UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Título del Proyecto:

Peer to Park

Autora:

Navarro Salvador, María Amparo

Director:

Tomás Gironés, Jesús

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

Diploma de Especialización en Desarrollo de Aplicaciones para Android

Septiembre de 2015

Contenido

Introducción 3
Descripción del problema 3
Objetivos 4
Motivación 5
Tecnologías utilizadas 5
Arquitectura de la aplicación 6
Esquema del diseño 6
Modelo de datos 7
Vistas 10
Conclusiones 15
Anexo fuentes 16
Listado de fuentes entregadas o enlace a GitHub 16
Manual de usuario 16

Introducción

Descripción del problema

El principal problema que encuentran los conductores a la hora de desplazarse en coche por la ciudad reside en encontrar una plaza de aparcamiento. Primero por el tiempo que se pierde en encontrar un la plaza de aparcamiento en la calle. Segundo porque la mayoría de las veces, ante la desesperación, se acaba aparcando en zona azul o garajes públicos, donde el precio por hora es elevado. Además, buscar una plaza de aparcamiento supone en media un gasto de 216€ al año dando vueltas intentando encontrar una plaza de aparcamiento.

Cabe destacar que el 64% de la población española se desplaza en coche a trabajar. Por otro lado, en las ciudades donde el problema de aparcar es más alto como Madrid, Barcelona y Valencia, hay más de 5.000 M de desplazamientos al año[1] en compra, cultura y trabajo. Y estos son los sectores que vamos a tomar como punto de partida. Esto se corresponde a un mercado de más de 17.000M de € en tan sólo tres ciudades.

Nuestro mercado se quiere dirigir a alquileres de media jornada a 10 días y así la plataforma está optimizada para la entrega de la llave.

Ilustración 1 – Competencia por tiempo / coste

[1] Instituto de Fomento, 2012

Actualmente nuestra competencia trabaja en alquileres por horas como el caso de los parkings públicos. El problema que tienen es los precios excesivos que pagan los conductores.

Por otro lado están otras Startups que también se dirigen a horas, pero para abrir un garaje particular introducen sus propios mecanismos de apertura, y esto es un handicap ya que tienen que llegar a acuerdos con la comunidad y son procesos lentos y costos, lo que les hace mucho menos escalables que nosotros.

También nos encontramos a plataformas como Idealista y Segundamano, donde ofrecen alquileres de garajes convencionales de mínimo de un mes, perdiendo así esos ingresos que se podrían obtener alquilado por días.

Por último está JustPark, nuestro competidor inglés, que tiene el mismo modelo que nosotros pero que no se ha introducido en el mercado español.

Teniendo en cuenta todo esto, nace Peer to Park, el Airbnb de los garajes. Una plataforma que permite alquilar y buscar garajes, por horas, días, semanas y meses. Nosotros nos centramos en alquileres de media jornada hasta 10 días, ya que hemos visto que en este margen obtenemos mayor beneficio.

Objetivos

Peer to Park cuenta con una aplicación web desarrollada en PHP y MySQL. Partiendo de la misma base de datos el objetivo del proyecto es desarrollar una aplicación para Android complementaria a la aplicación web existente.

Hay que explicar que en esta aplicación hay dos tipos de roles de usuario: usuario que busca aparcar (**conductor**) y usuario **propietario** de un garaje que desea alquilarlo. Cabe destacar que el mismo usuario puede poseer ambos roles a la vez.

La aplicación tiene que constar de las siguientes funcionalidades:

- Búsqueda de garajes por lugar y horas, días, semanas. Los garajes deben de aparecer en el mapa, indicando el precio de cada garaje. Al seleccionar el garaje se verá información más precisa del mismo (foto, dirección, accesibilidad…)
- *Login* de usuario y crear nuevo usuario. Para realizar una reserva o introducir un garaje es necesario estar *logueado* en la plataforma. Los garajes se pueden consultar sin necesidad de *loguearse*.
- Consultar / modificar el perfil de usuario.
- Como conductor se puede realizar:
    - Reserva del garaje. Se pagará para ello con una tarjeta de crédito utilizando el método de pago MangoPay. Si el propietario del garaje no acepta la reserva, se devolverá el dinero.
    - Consultar las reservas realizadas.

  - Cancelar una reserva.
- Como propietario se puede:
  - Introducir los datos de un garaje.
  - Consultar los garajes introducidos, modificando la información y disponibilidad del mismo.
  - Consultar las el estado de las reservas.
  - Cancelar / aceptar una reserva.

Motivación

Este proyecto lo estoy realizando en la empresa Shareparking, Startup de la que soy cofundadora. Actualmente nos encontramos dentro del programa de aceleración para Startups de BBooster. El próximo 6 de octubre se realizará en Madrid el "Demo Day", donde presentaremos el proyecto delante de los inversores, para que lo conozcan y puedan invertir en él si lo desean. El proyecto ha recibido el premio promoción de "Marlboro, es tu oportunidad, 2015", dotado de una cantidad de 20.000€.

Mi cargo dentro de la empresa es el de CTO. Actualmente me encargo de la programación tanto de la aplicación web como de las aplicaciones para móvil (en un futuro queremos realizar la aplicación para iPhone). Por falta de tiempo tuvimos que contar con la ayuda de otro programador que se encargara de finalizar la web y así yo centrarme en la aplicación para Android. Aún falta terminar algunos aspectos de la web, que también han retrasado parte del diseño de la aplicación para Android, al tratarse de la misma base de datos. Aunque no he podido terminar la aplicación por completo dentro del plazo de presentación del proyecto, en breve estará terminado para lanzar el mínimo producto viable, a expensas de ir retocándolo, resolviendo bugs e ir y ampliando funcionalidades según sean las necesidades de los usuarios en un futuro.

Tecnologías utilizadas

Dentro de las tecnologías utilizadas para el desarrollo del proyecto cabe destacar:

- **API Google Maps** para Android, que se utiliza para mostrar en el mapa iconos con el precio de los garajes y su ubicación.
- **PHP y MySQL**: El servidor web que aloja la DB y la aplicación web está desarrollado en PHP con el Framework Codeigniter. Este framework está basado en el Modelo Vista Controlador.
- **JSON**: es un formato ligero para el intercambio de datos. Se utiliza para recuperar la información no sensible de la base de datos, como por ejemplo los detalles de un garaje o la lista de garajes para representar en el mapa.

- Seguridad: Para dotar de seguridad a las transmisiones entre el cliente y el servidor, se ha incorporado el algoritmo **SSL** en el servidor, y la información sensible se manda usando el método POST. SSL, *Secure Sockets Layer* son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

- **Mangopay**: es una solución de pago por Internet con funciones específicas de marketplaces y que cobra una menor comisión que Paypal.

Arquitectura de la aplicación

Esquema del diseño

La plataforma completa de Peer to Park está formada por un servidor Web en el que se almacena toda la información de Peer to Park, el servidor de *Google Maps*, que proporciona la información referente a los mapas, y los usuarios, los cuales pueden acceder mediante navegador Web o usando un smartphone.

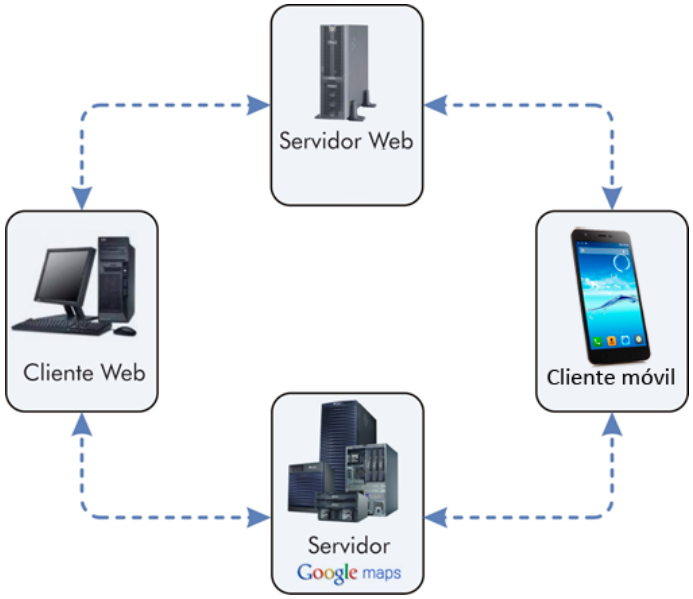


Ilustración 2- Arquitectura del sistema completo

Diagrama de clases

En la Ilustración 3 se aprecia el diagrama de clases resumido de la aplicación, ya que hay clases que poseen muchos más métodos pero que no son relevantes para la explicación del funcionamiento básico de la aplicación. En color verde se encuentran las actividades y en amarillo las clases de ayuda.

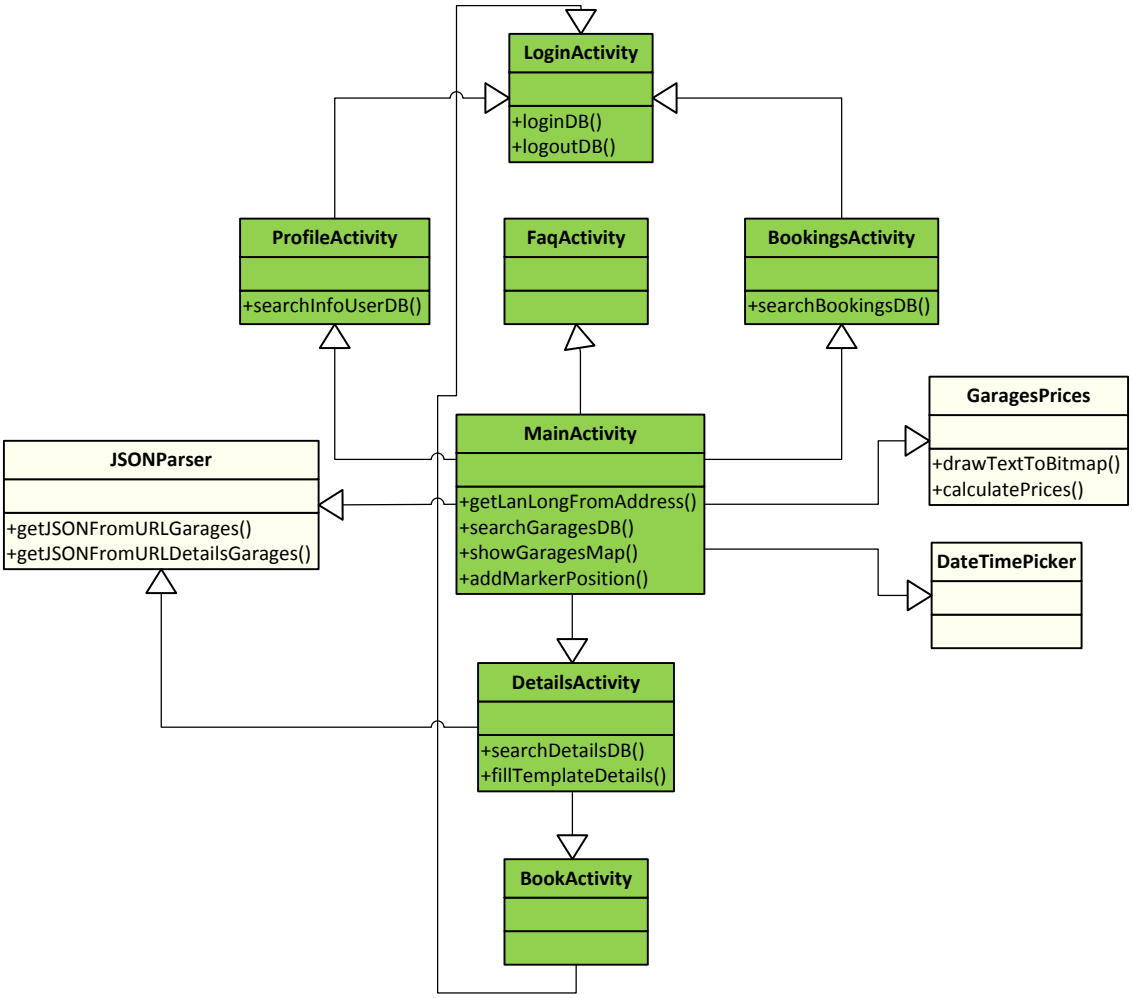


Ilustración 3 - Diagrama de clases

Para todas las comunicaciones con nuestra DB y el servidor de Google se han utilizado AsyncTask para así no bloquear la actividad principal y realizar la petición en hilos secundarios.

Resumiendo un poco el funcionamiento de las clases se tiene:

- ***MainActivity***: es la clase clase principal. Contiene el mapa de google y es la que realiza la búsqueda de los garajes en la DB y los muestra en el mapa.
    - Búsqueda garajes: Se introduce una dirección de destino y con el método *getLangLongFromAddress* se obtiene la dirección de Google Maps y se centra el

mapa en la misma. Con la clase ***dateTimePIcker*** que extiende de DialogFragment se muestran dos cuadros de diálogo para introducir el día y la hora de llegada y de salida.

- Una vez tenemos los datos de búsqueda se realiza la búsqueda con un Asyntask, usando la función *searchGaragesDB* y con *showGaragesMap* se añaden los marcadores en el mapa.
- Para mostrar los garajes en el mapa, esta función se ayuda de la clase ***GaragePrices*** que con la función *calculatePrices* calcula el precio del garaje en función del intervalo de tiempo que ha introducido el usuario y el precio obtenido de la DB por hora/día/semana de cada garaje. Con la ayuda de la función *drawTextToBitmap* se dibuja en un Bitmap el precio del garaje sobre el icono que luego se muestra en Google Maps.

- Los garajes que se obtienen en cada búsqueda son en un radio de 1,5km de donde está centrado el mapa. Si el usuario mueve el mapa dentro de ese radio no se recalculan los garajes pero si sale de ahí, se vuelve a hacer la llamada a la DB. Así los garajes mostrados cada vez no son muchos y el algoritmo es más rápido ya que no sería óptimo buscar una plaza de aparcamiento en Valencia y obtener los garajes de la DB de Madrid y Barcelona a su vez. Esto se realiza en el servidor, en la función que obtiene los garajes de la DB y a la que se le pasa la latitud y la longitud de la dirección de búsqueda.

- La clase ***DetailsActivity*** se encarga de buscar en la DB con el *id_garaje* los detalles del garaje (foto, dirección, descripción, extras…)

- La clase ***BookActivity*** es la que realiza la reserva del garaje y el pago y la clase **BookingsActivity** muestra las reservas del usuario. Estas dos clases se implementarán en breve ya que falta ultimar funciones con el método de pago. Si cuando se quiere realizar la reserva o bien consultar el perfil de usuario, el usuario no se ha logueado, se llama a la clase ***LoginActivity***, que guarda el usuario y el password en las *SharedPreferences*.

- Con la clase ***ProfileActivity*** el usuario puede ver su perfil y editarlo.

- ***FAQActivity***: hace uso de un *webView* para consultar las preguntas frecuentes desde la web, indicaciones de cómo reservar un garaje, cómo ser un buen anfitrión, entre otras cosas.

Modelo de datos

El esquema de la base de datos se puede ver a continuación en la ilustración 4. Actualmente es un diseño reducido puesto que no se dispone aún de todas las funcionalidades deseadas. Poco a poco se irá expandiendo conforme vayamos añadiendo funcionalidades.

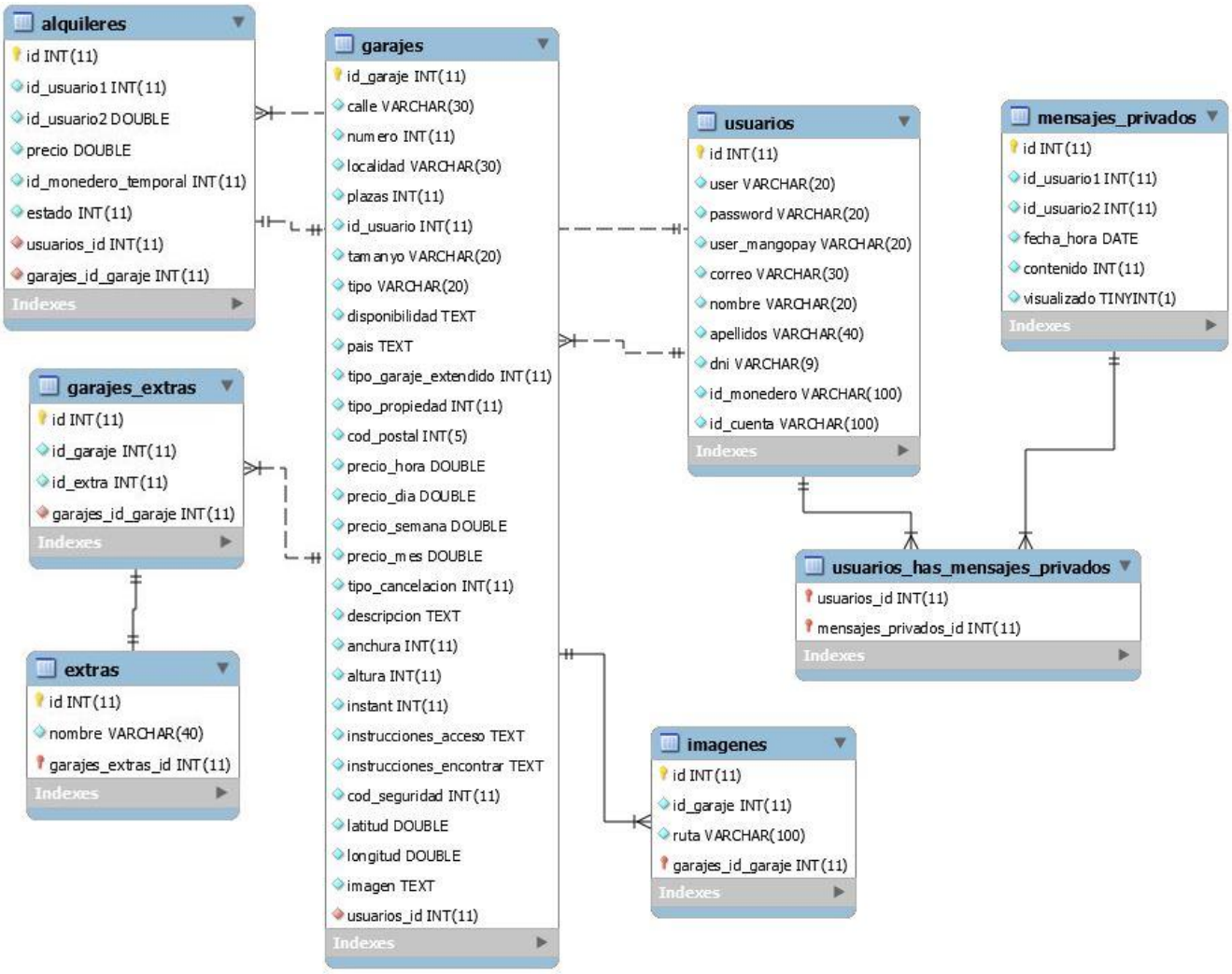
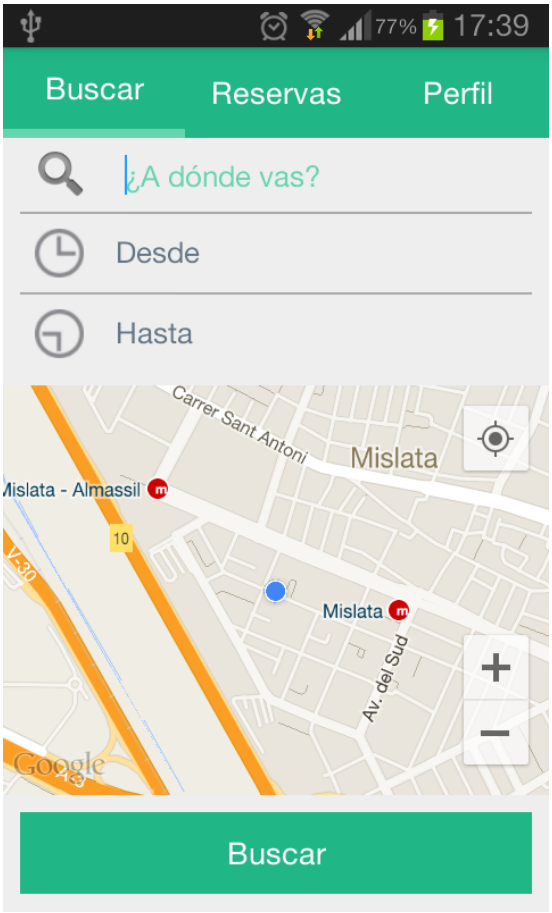


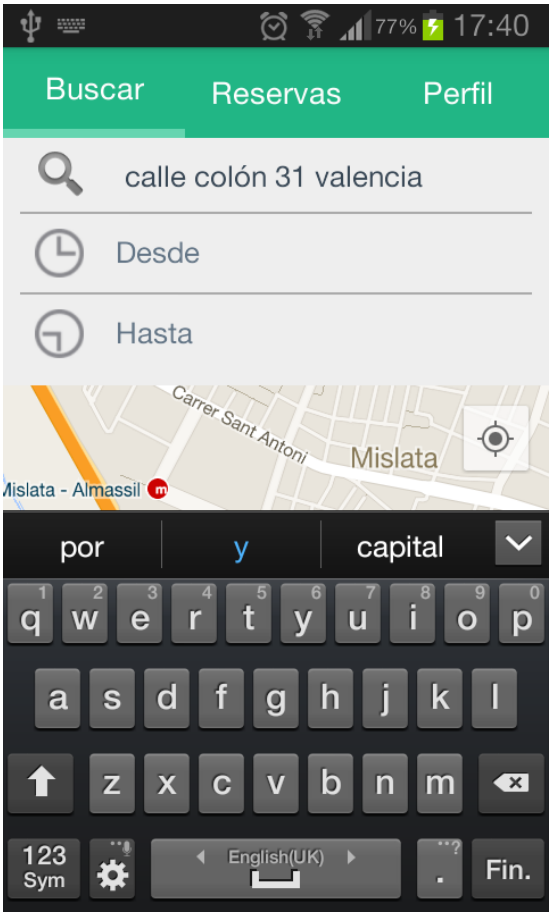
Ilustración 4 - Esquema de la Base de Datos

Vistas

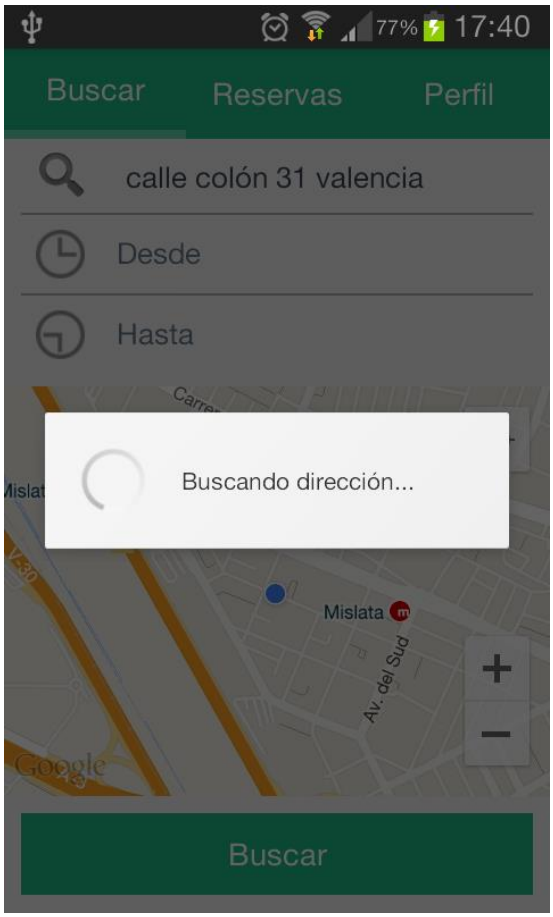
A continuación se muestran las principales vistas de la aplicación, con un resumen del funcionamiento.

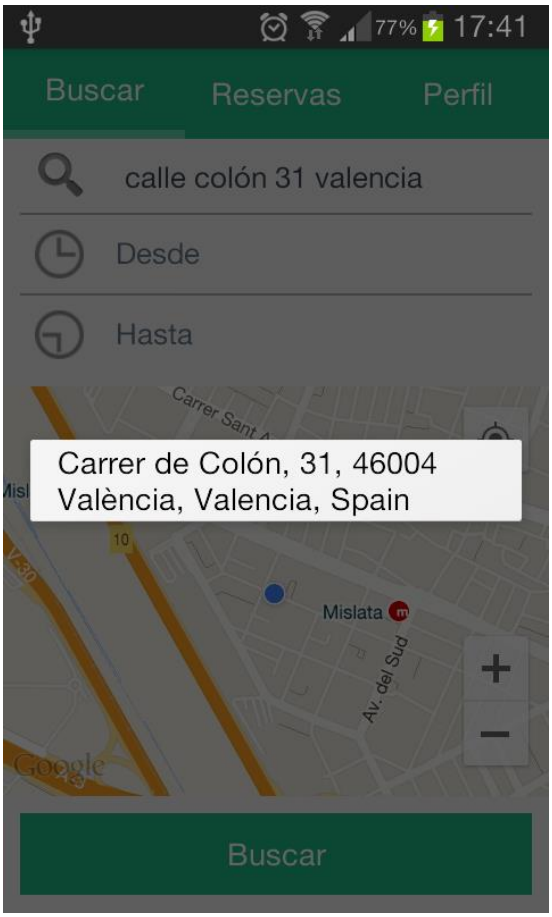
En la pantalla inicial, el usuario introduce la dirección en la cual desea encontrar una plaza de aparcamiento.





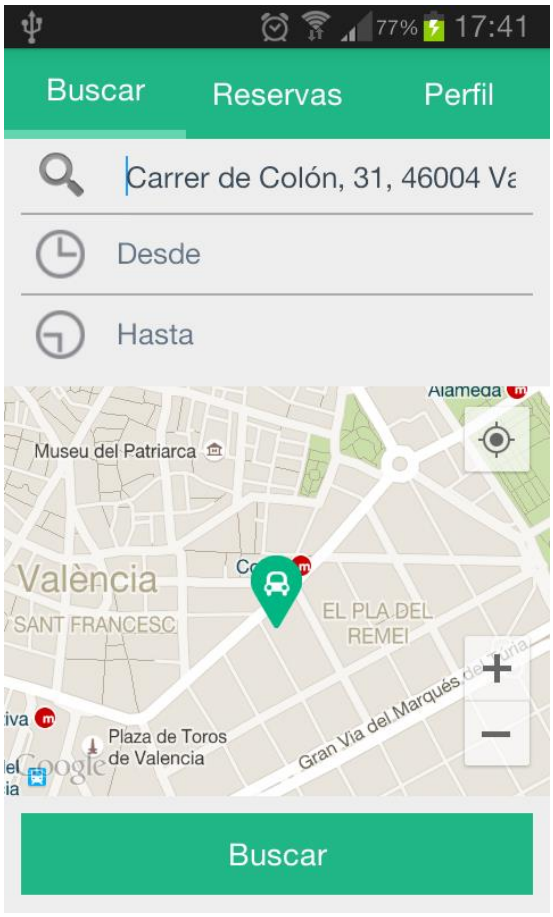
Tras darle al botón del “fin” del teclado, la aplicación busca la “traducción” de esa dirección a las direcciones de Google, y le muestra al usuario en una lista las direcciones que ha obtenido parecidas, para que él elija la correcta:

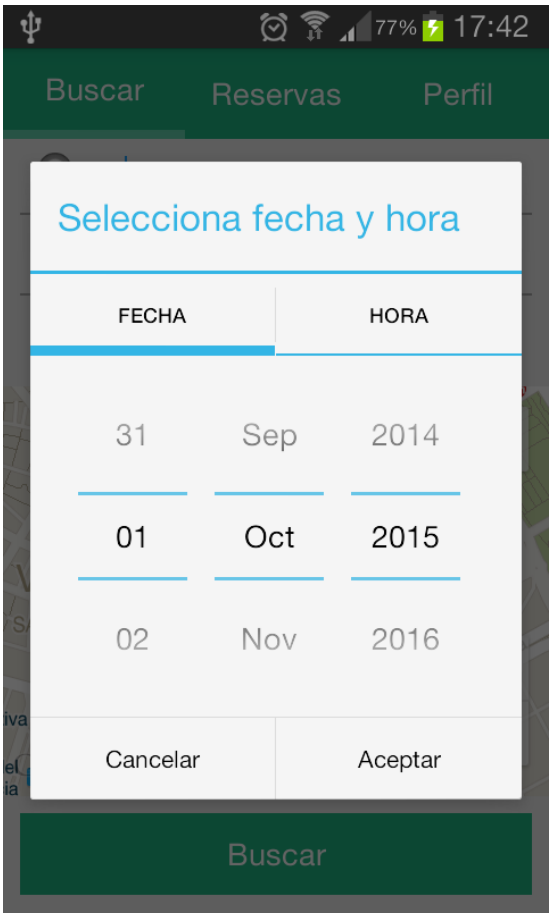




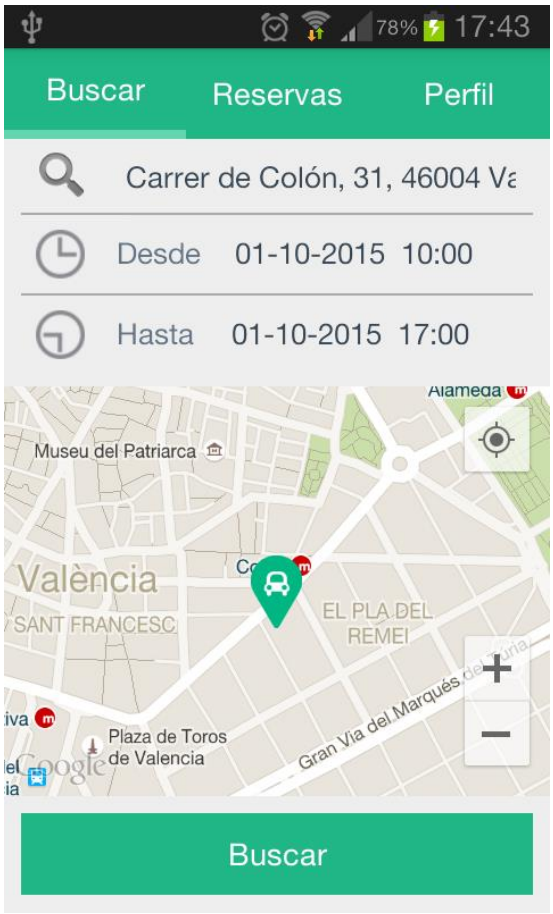
Una vez elegida la dirección, se centra el mapa en la dirección, mostrando el icono de Peer to Park en esa posición.

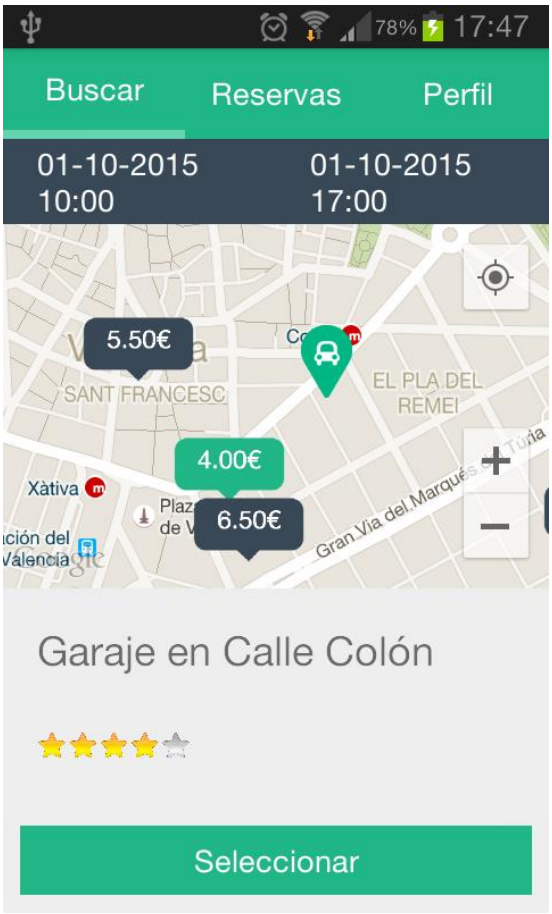
Al pinchar sobre “Desde” y “Hasta” se muestra un *DialogFragment* para seleccionar las fechas y las horas de inicio y fin del aparcamiento.



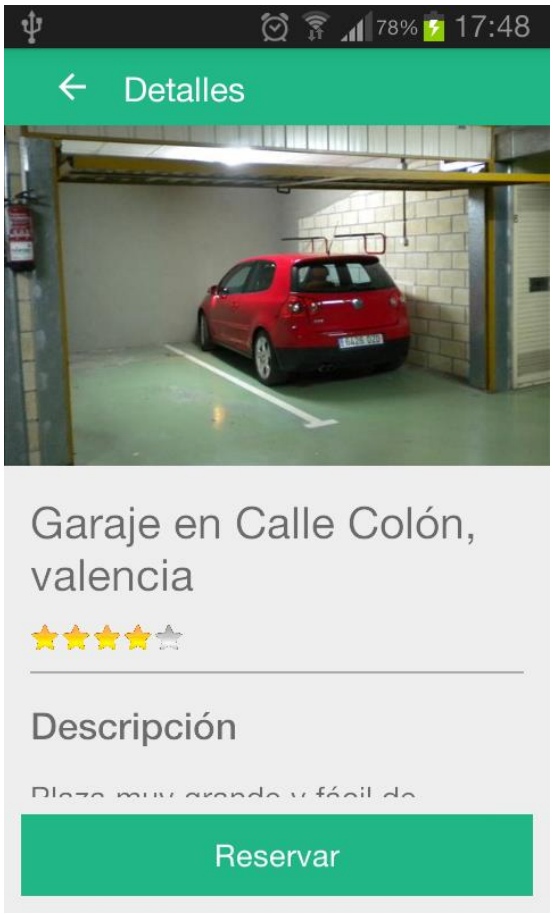


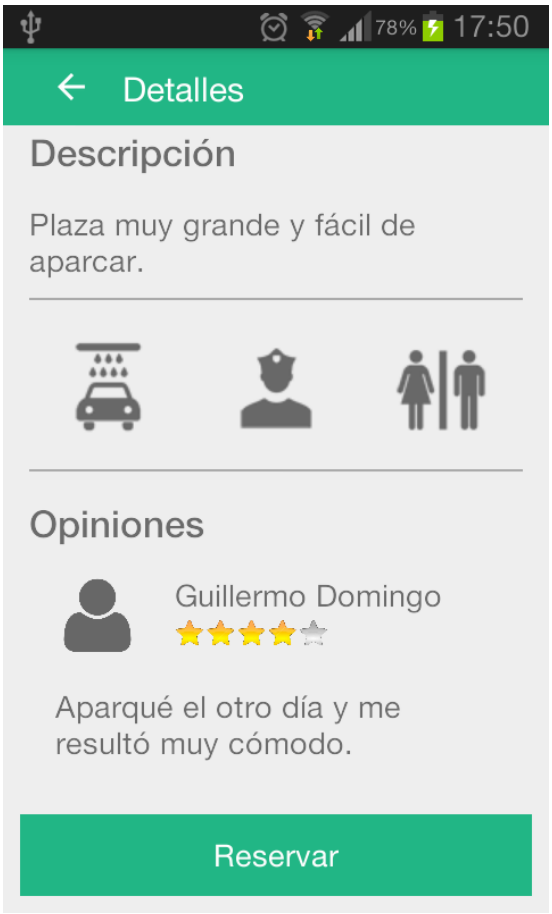
Una vez seleccionadas las horas, se pulsa el botón de "Buscar" y aparecen los garajes en el mapa, con el precio que cuesta el alquiler en ese periodo de tiempo y uno de ellos seleccionado (en verde claro) con una breve información del mismo. Si volvemos a pulsar sobre el resumen de las fechas de entrada/salida, se vuelve a la pantalla anterior y se pueden volver a cambiar las entradas de la búsqueda.



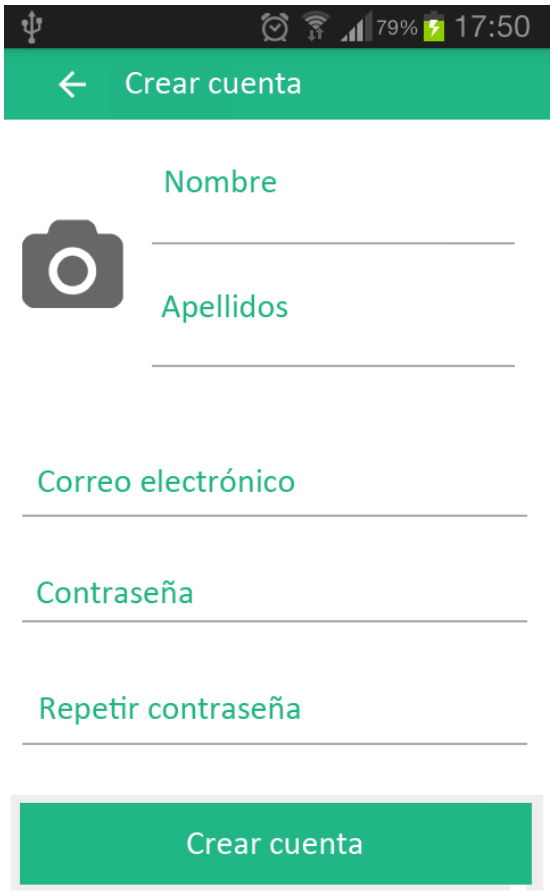


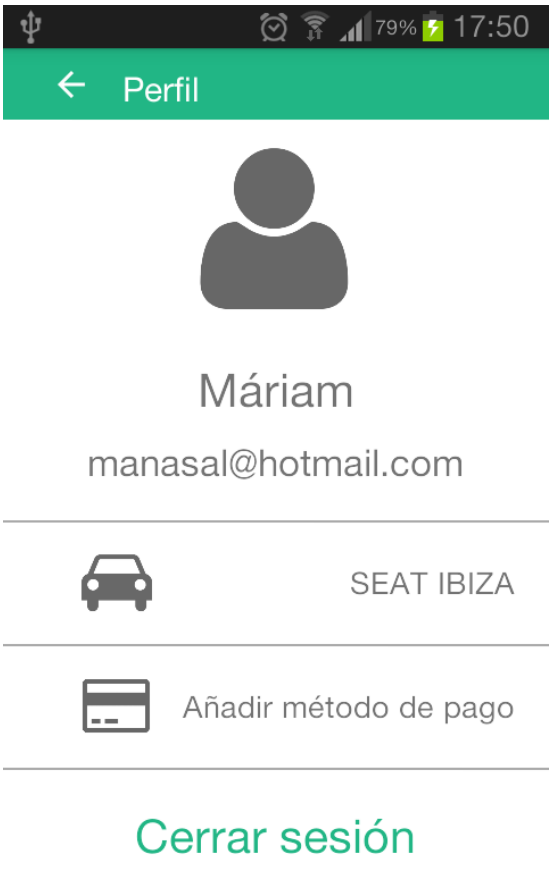
Una vez seleccionado el icono del garaje deseado en el mapa, al pulsar sobre el botón “Seleccionar”, se accede a la pantalla que muestra los detalles del garaje.





Si pinchamos sobre las pestañas superiores de “Reservas” o “Perfil” y no nos hemos dado de alta en la aplicación, aparece la pantalla de login, para darse de alta como un usuario nuevo o para introducir los datos de un usuario ya existente. En la pestaña de “Perfil”, se accede al perfil de usuario.





Conclusiones

La aplicación planteada en un principio es muy completa por lo que para la entrega del proyecto, dado que también he estado anteriormente desarrollando la web, no he podido terminarla por completo. Cabe destacar que las funcionalidades básicas ya están desarrolladas. Y las funciones de comunicación con el servidor para consultar o insertar datos en la base de datos ya están definidas, a falta de adecuarlas a las llamadas que se requieran en un futuro.

Para el desarrollo de este proyecto se han cumplido los siguientes objetivos:

- Desarrollo de una aplicación para Android, complementaria a la aplicación web existente.
- Funcionalidades de la aplicación:
    - Búsqueda de garajes por lugar y horas, días, semanas.
    - Visualización de los detalles de garaje

- Login de usuario existente (se puede probar con usuario: manasal / pwd: p1234)
- Consultar el perfil de usuario.
- FAQ

Como líneas futuras, cabe destacar las siguientes dependiendo del rol del usuario:

- Rol conductor:
    - Reserva del garaje.
    - Implementar Mangopay.
    - Consultar las reservas realizadas.
    - Cancelar una reserva.
    - Insertar una reseña del garaje.
- Desarrollar el rol propietario:
    - Introducir los datos de un garaje.
    - Consultar y modificar los garajes introducidos.
    - Consultar las el estado de las reservas.
    - Cancelar / aceptar una reserva.

La aplicación va a estar en continua mejora, para ir cubriendo las necesidades que el cliente requiera, además de ir adecuándose a las nuevas tecnologías que vayan surgiendo.

Anexos

Listado de fuentes entregadas

Se ha entregado anexo al proyecto el código fuente de la aplicación, si bien, por motivos de protección de datos, no deseamos hacerlo público.