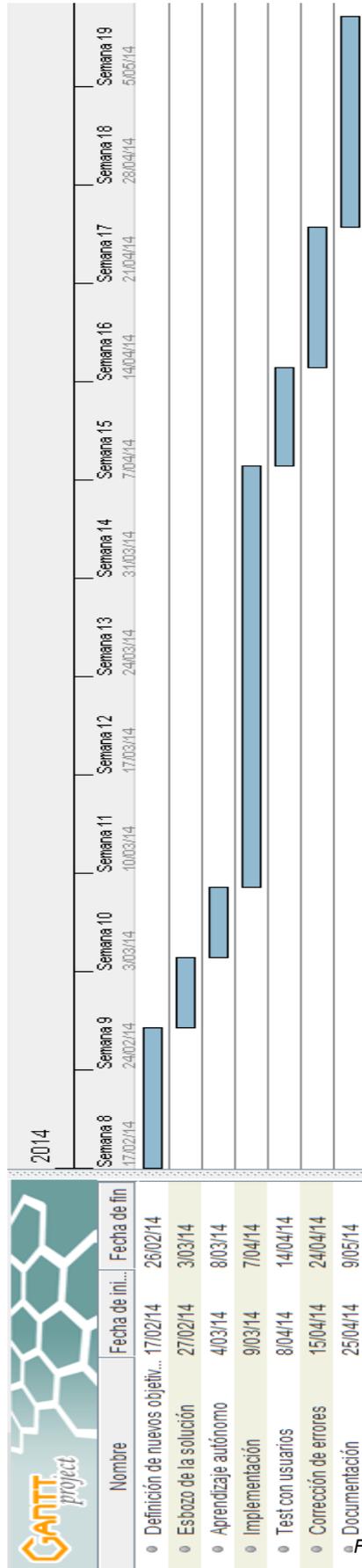


Figura 24

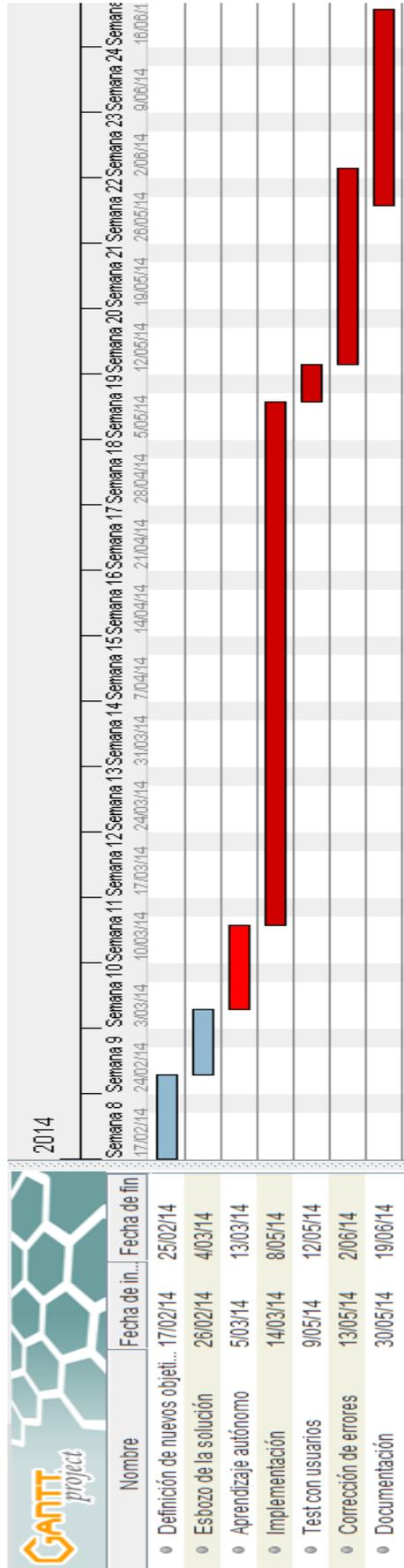


Figura 25

Etapa 1

Tal y como se observa en la primera etapa, a partir del aprendizaje autónomo se empieza a trabajar a contracorriente, ajustando los tiempos de ejecución de cada etapa del proceso, haciendo imposible cuadrar los “timings” previstos, debido a una lenta adaptación. Es por ello que tal y como se ha comentado anteriormente, se decide cambiar de rumbo y aplazar la entrega hasta Junio del 2014.

Etapa 2

De igual manera que en la primera etapa del proyecto, en esta segunda los tiempos de realización vuelven a ser alterados en la fase de aprendizaje autónomo.

En este caso, se debe a que se planificó en un escenario sin exámenes ni ningún tipo de inconveniente que pudiera afectar al desarrollo del proyecto, ya que no estaba previsto alargarlo durante un semestre más, y se encontró el inconveniente de tener otras 4 asignaturas con gran volumen de materia, y que dificultaron e interfirieron en los procesos de desarrollo volviendo a alterar los “timings” de esta segunda etapa.

6.- Desarrollo

En este capítulo se explica como se ha planteado y desarrollado PhoneFlat. Se mostrará como configurar el PC y todos los frameworks y programas necesarios para poder empezar a codificar y la manera en como PhoneFlat lo lleva incorporado.

6.1.- Casos de uso

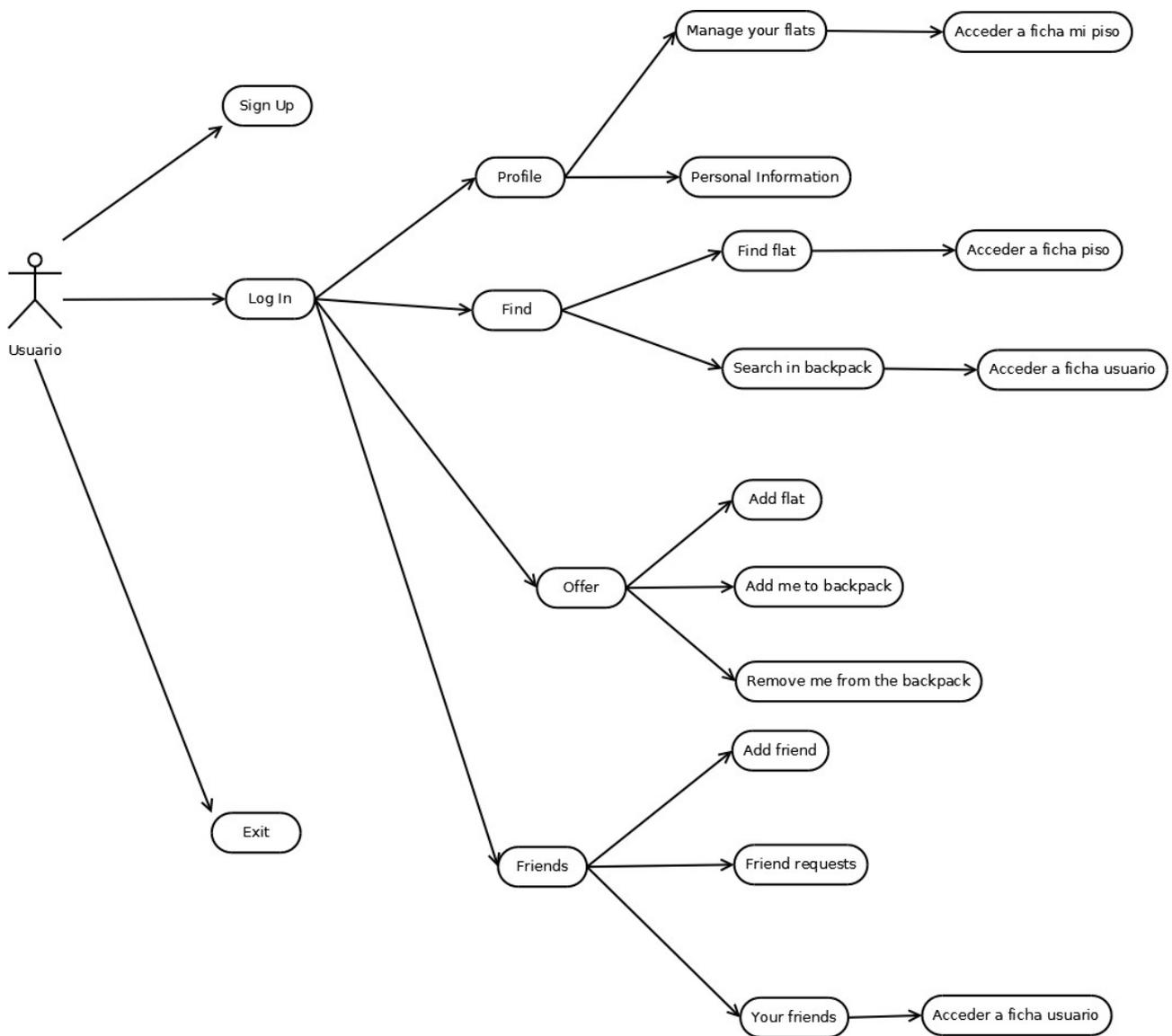


Figura 26

PhoneFlat dispone de un solo tipo de usuario, así que tendrá exactamente los mismos privilegios y posibilidades de acceso en todos los casos.

6.1.1.- Explicación de los casos de uso

6.1.1.1.- Sign Up

Descripción: El usuario tendrá que rellenar tres campos obligatorios y presionar sobre submit para realizar el registro.

Atributos:

- Username (type="text")
- Password (type="password")
- Repeat your password (type="password")

Requisitos: Ninguno.

Estado final:

- Acierto: El usuario estará registrado en la aplicación.
- Error: El usuario vuelve al estado Sign Up sin ninguna modificación en el sistema si:
 - Los atributos *Password* y *Repeat password* no son idénticos.

6.1.1.2.- Log In

Descripción: El usuario tendrá que rellenar dos campos obligatorios y presionar sobre submit para acceder al servicio.

Atributos:

- Username (type="text")
- Password (type="password")

Requisitos: Ninguno.

Estado final:

- Acierto: El usuario accederá al servicio si está registrado.
- Error: El usuario vuelve al estado Log In sin ninguna modificación en el sistema si:
 - No esta registrado
 - Los datos introducidos son erróneos

6.1.1.3.- Profile

6.1.1.3.1.- Personal information

Descripción: El usuario puede cambiar su información personal de su perfil.

Atributos:

- Age (type="select")
- Gender (type="select")
- Smoker (type="select")
- Employed (type="select")
- Description (type="textarea")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.

Estado final:

- Acierto:
 - Ya tenia un perfil creado:
 - El usuario modificará su información personal.
 - No tenia perfil creado:
 - Se creará un perfil con la información introducida.
- Error: El usuario permanece en el mismo estado sin ninguna modificación.

6.1.1.3.2.- Manage your flats

Descripción: El usuario visualizará una lista con sus anuncios de pisos ofrecidos y accederá a cualquiera de ellos mediante el botón “Go In!”.

Atributos: Ninguno

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.

Estado final:

- Acierto: El usuario accederá a uno de sus anuncios y accederá a su ficha.
(Caso: Acceder a ficha mi piso)
- Error: El usuario no posee ningún anuncio y el estado quedará intacto.

6.1.1.4.- Find

6.1.1.4.1.- Find flat

Descripción: El usuario puede buscar los pisos que estén anunciados en el sistema.

Atributos:

- City (type="text")
- Price/Month (type="select")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar logueado en el sistema.

Estado final:

- Acierto: El usuario visualizará una lista con los anuncios acordes a su búsqueda.

- Error: El usuario no encuentra ningún anuncio y permanece en el mismo estado.

6.1.1.4.2.- Search in backpack

Descripción: El usuario puede buscar todas aquellas personas que se adecuen a sus gustos rellenando unos filtros.

Atributos:

- Age (type="select")
- Gender (type="select")
- Smoker (type="select")
- Employed (type="select")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.

Estado final:

- Acierto: El usuario visualizará una lista de usuarios con las características buscadas. (Caso: Acceder a ficha usuario)
- Error: El usuario permanecerá en el mismo estado.

6.1.1.5.- Offer

6.1.1.5.1.- Add flat

Descripción: El usuario puede añadir un anuncio al sistema rellenando un formulario.

Atributos:

- Adress (type="text")
- City (type="text")

- Zipcode (type="text")
- Price/Month (type="select")
- Comments (type="textarea")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.

Estado final:

- El usuario añadirá una oferta al sistema.

6.1.1.5.2.- Add me to backpack

Descripción: El usuario podrá añadirse a la mochila para ser buscado.

Atributos: Ninguno. (Se definen previamente en "*Personal Information*")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.
- Se debe crear un perfil con tu información personal.

Estado final:

- Acierto: El usuario se añadirá a la mochila para ser buscado.
- Error: El usuario no cumple alguno de los requisitos y se queda en el mismo estado.

6.1.1.5.3.- Remove me from the backpack

Descripción: El usuario podrá eliminarse de la mochila y no podrá ser encontrado.

Atributos: Ninguno.

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.
- Se debe crear un perfil con tu información personal.
- Se debe estar añadido en la mochila

Estado final:

- Acierto: El usuario será eliminado de la mochila.
- Error: El usuario no cumple alguno de los requisitos y se queda en el mismo estado.

6.1.1.6.- Friends

6.1.1.6.1.- Add friend

Descripción: El usuario podrá enviar una solicitud de amistad a otro usuario.

Atributos:

- Username (type="text")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.
- Se precisa conocer el nombre del usuario a añadir.

Estado final:

- Acierto: El usuario enviará una petición de amistad y volverá a *Friends*.
- Error: El usuario no cumple alguno de los requisitos y se queda en el mismo estado.

6.1.1.6.2.- Friend requests

Descripción: El usuario visualizará todas las solicitudes de amistad y podrá aceptar o rechazarlas.

Atributos:

- Username (type="text")

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.

Estado final:

- El usuario aceptará o rechazará la solicitud y volverá a "*Friends*".

6.1.1.6.3.- Your friends

Descripción: El usuario podrá acceder a los perfiles de los usuarios que tenga agregados.

Atributos: Ninguno.

Requisitos:

- Se debe estar registrado en el sistema.
- Se debe estar identificado en el sistema.

Estado final:

- El entrará al perfil de uno de los usuarios. (Caso: Acceder a ficha usuario)

6.2.- Diseño gráfico de la aplicación

PhoneFlat hace uso de las hojas de estilo de jQuery para dar un aspecto juvenil e uniforme en toda su navegación.

PhoneFlat conserva el mismo estilo en cada estado o página en la que se encuentra el usuario (excepto en el menú de registro y acceso).

Las hojas de estilo estarán alojadas dentro del subdirectorio css tal y como se ve en la figura de la derecha.

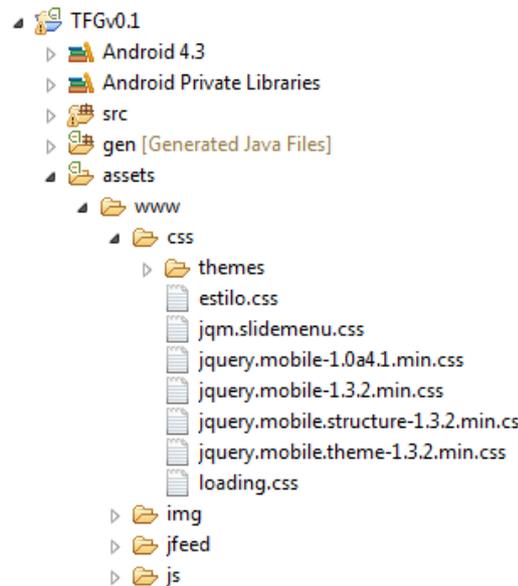


Figura 27

1.- Cabecera

Esta cabecera permanecerá intacta en todas las layouts de la aplicación, y mostrará en el punto exacto en el que se encuentra el usuario de la aplicación.

```
<header data-role="header">
  <h1>Personal information</h1>
</header>
```

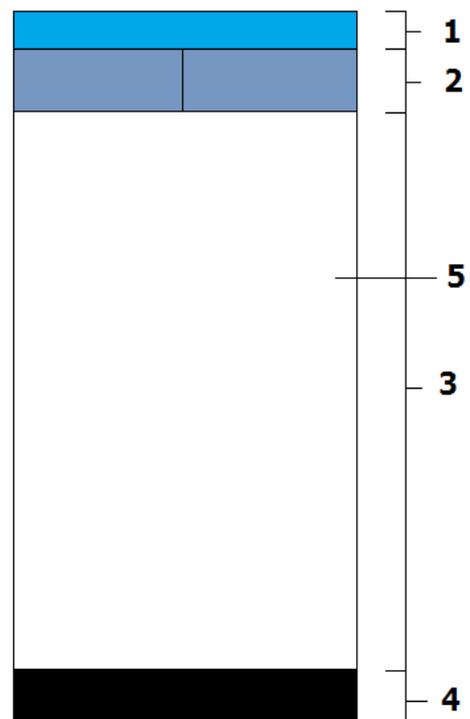


Figura 28

2.- Navbar

Ocupará el 100% del ancho de la pantalla ,y albergará las opciones en todos los casos de volver hacia atrás, o redirigir hasta el Home. Se ha incorporado un botón que permita ir hacia atrás, ya que el botón de back que llevan incorporados los móviles, han sido deshabilitas, con tal de evitar posibles errores en el sistema al acceder a páginas HTML sin parámetros en GET o POST, y sufrir un comportamiento no deseado.

Tal y como se ve en la figura, irá siempre de manera inmediata al <header> visto con anterioridad en el punto 1.

```
<div data-role="navbar" style="width:100%;">
  <ul>
    <a style="margin:0px;" rel="external" data-icon="arrow-l" href="profile.html">Back</a>
    <a style="margin:0px;" rel="external" data-icon="home" href="index.html#home">Home</a>
  </ul>
</div>
```

Figura 29

3.- Contenido

En la parte 3 de la figura, es donde PhoneFlat incorporará toda la información visual que apreciarán los usuarios.

4.- Botones del smartphone

La parte 3 de la figura, simplemente alberga los botones físicos de los que dispone cada smartphone.

6.3.- Multiplataforma

PhoneGap destaca por hacer que un mismo código HTML pueda ejecutarse en diferentes Sistemas Operativos disponibles para smartphones, y PhoneFlat se aprovecha de ello llevando su servicio más allá de Android.

Para ello se hace uso de PhoneGap Build, pero se deben de descargar e instalar varios ficheros para cada caso para que Build compile el proyecto.

Plataformas soportadas hasta PhoneGap 2.9



Figura 30

Plataformas soportadas con PhoneGap 3.0+



Figura 31

6.3.1.- Config.xml

Las aplicaciones que son compiladas mediante PhoneGap Build, pueden ser configuradas mediante la web que ofrece Adobe, o bien configurando el archivo **config.xml**.

El archivo **config.xml** permite a los desarrolladores especificar la metadata sobre sus aplicaciones de manera sencilla y debe encontrarse al mismo nivel que el fichero index.html, de lo contrario no se cargará de manera correcta.

6.3.1.1.- Propiedades esenciales

Widget

El elemento widget debe ser la raíz del documento XML. Si se desea hacer uso de PhoneGap Build para la compilación de los proyectos, el fichero debe contar con los siguientes atributos dentro de `<widget>`:

- id: Identificador único de la aplicación. Para dar soporte a todas las plataformas soportadas, se debe escoger un “reverse-domain style” (por ejemplo: 'com.tucompañia.tuapp')
- version: Declarar la versión que se está utilizando, preferentemente la última disponible, con tres números como por ejemplo “1.0.0”.

Name

Nombre de la aplicación `<name>`.

Description

Una descripción de tu aplicación `<description>`.

Ejemplo de Config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
  <widget xmlns = "http://www.w3.org/ns/widgets"
    xmlns:gap = "http://phonegap.com/ns/1.0"
    id = "com.phonegap.example"
    versionCode = "10"
    version = "1.0.0" >

    <!-- versionCode is optional and Android only -->

    <name>PhoneGap Example</name>

    <description>
      An example for phonegap build docs.
    </description>

    <author href="https://build.phonegap.com" email="support@phonegap.com">
      Hardeep Shoker
    </author>

  </widget>
```

Figura 32

6.3.1.2.- Selección de la plataforma de compilación

Por defecto PhoneGap Build compila las aplicaciones para cada plataforma. Si solo se desea compilar para una plataforma concreta, se puede especificar esa plataforma mediante el tag: *gap:platform*

Ejemplo:

```
<gap:platform name="ios" />
<gap:platform name="android" />
<gap:platform name="winphone" />
```

Figura 33

6.3.2.- Android

Es la plataforma base y la más sencilla de obtener, ya que en la mayoría de casos se va a desarrollar la aplicación en Eclipse IDE, y este genera el .apk de manera automática, al igual que con cualquier aplicación Android.

En el anexo se explica con detalle como configurar XAMPP y Eclipse para poder trabajar con entorno Android.

6.3.3.- iOS

Debido a problemas de licencias, PhoneFlat no se encuentra disponible para dispositivos iOS, pero a continuación se explica como configurar tu proyecto para obtener tu aplicación lista para ser publicada para iOS, en el caso de tener cuenta de desarrollador Apple.

6.3.3.1.- Requerimientos

Las herramientas de Apple necesarias para crear aplicaciones de iOS, sólo pueden ser ejecutadas en en sistema operativo OS X en ordenadores Mac basados en Intel. Xcode 4.5 (la versión mínima requerida), solo funciona a partir de la versión 10.7(Lion) en OS X, e incluye el SDK de iOS 6 (Software Development Kit).

Se pueden probar algunas de las funcionalidades de Cordova usando el emulador de iOS instalado en el SDK de iOS y Xcode, pero se necesita un dispositivo actual para probar completamente todas las características de la aplicación antes de subirla al App Store.

El dispositivo deberá tener como mínimo iOS 5.x instalado, la mínima versión soportada por Cordova 2.3.

Para poder instalar aplicaciones en un dispositivo, se debe ser miembro de grupo Apple

de Desarrolladores, que cuesta 99\$ al año.

Así pues, un pequeño esquema de requerimientos será el siguiente:

- Xcode 4.3+
- Xcode Command Line tools
- Ordenadores basados en Intel con Mac OS X Lion o mayor (10.7+)
- Necesario para instalar en el dispositivo:
 - Dispositivo Apple iOS (iPhone, iPad, iPod Touch)
 - Certificado de desarrollador iOS

6.3.4.- Windows Phone 8

No será necesario ningún tipo de configuración previa para poder compilar el archivo “.xap” ejecutable dentro de un smartphone con Windows Phone 8 incorporado. PhoneGap Build proporciona soporte completo de compilación, independientemente de la plataforma en la que la app haya sido desarrollada.

6.3.5.- Symbian y WebOS

Para poder realizar el port a Symbian y WebOS, se tubo que modificar la versión de PhoneGap/Apache Cordova que se estaba utilizando hasta la fecha, debido a la falta de soporte que ofrece PhoneGap Build a partir de la versión 3.0.

6.3.6.- BlackBerry10

No se ha podido realizar el port debidamente a este sistema operativo, pese a obtener las claves de desarrollador oficial necesarias para realizar la firma digital de autenticación de cada aplicación BlackBerry.

El motivo de la ausencia de esta plataforma, se debe a que la herramienta RIM de BlackBerry que empaqueta la carpeta assets de la aplicación, tiene unas reglas muy

exigentes en cuanto a los caracteres permitidos en la composición de los nombres de cada subdirector, y se hace muy difícil poder incorporar algunos archivos, aunque Adobe ya ha insistido en que están trabajando para solucionar este problema.

7.- Conclusiones y trabajo futuro

Al iniciar este proyecto se pretendía mostrar como PhoneGap/Apache Cordova, era capaz de dar herramientas necesarias para poder llevar al programación web a dispositivos smartphones, haciendo uso de varios recursos de los dispositivos y enriqueciendo el catálogo de funcionalidades disponibles para su uso.

El mayor problema que se ha tenido, ha sido poder familiarizarse con el framework, ya que no ha sido tarea fácil entender como funciona todo, los requisitos necesarios para la creación de un proyecto y sobretodo la portabilidad a otros sistemas operativos, ya que en Julio de 2013, PhoneGap dejaba de dar soporte a varios sistemas operativos tal y como se ha comentado en otros capítulos, y eso dificultó enormemente la tarea de la compilación en la nube.

PhoneGap/Apache Cordova, también demuestra que todos aquellos programadores Web que hasta ahora solo se podían dirigir a un público minoritario (todas aquellas personas que navegan delante de un PC), ahora también pueden desarrollar apps para plataforma móvil, aumentando su rango de actuación enormemente, ya que actualmente la mayor parte del mercado se mueve a través de los smartphones y sus aplicaciones.

Así pues, aprovechando la realización de este proyecto, deseo continuar con el desarrollo del servicio para un futuro gracias a toda la experiencia obtenida en todos los meses de familiarización y desarrollo, ya que encuentro que bien desarrollado puede llegar a tener potencial en el mercado, siendo posible extenderlo a más sectores a parte de la compartimiento de pisos.

7.1.- Errores y posibles mejoras

Pese a que PhoneFlat ofrece al usuario una experiencia “casi” completa respecto a la competencia en el sector, a continuación se mostrarán las carencias que hacen que PhoneFlat no sea aún un servicio totalmente completo respecto a sus funcionalidades, ya

que hay funcionalidades que no se han podido implementar y que la competencia si tiene.

7.1.1- Errores a mejorar

Márgenes en la descripción de perfil y descripción del piso

Cuando el usuario introduce comentarios o descripciones en su perfil o en la descripción de un piso, si contiene frases excesivamente largas, se desplegará un scroll horizontal que obliga al usuario a desplazarse y ocultar la interfaz principal, haciendo más incómoda la experiencia y la navegabilidad.

Visualización en tablets

Pese a que PhoneFlat se puede ejecutar sin problemas en dispositivos tablet, las interfaces que incorpora no se adaptan de manera correcta a medidas grandes, pese a que incorpora todos sus elementos con tamaños porcentuales. Es un aspecto a mejorar ya que se desea que la experiencia en tablets sea exactamente la misma que al ejecutarse en un smarthphone.

Eliminar un amigo inexistente

Pese a que no afecta en nada en el funcionamiento normal del servicio, PhoneFlat permite poder borrar a un usuario desde su ficha aunque no se tenga agregado en la lista de amigos.

Peticiones dobles

El servicio te permite poder enviar solicitudes de manera indiscriminada a cualquier usuario, pudiendo recibir más de una del mismo emisor. No obstante, una vez se acepta una de ellas, desaparecen todas evitando tener que ir aceptándolas o eliminándolas una por una. Es otro error sin ningún tipo de relevancia funcional pero a arreglar para el futuro.

Controlar nombres de usuario

Se ha notado que pueden haber problemas si el usuario introduce un número como nombre de usuario, ya que se creará una carpeta con su perfil con el nombre "1", por ejemplo, y los pisos se guardan con el nombre de su id e intentarían crear sus carpetas con el mismo nombre. Es algo necesario de controlar.

7.1.2.- Posibles mejoras

Sistema de geolocalización

En un inicio, PhoneFlat incorporaba un sistema de geolocalización que facilitaba la búsqueda de viviendas mediante un script de google-maps, pero finalmente se decidió prescindir de esta funcionalidad, debido a errores en la precisión que podrían confundir al usuario y que no se pudo corregir con exactitud por problemas de horario y la necesidad de desarrollar otros servicios más necesarios, pero que sin duda es una funcionalidad totalmente necesaria para el futuro, y ya está muy avanzada en su desarrollo.

Sonidos y efectos visuales

Aunque no es una mejora realmente necesaria, sería de gran utilidad aplicar algunos efectos visuales o sonidos cada vez que se interaccione con el entorno. Eso podría ofrecer al usuario una mejor experiencia de navegación y eso es algo importante en un servicio, ya que si está cómodo pasará más tiempo haciendo uso de ella.

Borrar anuncio de piso

Pese a que los usuarios pueden añadir sin problema tantos anuncios como se desee, PhoneFlat no incorpora actualmente una opción para eliminar un anuncio debido a que

nunca se llegó a contemplar cuando se definieron las tareas y objetivos del servicio, y ha sido en la realización de este documento cuando se ha podido observar esta ausencia de funcionalidad.

Ciudades

Sería de gran utilidad que el usuario pudiera introducir el nombre de su ciudad mediante un menú desplegable, para facilitar la ausencia de errores al introducir los datos, aunque PhoneFlat evita algunos errores de coincidencia mediante la función strtolower() en sus ficheros PHP, que convierte toda la cadena de entrada en letras minúsculas para que no hay ninguna confusión.

Persistencia de sesión

Algo en lo que se debe trabajar también es en la persistencia de datos para mantener las cuentas abiertas. Es muy molesto tener que poner cada vez tus datos para ser identificado.

Correo electrónico

Es de crucial importancia, poder sincronizar el servicio con los sistemas de correo electrónico para poder tener un método de recuperación o restablecimiento de contraseñas en caso de olvidarlas o hayan sido robadas.

7.1.3.- Trabajo futuro

Este proyecto presenta una primera versión del código que no está lista para ser distribuida, pero viendo el potencial del servicio y las posibilidades que puede abarcar en otros sectores, es mi intención el seguir desarrollándolo para poder sacarlo en un futuro al mercado cuando se hayan arreglado los fallos que presenta (comentados anteriormente), y se hayan implementado las funcionalidades restantes necesarias para competir con los

servicios parecidos actuales del mercado, y debido a que toda esta plataforma puede trasladarse a otro tipo de necesidades, es mi intención también poder seguir aprovechando todas las facilidades de desarrollo que ofrece PhoneGap/Apache Cordova para desarrollar otros servicios similares.

8.- Bibliografía

Aquí el libro

<http://www.adictosaltrabajo.com/detalle-noticia.php?noticia=422>

<http://www.rmunoiz.net/introduccion-a-node-js.html>

<http://pineappslab.wordpress.com/2013/08/06/como-configurar-phonegap-apache-cordova-y-no-morir-en-el-intento/>

<http://ajbalmon.wordpress.com/2008/07/01/configurando-xampp/>

<http://developer.android.com/sdk/installing/installing-adt.html>

<http://necsia.es/frameworks-multiplataforma-para-el-desarrollo-movil/>

<http://es.wikipedia.org/wiki/PhoneGap>

<http://es.wikipedia.org/wiki/Appcelerator>

<http://developer.android.com/tools/sdk/eclipse-adt.html>

http://docs.phonegap.com/en/3.5.0/guide_platforms_android_config.md.html#Android%20Configuration

http://docs.phonegap.com/en/3.5.0/guide_platforms_ios_index.md.html#iOS%20Platform%20Guide

http://docs.phonegap.com/en/2.0.0/guide_getting-started_ios_index.md.html

http://es.wikipedia.org/wiki/Inyecci%C3%B3n_SQL

<http://www.md5online.es/>

<http://rafaelurbanoblog.blogspot.com.es/2014/04/phonegap-introduccion.html?m=1>

9.- Anexo

9.1.- Preparar el entorno

A continuación se explica como configurar el entorno para poder desarrollar con PhoneGap/Apache Cordova para Android, mediante el uso de Eclipse IDE.

9.1.1.- Configurar XAMPP

Primero de todo, deberemos descargar XAMPP (en nuestro caso con el Sistema Operativo Windows) desde el siguiente enlace:

XAMPP: <https://www.apachefriends.org/es/download.html>

Una vez descargado, lo ejecutamos para instalarlo. En nuestro caso, tan solo usaremos Apache y MySQL, así que podemos descartar FileZilla, Mercury y Tomcat de la instalación. Se recomienda instalarlo en el directorio "[C:\xampp](#)".

Cuando lo instalemos, nos aparecerá un panel de control como este:

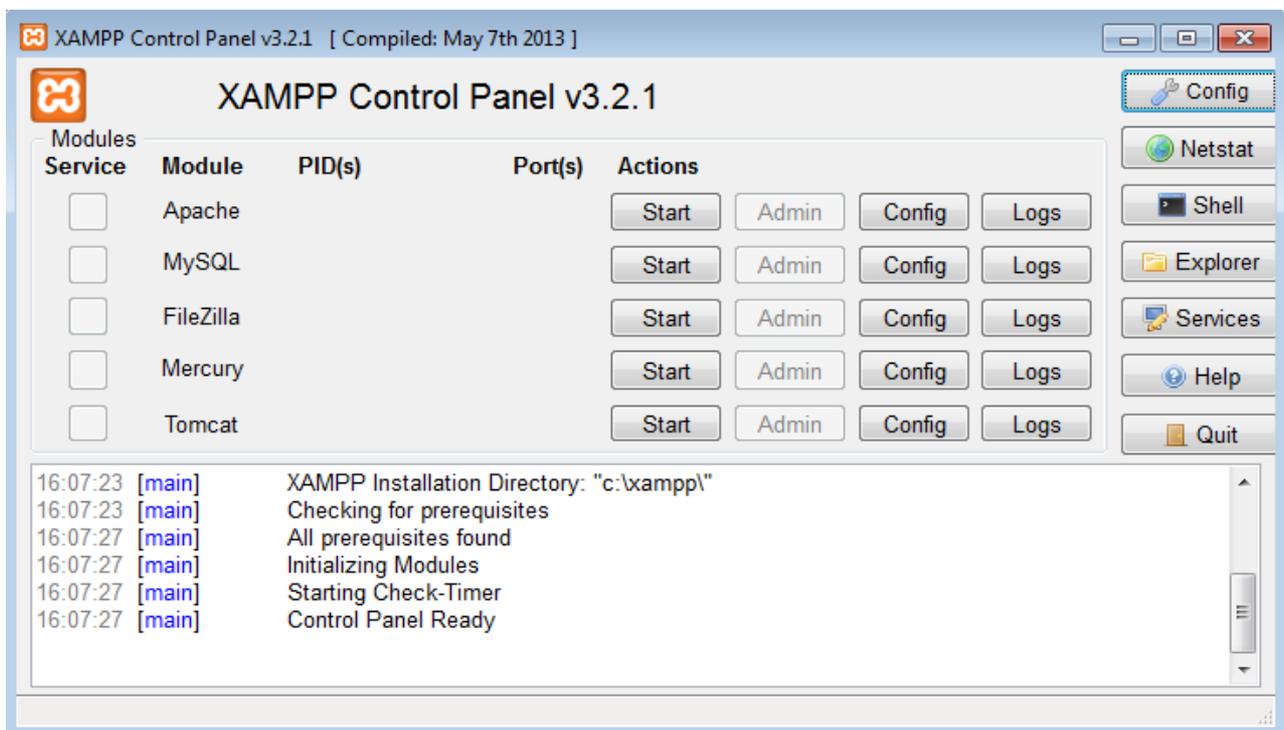


Figura 34

Una vez nos encontremos en este panel, tendremos que presionar sobre el botón Start tanto de Apache como MySQL, y ya podremos acceder a nuestro PHPMyAdmin desde la url: <http://localhost/xampp/>

Ahora ya podremos acceder a nuestro panel phpMyAdmin, que utilizaremos para poder crear nuestra base de datos, a la que accederemos desde la aplicación posteriormente. Sin embargo, aún debemos configurar los puertos y los permisos para que otros usuarios puedan acceder desde una red externa. Para ello, primero accederemos al fichero httpd-xampp.conf, ubicado en "C:\xampp\apache\conf\extra", en el caso de haber instalado xampp en el directorio comentado con anterioridad. Una vez abramos este archivo, debemos ir a la parte inferior, y eliminar el tag <LocationMatch, para sustituirlo por el siguiente:

```
<LocationMatch    "^(?i(?:xampp|licenses|phpmyadmin|webalizer|server-status|server-  
info))">  
    Order deny,allow  
    Allow from all  
    ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var  
</LocationMatch>
```

Una vez hecho esto, xampp permitirá a todas las ip el acceso a nuestras bases de datos mediante un usuario y una contraseña, para evitar que intrusos accedan sin permiso. Esto es necesario ya que nuestra aplicación tendrá que acceder continuamente a la información de manera remota mediante consultas PHP.

Finalmente, debemos mirar en que puerto esta siendo ejecutado xampp, y tendremos que acceder a nuestro router para abrirlo.

Finalmente, una vez hayamos hecho todo, presionaremos sobre el botón "Start", que hay junto a Apache y MySQL, y nuestro servicio estará listo para ser utilizado.

9.1.2.- Crear un proyecto PhoneGap con Eclipse

Una vez instalado Eclipse, ya podemos empezar a crear un proyecto con PhoneGap.

Nota: Si no tenemos Java en nuestro ordenador, tendremos que instalar el JDK y JRE desde <http://docs.oracle.com/javase/7/docs/webnotes/install/> o <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Una vez instalados el JDK y JRE, tendremos que descargar el Android SDK e instalarlo.

Una vez dentro de Eclipse, instalaremos el plug-In ADT. Iremos a Help > Install New Software. Una vez allí, presionaremos sobre el botón Add, e introduciremos el siguiente directorio, asignándolo a cualquier nombre que se quiera poner e instalaremos todos los componentes que queramos utilizar:

<https://dl-ssl.google.com/android/eclipse/>

Una vez tenemos todo instalado, reiniciaremos Eclipse, y ya podremos crear un nuevo proyecto Android.

Ahora pasemos a la configuración de PhoneGap.

Primero de todo, añadiremos una nueva carpeta llamada *www* dentro de la carpeta *assets* (*assets/www*), y dentro de la carpeta *www* crearemos un archivo llamado *index.html*.

Debería quedar así: *NombreAplicacion > assets > www > index.html*

Ahora extrae PhoneGap-X.X.X.zip i ves hasta /PhoneGap-X.X.X/lib/android/ y

- 1.- Copia *cordova-X.X.X.jar* y pegalo en la carpeta *libs* de tu proyecto Android.
- 2.- Copia *cordova-X.X.X.js* en la carpeta *www* de tu proyecto Android.
- 3.- Copia la carpeta *xml* en la carpeta *res* de tu proyecto Android.

Una vez hecho esto, deberemos ir a nuestro archivo *MainActivity.java*, y añadir el

siguiente código:

```

import org.apache.cordova.DroidGap;
import android.os.Bundle;

public class MainActivity extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}

```

Finalmente y para acabar, abriremos AndroidManifest.xml y añadiremos el siguiente código:

```

<supports-screens
    android:anyDensity="true"
    android:largeScreens="true"
    android:normalScreens="true"
    android:resizeable="true"
    android:smallScreens="true" />

<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
    android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.RECORD_VIDEO"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />

```

```
<activity
  android:name="org.apache.cordova.DroidGap"
  android:label="@string/app_name"
  android:configChanges="orientation|keyboardHidden">
  <intent-filter></intent-filter>
</activity>
```

Ya tenemos configurada nuestra aplicación PhoneGap. Para poder probarla en el emulador, debemos configurar el simulador de android que nos ofrece el ADT Plug-In.

9.2.- DynDNS

Para que PhoneFlat funcione, deberá realizar peticiones desde ficheros alojados en la parte del servidor localizables por una IP pública. El problema es que se dispone de una IP dinámica, cada vez que el router se reinicie o se apague se deberán cambiar las peticiones de llamada a la nueva IP asignada.

Para solucionar esto, se ha optado por crear un DynDNS con la dirección “*kl0d.no-ip.biz*”, que buscará cual es la IP pública actual y hará una re-dirección a dicha IP. De esta manera evitamos posibles problemas de conexión hacia la base de datos.

9.3.- Manual de Usuario

Al abrir PhoneFlat, el usuario encontrará una interfaz sencilla que ofrece dos opciones:

- Entrar a la aplicación (Log In)
- Registrarse al servicio (Sign Up)

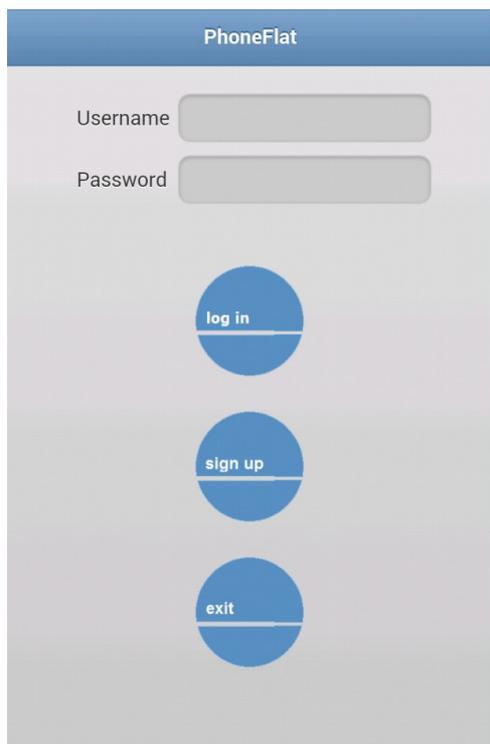


Figura 35

Sign Up

Aparecerá un formulario a rellenar con tu nombre de usuario y la contraseña, que deberás repetir para comprobar que no te has equivocado al escribirla.

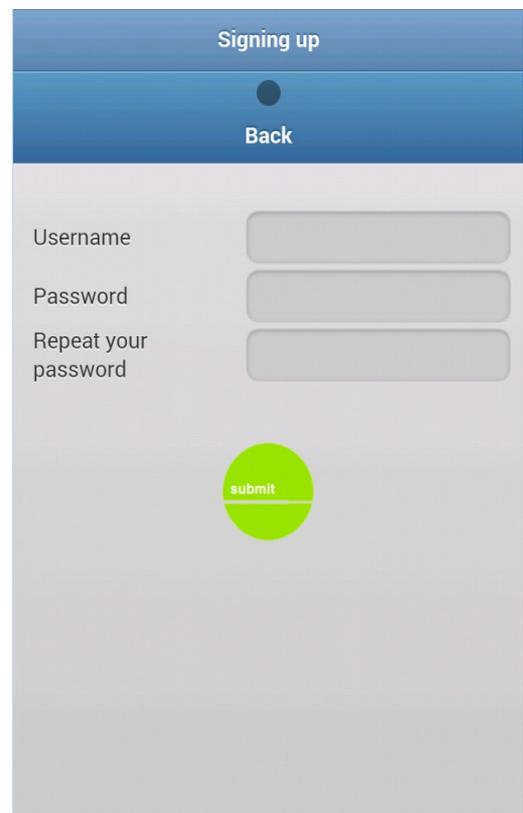


Figura 36

Home

Una vez dentro del servicio, veremos 4 opciones a realizar, a excepción del navbar, que en este caso nos permitirá desconectarnos.

Estas 4 acciones serán:

- Offer
- Find
- Profile
- Friends

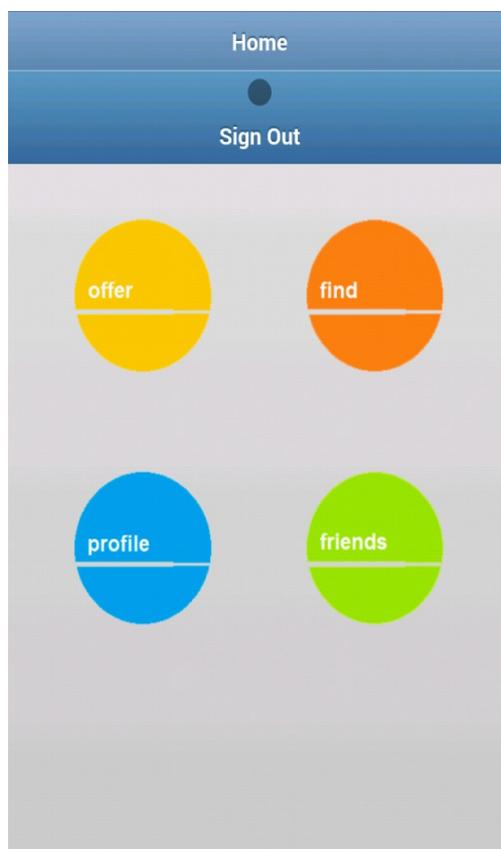


Figura 37

Profile

En este apartado podremos cambiar nuestros datos en el perfil personal o gestionar todo lo relacionado con las ofertas de piso que hayas añadido.

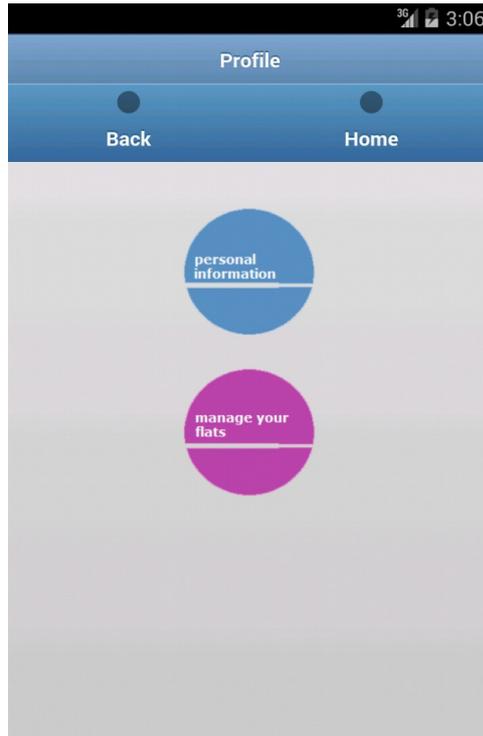


Figura 38

Personal information



Figura 38

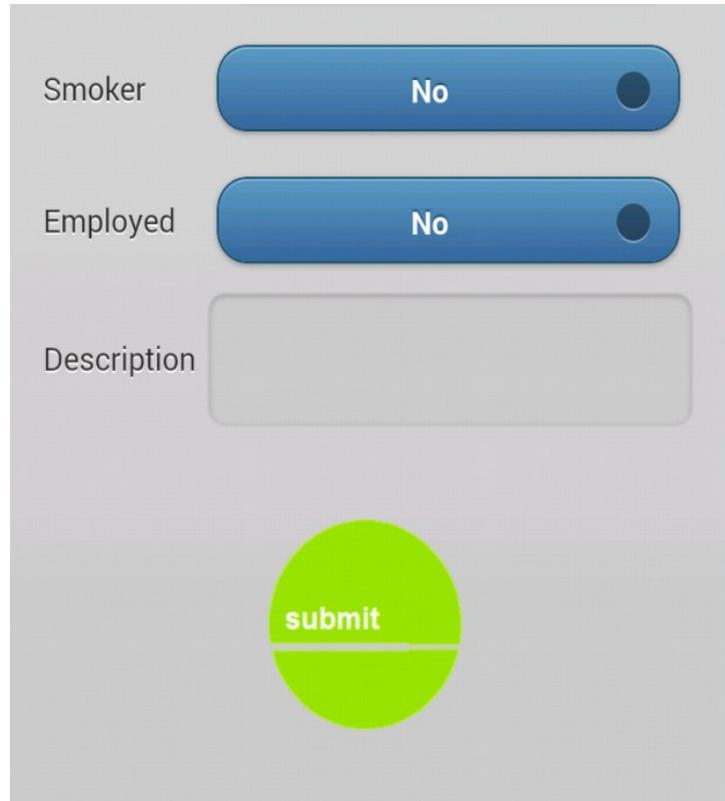


Figura 39

Manage your flats

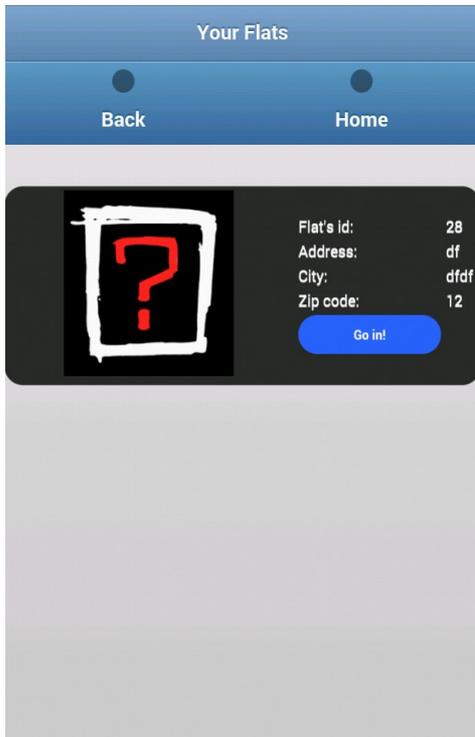


Figura 40

Offer

Podremos añadir un nuevo anuncio para buscar gente para tu piso, además de la posibilidad de ofrecerse en la mochila como posible inquilino, o de eliminarse de la mochila si ya no se desea encontrar ninguna residencia.

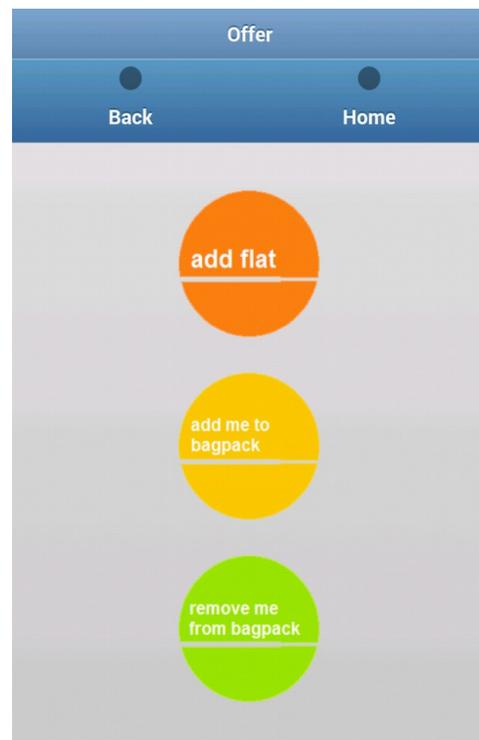
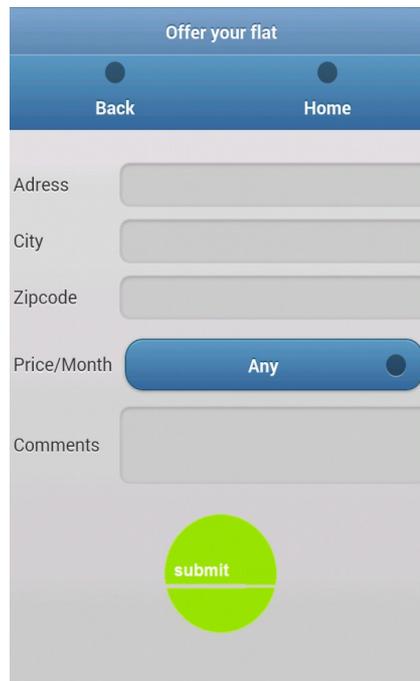


Figura 41

Add flat



The screenshot shows a mobile application interface for adding a flat. At the top, there is a blue header with the text "Offer your flat". Below the header is a navigation bar with two buttons: "Back" and "Home". The main content area contains several input fields: "Address", "City", and "Zipcode", each with a corresponding text input box. Below these is a "Price/Month" field with a dropdown menu currently set to "Any". At the bottom of the form is a "Comments" text area. A large green circular button with the text "submit" is positioned at the bottom center of the form.

Figura 42

Find

Este apartado esta dedicado a la búsqueda de personas o pisos.

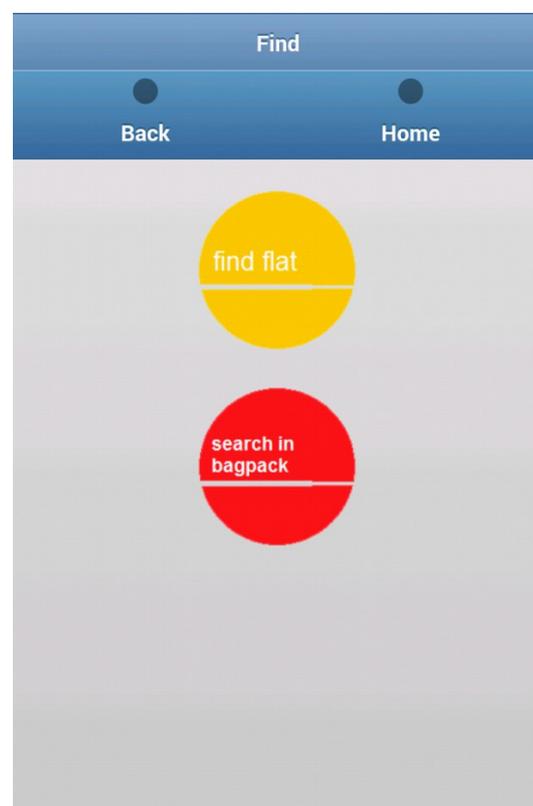


Figura 43

Find flat

En este menú podremos encontrar anuncios de pisos que nos interesen, según la ciudad que deseemos o su precio de alquiler por mes.

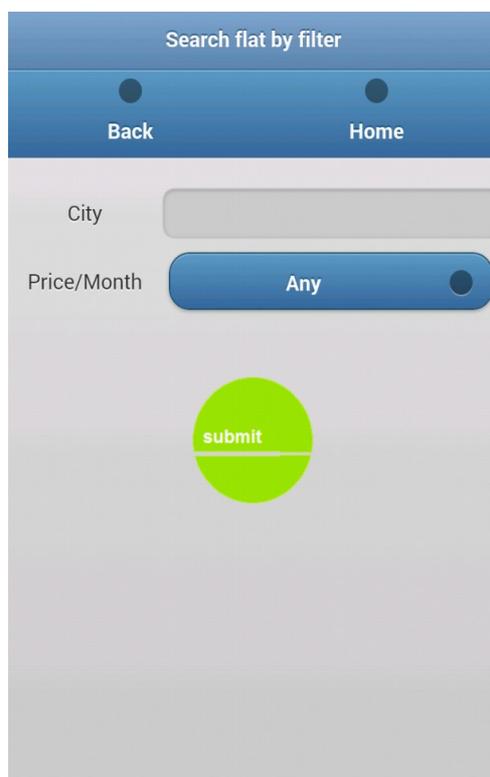


Figura 44

Search inside the backpack

Podremos buscar compañeros para compartir en tu piso o residencia, según las siguientes características:

- Edad
- Sexo
- Fumador
- Con trabajo

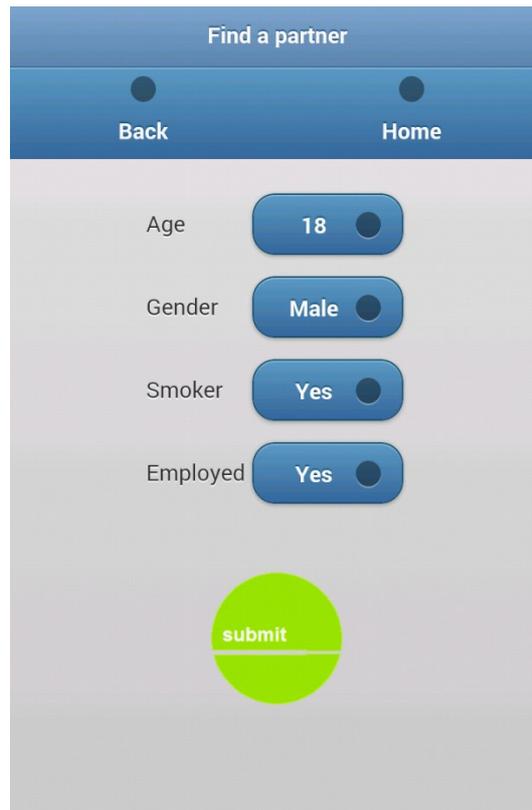


Figura 45

Friends

Este menú almacenará todo lo relacionado con tu lista de contactos de la aplicación

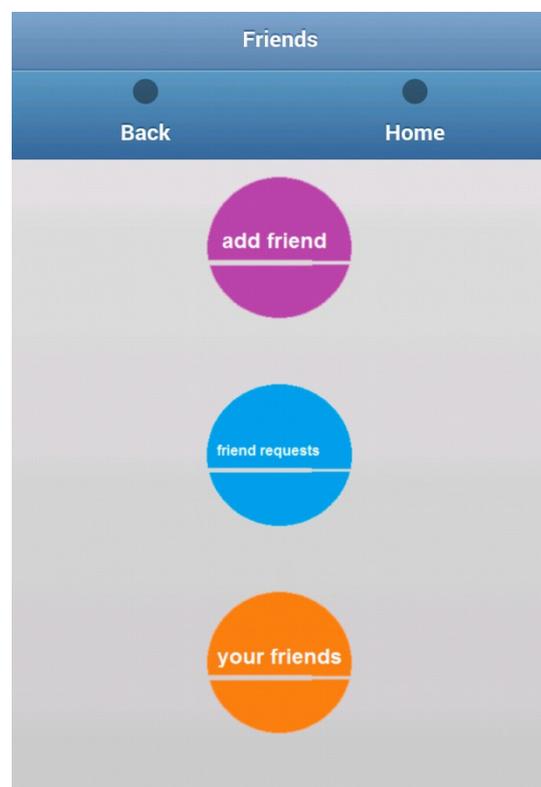


Figura 46

Add friend

Te permitirá enviar peticiones de amistad a otros usuarios de la aplicación.

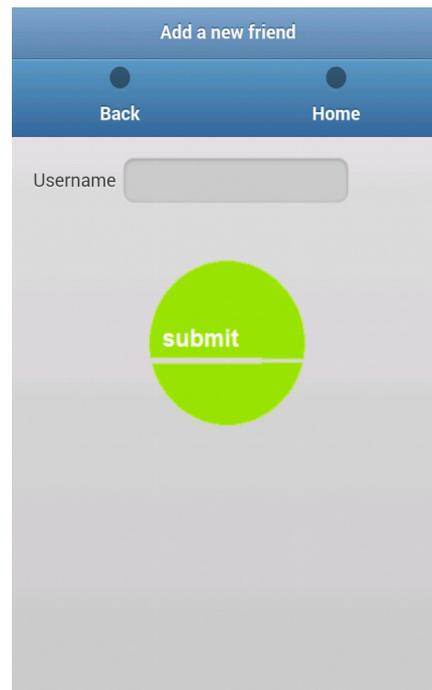


Figura 47

Friend requests

Aparecerán todas las solicitudes entrantes de otros usuarios, y podrás aceptarlas o denegarlas.

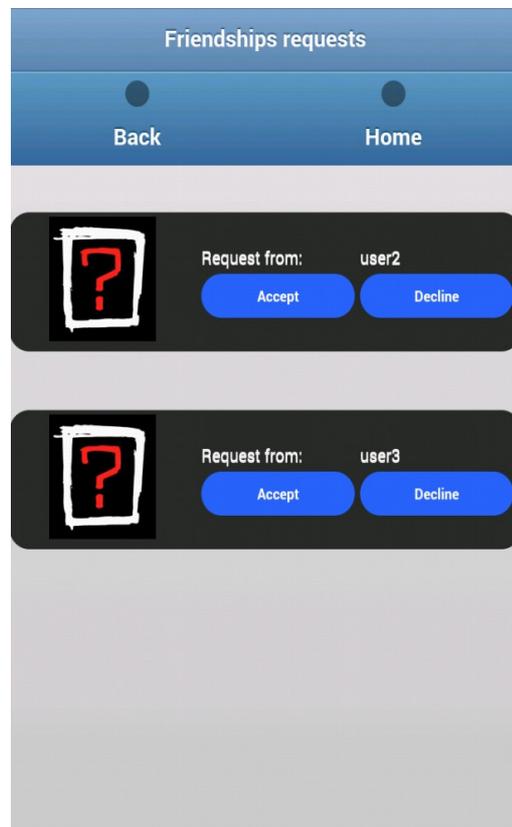


Figura 48

Your friends

Lista con todos tus contactos de la aplicación, que te permitirá acceder a la ficha de cada uno de ellos.

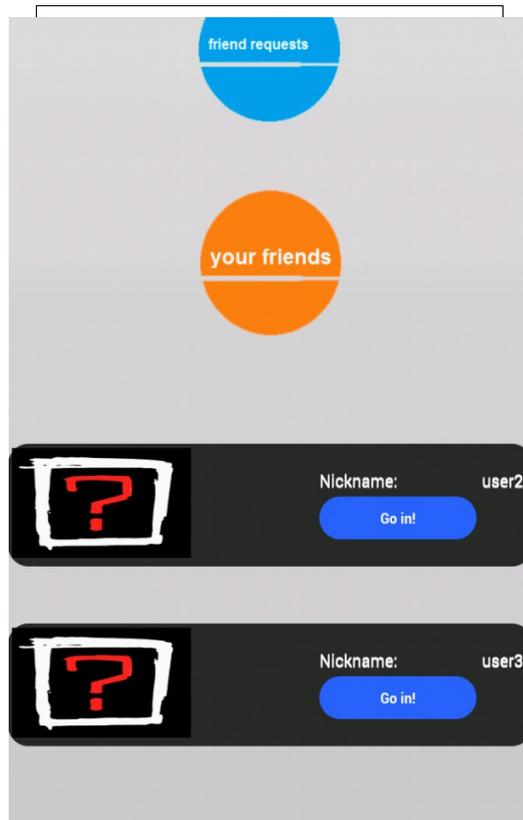


Figura 49

Ficha usuario

Cuando accedamos a la ficha de uno de nuestros “amigos”, esto será lo que veremos:

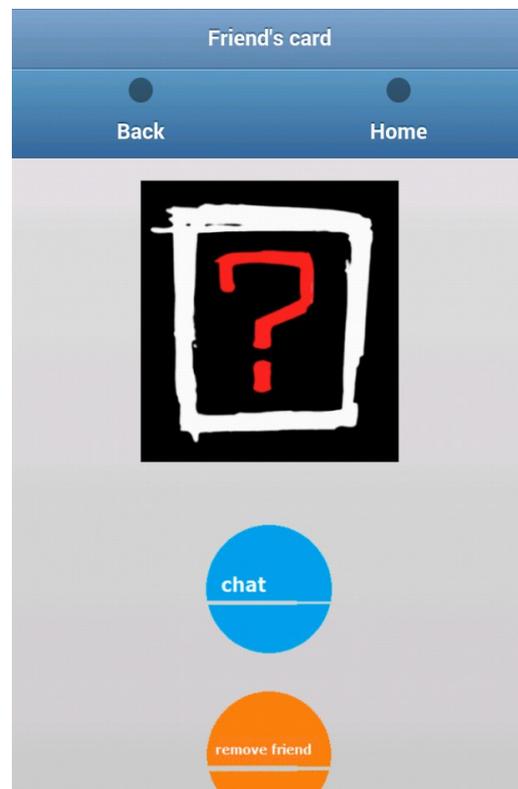


Figura 50

Remove Friend

Botón que permite al usuario borrar a su contacto.

Chat

Este chat nos permitirá hablar en tiempo real con los usuarios agregados, aunque para refrescar los mensajes, se debe salir de la ventana de chat y volver a entrar.

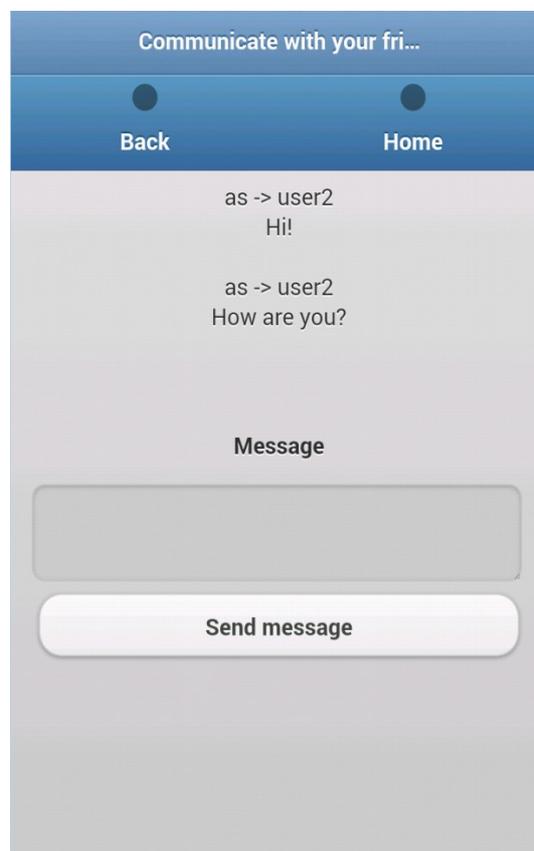


Figura 51

Ficha anuncio piso (Tu anuncio)

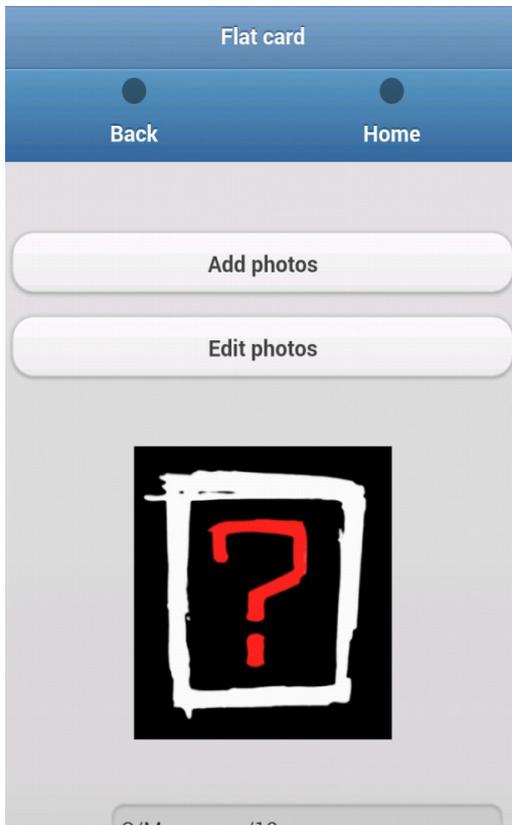


Figura 52

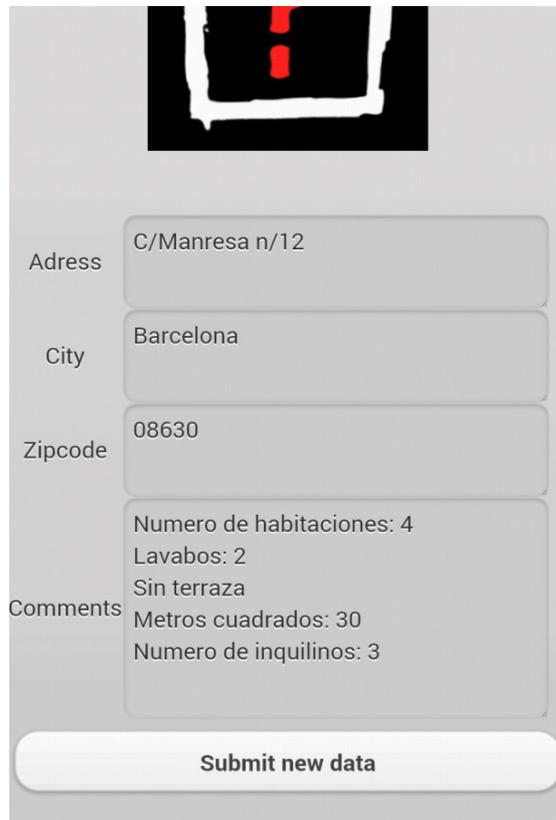


Figura 53

Ficha anuncio piso (Anuncio ajeno)



Figura 54

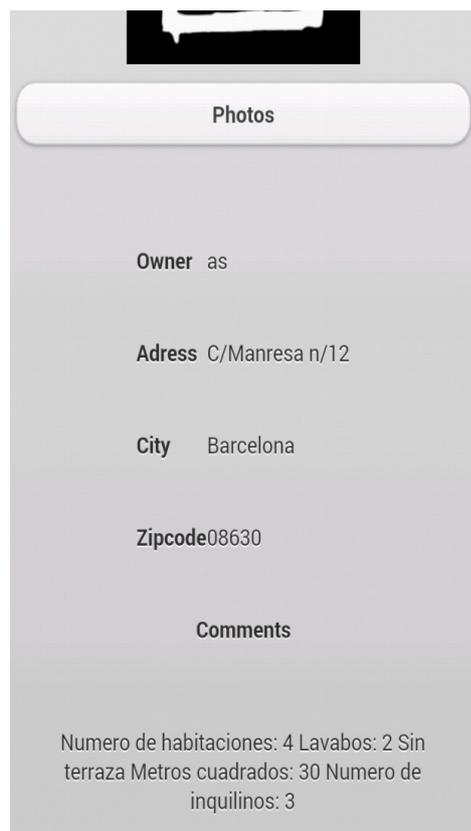


Figura 55

Navbar

Es el menú superior, que nos permite retroceder sobre tus pasos, o acceder directamente al menú principal de la aplicación. Está dedicado a ofrecer una mejor experiencia de navegabilidad al usuario, ya que PhoneFlat tiene deshabilitado el botón físico de Back cuando esta siendo ejecutado en primer plano, para tener un mayor control sobre las acciones del usuario.

9.4.- Instalar el SDK de iOS y Apache Cordova

- Instalar Xcode desde la Mac App Store
- Instalar la “*Command Line Tools*” de Xcode (Xcode Preferences -> Downloads -> Components -> Command Line Tools -> Install).
- Descargar la última versión de PhoneGap
 - Extraer su contenido
 - Apache Cordova iOS se encuentra en *lib/ios*

9.4.1.- Crear Nuevo Proyecto

- Ejecute el instalador 'Cordova-x.x.x.pkg'.
- Copia la carpeta **bin** en algún lugar de tu disco duro.

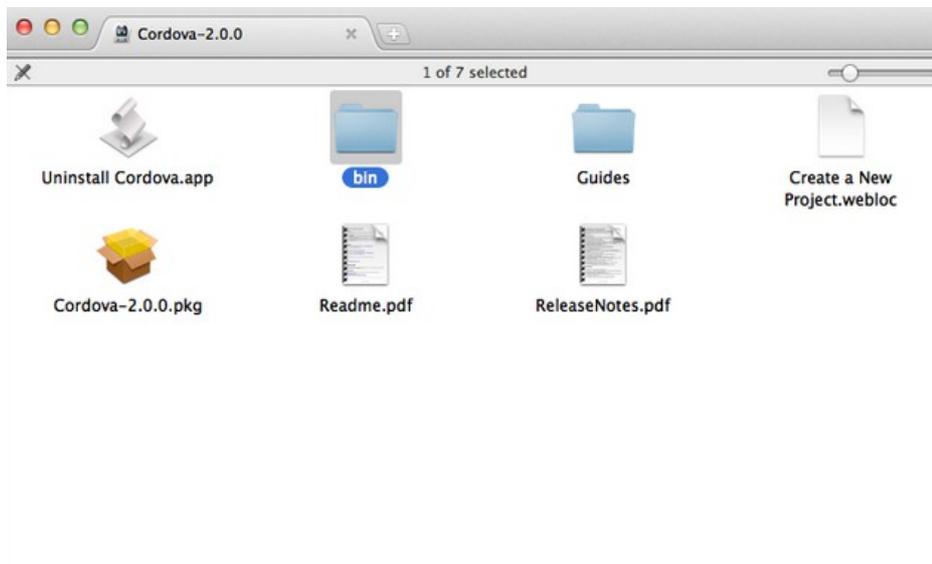


Figura 56

- Inicia **Terminal.app**
- Arrastrar la carpeta bin que se ha copiado, hasta **Terminal.app**, eso debería iniciar una nueva ventana de Terminal.
- Teclar `./create <project_folder_path> <package_name> <project_name>` y pulsar “**Enter**”.

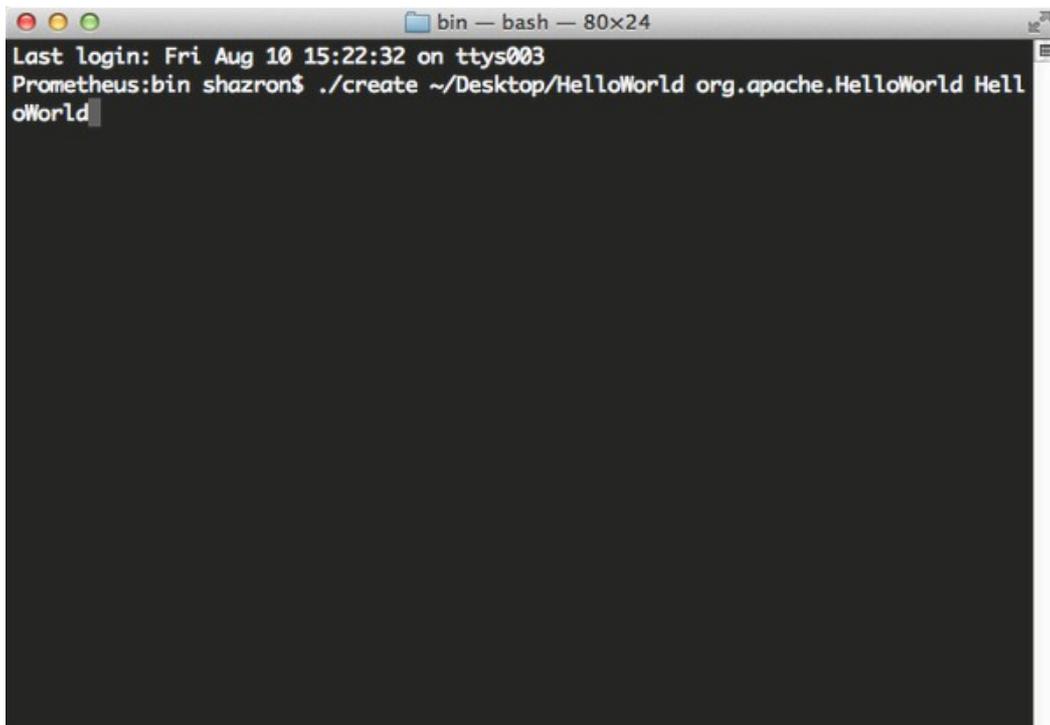


Figura 57

- **Ubica** la carpeta del nuevo proyecto
- **Ejecuta** el fichero `.xcodeproj` en la carpeta.

9.4.2.- “Testing” en Emulador

- Cambia el “*Target*” en el menú desplegable de la barra de herramientas
- Cambia el “*Active SDK*” en el menú desplegable de la barra de herramientas a ***iOS Simulator***
- Presionar sobre el botón **Run**



Figura 58

- Al final del proceso, el usuario deberá ver algo parecido a la siguiente imagen



Figura 59