



Eldelbar

Nombre Estudiante: Alejandro Hernández Doménech
Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: David Escuer Latorre
Profesor/a responsable de la asignatura: Carles Garrigues Olivella

Fecha de Entrega: 06 de Junio de 2018



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Eldelbar</i>
Nombre del autor:	<i>Alejandro Hernández Doménech</i>
Nombre del consultor/a:	<i>David Escuer Latorre</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2018
Titulación::	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>copas, ahorro, eldelbar</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>La presente aplicación se realiza con la finalidad de dotar al usuario de una herramienta con la que poder controlar sus gastos de ocio.</p> <p>Es sabido que cada vez más, tras la crisis que ha sucedido en los últimos años y de la que aún la sociedad se está recuperando, es cada vez más necesario llevar un control de gastos a nivel personal.</p> <p>Existen multitud de aplicaciones de control general de gasto, sin embargo esta aplicación se centra exclusivamente en el control de gastos de los momentos de ocio.</p> <p>El ocio es algo muy necesario, sobre todo en esta sociedad tan estresante, con lo que introducir una herramienta capaz de ayudar a dicho control, es algo útil y necesario.</p> <p>Para la realización del proyecto se ha realizado una planificación previa, y un posterior estudio de mercado y de los posibles usuarios.</p> <p>Hecho esto se ha procedido al desarrollo de un prototipo, y de la aplicación final después.</p> <p>Como conclusión cabe destacar por un lado el grado de conocimiento aprendido, al ser la primera aplicación realizada de principio a fin, así como la gran carga de trabajo que ha supuesto.</p> <p>Ha sido muy difícil y costoso seguir la temporización, por no decir imposible,</p>	

pero el resultado ha sido satisfactorio.

Abstract (in English, 250 words or less):

The present application is made with the purpose to provide a tool to the userch to control their leisure expenses.

After the crisis that we have happened in recent years, and which society is still recovering, it is very necessary to keep track of expenses on a personal level.

There are many general expenditure control applications, which are focused exclusively on the general expenditure control. This application is focused in expenditure control only y leisure time.

Leisure is very necessary, especially in this stressful society, so this kind of tool, capable of helping this control is something useful and necessary.

For the realization of the project, a previous planning has been carried out, and a later study of the market and of the possible users.

Once this has been done, a prototype has been developed, and the final application later.

In conclusion, on the one hand, the degree of knowledge learned is noteworthy, since it is the first application carried out from the beginning to the end, as well as the great workload that it has entailed.

It has been very hard and difficult to follow the timing, but almost impossible, but the result has been satisfactory.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2.- Estudio de mercado.....	2
1.3 Funcionalidades y Objetivos del Trabajo.....	4
1.4 Enfoque y método seguido.....	4
1.5 Planificación del Trabajo.....	7
1.6 Breve resumen de productos obtenidos.....	8
1.7 Breve descripción de los otros capítulos de la memoria.....	9
2. Análisis, diseño y evaluación.....	10
2.1.-Análisis de usuarios y contexto de uso.....	10
2.1.1.- Análisis de usuarios.....	10
2.1.2.- Necesidades.....	13
2.1.3.- Perfiles de usuario.....	13
2.1.4.- Fichas persona.....	13
2.2- Diseño.....	17
2.2.1.- Árbol de navegación.....	17
2.2.2.- Diagrama de flujo.....	18
2.2.3.- Prototipo.....	19
2.2.4.- Diagrama de casos de uso.....	25
2.2.5.- Arquitectura.....	27
2.2.6.- Base de datos.....	29
3.- Desarrollo.....	31
3.1.- Firebase.....	31
3.1.1.- Autenticación y registro de usuarios.....	32
3.2.- Base de datos SQLite.....	33
3.3.- Geolocalización.....	36
4.- Pruebas.....	37
4.1.- Pruebas de la interfaz.....	38
4.2.- Pruebas de navegación.....	38
5.- Viabilidad económica.....	38
5.1.- Inversión inicial.....	38
5.2.- Hardware.....	39
5.3.- Software.....	39
5.4.- Marketing.....	39
5.5.- Otros.....	39
5.6.- Resumen de costes.....	40
5.7.- Recuperación de la inversión.....	40
6.- Manual de instalación.....	41
7.- Manual de usuario.....	42
8.- Revisión de la planificación.....	44
9.- Conclusiones.....	45
10.- Glosario.....	46
11.- Bibliografía.....	47
12.- Anexos.....	48

1. Introducción

1.1 Contexto y justificación del Trabajo

Actualmente, y sobre todo debido a la crisis económica que estamos sufriendo, con falta de trabajo y sueldos mileuristas, la gente ha de trabajar para sacar adelante a su familia y su hogar.

Es necesario trabajar para poder pagar las facturas a final de mes, pero también son necesarios los momentos de ocio con los amigos, momentos en los que poder salir a tomar algo y relacionarse.

El siguiente trabajo se centra en esos momentos de ocio, ayudando al usuario a ahorrar cuando sale a tomar algo, ya sea solo o con los amigos.


Mediante esta aplicación, el usuario sabrá el precio de las copas/tapas de los locales de ocio que tiene cerca (o no tan cerca, según le interese), con lo que tendrá un mejor control de gastos, ya que podrá acudir al local que tenga mejor precio o mejor oferta en ese momento.


La aplicación está pensada para un usuario con una media de edad comprendida entre 20 y 40 años, sin familia, con sueldo medio de unos 1000€/mes, ya que son quienes tienen más dificultades, y por tanto deben ser más mirados a la hora de gastar, sobre todo en ocio.


Se trata de gente joven, que además son usuarios de teléfono móvil. Por tanto, nuestra aplicación se centraría en su uso mediante smartphone, aunque lógicamente se adaptaría para su uso en tableta.

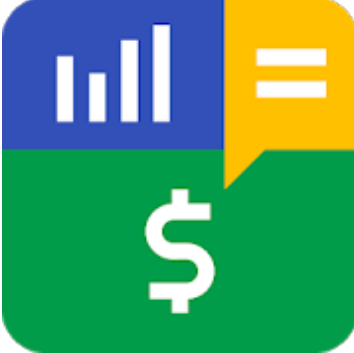
1.2.- Estudio de mercado

Según un estudio realizado en *Play Store*, se han encontrado las siguientes aplicaciones:

Soysuper - Supermercados	
	<p>Puntos fuertes:</p> <ul style="list-style-type: none">• Compara precios de supermercados online. <p>Puntos débiles:</p> <ul style="list-style-type: none">• Solamente da servicio para comprar en supermercados online.
Se diferencia de mi aplicación	
Solo sirve para supermercados online. Nuestra aplicación serviría para locales de ocio, tanto con presencia online, como sin ella.	

Tiendeo – Catálogos y ofertas	
	<p>Puntos fuertes:</p> <ul style="list-style-type: none">• Descarga catálogos de tiendas online. <p>Puntos débiles:</p> <ul style="list-style-type: none">• No sirve para comprar en supermercados online.
Se diferencia de mi aplicación	
No compara los precios, sino que da información de cada tienda por separado. Nuestra aplicación se centraría en locales de ocio de copas/tapas.	

Bluecoins – Finanzas y Presupuesto	
	<p>Un buen programa para llevar el control general de gastos.</p> <p>Puntos fuertes:</p> <ul style="list-style-type: none"> • Interfaz amigable • Diseño robusto. Muy completo.
Se diferencia de mi aplicación	
<p>Nuestra aplicación no es tan general. Se centra en momentos muy concretos de la vida del usuario, como son los momentos de ocio.</p>	

Mobills – Control de gastos y Finanzas Personales	
	<p>Al igual que la anterior, un buen programa para llevar el control general de gastos,</p> <ul style="list-style-type: none"> • muy general.
Se diferencia de mi aplicación	
<p>Nuestra aplicación es más específica en su función.</p>	

1.3 Funcionalidades y Objetivos del Trabajo

Funcionales

1. Localizar automáticamente la ubicación del usuario.
2. Mostrar un listado de los locales que hay cerca, y su ubicación.
3. Integración con *Google Maps* para encontrar una ruta adecuada desde el sitio actual al local escogido.
4. Mostrar la información del local elegido.
5. Realización de una base de datos con las sesiones de gastos en los locales visitados y su fecha.

No Funcionales

1. Disponible para su descarga en *Play Store*.
2. Sincronización automática con la base de datos del servidor para la gestión de usuarios.

1.4 Enfoque y método seguido

Tras realizar un estudio de la posible competencia, se observa que hay diversas aplicaciones que tratan de alcanzar un mismo objetivo, hacer que el usuario ahorre y lleve un mejor control de sus gastos. Todas las analizadas son de uso general, que permiten al usuario el control de todos sus gastos, convirtiéndose muchas veces en una aplicación muy compleja y difícil de utilizar.

Esta aplicación, se centra en un sector que por el momento no tiene mucha competencia, como es el sector del ocio, y desde luego, actualmente no existe ninguna aplicación que realice la misma función que la nuestra, con lo que la estrategia más apropiada es la de desarrollarla desde cero, consiguiendo cada uno de los objetivos.

Según el informe '*Google Consumer Barometer Report*', presentado en la *Mobile World Congress 2017* de Barcelona, el 87% de los españoles son usuarios de smartphone.

Desarrollando la aplicación para smartphone obtendríamos las siguientes ventajas:

- El smartphone es un dispositivo cotidiano que, tal y como revela el informe mencionado anteriormente, lleva consigo el 87% de la población, siendo esta de todas las edades.
- Cada nueva publicación será recibida por el usuario de forma instantánea.
- El uso del smartphone se ha extendido prácticamente a todos los ámbitos, desde comunicación, entretenimiento, organización, etc..

La siguiente tabla compara las características principales de las aplicaciones nativas, híbridas y aplicaciones web.

	Aplicaciones Nativas	Aplicaciones Web	Aplicaciones Híbridas
Acceso a Hardware	Completo	Parcial	Completo
Velocidad	Muy rápida	Rápida	Rápida
Costo	Elevado	Moderado	Moderado
Gestos <u>Multitouch</u>	Disponible	No	Disponible
Instalación	Requerida	No, solo browser	Requerida
<u>Market Place</u>	Disponible	No	Disponible
Necesita Aprobación	Obligatorio	No	Obligatorio
Multiplataforma	No	Si	No
Conexión a Internet	Offline	Si	No necesaria
Gráficos y Efectos	Superior, solo limitado por el dispositivo	Limitado	De acuerdo a implementación
Experiencia de usuario	Robusta	Limitada	Aceptable

Figura 1: Comparativa entre aplicaciones nativas, híbridas y web

Atendiendo a la información vista en la anterior tabla donde se compara las diferencias entre los diferentes tipos de aplicaciones, se ha decidido realizar de forma nativa para Android.

Dado que se busca conseguir buenos tiempos de respuesta en la sincronización con la base de datos de donde se actualizará el dispositivo, geolocalización y otras características, se ha optado por realizarla de forma nativa con objeto de poder optimizar estas tareas al máximo.

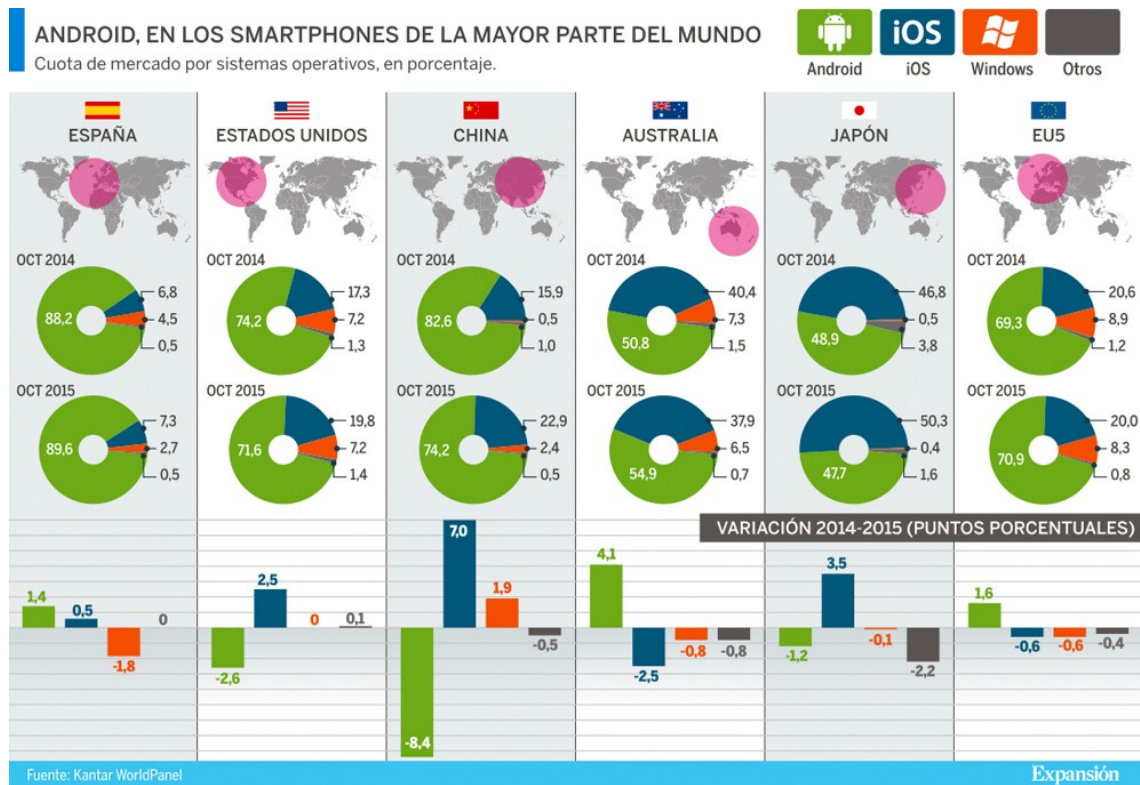


Figura 2: Uso de los distintos Sistemas Operativos móviles. Foto sacada de Expansión Digital.

Tal y como se observa en la comparativa de uso de los distintos sistemas operativos, el sistema operativo más utilizado es Android (cerca del 90%), frente a iOS o Windows Phone.

1.5 Planificación del Trabajo

El TFM se ha planificado en diversas tareas, basándonos en las PEC propuestas. Mediante la herramienta GanttProject, se muestra dicha planificación, mediante una sucesión de tareas, seguida del consiguiente diagrama de Gantt.

Una primera tarea sería la presente PEC, dedicada a la planificación del TFM :

PEC1: Planificación y temporización del TFM:

Inicio: 21/02/18

Fin: 14/03/18

Nombre	Fecha de inicio	Fecha de fin
PEC1	21/02/18	15/03/18
PEC1_Contexto y justificación	21/02/18	25/02/18
PEC1_Ojetivos	26/02/18	2/03/18
PEC1_Planificación y temporización	3/03/18	10/03/18
PEC1_Otros capítulos	11/03/18	15/03/18
PEC2	15/03/18	4/04/18
PEC2_Analisis de usuarios y contextos de uso	15/03/18	15/03/18
PEC2_Diseño conceptual	16/03/18	16/03/18
PEC2_Prototipado	17/03/18	22/03/18
PEC2_Evaluacion	23/03/18	24/03/18
PEC2_Def casos de uso	25/03/18	29/03/18
PEC2_Diseño de la arquitectura	30/03/18	4/04/18
PEC3_Desarrollo	5/04/18	16/05/18
PEC3_Desarrollo de la GUI	5/04/18	7/04/18
PEC3_Desarrollo de la bd	8/04/18	12/04/18
PEC3_Desarrollo del programa	13/04/18	16/05/18
PEC4_Realización de la memoria y vídeo	17/05/18	6/06/18

Figura 3: Planificación de sesiones por días.

Nombre	Horas estimadas
• PEC1	56 horas
• PEC1_Contexto y justificación	12 horas
• PEC1_Ojetivos	14 horas
• PEC1_Planificación y tempori...	16 horas
• PEC1_Otros capítulos	14 horas
• PEC2	50 horas
• PEC2_Analisis de usuarios y ...	4 horas
• PEC2_Diseño conceptual	2 horas
• PEC2_Prototipado	14 horas
• PEC2_Evaluacion	6 horas
• PEC2_Def casos de uso	8 horas
• PEC2_Diseño de la arquitect...	14 horas
• PEC3_Desarrollo	98 horas
• PEC3_Desarrollo de la GUI	8 horas
• PEC3_Desarrollo de la bd	14 horas
• PEC3_Desarrollo del programa	76 horas
• PEC4_Realización de la memoria...	47 horas

Figura 4: Planificación de sesiones por horas.

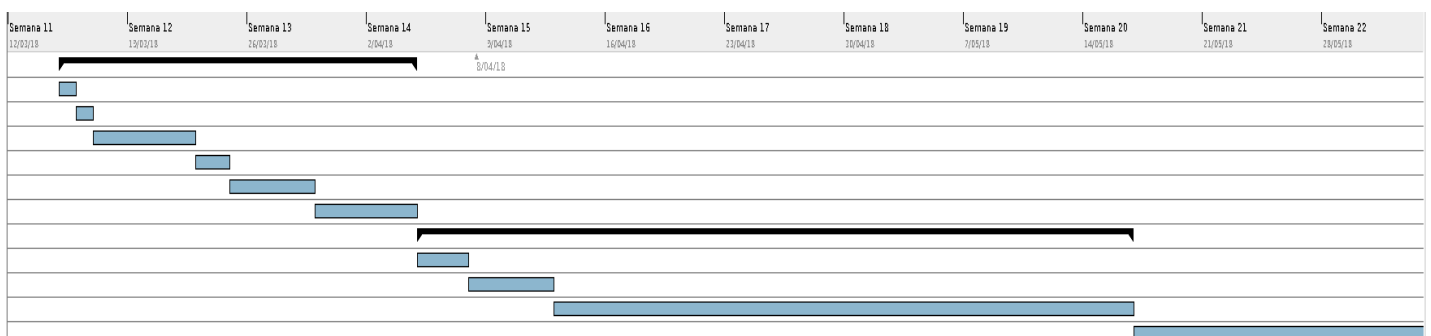


Figura 4: Diagrama de Gantt con la temporización del TFM

1.6 Breve sumario de productos obtenidos

Finalizado el TFM, obtendremos los siguientes entregables:

- Una archivo comprimido (.zip) con el código de la aplicación.
- Un ejecutable (.apk) instalable en dispositivos Android.
- Memoria del TFM realizada en PDF.
- Vídeo de presentación del proyecto.

1.7 Breve descripción de los otros capítulos de la memoria

Capítulos:

- 2.1.- Diseño.
- 2.2.- Implementación y desarrollo.
- 2.3- Pruebas de evaluación.
- 2.4.- Propuestas de mejora.
- 2.5.- Vídeo explicativo.

2.1.- Diseño.

Descripción de las partes más importantes que se han tenido en cuenta en cuanto al diseño de la aplicación.

2.2.- Implementación y desarrollo.

Aspectos importantes en cuanto a la implementación y desarrollo del proyecto.

2.3.- Pruebas y evaluación.

Pruebas de calidad realizadas a la aplicación y sus conclusiones.

2.4.- Propuestas de mejora.

Propuestas de posibles mejoras a realizar *a posteriori* en la aplicación.

2.5.- Vídeo explicativo.

Acerca de cómo se ha realizado el vídeo de presentación de la aplicación, así como las herramientas utilizadas en su elaboración.

2. Análisis, diseño y evaluación

En el siguiente capítulo se analizarán y especificarán las características de los usuarios, sus necesidades y el contexto de uso. Se describirán también las funcionalidades que tendrá la aplicación. También se procederá a realizar el diseño conceptual de la misma.

2.1.-Análisis de usuarios y contexto de uso

2.1.1.- Análisis de usuarios

¿Qué edad tiene?

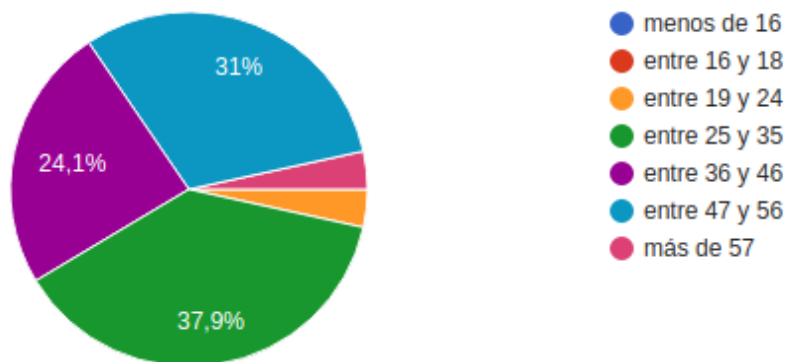


Figura 9: Encuesta 1

¿Qué tipo de teléfono móvil utiliza?

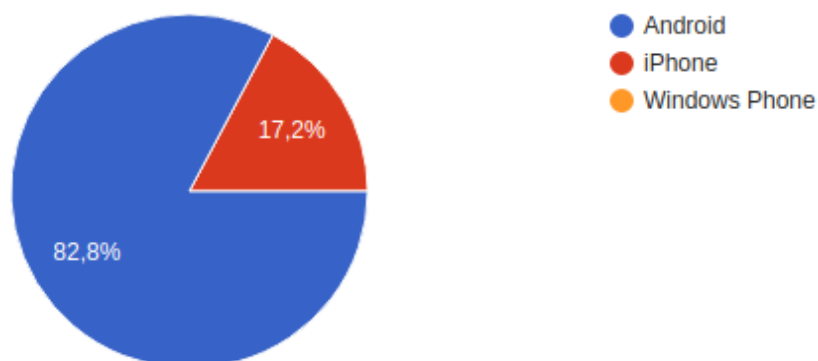


Figura 10: Encuesta 2

¿A qué tipo de local de ocio le gusta ir con sus amigos?

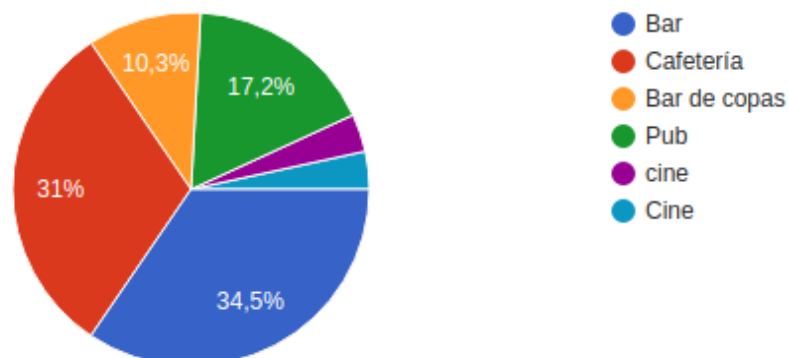


Figura 11: Encuesta 3

¿Suele salir siempre a los mismos lugares?

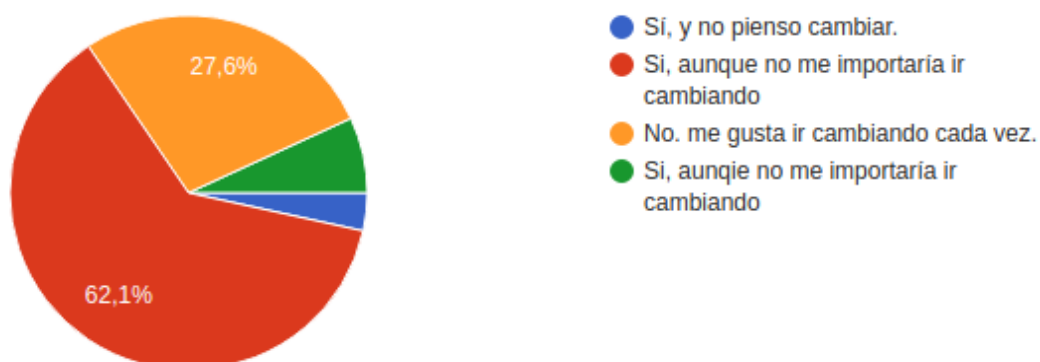


Figura 12: Encuesta 4

¿Suele fijarse en las ofertas o precios en copas o tapas que tienen los locales de ocio?

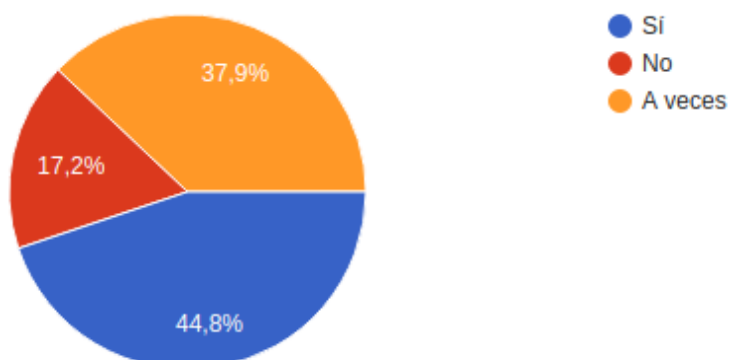


Figura 13: Encuesta 5

¿Le parecería interesante conocer las ofertas o precios en bebida/copas de los locales de ocio que tiene cerca?

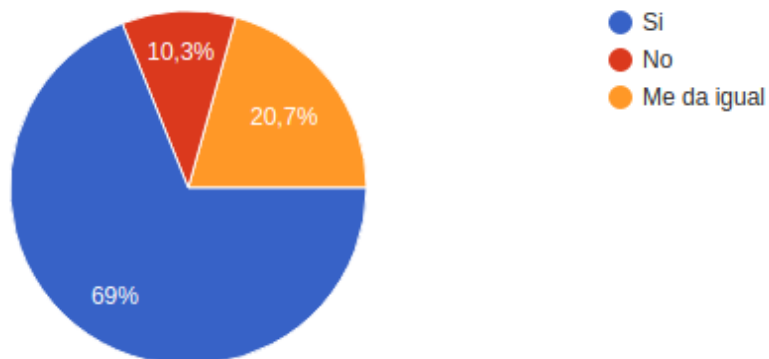


Figura14:Encuesta 6

Cuando sale con sus amigos a una cafetería, bar, etc.. ¿Qué tipo de bebida suele tomar?

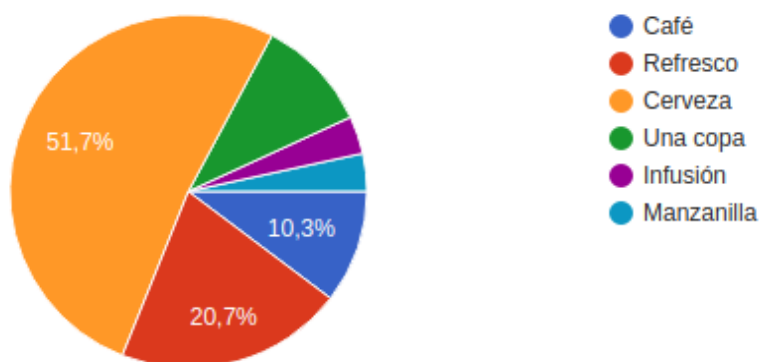


Figura 15: Encuesta 7

La encuesta ha sido realizada por un total de 29 personas. El rango de edad más significativo es el comprendido entre 25 y 35 años, ya que el 37,9% de los encuestados tenían edad comprendida en este rango.

El 100% de los usuarios utiliza el teléfono móvil, destacando un 82,8% que utiliza smartphone con sistema operativo Android, frente a un 17,2% que utiliza habitualmente iPhone.

2.1.2.- Necesidades

Tras la obtención y análisis de los resultados de la encuesta, se observa:

- La mayor parte de los usuarios, un 44,8%, suelen fijarse en los precios de los productos de los locales donde van a consumir y el resto, el 37,9%, lo hacen a veces.
- Además, a un 69% de los encuestados les parecería interesante conocer los precios de antemano.

Se observa por tanto:

- La necesidad de tener una herramienta para poder conocer los precios de los productos de los locales de ocio.
- Una herramienta que sea fácil de utilizar y muestre de forma clara la información que le interesa al usuario.

2.1.3.- Perfiles de usuario

En cuanto al perfil del usuario que utilizaría nuestra app, podemos decir tras analizar los datos de la encuesta:

Usuario joven, de edades comprendidas entre 25 y 35 años, con trabajo. Además, son usuarios que no son asiduos a salir siempre a un mismo local, sino que les gusta ir cambiando de local de ocio de vez en cuando y siempre llevando el control de sus gastos.

2.1.4.- Fichas persona

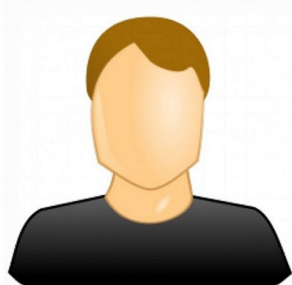
Principales motivaciones y/o necesidades para usar nuestra app:

- La principal motivación que puede llevar a los usuarios potenciales a usar esta aplicación es el hecho de poder ahorrar o por lo menos no gastar de más a la hora de tener momentos de ocio con las amistades, normalmente yendo a una terraza de cafetería/bar, de tapas, etc..
- Otra motivación puede ser aprovechar las ofertas que hay en los diferentes locales en cada momento para no ir siempre al mismo, es decir, conocer nuevos locales, los cuales tienen otros entornos y especialidades.

Contexto de uso

Esta aplicación está pensada para ser usada en cualquier momento, por lo que principalmente se desarrolla para dispositivos smartphone con conexión a datos y GPS para poder localizar el dispositivo y proporcionar la información de los locales que hay cerca del usuario.

Cuando un usuario está en casa o en la calle, en el momento en el que esté decidiendo donde ir, esta aplicación le será de gran ayuda.

	<p>Nombre: Adrián Edad: 26 años Nivel de estudios: Ciclo Formativo Profesión: Empleado de la ITV</p>
Ficha persona	
<p>Adrián es un chico de 26 años, el cual le gusta tener una vida independiente. Es titulado de ciclo formativo de grado medio de mecánica, y hace tres años consiguió un trabajo en la ITV. Por fin este año le han hecho fijo y ha decidido independizarse.</p> <p>Después de buscar, ha encontrado un piso de unos 65 metros cuadrados donde ha empezado su nueva vida.</p> <p>En casa Adrián no tiene conexión a Internet, pero hizo amistad con un vecino, y junto a un tercer vecino han decidido compartir entre los tres una conexión a Internet. Uno de ellos que tiene conocimientos de informática es quien les ayudó a realizarlo.</p> <p>No posee unos ingresos muy altos, por lo que cada vez es más consciente de que si quiere tener dinero para poder pagar todos los gastos, salir de vez en cuando con los amigos algún fin de semana, y ahorrar un poco, va a tener que organizarse y administrar muy bien todos sus gastos, intentando ahorrar al máximo cada mes.</p> <p>Le gusta salir de fiesta con sus amigos, sobre todo a una cafetería o pub a tomar unas cervezas, pero últimamente han subido el precio. Se están planteando cambiar de local para probar cosas nuevas, y a ser posible un poco más económicas. De vez en cuando se enteran de locales que hacen alguna oferta de "bebida+tapa" y aprovechan, pero no siempre.</p>	

A Adrián le gusta mucho trastear con el móvil, suele usarlo mucho tanto para comunicarse, como para organizarse el tiempo libre. Últimamente lo usa para controlar todos sus gastos, aunque de momento no ha encontrado una aplicación que realmente le guste en ese sentido.

Ficha escenario

Adrián hace poco que se ha independizado de sus padres y vive solo en un piso en alquiler. Está preparando oposiciones a Cuerpos de Seguridad del Estado, por lo que acude 2 veces por semana a una academia a prepararse.

Posee vehículo propio, y lo tiene totalmente pagado. Trabaja a jornada completa como empleado de la ITV y su salario mensual no llega a los 1000€, con lo que ha de tener cierto cuidado para poder llegar a final de mes sin dificultades.

Es jueves por la noche, y Adrián ha terminado muy pronto de trabajar. Ha aprovechado para ir a estudiar un rato a la biblioteca, y ahora, a las 20:30 no le apetece nada irse a casa. De camino a casa ve un cartel en una cafetería donde esta noche y mañana por consumir una cerveza o vino, regalan una tapa, así que avisa a sus amigos Pedro y Juan, quienes gustosamente avisan al resto de amigos para ir a tomar unas cervezas y beneficiarse de la oferta.

Al llegar al local, se dan cuenta que incluso regalando la tapa, la cerveza es un poco más barata que en el local donde van habitualmente, incluso les gusta más.

Todos están contentos, porque con menos dinero del que habitualmente gastan cada vez que salen, han estado disfrutando de un buen rato, y casi casi han podido cenar.



Nombre: Elisa
Edad: 21 años
Nivel de estudios: Bachiller.
Profesión: Estudiante de Bellas Artes.

Ficha persona

Elisa es una estudiante universitaria de 21 años. Vive en un piso compartido en Valencia, con sus 3 compañeras de piso. Cada día acude a clase a la universidad por la mañana, y algunas tardes le toca quedarse para hacer alguna práctica o estudiar.

Las tardes que no tiene prácticas que realizar, las pasa en la biblioteca estudiando y preparando sus asignaturas. Cuando acaba de estudiar, vuelve a su piso, donde se prepara la cena, y si es necesario, ultima algún trabajo para el día siguiente.

En su piso disponen de conexión a internet de fibra óptica. Cada compañera (incluida Elisa) tiene teléfono móvil con conexión de datos, aunque en el piso usan la conexión wifi para no consumir los datos.

Los jueves es la noche universitaria, y por la tarde/noche suelen quedar con amigos/as o compañeros/as para salir a tomar algo. Les gusta ir a varios lugares de copas, para ir cambiando cada semana.

Una de las principales dificultades es racionalizar el dinero que sus padres le dan cada semana para cubrir gastos, con lo que a la hora de salir siempre mira cuánto puede gastar.

Ficha escenario

Es jueves, y Elisa sale de la biblioteca a eso de las 19:30. Ha quedado con sus amigos/as para ir a tomar una cerveza, cenar y después pasar un rato agradable en un bar de copas o en un pub.

Han quedado dentro de 30 minutos delante de la cafetería que hay al lado del piso de su amiga Elena. Una vez allí, decidirán donde ir a tomar una cerveza y relajarse un rato antes de decidir dónde ir a cenar. Tras reunirse todo el grupo, deciden quedarse en esa misma cafetería, ya que Juan y Silvia están cansados y no les apetece caminar buscando un local donde tomar la cerveza. Tras pasar un rato agradable, deciden cenar allí mismo. Terminada la cena, se marchan.

De camino al bar de copas, pasan por delante de un pub, a escasos 200 metros de donde estaban donde esa semana las cervezas están a 2x1. "Qué rabia!" piensa Elisa (al igual que los demás). Si lo hubieran sabido hubieran ido a ese pub y las cervezas les hubieran salido a mitad de precio.

2.2- Diseño

Analizadas las funcionalidades que ha de tener la app, procedemos al diseño conceptual de la misma.

2.2.1.- Árbol de navegación

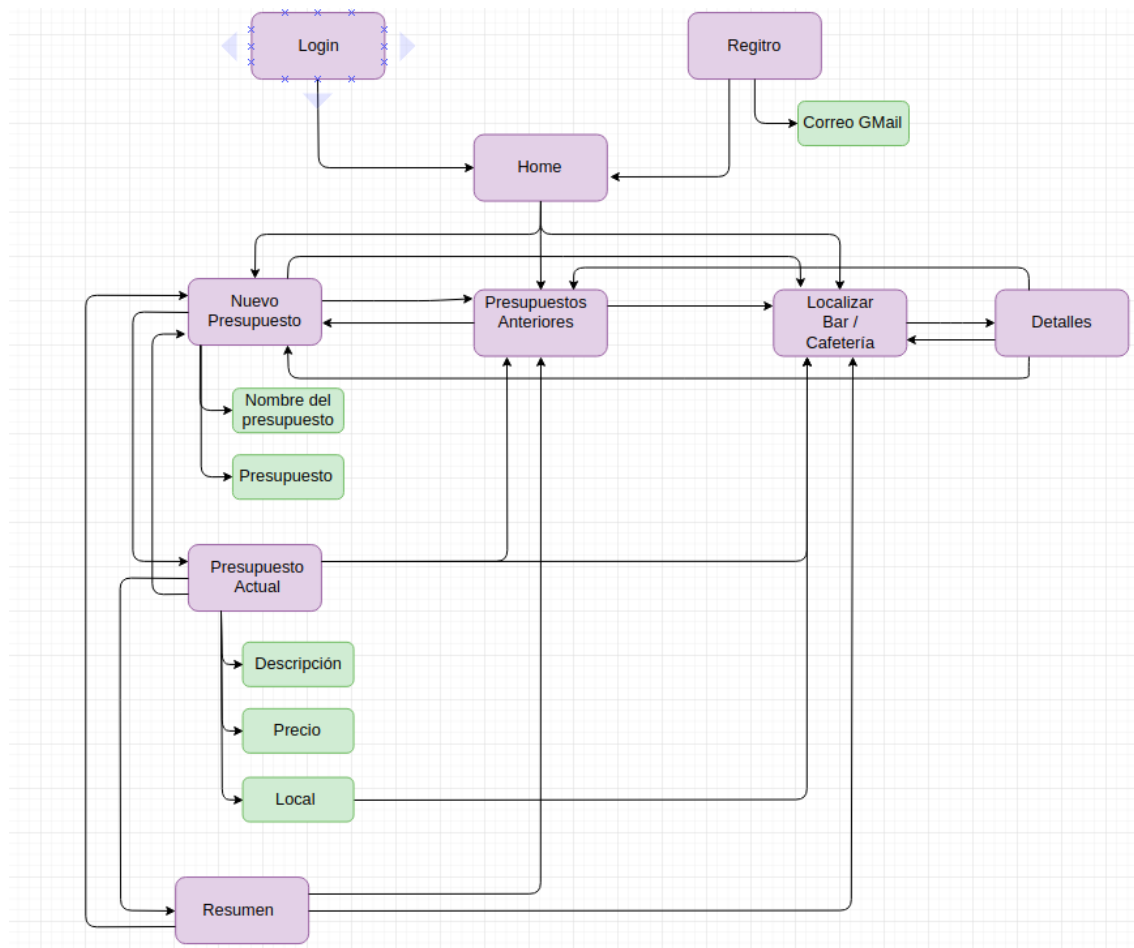


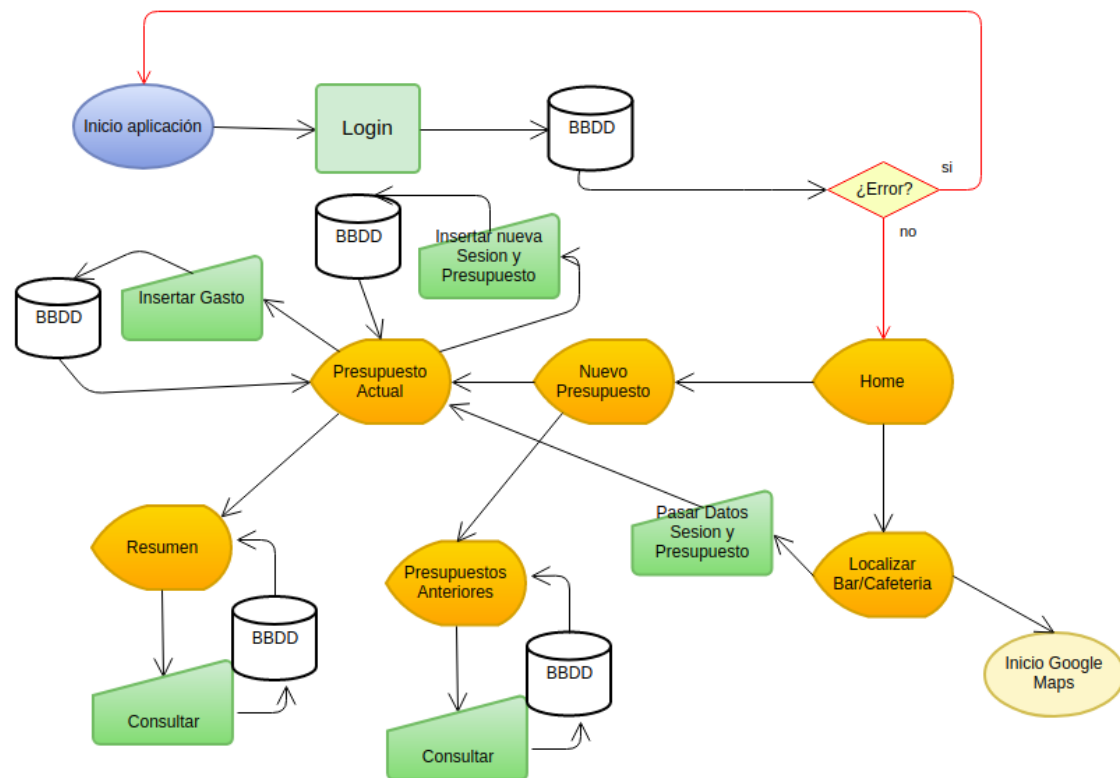
Figura 16: Árbol de Navegación

Cuando el usuario inicia la aplicación, esta muestra un menú con opciones para planificar una nueva sesión de gastos. Puede por tanto escoger un local a donde dirigirse, comenzar una nueva sesión de gastos o bien consultar sesiones anteriores.

La localización del usuario y de los locales que tiene cerca se realiza mediante sincronización de la aplicación con *Google Maps*, mediante la cual puede ver los detalles de los locales: dirección, teléfono, web.. y trazar una ruta para llegar al local elegido.

La gestión de usuarios se realiza mediante el uso de una base de datos online, y la de sesiones y gastos mediante una base de datos local.

2.2.2.- Diagrama de flujo



Leyenda



Figura 17: Diagrama de Flujo

El presente diagrama de flujo muestra la interacción del usuario con la aplicación.

Se creará una base de datos en *Firebase* para gestionar toda la información referente tanto a la gestión y registro de los usuarios.

Cuando se inicia la aplicación por primera vez se registra automáticamente al usuario en *Firebase*, mediante la cuenta de *Gmail* que lleva asociada a su teléfono.

Hecho esto, accede automáticamente a la pantalla principal, donde podrá elegir una opción de entre las tres del menú:

- “Nuevo Presupuesto”
- “Presupuestos anteriores”
- “Localizar Bar/Cafetería”

Nuevo Presupuesto

Accede a otra pantalla donde introducir un título de la sesión y el presupuesto a gastar.

Tras validar la información, accede a “**Presupuesto Actual**”, donde se valida la sesión y se crea la misma en la base de datos. Aparece un formulario donde introducir los gastos que se efectúan en dicha sesión, los cuales son: *Descripción del gasto, precio, local* donde se ha efectuado el mismo.

2.2.3.- Prototipo

A continuación pasamos al diseño del prototipo de la aplicación. Las principales interfaces de que constará nuestra aplicación son las que se muestran a continuación. Estas podrán sufrir algún tipo de variación durante el desarrollo del proyecto.

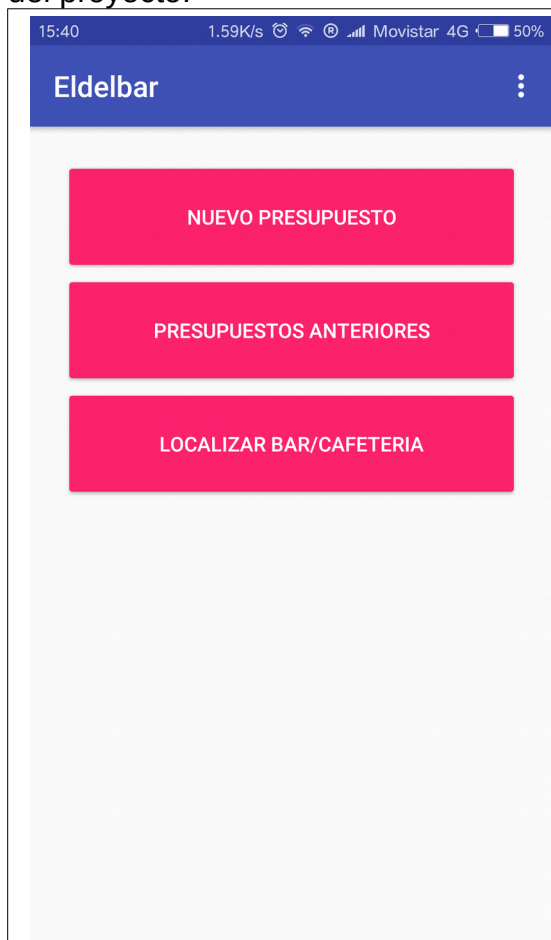


Figura 18: Main Activity

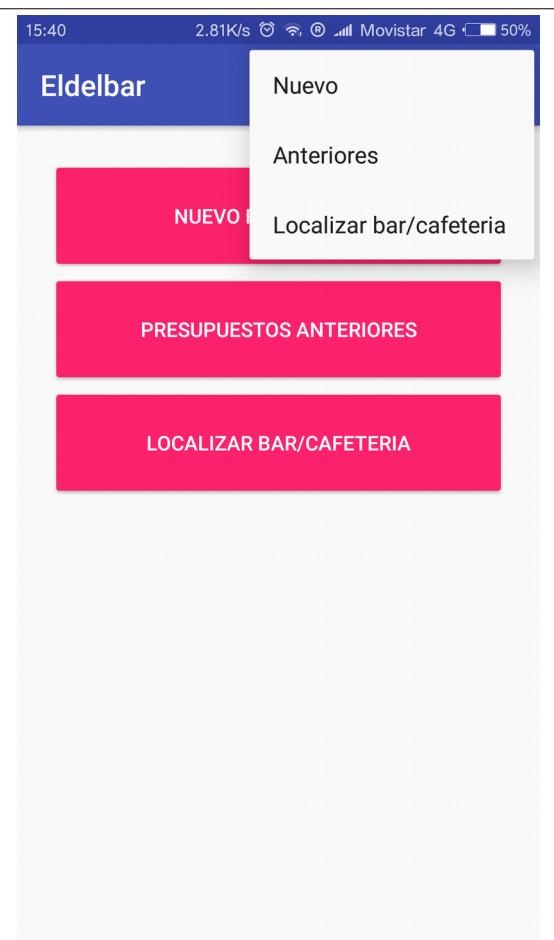


Figura 19: Menú contextual

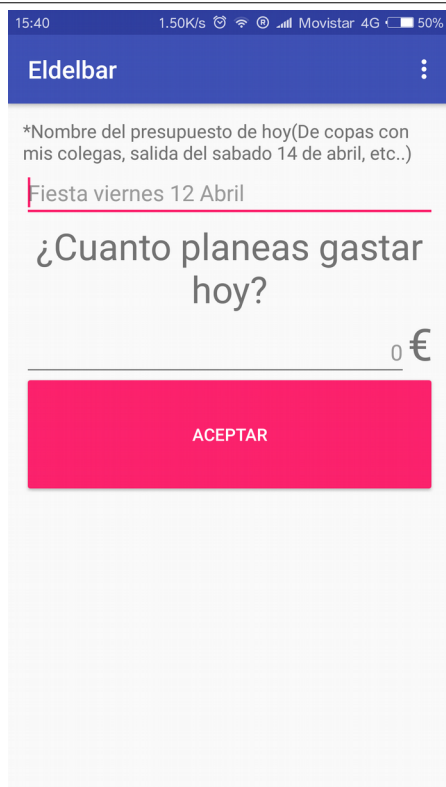


Figura 20: Nuevo presupuesto .

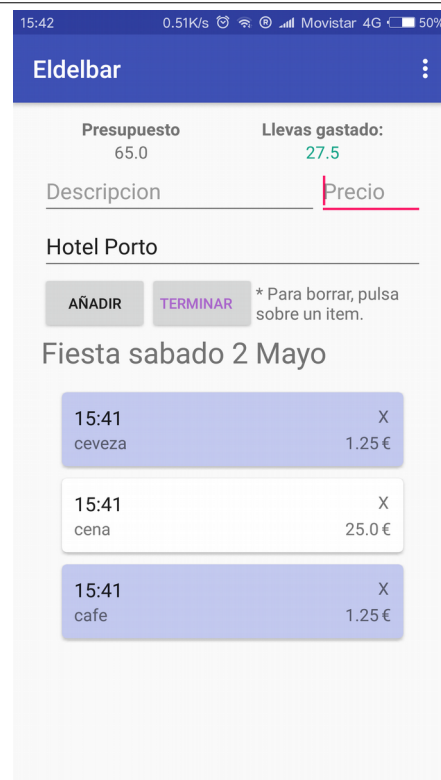


Figura 21: Introduccón de gastos.

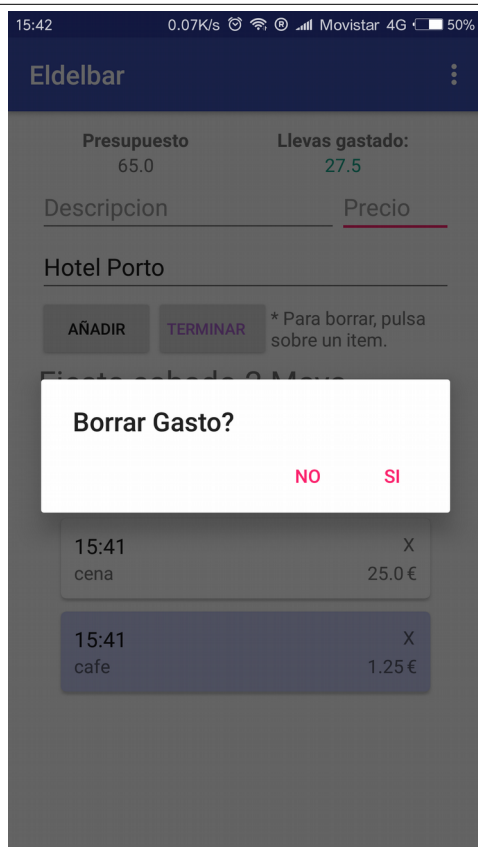
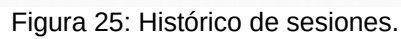
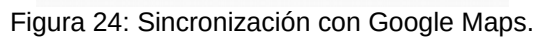
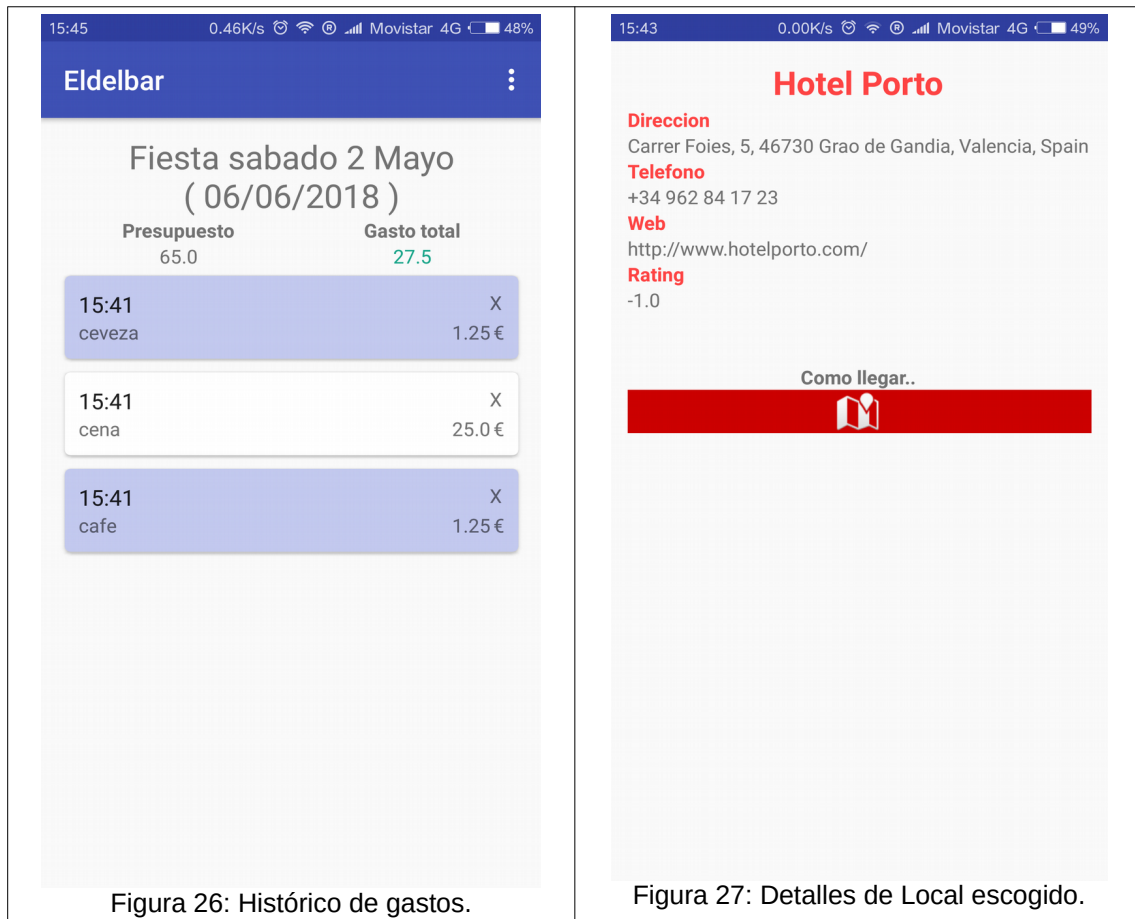


Figura 22: Borrado de gasto.



Figura 23: Resumen de la sesión.





Todas las interfaces de la aplicación se han diseñado para un dispositivo Android, consiguiendo un modelo simple, claro y limpio.

El objetivo es que el usuario sea capaz de utilizar la aplicación de una manera cómoda e intuitiva.

Para lograrlo, se ha optado por un modelo simple, donde se utiliza un menú lateral sobre todo para las funciones anteriormente descritas.

El resto de la pantalla se utiliza para representar toda la información de manera clara y limpia. Se ha optado para ello la representación mediante tablas y listas.

El objetivo es que un usuario de conocimientos básicos sea capaz de utilizar la aplicación con facilidad y comodidad.

Para ello se ha prestado especial atención a:

- Colores: Se han usado colores neutros y con suficiente contraste para dar mejor comodidad al usuario.
- Tipografía: Se ha usado tamaños de letra amplios para poder ser legibles por cualquier usuario.
- Continuidad: En todo momento se informa al usuario de donde se encuentra a través del menú de opciones.
- Uso de botones simples, intuitivos y de suficiente tamaño.
- Intuición: interfaces con gestos intuitivos para el usuario: tap, drag, pinch, spread...

Arquitectura entre pantallas

Para lograr una navegación lo más cómoda e intuitiva posible, se ha seguido la guía que ofrece Android para el desarrollo de interfaces de usuario:

["https://developer.android.com/guide/practices/ui_guidelines/index.html"](https://developer.android.com/guide/practices/ui_guidelines/index.html).

Se buscan las siguientes directrices que han de seguir el diseño de las interfaces:

- Claridad y simplicidad
- Concisión y consistencia
- Ligereza visual
- Navegación intuitiva

Se trata de seguir los principios básicos de la "ley de Fitts" 1 aplicada al diseño para dispositivos móviles.

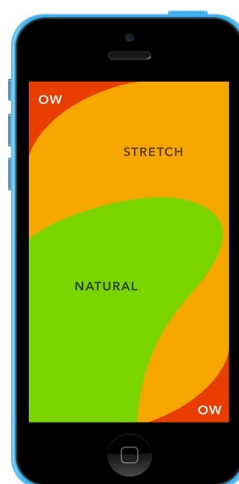


Figura 28: Ley de Fitts.

La zona verde es una zona de acceso natural para el pulgar del usuario, por tanto es donde debería aparecer los controles e información que el usuario necesita usar con mayor asiduidad.

La zona naranja es una zona menos cómoda que la verde. La zona roja es de difícil acceso, y por tanto donde deberían situarse los controles que el usuario no deba tocar por error.

La interfaz gráfica se ha diseñado de esta forma para poder tener una jerarquía entre todas las interfaces del prototipo. De esta manera, los listados son de fácil acceso y las opciones de menú y las acciones sobre las interfaces requieren total atención por parte del usuario.

Con objeto de evitar navegación entre interfaces y disminuir al máximo el número de interacciones de usuario, se ha decidido el diseño u uso de navegación del tipo “experience-driven navigation”.

Figura 29: Experience-Driven Navigation

2.2.4.- Diagrama de casos de uso

A continuación se presenta el diagrama de casos de uso para mostrar cómo debe interactuar la aplicación con los usuarios.

Usuario

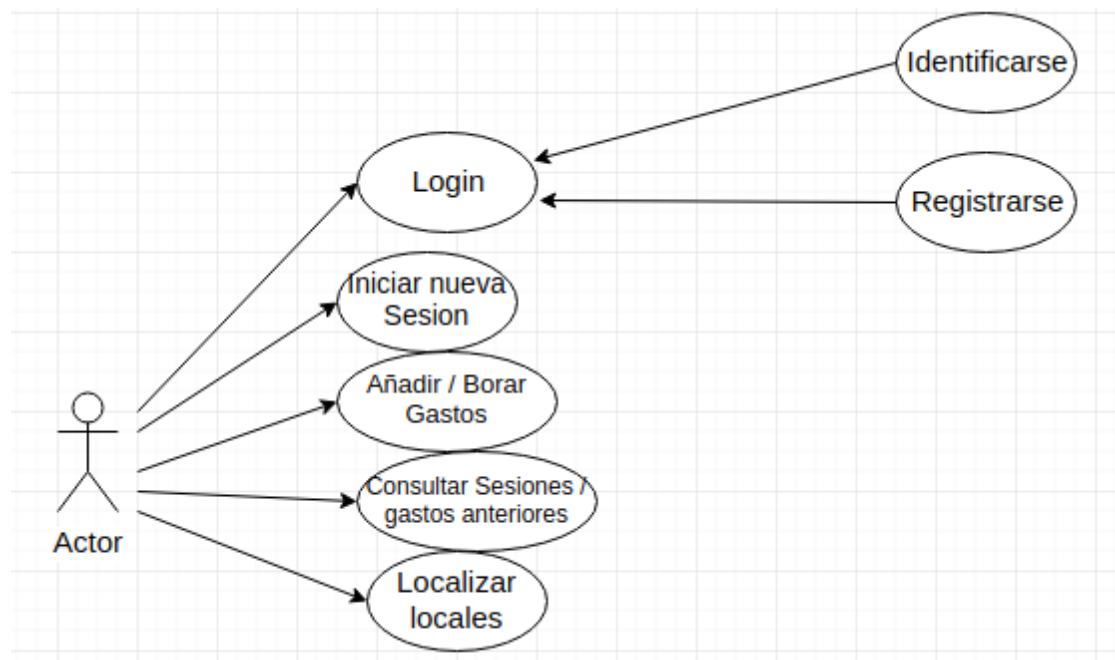


Figura 30: Casos de uso.

El usuario, además de consultar la información, si es necesario, la modifica e incluso añade nueva.

Como se puede observar, solamente hay un actor: el usuario, el cual se instala la aplicación en su *smartphone* e interactúa directamente con ella. Al iniciar la aplicación, el usuario dispone de un menú de opciones, el cual mantiene en todas las pantallas en forma de menú contextual.

A continuación se procede a la explicación de los casos de uso:

Caso de uso “Login”

Al iniciar la aplicación, Por medio de *Firebase* y *Google Smart Lock*, el usuario se registra/identifica de forma automática.

Caso de uso “Iniciar sesión nueva”

El usuario ingresa un título para la nueva sesión y un presupuesto inicial de gasto.

Caso de uso “Añadir / Borrar gastos”

Tras iniciar la nueva sesión, el usuario accede a una pantalla con un formulario donde introducir los datos de cada gasto:

- *Descripción*: descripción del gasto que realiza (Ej. cerveza, café, ..)
- *Local*: Nombre del local donde realiza el gasto. El nombre lo podrá introducir de forma manual, o elegirlo de la aplicación Google Maps de forma automática.
- *Precio*: Precio del gasto o consumo.

En todo momento se mantendrán visibles tanto el presupuesto inicial, como la acumulación de gastos que lleva en la sesión, a fin que el usuario pueda llevar el control en todo momento sobre los mismos.

Cada gasto será introducido en la base de datos y se verá reflejado en la parte inferior, con la posibilidad de borrarlo si fuera necesario.

Al finalizar la sesión, se le mostrará un resumen de los gastos realizados.

Caso de uso “Consultar sesiones / gastos anteriores”

El usuario puede consultar las sesiones anteriores, y para cada sesión, los gastos realizados.

Caso de uso “Localizar locales”

La aplicación geolocaliza el dispositivo del usuario y le muestra los locales que tiene cerca. El usuario puede seleccionar uno para ver los detalles tales como teléfono, web, ranking (según Google Maps), dirección.. y mediante sincronización con Google Maps trazar una ruta para llegar, o bien comenzar una nueva sesión.

2.2.5.- Arquitectura

En este punto, se explicará el funcionamiento de la aplicación. También se detallan las clases que la forman, y cómo están relacionadas. Para ello se ha optado por la utilización de un diagrama MVC.

Diagrama de clases y patrón MVC de las listas

El siguiente diagrama representa las listas existentes en la aplicación.

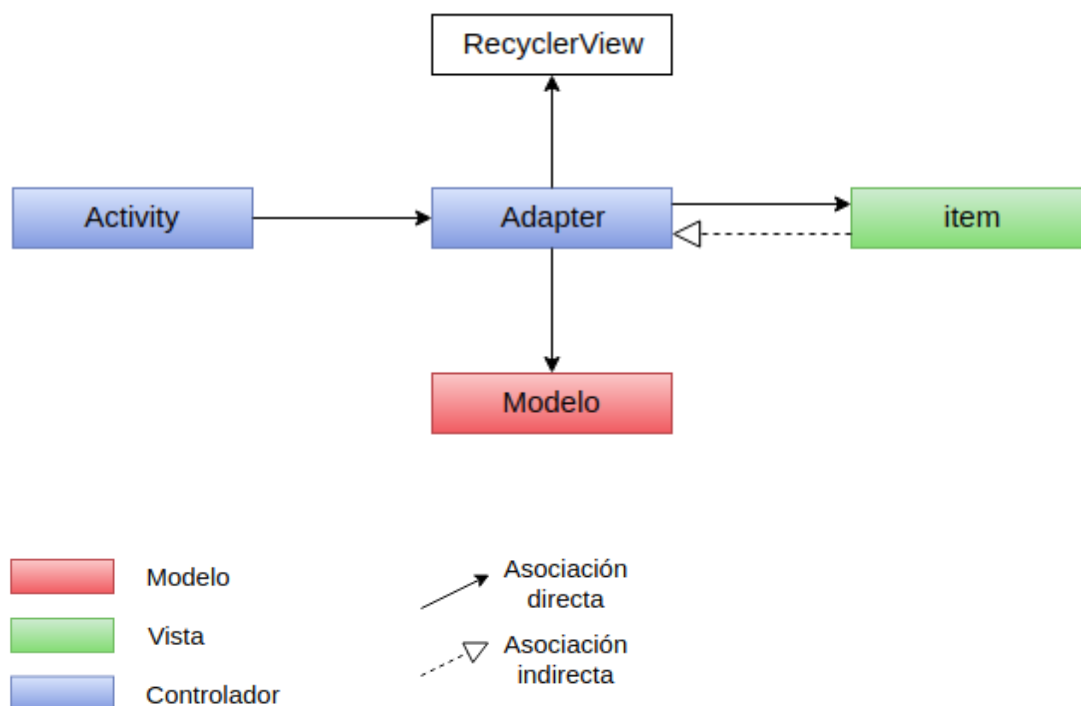


Figura 31: Diagrama MVC

- En azul, las *activities* son del tipo AppCompatActivity.
- En verde, los *layouts*, los cuales contienen la definición de cómo son y de la estructura que tiene la vista a la que corresponde.
- En naranja, las clases pertenecientes al modelo. Contienen la información necesaria para la correcta gestión de base de datos y las vistas.

Diagrama de clases y MVC general

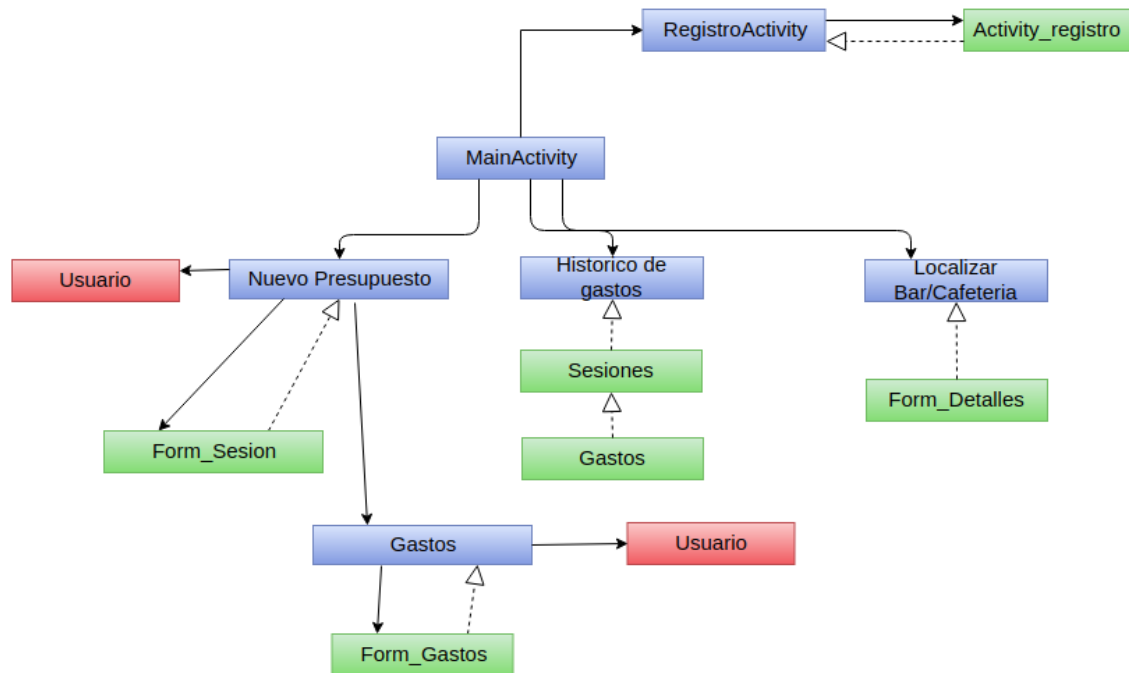


Figura 32: Diagrama MVC General

2.2.6.- Base de datos.

La aplicación se conectará a una base de datos local *sqlite* creada, dado que lo que se pretende es que todos los usuarios tengan el control de sus gastos de manera local.

Dicha base de datos está preparada para posibles actualizaciones y mejoras, dado que a parte de los datos que el usuario visualiza, recoge otros, tales como la localización (coordenadas) del local donde realiza los gastos, para posible sincronización con una base de datos remota que gestione datos y opiniones sobre dicho local y que éstas puedan ser compartidas en redes sociales.

Por tanto se creará una base de datos local, la cual estará formada por las siguientes tablas.

La tabla “Sesiones” almacenará los datos referentes a la sesión actual:

- clave primaria: formada por la presupuesto+ fecha+hora
- fecha y hora de inicio de la sesión.
- presupuesto propuesto para la sesión.
- título para identificar la sesión de forma más intuitiva para el usuario.

La tabla “gastos”, donde se almacenan los gastos que ha habido en cada sesión:

- Clave primaria.
- Fecha y hora del gasto.
- Descripción: Descripción del tipo de gasto realizado.
- Precio.
- Dirección del local donde se ha realizado el gasto.
- Localización: Coordenadas de *Google Maps* para la localización del local en futuras mejoras de la aplicación.

La base de datos se ha diseñado y planteado de tal forma que en una sesión se pueden realizar muchos gastos.

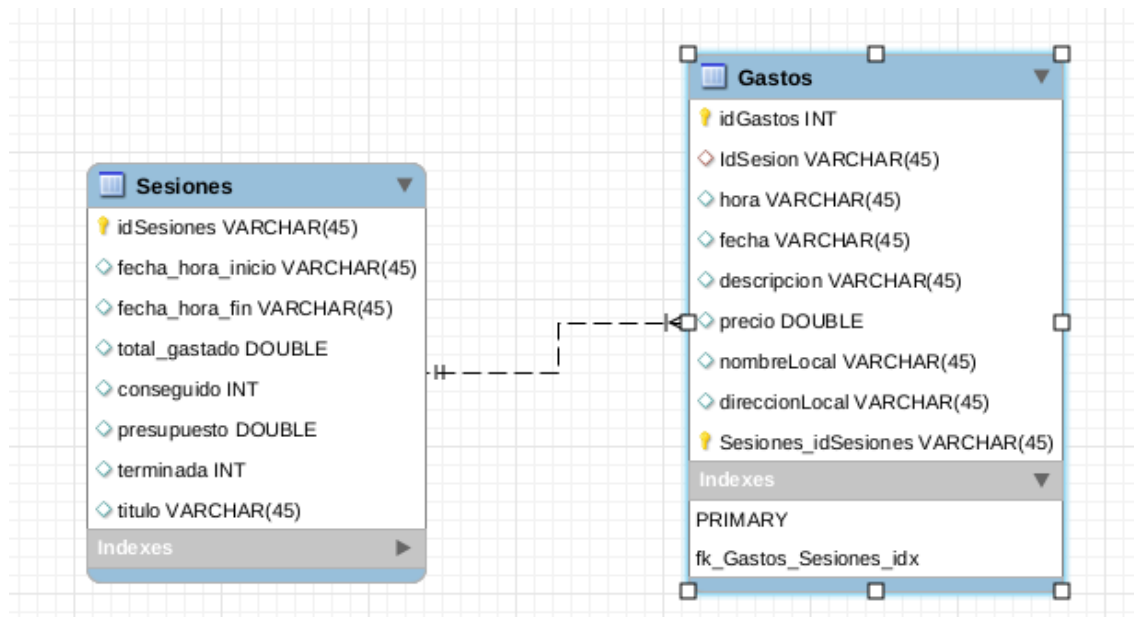


Figura 33: Base de Datos

3.- Desarrollo

En el presente capítulo se explicará cómo se ha realizado el proceso de desarrollo del trabajo, destacando el código de desarrollo de las partes más importantes del mismo.

Esta aplicación se ha desarrollado en Android debido a que es el sistema operativo que tiene una mayor cuota de usuarios, tal y como se ha mencionado en capítulos anteriores. Teniendo eso en cuenta y con el fin de poder acceder al mayor número de usuarios posible, se ha decidido implementar la aplicación para una versión mínima 19 de Android.

3.1.- Firebase



Se trata de una aplicación de bases de datos en tiempo real, que además proporciona herramientas para la creación, uso y mantenimiento de bases de datos sincronizables en dispositivos móviles.

En esta aplicación, se ha hecho uso de *Firebase* para la gestión de los usuarios: registro y *login*, con el fin de tener un control sobre los mismos y poder llevar a cabo futuras aplicaciones.

3.1.1.- Autenticación y registro de usuarios

Para facilitar el registro de los usuarios, se ha decidido implementar la API de gestión de usuarios que proporciona *Firebase*. Esta API da opción, entre otros, a poderse identificar mediante las credenciales que se poseen con una cuenta en *Gmail*, o mediante la cuenta de *Facebook*.

Para poder hacer uso de estas funcionalidades que proporciona *Firebase* se deben habilitar los proveedores de inicio de sesión correspondientes: *GMail* y *Facebook*.








 Teléfono	Inhabilitado
 Google	Habilitada
 Play Juegos	Inhabilitado
 Facebook	Habilitada
 Twitter	Inhabilitado
 GitHub	Inhabilitado
 Anónimo	Inhabilitado

Figura 34: Autenticación de usuarios en Firebase

3.2.- Base de datos SQLite

SQLite es una base de datos que permite almacenar datos de forma local en dispositivos poco potentes, como son smartphones, PDAs y Tablets.

Implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte, desde las consultas más básicas, hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

En cuanto a la estructura de la base de datos de la aplicación:

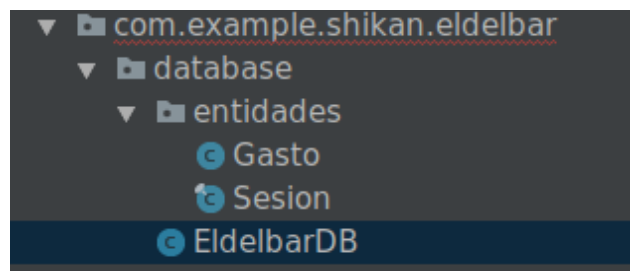


Figura 35: Estructura de la Base de Datos

Tal y como muestra la figura 32, la base de datos de la aplicación se estructura en dos entidades:

- Sesión: que contiene
 - Id_Sesión: clave primaria
 - fecha_hora_inicio: fecha y hora de inicio de la sesión
 - fecha_hora_fin: fecha y hora de fin de la sesión
 - total_gastado: El gasto total en euros de la sesión
 - presupuesto: la cantidad que el usuario había fijado como objetivo.
 - Conseguido: Si el usuario ha conseguido el objetivo o no
- Gasto: que contiene:
 - Id_sesion: clave foránea que apunta a sesión
 - hora: hora en que se ha producido el gasto o consumición.
 - Fecha: fecha en que se ha producido el gasto o consumición.
 - Descripción: descripción de la consumición.
 - Precio: el precio de la consumición.

La base de datos se denomina EldelbarDB. Esta extiende la clase **SQLiteOpenHelper**, y almacena toda la estructura de la base de datos, tal y como podemos ver a continuación:

```
//Campos tabla Gastos
public static final String TABLA_GASTOS = "Gastos";
public static final String GASTOS_ID_GASTO = "id_gasto";
public static final String GASTOS_ID_SESION_FK = "id_sesion";
public static final String GASTOS_HORA = "hora";
public static final String GASTOS_FECHA = "fecha";
public static final String GASTOS_DESCRIPCION = "descripcion";
public static final String GASTOS_PRECIO = "precio";
public static final String GASTOS_NOMBRE_LOCAL = "nombre_local";
public static final String GASTOS_DIRECCION_LOCAL = "direccion_local";
public static final String GASTOS_LOCALIZACION_LOCAL = "localizacion_local";

public static final String CREAR_TABLA_GASTOS = "CREATE TABLE "+
    TABLA_GASTOS+" (" +
    GASTOS_ID_GASTO+" INTEGER PRIMARY KEY AUTOINCREMENT,"+
    GASTOS_FECHA+" TEXT,"+
    GASTOS_HORA+" TEXT,"+
    GASTOS_PRECIO+" DOUBLE,"+
    GASTOS_DESCRIPCION+" TEXT,"+
    GASTOS_ID_SESION_FK+" TEXT,"+
    GASTOS_NOMBRE_LOCAL+" TEXT,"+
    GASTOS_DIRECCION_LOCAL+" TEXT,"+
    GASTOS_LOCALIZACION_LOCAL+" TEXT )";
```

Figura 36: Código para crear la tabla Gastos

```
//Campos tabla Sesion
public static final String TABLA_SESION = "Sesion";
public static final String SESION_ID_SESION_PK = "id_sesion";
public static final String SESION PRESUPUESTO = "presupuesto";
public static final String SESION_TOTAL_GASTADO = "total gastado";
public static final String SESION CONSEGUIDO = "conseguido";
public static final String SESION_FECHA_HORA_INICIO = "fecha_hora_inicio";
public static final String SESION_FECHA_HORA_FIN = "fecha_hora_fin";
public static final String SESION_TERMINADA = "terminada";
public static final String SESION_TITULO = "titulo";

public static final String CREAR_TABLA_SESION = "CREATE TABLE "+
    TABLA_SESION+" (" +SESION_ID_SESION_PK+" TEXT PRIMARY KEY,"+
    SESION PRESUPUESTO+" DOUBLE,"+
    SESION_TOTAL_GASTADO+" DOUBLE,"+
    SESION CONSEGUIDO+" INTEGER,"+
    SESION_FECHA_HORA_INICIO+" TEXT,"+
    SESION_FECHA_HORA_FIN+" TEXT,"+
    SESION_TERMINADA+" INTEGER,"+
    SESION_TITULO+" TEXT )";
```

Figura 37: Código para crear la tabla Sesión.

Mediante el objeto *db* de tipo *EldelbarDB* accedemos a los métodos *AgreGargasto* y *AgregarSesion*, mediante los cuales se introducen los datos en la Base de Datos.

```
db.AgregarGasto(Id_sesion,fecha_gasto,hora_gasto,precio,descripcion,nombreLocal,direccionLocal,localizacion);
```

Figura 38: Llamada al método *AgregarGasto* de la clase *EldelbarBD*

```
//Agregamos datos a la tabla Gastos
public void AgregarGasto(String Id Sesion, String fecha, String hora, double precio, String descripcion){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(CAMPO_ID_SESION_FK, Id_Sesion);
    values.put(CAMPO_FECHA, fecha);
    values.put(CAMPO_HORA, hora);
    values.put(CAMPO_PRECIO, precio);
    values.put(CAMPO_DESCRIPCION, descripcion);

    db.insert(TABLA_GASTOS, nullColumnHack: null, values);

    db.close();
}
```

Figura 39: Método *Agregargasto* de la clase *EldelbarBD*

3.3.- Geolocalización

La geolocalización es una función mediante la cual el usuario puede localizar locales cercanos a su ubicación.

Dado que lo que se pretende es localizar comercios deo tipo cafetería o bar, se ha optado por la utilización de la API que proporciona Google Places. Esta obtiene mediante el objeto de tipo *PlaceDetectionClient* la localización del dispositivo del usuario.

Mediante el objeto *GeodataClient* crea un mapa donde se muestran todos los comercios que el usuario tiene cerca, mostrando un nivel de zoom determinado. En este caso, se realiza un zoom de 15, el cual tiene el suficiente detalle para mostrar los locales que hay cerca del usuario.

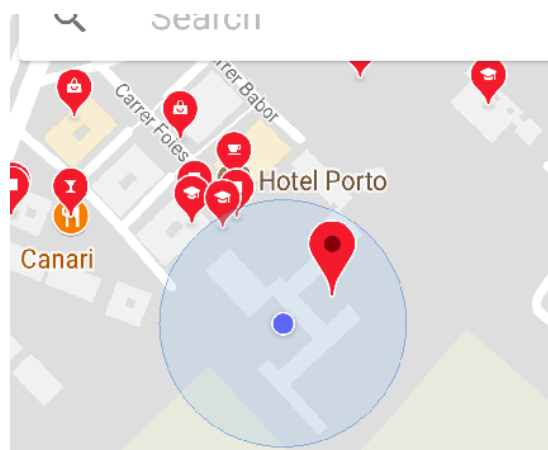


Figura 40: Detalle de Zoom.

```
//Construimos un geodataClient
mGeoDataClient = Places.getGeoDataClient( activity: this);

// Construct a PlaceDetectionClient.
mPlaceDetectionClient = Places.getPlaceDetectionClient( activity: this);

//Conectamos ala API de Google Places
googleApiClient = new GoogleApiClient
    .Builder( context: this)
    .addApi(Places.GEO_DATA_API)
    .addApi(Places.PLACE_DETECTION_API)
    .enableAutoManage( fragmentActivity: this, onConnectionFailedListener: this)
    .build();

googleApiClient.connect();
```

Figura 41: Conexión a la API de Google Places

Mediante el objeto de tipo Place, se obtienen todos los datos del lugar seleccionado, tales como nombre, tipo de local, ranking, dirección, teléfono, localización, web, etc..

Estos datos se se almacenan en variables para su posterior uso.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (requestCode == PLACE_PICKER_REQUEST && resultCode == RESULT_OK) {
        Place place = PlacePicker.getPlace( context: this, data);
        final CharSequence name = place.getName();
        final CharSequence direcc = place.getAddress();
        final CharSequence tfn = place.getPhoneNumber();
        final Float rating = place.getRating();
        final Uri web = place.getWebsiteUri();
        final LatLng ubicacion = place.getLatLng();

        final Uri localizacion = Uri.parse("geo:"+ubicacion.toString()+"?q="+Uri.encode(name.toString()));
    }
}
```

Figura 42: Obtención de datos de Places.

```
Place place = PlacePicker.getPlace( context: this, data);
final CharSequence name = place.getName();
```

Figura 43: Objeto Place

4.- Pruebas

Durante este capítulo se exponen las diferentes pruebas a las que ha sido sometida la aplicación.

Se consideran satisfactorias si cumplen con los requisitos previstos, en cambio si no los cumple, se consideran no satisfactorias.

Se trata de una de las etapas más importantes dentro del desarrollo de una aplicación y es donde se muestra la parte de la calidad de ésta.

4.1.- Pruebas de la interfaz.

Estas pruebas validan todo lo referente al aspecto visual de la aplicación: se han hecho pruebas de la interfaz visual con el prototipo establecido en el apartado [2.2.3 Prototipo](#).

Los usuarios estaban cómodos con los colores claros que presenta el aspecto de la interfaz, además de con los botones grandes del menú principal, por lo que se ha decidido mantener el aspecto propuesto en el prototipo.

4.2.- Pruebas de navegación

Estas pruebas permiten comprobar la accesibilidad de los diferentes menús y la navegación por las diferentes vistas de la aplicación.

Para comprobar que la navegación por los menús funciona correctamente se ha navegado por cada uno de ellos de todas las formas posibles, obteniendo un resultado satisfactorio excepto al intentar crear una nueva sesión desde la pantalla de localización de bares/caféterías, donde fallaba el paso de datos.

Esto se daba porque al intentar crear dicha sesión, los datos se rellenaban en un formulario mostrado en un *Alert Dialog*, y se intentaban recoger de un formulario de la propia *Activity*.

5.- Viabilidad económica

En este punto se realizará un análisis sobre la inversión que habrá que realizar en el proyecto, la forma con la que se intentará recuperar dicha inversión y también la forma de obtener beneficios.

Dado que actualmente soy yo quien ha realizado el proyecto, se va a suponer que soy el único componente de la empresa.

Por tanto, el modelo de negocio se basa en que todos los costes de la inversión, hardware, software y costes de marketing, los asume la propia empresa sin financiación externa.

5.1.- Inversión Inicial

Mano de obra.

Los gastos de desarrollo de la aplicación tratan dos elementos:

- Precio/hora del programador.
- Número de horas para realizar la aplicación.

Basándonos en el *Convenio de Oficinas y Despachos de la Provincia de Valencia*:

Actualmente el salario bruto medio de un ingeniero técnico informático es de 18.500€/año.

Sumando a esta cantidad el 30% que la empresa paga a la Seguridad Social (5.500€/año), el coste anual del trabajador para la empresa es: $24.000 + 7.200 = 31.200$ €.

El número de horas por trabajador al año según este convenio son 1.742. Por tanto:

$$31.200€ / 1.742h = 17,85 €/h.$$

La duración del proyecto es de 251 horas. Por lo tanto, el coste de la mano de obra será:

Mano de obra = 251h x 14 €/hora (salario) = 3.514€

5.2.- Hardware

Para la realización de este proyecto ha utilizado un ordenador Asus 1000 € y un dispositivo Android valorado en 199 €.

Considerando una vida útil para el ordenador de 5 años y de 3 años para el dispositivo Android:

- Amortización anual del ordenador $1000/5 = 200$ €
- Amortización anual dispositivo Android $199/3 = 66,3$ €

• Con una duración del proyecto de 15 semanas aproximadamente y un uso para el ordenador de un 40% del tiempo y un 10% del tiempo para el teléfono:

• Gasto amortización del ordenador: $(200 \text{ €} \times 15 \text{ semanas} \times 0,4 \text{ (porcentaje de uso)}) / 52 \text{ semanas del año} = 23 \text{ €}$

• Gasto amortización del dispositivo: $(66,3 \text{ €} \times 15 \text{ semanas} \times 0,1 \text{ (porcentaje de uso)}) / 52 \text{ semanas del año} = 1,91 \text{ €}$

En resumen:

Gasto total amortización hardware = 25€

5.3.- Software

Todo el Software utilizado es libre, por tanto no hay gasto de software.

5.4.- Marketing

25\$ la licencia Google Play → 25€

5.5.- Otros

Luz (40€), agua(6€), teléfono(8€), internet(35€): $40 + 6 + 8 + 35 = 89\text{€}$

5.6.- Resumen de costes

Partida	Costes
Mano de obra	3514
Hardware	25
Software	0
Otros gastos	89
Costes de Marketing	25
TOTAL	3653€

5.7.- Recuperación de la inversión.

Para poder distribuir la aplicación en la tienda virtual de Google, *Google Play*, se necesita la licencia de desarrollador correspondiente, la cual tiene duración de 1 año, y después hay que renovarla.

El propietario recibe una remuneración del 85% del precio de la aplicación.

Teniendo esto en cuenta, planteamos un posible escenario:

Suscripciones en un año	Precio	Ingresos anuales	15% para Google	Ingresos 85%
100	0,99	1200	180	1020
500	0,99	6000	900	5100
1.000	0,99	12.000	1.800	10.200

Por tanto, para recuperar la inversión, de 3.653€, necesitaríamos al menos 457 descargas mensuales para cubrirlo en un año, o 230 para cubrirlo en 2 años.

6.- Manual de instalación


En el siguiente punto se procederá a explicar cómo instalar la aplicación en un dispositivo con sistema operativo Android.

Para poder instalar la aplicación 'Eldelbar' en un dispositivo, se puede realizar de varias formas:

1. Desde Android Studio.
 1. Importando la aplicación a un emulador AVD.
 2. Importando la aplicación a un dispositivo móvil conectado.
2. Grabando el archivo .apk en un dispositivo móvil.

1.- Instalando desde Android Studio

Importando la aplicación a un emulador AVD

Abierto Android Studio y habiendo abierto el proyecto, se presiona el icono Run 'app' .

Hecho esto, aparece la ventana para escoger emulador.

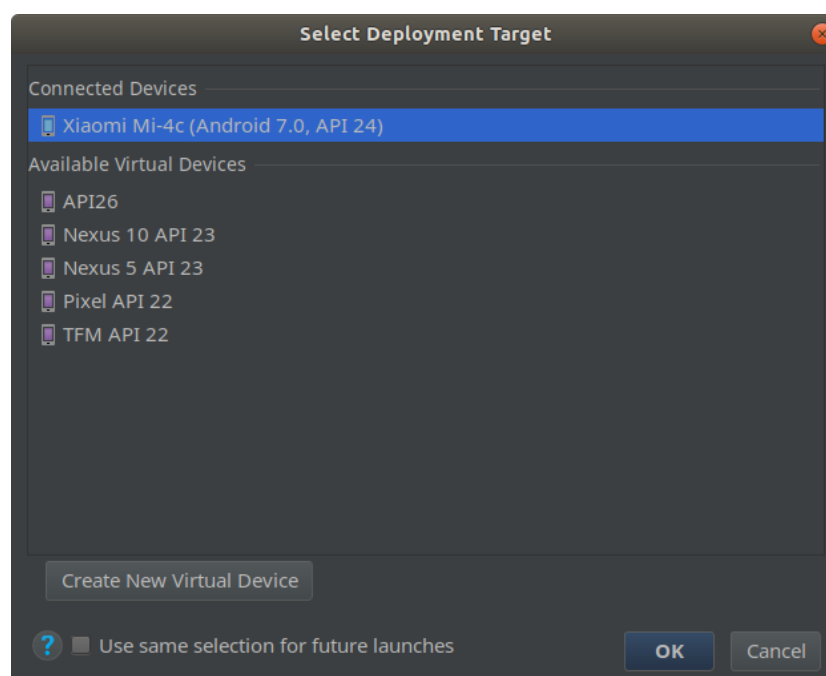


Figura 44: Elección del dispositivo

Seleccionamos el emulador deseado y pulsamos 'ok'.

En caso de haber elegido importar la aplicación a un dispositivo, hay que asegurarse de:

- Habilitar opciones para programadores y depuración.
- Opción *Ajustes* → *Seguridad* → *origenes desconocidos* esté activada.

7.- Manual de usuario

En el apartado [2.2.3 Prototipo](#) se pueden observar capturas de pantalla de la aplicación.

Al iniciar la aplicación, esta se registrará de forma automática en *Firebase* a través de la cuenta de correo *Gmail* que lleva asociada el dispositivo.

Tras aceptar el registro, se accederá a la pantalla principal con las opciones disponibles, bien en los botones del centro de la pantalla, o bien en el menú contextual situado en la parte superior.

Hecho esto tenemos 2 opciones:

1. Buscar car/cafetería cercanos para ir a tomar algo.
2. Ya sabemos a qué local iremos, con lo que iniciamos un nuevo presupuesto.

Opción 1: Buscar car/cafetería cercanos para ir a tomar algo.

Seleccionamos *LOCALIZAR BAR/CAFETERIA* y se ejecuta una pantalla donde aparece nuestra ubicación (la del dispositivo), y los locales de ocio que hay cerca. Si la geolocalización tardase, se puede acelerar presionando el siguiente botón, situado en la parte inferior derecha de la pantalla.



Tras escoger un local, se abre un diálogo para escoger entre 3 opciones:

‘Añadir local a gasto’ : Se asocia el local al gasto que se va a realizar. En caso de no haber creado una sesión, se abrirá un diálogo para introducir los datos necesarios para iniciar una nueva sesión de gastos.

‘Ver detalles del local’: Se mostrarán los detalles del local seleccionado, tales como nombre, dirección, teléfono.. y su localización por medio de un botón para trazar una ruta hasta el mismo para poder llegar.

‘Cómo llegar’: Se trazará una ruta desde la localización del dispositivo hasta el local seleccionado, todo ello a través de *Google Maps*.

Opción 2: Ya sabemos a qué local iremos, con lo que iniciamos un nuevo presupuesto.

Pasamos a la pantalla de inicio de sesión, donde se nos pedirá los datos necesarios para iniciar una nueva sesión de gastos: Nombre de la sesión y presupuesto a gastar.

En ambas opciones es necesario rellenar todos los datos.

Entramos entonces en la pantalla de gastos, donde podemos introducir cada uno de los gastos que realicemos, y asociarlo al local/es en cuestión rellenando los campos y presionando el botón *AÑADIR*.

Al presionar el campo *Local*, se procede a la geolocalización del dispositivo y a mostrar los locales que tiene cerca, permitiendo seleccionar el local deseado.

Cada gasto que se vaya añadiendo irá apareciendo en la parte inferior, permitiendo borrarlos si es necesario.

En la parte superior aparece en todo momento el presupuesto inicial y el gasto que se lleva acumulado, para poder llevar el mejor control posible del mismo. En caso de mantenerse dentro del presupuesto, la cifra del gasto acumulado se mantendrá en verde. En caso contrario aparecerá en rojo.

Finalizada la sesión, se presiona el botón *TERMINAR*, mostrándose un resumen de los gastos realizados en la sesión.

SESIONES ANTERIORES

Se muestra un listado con las sesiones anteriores; para cada sesión, al hacer click aparecen los gastos de dicha sesión, incluyendo el presupuesto y el gasto realizado.

LOCALIZAR BAR/CAFETERIA

Tras seleccionar el local, hay varias opciones: añadirlo al gasto, explicado anteriormente, ver detalles, con lo que nos aparece una tabla con los datos sobre el local: dirección, tfn, web.. y un botón en la parte inferior que nos muestra cómo llegar; al presionarlo por medio del *Google Maps*, aparece un mapa con una ruta trazada para llegar al local.

8.- Revisión de la planificación

Realizar el Trabajo de Final de Máster no es una tarea fácil, máxime cuando se compatibiliza trabajo y estudios. Requiere de gran dedicación, lucidez de ideas, ganas e ilusión.

La principal dificultad a la hora de realizarlo ha sido el saber qué hacer, cómo hacerlo y el tiempo a dedicar para cada una de las fases en las que se compone el presente proyecto.

Al iniciar la planificación, se fijaron unos objetivos iniciales y una temporización para distribuir las distintas fases de creación del proyecto, las cuales coincidían con las fechas y contenidos de las PEC propuestas.

Dos fueron los problemas principales:

Falta de experiencia en el desarrollo de aplicaciones:

En la propuesta inicial, se propusieron unos objetivos funcionales que por falta de experiencia, hubo que hacer pruebas antes de implementarlos. Ello llevó a invertir un tiempo que ocasionó un retraso en el progreso del proyecto, ya que por un lado se invirtió tiempo en probar, y por otro se llegó a la conclusión de que no se podía hacer, con lo que se optó por una variación en los objetivos.

Falta de horas que invertir al proyecto:

Tal y como se ha mencionado anteriormente, compaginar trabajo y realización del proyecto no es tarea fácil, ya que no permite invertir todas las horas que se consideran necesarias, precisamente porque no se dispone de ellas.

La única manera de recuperar las horas 'a priori mal invertidas' al inicio del proyecto fue mediante una serie de retrasos en los plazos de entrega.

Al final de todo, dichos cambios y esfuerzo dieron su fruto plasmado en el presente Trabajo de Final de Máster.

9.- Conclusiones

Concluído el proyecto se pueden sacar varias conclusiones respecto a la planificación y realización del mismo, fruto de la experiencia adquirida en estas fases de planificación y desarrollo.

Inicialmente se planteó para una duración de 251 horas, sin embargo la realidad es que se han invertido **más de 400 horas**, debido principalmente a la inexperiencia en el desarrollo de aplicaciones.

En cuanto al proyecto, por un lado ha sido una muy buena experiencia, ya que se trata de la primera aplicación que he realizado de principio a fin. Han sido muchas las cosas aprendidas, como el diseño de interfaces, diseño, creación y manejo de bases de datos locales, y remotas, por ejemplo.

Ha habido también experiencias no tan buenas, como que me ha sido imposible seguir la temporización establecida, debido por un lado a la inexperiencia en cuanto al entorno Android, ya que al principio me veía muy perdido.

Todo eso ha hecho que se modificasen objetivos inicialmente establecidos para poder llevar a buen puerto este proyecto, lo que no hubiera sido posible sin la gran colaboración y comprensión del tutor del proyecto.

En lo personal:

Realizar un proyecto de fin de máster no es tarea fácil, y hay muchos factores a tener en cuenta: la realidad personal del alumno, el entorno, las dificultad del proyecto, experiencia y conocimientos previos.

En este caso, debido a diversas dificultades, por un lado se está compaginando un trabajo a jornada completa con el máster, del cual se están cursando 2 asignaturas y el proyecto, son factores determinantes a la hora de una adecuada viabilidad del mismo.

Aún así, y teniendo todo en cuenta, creo que lo positivo supera con creces a lo negativo.

10.- Glosario

Activities: Componente de la aplicación Android que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción.

.apk: Instalable de una aplicación para Android.

API: *Application Programming Interface*. Es un conjunto de funciones y procedimientos ofrecidos por diferentes plataformas para ser usados por otros softwares.

EDP: Estructura de descomposición del proyecto.

Fragments: Representación de una parte de la interfaz de usuario en una *Activity*.

FK: *Foreign Key*.

ID: Identificador.

iOS: Iphone OS (sistema operativo de Apple).

Layouts: archivos de Android que sirven para especificar la interfaz visual que ve el usuario en una aplicación.

Listener: Objeto que se utiliza para detectar eventos o acciones.

MVC: Modelo-Vista-Controlador. Es un patrón de diseño de software que se utiliza para la implementación de sistemas donde se requiera el uso de interfaces de usuario. Sirve para separar de una manera clara las tres partes más importantes del software: los modelos o capa de datos, las vistas que permitirán la visualización de las interfaces y los controladores que contiene el código necesario para responder a las acciones o eventos que se solicitan en una aplicación.

PK: *Primary Key*.

TFM: Trabajo final de máster.

UML: *Unified Modeling Language*. Es un lenguaje de modelado unificado y visual que sirve para la creación de diagramas que presenten la arquitectura, el diseño y la implementación de diferentes softwares.

.zip: Extensión asociada a archivos comprimidos.

11.- Bibliografía

Firestore.

<https://firebase.google.com/>

Ley de Fitts.

<http://emmallensa.com/la-ley-fitts-aplicada-dispositivos-moviles/>

GoogleSheet.

<https://docs.google.com/spreadsheets/u/0/>

Android Design

<http://developer.android.com/design/index.html>

Justinmind

<http://www.justinmind.com/>

Wikipedia

<http://en.wikipedia.org>

Parse

<https://www.parse.com/>

12.- Anexos

Clase gasto

```
package com.example.shikan.eldelbar.database.entidades;
public class Gasto {
    private int id_gasto;
    private String Id_sesion;
    private String hora;
    private String fecha;
    private String descripcion;
    private double precio;
    private String nombreLocal;
    private String direccionLocal;
    private String localizacionLocal;
    public Gasto(){}
    public Gasto(int id_gasto, String Id_sesion, String fecha, String
hora, double precio, String descripcion, String nombreLocal, String
direccionLocal, String localizacionLocal){
        this.id_gasto = id_gasto;
        this.Id_sesion = Id_sesion;
        this.fecha = fecha;
        this.hora = hora;
        this.descripcion = descripcion;
        this.precio = precio;
        this.nombreLocal = nombreLocal;
        this.direccionLocal = direccionLocal;
        this.localizacionLocal = localizacionLocal;
    }
    //Getters
    public int getId_gasto() {
        return id_gasto;
    }
    public String getId_sesion() {
        return Id_sesion;
    }
    public String getFecha() {
        return fecha;
    }
    public String getHora() {
        return hora;
    }
    public String getDescripcion() {
        return descripcion;
    }
    public double getPrecio() {
        return precio;
    }
    public String getNombreLocal() {
        return nombreLocal;
    }
    public String getDireccionLocal() {
        return direccionLocal;
    }
}
```

```
}
public String getLocalizacionLocal() {
    return localizacionLocal;
}
//Setters
public void setFecha(String fecha) {
    this.fecha = fecha;
}
public void setHora(String hora) {
    this.hora = hora;
}
public void setDescription(String descripcion) {
    this.descripcion = descripcion;
}
}
public void setPrecio(double precio) {
    this.precio = precio;
}
}
public void setNombreLocal(String nombreLocal) {
    this.nombreLocal = nombreLocal;
}
}
public void setDireccionLocal(String direccionLocal) {
    this.direccionLocal = direccionLocal;
}
}
public void setLocalizacionLocal(String localizacionLocal) {
    this.localizacionLocal = localizacionLocal;
}
}
```

Clase Sesión

```
package com.example.shikan.eldelbar.database.entidades;
import com.orm.SugarRecord;
import java.sql.Time;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
//En una sesión almacenamos los datos de ESA sesión
public final class Sesión {
    private String Id_sesión; /*Presupuesto+fecha_comienzo+hora_comienzo*/
    private String fecha_hora_inicio;
    private String fecha_hora_fin;
    private double total_gastado;
    private int conseguido;
    private double presupuesto;
    private int terminada;
    private String titulo;
    public Sesión(){}
    public Sesión(String Id_sesión, String fecha_hora_inicio, String
fecha_hora_fin, double total_gastado, int conseguido, double
presupuesto, int terminada, String titulo){
        this.Id_sesión = Id_sesión;
        this.fecha_hora_inicio = fecha_hora_inicio;
        this.fecha_hora_fin = fecha_hora_fin;
    }
}
```

```
        this.total_gastado = total_gastado;
        this.conseguido = conseguido;
        this.presupuesto = presupuesto;
        this.terminada = terminada;
        this.titulo = titulo;
    }
    public String getFecha_hora_inicio() {
        return fecha_hora_inicio;
    }
    public String getFecha_hora_fin() {
        return fecha_hora_fin;
    }
    public int getConseguido() {
        return conseguido;
    }
    public double getTotal_gastado() {
        return total_gastado;
    }
    public double getPresupuesto() {
        return presupuesto;
    }
    public int getTerminada() {
        return terminada;
    }
    public String getTitulo() {
        return titulo;
    }
    public String getId_sesion() {
        return Id_sesion;
    }
    public void setFecha_hora_inicio(String fecha_hora_inicio) {
        this.fecha_hora_inicio = fecha_hora_inicio;
    }
    public void setFecha_hora_fin(String fecha_hora_fin) {
        this.fecha_hora_fin = fecha_hora_fin;
    }
    public void setPresupuesto(double presupuesto) {
        this.presupuesto = presupuesto;
    }
    public void setTerminada(int terminada) {
        this.terminada = terminada;
    }
    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }
}
```

Clase localización

```
package com.example.shikan.eldelbar;
import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Dialog;
```

```
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import com.google.android.gms.common.ConnectionResult;
import
com.google.android.gms.common.GooglePlayServicesNotAvailableException
;
import
com.google.android.gms.common.GooglePlayServicesRepairableException;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.PendingResult;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.location.places.GeoDataClient;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.PlaceDetectionClient;
import com.google.android.gms.location.places.PlaceFilter;
import com.google.android.gms.location.places.PlaceLikelihood;
import com.google.android.gms.location.places.PlaceLikelihoodBuffer;
import com.google.android.gms.location.places.Places;
import com.google.android.gms.location.places.ui.PlacePicker;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.common.api.GoogleApiClient;
import
com.google.android.gms.common.api.GoogleApiClient.OnConnectionFailedL
istener;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
```



```
import java.net.URI;
import java.util.ArrayList;
import java.util.List;
public class Localizacion extends FragmentActivity implements
OnConnectionFailedListener {
    private GoogleApiClient googleApiClient;
    private int PLACE_PICKER_REQUEST = 1;
    // The entry points to the Places API.
    private GeoDataClient mGeoDataClient;
    private PlaceDetectionClient mPlaceDetectionClient;
    // A default location (Sydney, Australia) and default zoom to use
    when location permission is
    // not granted.
    private final LatLng mDefaultLocation = new LatLng(-33.8523341,
151.2106085);
    private static final int DEFAULT_ZOOM = 15;
    private static final int PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION
= 1;
    private boolean mLocationPermissionGranted;
    private TextView tv_nombreLocal;
    private TextView tv_direccion;
    private TextView tv_telefono;
    private TextView tv_rating;
    private TextView tv_web;
    private EditText et_titulo;
    private EditText et_presupuesto;
    private ImageButton ib_comoLlegar;
    private String id_sesion;
    private Double total_gastado;
    private Double presupuestoActual;
    private String descripcionSalida;
    private Place lugar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_localizacion);
        tv_nombreLocal = findViewById(R.id.nombreLocal);
        tv_direccion = findViewById(R.id.direccion);
        tv_telefono = findViewById(R.id.telefono);
        tv_rating = findViewById(R.id.rating);
        tv_web = findViewById(R.id.web);
        ib_comoLlegar = findViewById(R.id.comoLlegar);
        id_sesion = getIntent().getStringExtra("id_sesion");
        total_gastado = getIntent().getDoubleExtra("total_gastado",
0.0);
        presupuestoActual =
getIntent().getDoubleExtra("presupuestoActual", 0.0);
        descripcionSalida =
getIntent().getStringExtra("descripcionSalida");
        //Construimos un geodataClient
        mGeoDataClient = Places.getGeoDataClient(this);
        // Construct a PlaceDetectionClient.
        mPlaceDetectionClient = Places.getPlaceDetectionClient(this);
        //Conectamos ala API de Google Places
        googleApiClient = new GoogleApiClient
.Builder(this)
```

```
        .addApi(Places.GEO_DATA_API)
        .addApi(Places.PLACE_DETECTION_API)
        .enableAutoManage(this, this)
        .build();
    googleApiClient.connect();
    //Iniciamos el seleccionador de sitios
    PlacePicker.IntentBuilder builder = new
PlacePicker.IntentBuilder();
    try {
        startActivityResult(builder.build(this),
PLACE_PICKER_REQUEST);
    }
    catch (GooglePlayServicesRepairableException e) {
        e.printStackTrace();
    }
    catch (GooglePlayServicesNotAvailableException e) {
        e.printStackTrace();
    }

    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then
overriding
        //    public void onRequestPermissionsResult(int
requestCode, String[] permissions,
        //                                     int[]
grantResults)
        // to handle the case where the user grants the
permission. See the documentation
        // for ActivityCompat#requestPermissions for more
details.
        return;
    }

    @SuppressWarnings("MissingPermission")
    PendingResult<PlaceLikelihoodBuffer> result =
Places.PlaceDetectionApi
        .getCurrentPlace(googleApiClient, null);
    result.setResultCallback(new
ResultCallback<PlaceLikelihoodBuffer>() {
        @Override
        public void onResult(PlaceLikelihoodBuffer likelyPlaces)
        {
            for (PlaceLikelihood placeLikelihood : likelyPlaces)
            {
                String TAG="Sitios";
                Log.i(TAG, String.format("Place '%s' has
likelihood: %g Tipo: '%s'",
                placeLikelihood.getPlace().getName(),
                placeLikelihood.getLikelihood(),
                placeLikelihood.getPlace().getPlaceTypes()
            ));
            }
            likelyPlaces.release();
        }
    });
}
```

```
});

}

private void getDeviceLocation() {
    if (ContextCompat.checkSelfPermission(this, getApplicationContext(),
        android.Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
        mLocationPermissionGranted = true;
    } else {
        ActivityCompat.requestPermissions(this,
            new String[]
            {android.Manifest.permission.ACCESS_FINE_LOCATION},
            PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION);
    }
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult
connectionResult) {
    //TODO: Falta notificar error de conexión -> HECHO
    String toastMsg = "Error de Conexión";
    Toast.makeText(this, toastMsg, Toast.LENGTH_LONG).show();
}

//Una vez seleccionado el sitio, aparecen aquí los detalles
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (requestCode == PLACE_PICKER_REQUEST && resultCode ==
    RESULT_OK) {
        Place place = PlacePicker.getPlace(this, data);
        final CharSequence name = place.getName();
        final CharSequence direcc = place.getAddress();
        final CharSequence tfn = place.getPhoneNumber();
        final Float rating = place.getRating();
        final Uri web = place.getWebsiteUri();
        final LatLng ubicacion = place.getLatLng();
        final Uri localizacion =
        Uri.parse("geo:"+ubicacion.toString()+"?
        q="+Uri.encode(name.toString()));
        //Codigo del AlertDialog
        AlertDialog.Builder builder = new
        AlertDialog.Builder(Localizacion.this);
        // Add the buttons
        builder.setTitle("Eldelbar")
            .setItems(R.array.Opciones, new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog,
            int which) {
                switch (which){
                    case 0:
                        //Hay una sesion creada
                        if(id_sesion != null){
                            //Pasamos el nombre del local
                            Intent i = new
                            Intent(getApplicationContext(), ActualPresupuesto.class);
                            startActivity(i);
                        }
                    }
                }
            }
        });
    }
}
```

```
name);
, localizacion.toString());
recc);
on);
fromLocalizacion",presupuestoActual);
Localizacion",total_gastado);
fromLocalizacion",descripcionSalida);
startActivity(i);
}
else{
    String localizacionString =
    dialogo(name.toString() ,direcc
    Log.d("Dialogo","Intentando
    acceder a dialogo()");
}
break;
case 1:
    if(web == null){
        tv_web.setText("No
especificado");
    }
    else {
        String url = web.toString();
        tv_web.setText(url);
    }
    String valoracion =
Float.valueOf(rating).toString();
tv_nombreLocal.setText(name);
tv_direccion.setText(direcc);
tv_telefono.setText(tfn);
tv_rating.setText(valoracion);
Log.d("LOCALIZACION",ubicacion.toS
tring());
new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new
Intent(Intent.ACTION_VIEW, localizacion);
        intent.setPackage("com.goo
gle.android.apps.maps");
        startActivity(intent);
    }
});
break;
case 2:
    Intent intent = new
```

```
Intent(Intent.ACTION_VIEW, localizacion);
                                                    intent.setPackage("com.google.android.apps.maps");
                                                    startActivity(intent);
                                                    break;
                                                    default:
Toast.makeText(getApplicationContext(),"Error Desconocido.",
Toast.LENGTH_SHORT);
                                                    Toast msg =
                                                    msg.show();
                                                    }
                                                    }
                                                    });
builder.create();
builder.show();
//Fin Codigo AlertDialog
}
}
/*
 *      MENU
 * */
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.layout.menu, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.menu_NuevoPresupuesto:
            nuevoPresupuesto();
            return true;
        case R.id.menu_Historico:
            Historico();
            return true;
        case R.id.menu_Localizar:
            Localizar();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
public void nuevoPresupuesto(){
    Intent presupuesto = new
Intent(getApplicationContext(),NuevoPresupuesto.class);
    startActivity(presupuesto);
}
public void Historico(){
    Intent i = new
Intent(getApplicationContext(),HistoricoSesiones.class);
    startActivity(i);
}
public void Localizar(){
    Intent localizar = new
```

```
Intent(getApplicationContext(), Localizacion.class);
    startActivity(localizar);
}
    public void dialogo (final String nombreLocal, final String
direccion, final String localizacion) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(Localizacion.this);
        Log.d("Dialogo", "Estamos en dialogo()");
        // Get the layout inflater
        LayoutInflater inflater =
Localizacion.this.getLayoutInflater();
        Toast msg =
Toast.makeText(getApplicationContext(), "dialogo()",
Toast.LENGTH_SHORT);
        msg.show();
        // Inflate and set the layout for the dialog
        // Pass null as the parent view because its going in the
dialog layout
        builder.setView(inflater.inflate(R.layout.alert_localizacion,
null));
        et_titulo = inflater.inflate(R.layout.alert_localizacion,
null).findViewById(R.id.tituloLocalizacion);
        et_presupuesto =
inflater.inflate(R.layout.alert_localizacion,
null).findViewById(R.id.presupuestoLocalizacion);
        builder.setTitle("Eldelbar")
        // Add action buttons
        .setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
id) {
                //et_titulo =
findViewById(R.id.tituloLocalizacion);
                //et_presupuesto =
findViewById(R.id.presupuestoLocalizacion);
                // Pasamos Titulo, presupuesto, nombreLocal,
direccion y localizacion
                descripcionSalida =
et_titulo.getText().toString();
                presupuestoActual =
Double.parseDouble(et_presupuesto.getText().toString());
                Toast msg =
Toast.makeText(getApplicationContext(), "descripcion"+descripcionSalid
a, Toast.LENGTH_SHORT);
                msg.show();
                Intent i = new
Intent(getApplicationContext(), ActualPresupuesto.class);
                i.putExtra("presupuesto", presupuestoActual);
                i.putExtra("descripcionSalida",
descripcionSalida);
                i.putExtra("nombreLocal", nombreLocal);
                i.putExtra("direccionLocal", direccion);
                i.putExtra("localizacionLocal", localizacion);
                startActivity(i);
            }
        })
    }
```

```
        })
        .setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int
id) {
        Toast msg =
Toast.makeText(getApplicationContext(), "Cancelado",
Toast.LENGTH_SHORT);
        msg.show();
        }
    });
    builder.create();
    builder.show();
}
}
```