

Desarrollo de aplicación Android: Sinlosofía

Trabajo Fin de Máster

Máster Universitario en Ingeniería Informática



Enero 2018

Autor

José Antonio Crespo Toral

Consultores

Jordi Ceballos Villach

Jordi Almirall López

Copyright © Jose Antonio Crespo Toral

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de aplicación Android: Sinlosofía</i>
Nombre del autor:	<i>Jose Antonio Crespo Toral</i>
Nombre del consultor/a:	<i>Jordi Ceballos Villach Jordi Almirall López</i>
Nombre del PRA:	<i>Robert Clarisó Viladrosa</i>
Fecha de entrega (mm/aaaa):	01/2018
Titulación:	<i>Máster Universitario en Ingeniería Informática</i>
Área del Trabajo Final:	<i>Desarrollo de aplicaciones sobre dispositivos móviles</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Restricciones Alimentarias, Android, Firebase</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Hoy en día, cada vez hay más personas con restricciones alimentarias que limitan ciertos aspectos de sus vidas como comer fuera de casa. El presente trabajo pretende ayudar a este problema mediante una herramienta que les ayude en la labor de buscar establecimientos adaptados a ellos.</p> <p>El proyecto tiene como objetivo principal el desarrollo de un sistema formado por una aplicación Android, Sinlosofía, y un backend. La app de Sinlosofía permite buscar establecimientos que ofrezcan comida adaptada para diferentes restricciones alimentarias, así como añadir nuevos, y crear una red social para compartir tu opinión acerca de los establecimientos visitados y conocer las experiencias del resto de usuarios.</p> <p>La metodología utilizada en este proyecto ha sido el ciclo de vida en casada que se divide en diferentes etapas: Análisis, Diseño, Implementación, Pruebas y Mantenimiento. Merece la pena destacar la utilización del Diseño Centrado en el Usuario que sitúa las necesidades y características del usuario en el centro de cada una de las etapas de diseño.</p> <p>El desarrollo de la aplicación se ha realizado de forma nativa con Android Studio y el SDK Android, mientras que para el backend se ha utilizado Firebase que integra multitud de productos para este cometido. Por otra parte, también han sido utilizadas otras librerías y APIs (Google Maps, Glide y Espresso).</p> <p>Finalmente, el resultado obtenido es un producto funcional y depurado que cumple los objetivos definidos al inicio del proyecto. Además, nos ha permitido aumentar y afianzar los conocimientos adquiridos durante el máster.</p>	
Abstract (in English, 250 words or less):	
<p>Nowadays, there are more and more people with food restrictions that limit certain aspects of their lives like eating out. The present work aims to help this problem through a tool that helps them find establishments adapted to them.</p> <p>The main objective of the project is the development of a system formed by an Android application, Sinlosofía, and a backend. The Sinlosofía app allows you to search for establishments that offer adapted food for different food restrictions, as well as add new ones, and create a social network to share your opinion about the establishments visited and learn about the experiences of other users.</p> <p>The methodology used in this project has been the Waterfall model that is divided into different stages: Analysis, Design, Implementation, Testing and Maintenance. It is worth highlighting the use of User Centered Design that places the needs and characteristics of the user at the center of each of the design stages.</p> <p>The development of the application has been done natively with Android Studio and the Android SDK, while the backend has used Firebase that integrates a multitude of products for this purpose. On the other hand, other libraries and APIs have also been used (Google Maps, Glide and Espresso).</p> <p>Finally, the result obtained is a functional and refined product that meets the objectives defined at the beginning of the project. In addition, it has allowed us to increase and consolidate the knowledge acquired during the master's degree.</p>	

Tabla de contenido

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	4
1.3 Enfoque y método seguido	6
1.4 Planificación del Trabajo	7
1.4.1 Definición de tareas	7
1.4.2 Planificación temporal	9
1.4.3 Riesgos del proyecto	9
1.5 Breve resumen de productos obtenidos	10
1.6 Breve descripción de los otros capítulos de la memoria	11
2. Análisis, Diseño y Arquitectura	12
2.1 Diseño Centrado en el Usuario	12
2.1.1 Usuarios y Contexto de uso (Análisis)	12
2.1.1.1 Entrevista en profundidad	12
2.1.1.1.1 Diseño y desarrollo de la entrevista	13
2.1.1.1.2 Conclusiones	15
2.1.1.2 Benchmarking	15
2.1.1.3 Encuestas	16
2.1.1.3.1 Diseño del cuestionario y desarrollo	16
2.1.1.3.2 Resultados y conclusiones	19
2.1.1.4 Perfiles de usuario	21
2.1.2 Diseño Conceptual	23
2.1.2.2 Personajes	23
2.1.2.2 Escenarios de Uso	25
2.1.2.2 Flujo de Interacción	30
2.1.3 Prototipado	31
2.1.3.1 Sketches	31
2.1.3.2 Prototipos horizontales de alta fidelidad	32
2.1.4 Evaluación	34
2.1.4.1 Test de Usuarios	34
2.1.4.1.1 Perfil socio-demográfico y documento de <i>Screening</i>	34
2.1.4.1.2 Cuestionario Pre-test	35
2.1.4.1.3 Escenario y tareas	35
2.1.4.1.4 Cuestionario Post-test	36
2.1.4.1.5 Desarrollo y conclusiones	36
2.2 Diseño técnico	39
2.2.1 Casos de uso	39
2.2.2 Diseño de datos	49
2.2.3 Diseño de la arquitectura	51
2.2.3.1 Vista Estática	52
2.2.3.1.1 Arquitectura de la aplicación	52
2.2.3.1.2 Vista Física	53
3. Implementación	55
3.1 Herramientas y APIs utilizadas	55
3.1.1 Android	55
3.1.1.1 Android Studio	56
3.1.2 Firebase	56
3.1.3 Google Maps API	56
3.1.4 Glide	57
3.1.5 Espresso	57
3.2. Aspectos relevantes del desarrollo	57

3.2.1 Proyecto Android	57
3.2.2 Diseños <i>responsive</i> para diferentes tamaños de pantalla	58
3.2.3 Diseño de la <i>Navigation Drawer</i> Activity	58
3.2.4 Diseño de la <i>Bottom Navigation</i> Activity	59
3.2.5 Estructura de Realtime Database Firebase	59
3.2.6 Almacenamiento y recuperación de datos en Firebase RealTime Database.....	61
3.2.7 Gestión de usuarios con Firebase Authentication	62
3.2.8 Obtención y compresión de imágenes y almacenamiento en Firebase Storage.....	62
3.2.9 Descarga de imágenes de Firebase Storage con Glide y FirebaseUI	64
3.2.10 Obtención de direcciones con Google Maps	65
3.2.11 <i>Geocoder</i>	65
3.2.12 Obtener la localización del usuario.....	66
3.2.13 Búsqueda de establecimientos y visualización de resultados en mapa	67
3.2.14 Valoración y actualización de puntuación medida	68
3.2.15 <i>Custom Adapters</i> y <i>filter</i>	69
3.2.16 <i>Parcelable</i>	69
3.3 Pruebas	70
3.3.1 Pruebas manuales.....	70
3.3.2 Pruebas automáticas.....	72
3.4. Funcionalidad de la aplicación.....	74
3.4.1 Gestión de usuarios	74
3.4.1.1 Registro de usuario y visualización de la cuenta del usuario.....	74
3.4.1.1 Login y cierre de sesión	75
3.4.2 Búsquedas.....	75
3.4.3 Establecimientos.....	76
3.4.3.1 Visualización de los datos, comentarios y fotografías del establecimiento	76
3.4.3.2 Valorar un establecimiento	76
3.4.3.3 Añadir establecimiento	77
3.4.4 Listas personalizadas.....	77
3.4.4.1 Ver tus establecimientos favoritos y añadir o eliminar un establecimiento favorito	77
3.4.4.2 Ver tus establecimientos recomendados y añadir o eliminar un establecimiento recomendado	77
4. Conclusiones	78
4.1 Líneas de trabajo futuro.....	79
5. Glosario.....	80
6. Bibliografía.....	81
6.1 Recursos de texto	81
6.2 Recursos web	81
6. Anexos	83

1. Introducción

1.1 Contexto y justificación del Trabajo

La tecnología ha avanzado muy rápidamente en la última década y esto ha provocado cambios en nuestras costumbres y maneras de hacer las cosas. Hoy en día, no se concibe la idea de estar un día desconectados, de hecho, el móvil está presente prácticamente en todos los momentos de nuestro día utilizándolo tanto cuando realizamos actividades como comprar, trabajar, ver la televisión... como cuando nos tomamos un descanso. El siguiente gráfico muestra las actividades más comunes que realizamos al tiempo que usamos nuestro dispositivo móvil:

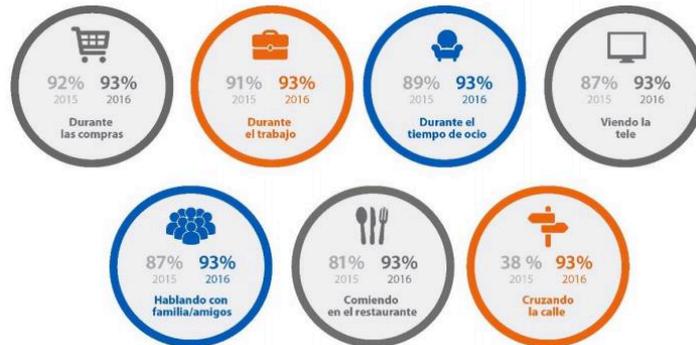


Ilustración 1. Uso del móvil mientras se realizan otras actividades en el mundo
(Fuente: Informe Amic 2017)

Según el estudio 'Google Consumer Barometer' en cinco años prácticamente se ha duplicado el número de españoles que usan un smartphone del 44% en 2012 al 87% en 2017. Entre los usos principales tenemos la mensajería instantánea, ver vídeos, jugar en el móvil, usar la banca móvil o los servicios de geolocalización.

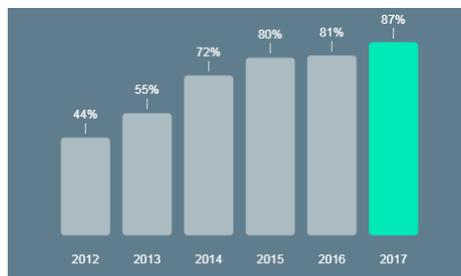


Ilustración 2. Uso de smartphone en España
(Fuente: Consumer Barometer)

De hecho, como refleja el informe Amic de 2017, España es líder mundial en penetración con un 88% de usuarios móviles siendo el promedio global de un 66%. Además, de estos usuarios un 94,6% utilizan el dispositivo móvil como primera opción para acceder a internet por encima de otros dispositivos como el ordenador portátil o la tablet.

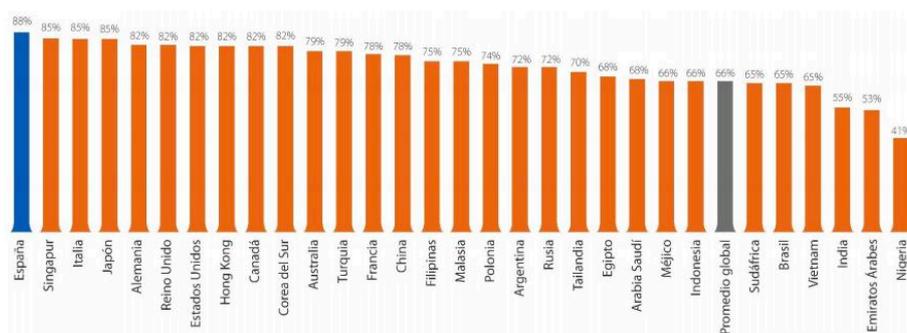


Ilustración 3. Número de usuarios con móviles por país.
(Fuente: Informe Amic 2017)

Otros datos relevantes que se desprenden de este informe son que el uso de aplicaciones móviles sigue disparándose aumentando un 11% en el último año y que el 99% de los jóvenes españoles accede a internet a diario desde su móvil, lo que supone sin duda una adopción total. De hecho, la generación *millennials* dedica al uso de apps una media de 93,5 horas al mes.

Como vemos, estamos evolucionando hacia un mundo digital, en el cual los dispositivos móviles avanzan a pasos agigantados formando completamente parte de nuestra vida cotidiana y llegando a ser incluso una necesidad. De hecho, de acuerdo con Eric Schmidt, ex CEO de Google, en el 2030 el mundo estará tan conectado que se borrarán las fronteras entre lo online y offline y la red estará tan integrada a nuestro quehacer diario que apenas percibiremos su presencia.

Unido a este auge, en los últimos años también nos enfrentamos a la existencia de gran cantidad de enfermedades que provocan restricciones alimentarias, como por ejemplo:

- **Celiaquía** que es una enfermedad sistémica inmunomediada provocada por el gluten que obliga a las personas que la padecen a llevar una dieta estricta sin gluten. El gluten es una proteína amorfa que se encuentra en la semilla de muchos cereales como trigo, cebada, centeno, etc. Similar a esta patología también encontramos la sensibilidad al gluten no celiaca o alergia al gluten, que también implican el mismo tratamiento, dieta estricta.
- **Intolerancia a la lactosa**, se trata de un trastorno digestivo que impide la digestión de la lactosa por una deficiencia de la enzima lactasa y cuyo tratamiento es una dieta sin/baja en lactosa dependiendo del nivel de intolerancia. La lactosa o azúcar de la leche es un disacárido que está presente en la leche que producen los mamíferos.
- **Diabetes** es una enfermedad crónica en la cual el cuerpo no puede regular la cantidad de azúcar en la sangre debido a una insuficiente producción de insulina, una resistencia a esta o ambas. Esto hace que en lo relativo a la ingesta de alimentos los diabéticos deban, generalmente, limitar y controlar el consumo de azúcar.

Estas enfermedades provocan un cambio en las rutinas, en mayor o menor medida, de las personas que las padecen debiendo prestar especial atención a los alimentos que consumen y dificultando todas aquellas situaciones que impliquen comer o beber fuera de casa. Esto se complica aún más cuando la persona padece estos problemas de manera combinada pues sus restricciones alimentarias son aún mayores.

Algunas preguntas frecuentes que les surgen estas personas son “¿En qué restaurante puedo comer con mis amigos o familiares sin peligro? ¿Dónde puedo tomar unas cervezas sin gluten o un café sin lactosa en mi ciudad? ¿En qué pastelerías o establecimientos puedo comprar dulces aptos para diabéticos?”. El presente trabajo pretende cubrir esta necesidad, ayudando a todas estas personas a llevar una vida común a pesar de sus restricciones alimentarias, de forma que a través de un vistazo a su smartphone pueda resolver estas dudas de forma rápida.

Si realizamos un análisis por la tienda de aplicaciones de Android, Google Play, podemos encontrar algunas aplicaciones relacionadas con esta problemática como las que se detallan a continuación:

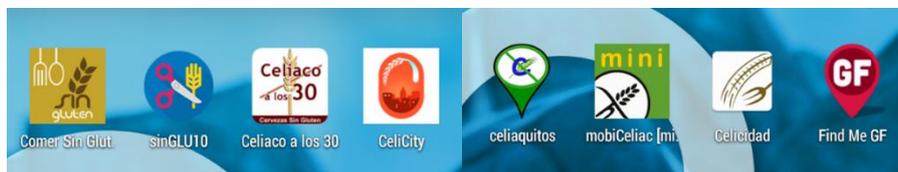


Ilustración 4. Apps analizadas

- **Comer Sin Gluten.** Aplicación dedicada a buscar restaurantes que ofrezcan platos sin gluten en una determinada ubicación geográfica o alrededor de la posición del usuario. Sin embargo, no permite interactuar con los establecimientos, valorándolos o escribiendo una opinión, ni tampoco añadir nuevos establecimientos por parte de los usuarios. La interfaz es bastante sencilla al ofrecer una única funcionalidad.
- **SinGlu10.** Se centra únicamente en personas que no pueden tomar gluten y su funcionalidad principal es comprobar qué productos están libre de gluten a partir de un listado integrado o mediante la lectura de un código de barras. Adicionalmente, también muestra una pequeña lista

de tiendas, hoteles y restaurantes de Madrid que ofrecen platos sin gluten, pero no permite la interacción de los usuarios con los mismos ni la adición de nuevos. La aplicación presenta una interfaz simple y fácil de usar.

- **Celiaco a los 30.** Se centra en el gluten únicamente y especialmente en cervezas. Su funcionalidad principal consiste en buscar en un mapa, mediante la ubicación o posición del usuario, bares con cervezas sin gluten o establecimientos con cartas con opción sin gluten. Los usuarios pueden añadir o quitar establecimientos, pero no valorarlos y la información que se muestra es escasa. También tiene un apartado dedicado a las cervezas sin gluten con un catálogo, una guía de cata, la opción de votar tu cerveza favorita y un ranking. En cuanto a la interfaz la considero poco usable, compleja y con una mala organización del contenido lo que dificulta su utilización por parte de los usuarios.
- **Celicity.** Está dirigida únicamente a personas que no pueden tomar gluten. Es una de las aplicaciones más completas que hemos encontrado en cuanto a funcionalidad y se centra en la búsqueda de establecimientos que ofrecen platos sin gluten a partir de la localización del usuario o en una ubicación concreta. Además, permite a los usuarios añadir nuevos sitios, marcar los favoritos, puntuar los establecimientos y añadir opiniones. La aplicación también presenta una interfaz cuidada, vistosa e intuitiva.
- **Celiaquitos.** Está destinada solamente para personas que siguen una dieta sin gluten. En general, tiene una funcionalidad bastante completa, ofreciendo la búsqueda de establecimientos que ofrezcan comidas sin gluten, pudiendo añadir ciertos filtros como “sin lactosa” aunque no permite la búsqueda de establecimientos sin lactosa únicamente. Los usuarios pueden compartir establecimientos a través de la app, puntuar y comentar los existentes. La interfaz no es muy agradable y considero que tiene una organización mejorable.
- **MobiCeliac [mini].** Está focalizada en tema del gluten y tiene dos funcionalidades principalmente: buscar alimentos sin gluten en una base de datos que se descarga y buscar tiendas, restaurantes y hoteles que tengan platos libres de gluten a partir de su nombre, ordenándolos por zona geográfica o por posición del usuario, pero no permite buscar por ubicación ni muestra los resultados en un mapa. La ficha de los restaurantes muestra poca información y no permite a los usuarios valorar los establecimientos ni tampoco añadir nuevos. Dispone de una versión de pago MobiCeliac [XL] con más funcionalidades para alimentos sin gluten como la lectura de código de barras. La interfaz de la aplicación es sencilla e intuitiva, aunque no muy llamativa.
- **Celicidad.** Está destinada a la problemática del gluten. Ofrece una funcionalidad muy completa centrándose en buscar un establecimiento donde comer sin gluten a partir de una ubicación o alrededor de la posición del usuario. También ofrece información de los establecimientos y la posibilidad de que los usuarios suban fotos, escriban un comentario o lo marquen como favorito. Sin embargo, no permite añadir nuevos sitios por parte de los usuarios. La interfaz que presenta la aplicación es generalmente intuitiva y sencilla.
- **Find me GF.** Como las anteriores se centra únicamente en la problemática del gluten, no está disponible en español y la interfaz no resulta muy intuitiva. Entre sus funcionalidades principales encontramos un tablón de noticias, buscar establecimientos libres de gluten por ubicación (aunque solo dispone de algunos situados en Madrid), valorar y comentar los establecimientos, una sección de recetas sin gluten y una lista de ingredientes indicando el contenido de gluten.

Como hemos podido observar todas las aplicaciones están destinadas a personas que no pueden tomar gluten, sin considerar a otros usuarios que presentan otras restricciones alimentarias como los intolerantes a la lactosa o diabéticos. Otro punto a destacar es que en la mayoría de las aplicaciones son los administradores quienes gestionan los establecimientos disponibles, impidiendo la creación de una comunidad y no permitiendo a los usuarios añadir nuevos lugares. Algunas de ellas tampoco permiten ningún tipo de interacción de los usuarios con los establecimientos, como la subida de fotos o valoraciones a través de una puntuación o de un comentario, y otras solo tienen algunas de estas funcionalidades. Por otra parte, destacar que muchas de las aplicaciones no cuentan con una interfaz diseñada pensando en el usuario con la finalidad de aumentar su usabilidad. En definitiva, ninguna de las aplicaciones analizadas se centra en la totalidad del problema que queremos abordar ni posee todas las funcionalidades que nos planteamos llevar a cabo.

Con todo ello, lo que queremos conseguir es crear una comunidad de usuarios mayor a las existentes uniendo a personas con diferentes restricciones alimentarias con el fin de que puedan ayudarse entre sí compartiendo nuevos establecimientos adaptados a sus necesidades y permitiendo la interacción con los existentes por diferentes medios como comentarios, puntuaciones, etc.

1.2 Objetivos del Trabajo

En este apartado se describen los objetivos del Trabajo Fin de Máster que aquí nos ocupa, los requisitos funcionales y no funcionales de la aplicación móvil y la matriz de rastreabilidad que detalla las dependencias entre los objetivos del sistema y los requisitos. También se detallan los objetivos personales de este trabajo y las plataformas y/o dispositivos a los que se dirige.

El objetivo principal de este trabajo es ofrecer una aplicación móvil que permita a personas con diferentes restricciones alimentarias encontrar establecimientos de hostelería adaptados a sus necesidades en cualquier lugar y momento, creando una comunidad que posibilite a los usuarios compartir los restaurantes, bares y cafeterías que conozcan.

Los objetivos más específicos de la aplicación a desarrollar en este trabajo son:

- OBJ1 – El sistema deberá permitir registrar a nuevos usuarios, así como iniciar sesión a los usuarios previamente registrados y acceder a su perfil.
- OBJ2 – El sistema deberá mostrar los establecimientos cercanos a la posición del usuario o buscar establecimientos en una ubicación específica, todo ello de acuerdo a unos filtros.
- OBJ3 – El sistema deberá mostrar en la ficha de cada establecimiento información como dirección, valoración, opiniones, etc.
- OBJ4 – El sistema deberá permitir que los usuarios valoren los establecimientos, escriban una opinión, añadan una fotografía desde sus dispositivos o recomienden un establecimiento.
- OBJ5 – El sistema permitirá a los usuarios guardar sus establecimientos favoritos o pendientes de visitar, así como visualizar sendas listas.
- OBJ6 – El sistema deberá permitir a los usuarios dar de alta nuevos establecimientos en la aplicación, estando disponible también para el resto de usuarios.

A continuación, se detallan los requisitos funcionales de la aplicación:

- RF1- Gestión de los usuarios
 - RF1.1- Registrarse en la aplicación
 - RF1.2- Iniciar sesión en la aplicación
 - RF1.3- Visualizar los datos de la cuenta del usuario
- RF2- Búsqueda de establecimientos de hostelería
 - RF2.1- Buscar en función de la posición actual del usuario
 - RF2.2- Buscar en una ubicación específica
 - RF2.3- Filtrar las búsquedas
- RF3- Ficha de establecimientos de hostelería
 - RF3.1- Proporcionar información sobre el establecimiento
 - RF3.2- Mostrar la puntuación media en función de todas las valoraciones del establecimiento
 - RF3.3- Mostrar las opiniones de los usuarios acerca del establecimiento
 - RF3.4- Mostrar las fotografías de los usuarios
- RF4- Interacción con los establecimientos
 - RF4.1- Añadir un nuevo restaurante, bar o cafetería a la aplicación
 - RF4.2- Valorar un establecimiento
 - RF4.3- Escribir una opinión
 - RF4.4- Añadir una fotografía
 - RF4.5- Guardar el establecimiento en favoritos
 - RF4.6- Eliminar un establecimiento de favoritos

- RF4.7- Guardar el establecimiento como pendiente de visitar
- RF4.8- Eliminar un establecimiento de pendiente de visitar
- RF4.9- Recomendar un establecimiento a un amigo
- RF5- Gestión de listas personalizadas
 - RF5.1 Visualizar lista de favoritos
 - RF5.2 Visualizar lista de pendiente de visitar

Por otra parte, los requisitos no funcionales son:

- RNF1 – La aplicación deberá implementarse para la plataforma Android de Google en Java.
- RNF2 – La aplicación cliente deberá obtener los datos a mostrar del backend.
- RNF3 – El sistema deberá estar disponible en todo momento y permitirá la autenticación de los usuarios.
- RNF4 – El sistema deberá ser robusto y fiable no produciendo errores críticos que produzcan el cierre de la aplicación.
- RNF5 – El sistema deberá proporcionar mensajes de error que sean informativos y en un vocabulario común entendible por cualquier usuario.
- RNF6 – El sistema deberá proporcionar la seguridad necesaria para evitar el acceso no autorizado a los datos del backend.
- RNF7 – El sistema deberá ser capaz de operar adecuadamente con hasta 100 usuarios concurrentes (licencia gratuita backend).
- RNF8 – El sistema deberá ser sencillo y fácil de usar con un tiempo de aprendizaje por parte de los usuarios no superior a 1 hora.

La matriz de rastreabilidad dados los objetivos y requisitos anteriores es:

MR-1	OBJ1	OBJ2	OBJ3	OBJ4	OBJ5	OBJ6
RF1.1	X					
RF1.2	X					
RF1.3	X					
RF2.1		X				
RF2.2		X				
RF2.3		X				
RF3.1			X			
RF3.2			X			
RF3.3			X			
RF3.4			X			
RF4.1						X
RF4.2				X		
RF4.3				X		
RF4.4				X		
RF4.5					X	
RF4.6					X	
RF4.7					X	
RF4.8					X	
RF4.9			X			
RF5.1					X	
RF5.2					X	

Ilustración 5. Matriz de rastreabilidad

Personalmente, los objetivos que pretendo conseguir con este trabajo son consolidar y aumentar los conocimientos aprendidos a lo largo del Máster y formaciones anteriores, en especial mis conocimientos acerca del desarrollo de aplicaciones móviles en Android, en el que no tengo mucha experiencia, y del desarrollo de backend para app móviles, que es totalmente novedoso para mí.

El sistema que se desarrolla en este trabajo consiste en una aplicación distribuida formada por:

- Un backend que se encarga de autenticar a los usuarios y de almacenar todos los datos del sistema como establecimientos, usuarios, opiniones, etc.
- Una aplicación cliente que obtendrá los datos del backend para proporcionar las diferentes funcionalidades a los usuarios.

Para la parte del backend se utilizará el servicio de Google Firebase, ya que es uno de los más potentes del mercado, con una documentación muy completa y relativamente sencillo de utilizar. En cuanto a la aplicación cliente se desarrollará de forma nativa utilizando la plataforma de Android. Su justificación radica en que esta plataforma posee la mayor cuota de mercado y puede ser subida a Google Play con un coste menor de licencia de desarrollo y sin mantenimiento, en contraposición a su mayor competidor, iOS de Apple. Además, al tratarse de una aplicación nativa, nos permitirá de forma sencilla acceder a todas las funciones del dispositivo y proporcionará un mejor rendimiento.

Por otra parte, este proyecto se dirige a dispositivos móviles para que nuestros usuarios puedan utilizar la aplicación en cualquier momento y lugar.

1.3 Enfoque y método seguido

Como hemos comentado en el primer apartado, existen algunas aplicaciones con más o menos funcionalidades focalizadas en esta temática, aunque no abarcan toda la problemática existente en cuanto a las restricciones alimentarias, centrándose exclusivamente en el gluten. Una opción para este proyecto podría ser adaptar una de estas aplicaciones al objetivo de este trabajo añadiéndole nuevas funcionalidades, corrigiendo otras e incorporando el resto de restricciones alimentarias que hemos citado. Sin embargo, estas aplicaciones son privadas, se comercializan de forma oficial en Google Play, no se dispone de su código y pueden no estar interesadas en nuestro proyecto.

Por lo tanto, la estrategia a seguir para llevar a cabo el trabajo es desarrollar un producto nuevo que aumente el público objetivo de la app al añadir otras restricciones de alimentación y que incorpore tanto funcionalidades que ya poseen algunas aplicaciones como otras nuevas que resuelvan las necesidades de los usuarios. Esta decisión también me permitirá aumentar mis conocimientos en la construcción de este tipo de sistemas.

En cuanto a la metodología de desarrollo de software que vamos a utilizar en este proyecto se ha elegido el modelo clásico en cascada combinado con el Diseño Centrado en el Usuario. La decisión de utilizar este modelo reside en que se trata de una metodología sencilla, los requisitos generalmente serán fijos y no se espera que cambien durante el ciclo de vida, ya que es un proyecto sin mucha incertidumbre.

El modelo en cascada consiste en un proceso de desarrollo que se basa en una serie de etapas que se ejecutan una tras otra. Estas etapas son:

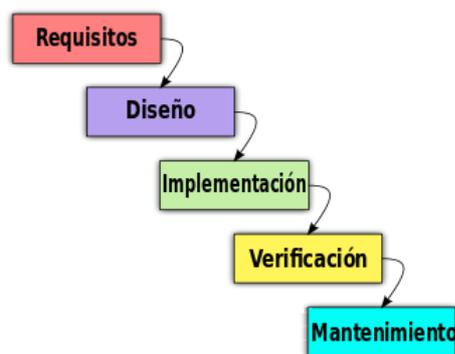


Ilustración 6. Ciclo de vida del desarrollo en cascada
(Fuente: Wikipedia)

- **Análisis de requisitos:** se realiza un análisis de las necesidades del cliente para determinar las características del software a desarrollar.
- **Diseño:** en esta etapa se describe la estructura interna del software y las relaciones entre las entidades que lo componen. Para ello, se descompone y se organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo.
- **Implementación:** se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior.

- **Verificación o pruebas:** se verifica que todos los componentes del sistema funcionen correctamente y cumplan con los requisitos.
- **Mantenimiento:** consiste en la modificación del producto una vez finalizado, ya sea para corregir errores o para mejorar su rendimiento o características.

Este modelo de desarrollo se combinará con el Diseño Centrado en el Usuario (DCU), de modo que en cada etapa del modelo en cascada se aplicarán los conceptos del DCU realizando iterativamente cada una de sus fases, las cuales serán descritas en capítulos posteriores de este documento. Esto nos permite situar al usuario en el centro de todo el proceso con el objetivo de obtener información sobre los usuarios, sus tareas y sus objetivos para orientar el diseño y desarrollo de productos, de modo que el resultado sea productos usables que se adapten a sus necesidades y con una buena experiencia de usuario.

1.4 Planificación del Trabajo

En este apartado se definen cada una de las tareas y actividades que se pretenden llevar a cabo y las fechas en las que éstas se deben ejecutar. Con esto, se pretende obtener un plan de desarrollo de tareas que permita un mayor control a la hora de desarrollar el proyecto, además de una estimación aproximada de la fecha de finalización del mismo.

En primer lugar, el equipo de trabajo de este TFM está formado por tres personas, el alumno y autor material del mismo, José Antonio Crespo Toral, y dos consultores: Jordi Ceballos Villach que realiza la función de apoyo y soporte general y Jordi Almirall López que también presta apoyo y guía, aunque más centrado en los aspectos relacionados con el diseño de interacción y usabilidad.

Los recursos de hardware necesarios y utilizados para este proyecto son:

- Un ordenador iMac con un Intel Core i5, 8GB de RAM, Samsung SSD EVO 500GB y MacOS Sierra (10.12.6) instalado.
- Un Smartphone Samsung Galaxy S6 con Android Nougat (7.0).

En cuanto a los recursos de software principalmente tenemos:

- Android Studio: entorno de desarrollo oficial de Android con el SDK necesario para implementar aplicaciones Android nativas.
- Microsoft Word: para la realización de la documentación asociada al trabajo.
- GanttProject: para la planificación temporal del proyecto.
- Firebase: se trata de un *backend as a service* (BaaS) de Google que ofrece un conjunto de herramientas para el desarrollo de aplicaciones como la autenticación, el almacenamiento y la sincronización de datos en la nube.
- Adobe Photoshop CS6: para la creación y edición de los recursos gráficos necesarios.

1.4.1 Definición de tareas

El plan de trabajo realizado cuenta con cinco fases que se basan en el modelo de evaluación continua propuesto por la UOC. En estas cinco etapas se definen una serie de tareas a realizar y se establecen una serie de hitos que se corresponden con las entregas parciales del proyecto (PECs).

0. Definición del proyecto

- Planteamiento de la idea general
- Redacción de la propuesta
- Propuesta TFM (hito)

En esta primera fase nace la idea en la que se basa el proyecto y se le presenta al consultor para su aprobación.

1. Plan de trabajo

- Contexto y motivación
- Objetivos y requisitos
- Enfoque y metodología
- Planificación del proyecto
- Documentación
- Entrega PEC1 (hito)

En esta fase inicial se definirá el contexto y justificación, los objetivos del Trabajo Final y los requerimientos funcionales y no funcionales a alto nivel de la aplicación. También se describirá la metodología y estrategias para llevar a cabo el trabajo y se detallará la planificación temporal del mismo. Esta fase finaliza con un hito que representa la entrega de la PEC1.

2. Análisis, Diseño y Arquitectura

- Diseño Centrado en el Usuario (DCU)
- Diseño Técnico
 - Definición de los casos de uso
 - Diseño de la arquitectura
- Documentación
- Entrega PEC2 (hito)

Esta es una de las fases más importantes que dan forma a nuestro proyecto, en primer lugar, se lleva a cabo el DCU de forma iterativa entre sus cuatro fases (Usuarios y contexto de uso, Diseño conceptual, Prototipado y Evaluación). Posteriormente, tras el DCU, se realiza el diseño técnico del sistema definiendo formalmente los casos de uso para establecer las funcionalidades de la aplicación y la arquitectura del sistema identificando: las entidades de la base de datos para realizar los diagramas de diseño correspondientes; las clases y objetos que intervienen en los procesos que se plasmarán en diagramas de diseño UML y la estructura API de comunicación entre el cliente y el servidor. Para finalizar, encontramos el hito de entrega de la PEC2.

3. Implementación

- Preparación del entorno
- Aprendizaje desarrollo backend
- Desarrollo backend
- Aprendizaje desarrollo Android
- Desarrollo App Cliente Android
- Testing
 - Definición de las pruebas
 - Realización de pruebas
- Documentación
- Entrega PEC3 (hito)

En esta fase es donde se produce la construcción del producto del Trabajo Fin de Máster. Al comienzo de la fase encontramos una tarea de preparación, en la que se gestionarán y configurarán todas las librerías necesarias, y una serie de actividades de aprendizaje en las tecnologías involucradas. A continuación, se encuentra el desarrollo de backend y la aplicación Android y, por último, tenemos la fase de pruebas en la que se definen y se realizan las mismas comenzado durante el desarrollo. Esta etapa finaliza con el hito de entrega de PEC3.

4. Entrega Final

- Finalización de la memoria
- Confección de la presentación
- Grabación vídeo presentación
- Preparación de la entrega
- Entrega final (hito)

En esta última fase se finalizan los productos desarrollados y la memoria del Trabajo Fin de Máster y se elabora una presentación con los detalles más relevantes del trabajo como los aspectos clave del desarrollo, las decisiones tomadas y las conclusiones y resultados obtenidos. Una vez finalizada se grabará un vídeo en el que se explicará y comentará la presentación realizada. Por último, se producirá la entrega final del Trabajo Fin de Máster.

1.4.2 Planificación temporal

En este apartado se muestra la planificación temporal de cada tarea del trabajo. Para estimar la duración de las mismas se ha partido de la premisa de dedicar al proyecto 3 horas en días laborables y 6 horas los días de fin de semana o festivos, aunque en la fase de implementación el número de horas de dedicación durante los días festivos y de fin de semana aumentará hasta las 8 horas, puesto que la carga de trabajo será mayor. Como existen tareas concurrentes las horas diarias se dividen entre las actividades planificadas durante ese período.

El Trabajo Fin de Máster comienza el 20 de septiembre de 2017 y la fecha final del proyecto es el 10 de enero de 2018. La duración total del proyecto se estima en 436 horas que serán realizadas en 108 días. Así mismo, también se ha teniendo en cuenta que los días festivos 24, 25 y 31 de diciembre y el 1 de enero no se trabajará en el proyecto.

A continuación, se muestra el diagrama de Gantt con las tareas a ejecutar y su estimación:

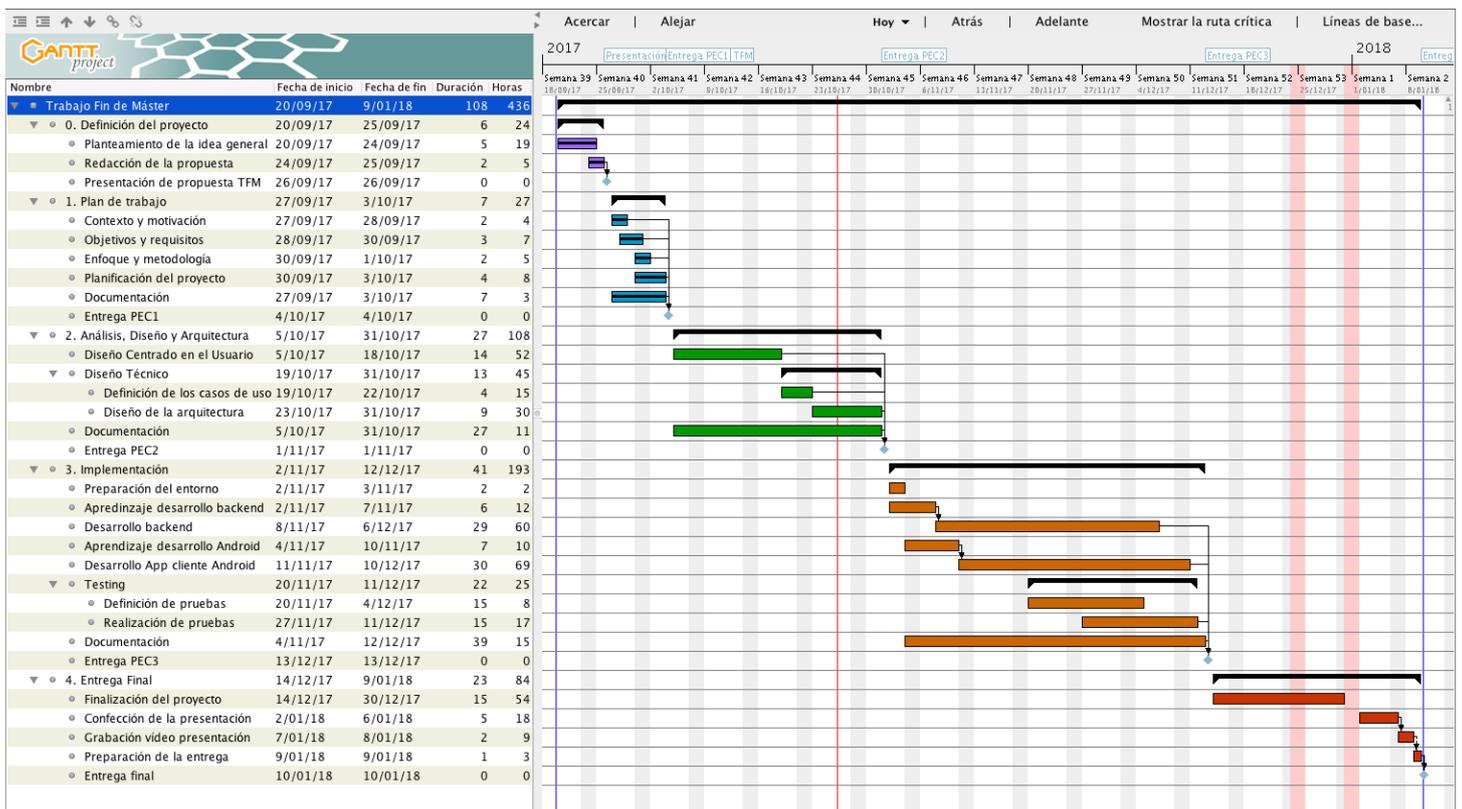


Ilustración 7. Diagrama de Gantt del trabajo

1.4.3 Riesgos del proyecto

Como en todo proyecto existen una serie de riesgos que pueden conducir al fracaso del proyecto. Para evitar esta situación y anticiparnos a los mismos, definimos un plan de riesgos en el que se detallan la probabilidad de que ocurran, el impacto en caso de materializarse y las acciones mitigadoras para reducir su aparición.

RIESGOS					
Nombre	Descripción	Probabilidad	Impacto	Riesgo	Acciones mitigadoras
Carga de trabajo laboral	Periodos en los que aumente la carga de trabajo laboral provocando jornadas laborales más extensas y mayor dedicación	Alta	Medio	Riesgo mayor	- Adelantar todo el trabajo posible en los períodos de menor carga - No posponer tareas ni dejar trabajo pendiente
Visitas familiares	La llegada de una visita familiar o la realización de un viaje para visitar a la familia y amigos durante el fin de semana reduciendo el tiempo de dedicación	Media	Medio	Riesgo significativo	- Adelantar todo el trabajo posible en los períodos de menor carga - No posponer tareas ni dejar trabajo pendiente - Intentar dedicar al menos un cuarto de las horas diarias estipuladas a pesar de la visita o viaje
Enfermedades	Contraer alguna enfermedad durante el período del proyecto o por parte de algún familiar	Media	Bajo	Riesgo menor	- Llevar una vida saludable con ejercicio semanal y una buena alimentación - Prestar atención a los primeros síntomas y acudir al médico tan pronto como se detecten
Pérdida del proyecto	Fallo de hardware o software que provoque la pérdida de todo el trabajo realizado	Media	Alto	Riesgo mayor	- Almacenar copias de seguridad periódicas del trabajo realizado en discos duros externos y en la nube
Mala planificación	Error en la estimación de la duración de las tareas a realizar durante el proyecto	Medio	Alto	Riesgo mayor	- Consultar documentación y proyectos similares que puedan servir de apoyo para dar una correcta estimación - Revisar la planificación periódicamente para replanificar y corregir los desvíos a tiempo - Dejar algo de holgura en las estimaciones para contemplar retrasos sin que afecte a la fecha final del proyecto
Conocimientos insuficientes	Aumento del tiempo de desarrollo al afrontar nuevos retos que supongan un mayor tiempo de aprendizaje	Media	Medio	Riesgo significativo	- Planificar la fase de desarrollo teniendo en cuenta esta situación - Anticiparnos a estos retos documentándonos correctamente antes de afrontarlos

Ilustración 8. Plan de riesgos del proyecto

1.5 Breve resumen de productos obtenidos

Los productos finales obtenidos de este trabajo son:

- El sistema desarrollado formado la aplicación cliente y el backend, se entregará tanto el apk de la aplicación cliente como el código fuente de esta y del backend.
- Una memoria del Trabajo Fin de Máster, en la que se describe exhaustivamente todo el trabajo realizado.
- Anexos, como el manual de uso que detalla todas las funcionalidades de la aplicación, el manual de instalación donde se explicará cómo instalar la aplicación en un dispositivo Android, etc.
- Presentación del proyecto, que resume las partes más relevantes del mismo.
- Video de presentación, donde se explicará el trabajo desarrollado a partir de la presentación y se realizará una demo de la aplicación.

1.6 Breve descripción de los otros capítulos de la memoria

Los siguientes capítulos se corresponden con la fase de diseño, la etapa de implementación y la finalización del Trabajo Fin de Máster.

Como hemos dicho, el primer capítulo que encontramos se detalla la fase de diseño y está compuesto por dos apartados:

- En el primero capítulo nos centraremos en la metodología Diseño Centrado en el Usuario, donde se detallan los resultados de llevar a cabo cada una de sus cuatro fases: Usuarios y contexto de uso, Diseño conceptual, Prototipado y Evaluación.
- En el segundo se dedica al diseño técnico donde se definen y detallan los casos de uso que nos servirán para establecer las funcionalidades de la aplicación y el diseño de la arquitectura, identificando las entidades que se representarán en la base de datos, las clases que gestionarán los diferentes procesos y la estructura API que servirá para realizar las peticiones entre el cliente y el servidor.

A continuación, tenemos el capítulo de implementación que se centra en el producto a desarrollar, exponiendo las tecnologías y herramientas utilizadas en esta etapa, los detalles más relevantes del desarrollo llevado a cabo y las pruebas definidas para comprobar el correcto funcionamiento del sistema y su realización.

Posteriormente, encontramos un apartado de conclusiones en el que se describen las lecciones aprendidas, una reflexión sobre de la consecución de los objetivos y su explicación, un análisis crítico acerca del seguimiento de la planificación y la metodología seleccionada y las líneas de trabajo futuro.

En las últimas secciones tenemos un apartado de glosario donde se definen los términos y acrónimos más relevantes que se han utilizado en este documento, la bibliografía que incluirá todas las referencias utilizadas para este trabajo y los anexos complementarios a esta memoria.

2. Análisis, Diseño y Arquitectura

2.1 Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario (DCU) es una filosofía y un proceso de desarrollo que sitúa las necesidades y características del usuario en el centro de cada una de las etapas de diseño. El objetivo es la creación de productos que los usuarios encuentren útiles y usables, es decir, que satisfagan sus necesidades teniendo en cuenta sus características.

El proceso se estructura en una serie de etapas que suelen consistir en la investigación y análisis de los usuarios, el diseño y la evaluación. Se trata, además, de un proceso iterativo dado que cada fase se retroalimenta de la respuesta de los usuarios para mejorar y adaptar los elementos diseñados hasta el momento.

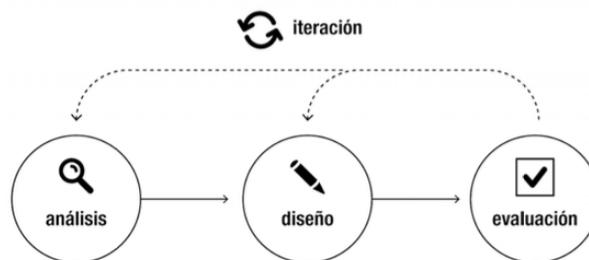


Ilustración 9. Flujo del DCU
(Fuente: DCU - UOC)

La aplicación del Diseño Centrado en el Usuario nos proporciona importantes beneficios como:

- Desarrollo más eficiente del producto, dado que la usabilidad es menos costosa y fácil de incorporar si se realiza desde las primeras etapas del desarrollo.
- Costes de soporte y documentación bajos, puesto que los productos usables son más sencillos de documentar y requieren un nivel bajo de soporte o ayuda.
- Reducir el tiempo y los costes de desarrollo, dado que solo se implementarán aquellas funcionalidades que el usuario necesita y de la forma que lo necesita.
- Reducir los costes de mantenimiento, pues se garantiza un desarrollo correcto a la primera.
- Incrementar la adopción y uso del sistema.
- Aumentar la satisfacción, la productividad y la eficiencia del usuario, puesto que los productos usables les permiten realizar sus tareas de manera más eficaz y eficiente.

2.1.1 Usuarios y Contexto de uso (Análisis)

Esta primera fase del DCU tiene como objetivo conocer las características de los usuarios, sus necesidades y objetivos. Para ello, se llevan a cabo los métodos de indagación que se basan en involucrar a los usuarios en las diferentes actividades y están orientados, como comentábamos, a la obtención de información para definir el producto o servicio.

El conocimiento sobre los usuarios, sus contextos de uso, sus necesidades, objetivos y actitudes son imprescindibles para un diseño centrado en el usuario y para desarrollar aplicaciones y entornos usables.

Los diferentes métodos de indagación se clasifican en función de la información que pueden recoger: Métodos Cualitativos (observación, entrevistas, etc.) y Métodos Cuantitativos (encuestas y cuestionarios). En las próximas secciones se detallarán los métodos de indagación realizados en este trabajo.

2.1.1.1 Entrevista en profundidad

El primer método llevado a cabo en esta etapa son las entrevistas en profundidad, que nos permite obtener información de tipo cualitativo. Con ello conseguimos una comprensión exhaustiva sobre las

necesidades, preferencias y experiencias con los productos o sistemas relacionados con este trabajo. Además, la ejecución de una fase cualitativa inicial nos ayudará a definir y realizar mejor los métodos cuantitativos que se aplicarán posteriormente.

2.1.1.1.1 Diseño y desarrollo de la entrevista

Para el desarrollo de este método se ha realizado un guion poco estructurado con cuatro apartados, que hemos considerado más relevantes en esta temática para conseguir comprender las necesidades, preferencias y experiencias de los entrevistados.

Guion	
1. Conocer al usuario	<ul style="list-style-type: none"> • Nombre y edad • Lugar de residencia, profesión y situación laboral • ¿Cuánto usas el móvil al día? ¿Dónde sueles utilizarlo? • ¿Cómo de fácil te resulta el uso de las aplicaciones móviles? • ¿Sueles puntuar y escribir comentarios de productos comprados o sitios que has visitado? • ¿Utilizas las valoraciones de otros usuarios cuando tomas decisiones?
2. Restricciones alimentarias	<ul style="list-style-type: none"> • ¿Tiene alguna restricción alimentaria? ¿Cuál? ¿A qué se debe? • ¿Cómo cambio esto tu vida? • ¿Cómo te sientes? ¿Cómo lo afrontas?
3. Momento de salir a comer fuera de casa	<ul style="list-style-type: none"> • ¿Con qué frecuencia comes fuera de casa en algún restaurante? • ¿Cómo lo compatibilizas con tus restricciones? ¿Te encuentras limitado/a? • ¿Sueles buscar sitios donde poder comer por internet? ¿Tienen éxito estas búsquedas?
4. Aplicación	<ul style="list-style-type: none"> • ¿Conoces alguna aplicación donde buscar sitios para comer? ¿Alguna está adaptada a tus necesidades? • ¿Usas alguna para este cometido? ¿Cuál? ¿Qué aspecto a mejorar de ellas destacarías? • ¿Te parece interesante una app en la que pudieras buscar establecimientos adaptados a tus restricciones para comer o tomar algo? ¿La utilizarías? • ¿Qué características o funcionalidades piensas que debería de incluir?

Ilustración 10. Guion de las entrevistas

El primer apartado se centra en conocer aspectos relacionados con el usuario y con la utilización que hace del dispositivo móvil (frecuencia, entorno y experiencia) y su comportamiento en la utilización de las aplicaciones y páginas webs en las que existe una comunidad de usuarios.

El siguiente apartado hace foco en las restricciones alimentarias del entrevistado y cómo esto le afecta en su vida diaria, para poder comprender mejor las necesidades de estas personas.

El tercer apartado se basa en el problema principal para estos usuarios, comer fuera de casa en algún establecimiento. Con ello se pretende saber la forma en la que estas personas afrontan esta problemática y cómo lo combinan con sus restricciones alimentarias.

El último apartado se dedica enteramente al tema de apps móviles relacionadas con esta temática para obtener información acerca de las aplicaciones que conocen o usan. Posteriormente, se les pide su opinión acerca de la idea del proyecto y las funcionalidades que consideran que debería presentar la aplicación.

Las entrevistas han transcurrido en un entorno cotidiano y distendido para que las personas participantes se sintieran cómodas y lo menos influenciadas posibles.



Ilustración 11. Entrevista en profundidad

El número de participantes de la entrevista han sido tres usuarios voluntarios, seleccionados entre conocidos con restricciones alimentarias que se encuentran con muchos de los problemas que giran en torno al tema del trabajo. Estas personas nos permitirán profundizar en el perfil de los posibles usuarios de nuestra aplicación y conocer sus necesidades, preferencias y experiencias.

Usando las notas tomadas durante cada entrevista podemos resumirlas de la siguiente manera:

- **Irene**

Chica de 28 años ingeniera informática que trabaja en una empresa de telecomunicaciones. Utiliza el móvil para el ocio alrededor de 1 o 2 horas al día cuando está en casa. Considera que tiene bastante experiencia con los móviles y normalmente se basa en las valoraciones de los usuarios para sus decisiones, pero no suele dejar comentarios.

Padece diferentes problemas digestivos que le hacen a seguir una dieta estricta sin gluten y esto limita en parte su vida social, en concreto, los planes de salir a comer (en ocasiones va al restaurante con sus amigos, pero no come nada). Todo ello la genera sensación de limitación e incompreensión por parte del sector de la hostelería.

Come fuera aproximadamente 1 vez a la semana, mirando en el restaurante al que acude si la carta tiene algún plato apto para su dieta. También realiza búsquedas de establecimientos donde poder comer, aunque no encuentra muchos sitios adaptados para ella.

No conoce apps específicamente adaptadas, pero sí Tripadvisor y utiliza El tenedor del que mejoraría el tema de las restricciones en las búsquedas. Le gusta la idea del trabajo y destaca características como buscar por posición o en una localización, ver la carta adaptada e información del restaurante, filtros de restricciones, puntuar los establecimientos y guardar los restaurantes favoritos.

- **Jose**

Varón de 58 años que trabaja en un centro de salud como enfermero en la localidad de Don Benito (Badajoz). Utiliza el móvil alrededor de 4 horas al día tanto en casa como en la calle. Generalmente se adapta bien a las aplicaciones, suele realizar valoraciones de los productos que compra o de los lugares que visita y presta atención a las del resto de usuarios antes de tomar una decisión.

No tiene ninguna restricción, pero su hijo es intolerante a la lactosa y por continuos problemas digestivos también sigue una dieta sin gluten. Esto hace que tengan que tener un mayor cuidado en la alimentación teniendo que preguntar en restaurantes la composición de los platos. Esta situación les provoca principalmente estrés.

No salen con mucha frecuencia a comer fuera, 1 vez cada 15 días más o menos, siempre conociendo los establecimientos donde van a comer, para que ofrezcan platos sin gluten y sin lactosa lo que les limita bastante sus posibilidades. Suelen realizar búsquedas cuando viajan o no están cerca de casa y estas no siempre tienen éxito.

Ha utilizado en alguna ocasión la app de Tripadvisor, en la que incluiría filtros para restricciones alimentarias, pero ninguna adaptada. Comenta que la aplicación objetivo de este trabajo les sería de gran utilidad y valoraría funcionalidades como buscar restaurantes de acuerdo a las restricciones, valorar y comentar los establecimientos por parte de los usuarios, guardar restaurantes en una lista y conocer la localización del establecimiento y verlo en un mapa.

- **Marina**

Chica de 27 años que trabaja de monitora escolar en Madrid. Realiza un uso diario del móvil de unas 3 horas cuando está en casa, en el transporte público y en la calle. Le resulta fácil utilizar las aplicaciones, suele realizar comentarios de los sitios que ha visitado y tiene en cuenta las aportaciones de los usuarios para decidir entre una opción u otra.

Es intolerante a la lactosa y celíaca por lo que lleva una dieta estricta sin gluten y sin lactosa, lo que le condiciona a la hora de comer fuera de casa o planear viajes teniendo que informarse sobre dónde puede comer. Esto le genera malestar, al no encontrar muchos establecimientos adaptados a estas necesidades cada vez más crecientes.

Suele comer en establecimientos una vez a la semana, pero siempre planeando previamente el sitio, sobre todo si es un lugar nuevo. Considera que es una limitación y realiza búsquedas, pero no logra encontrar restaurantes adaptados o no le muestran la información necesaria para sus restricciones.

Comenta que ha utilizado Tripadvisor y que le gustaría que los establecimientos mostraran su carta o si respetan sus restricciones. Sin embargo, no conoce ninguna app adaptada. La aplicación propuesta le parece una buena idea e indica que la utilizaría. En cuanto a sus características incluiría búsqueda de restaurantes cercanos o en una ciudad, filtros de restricciones tanto en comida como en bebida (Ej. Leche sin lactosa para un café), información acerca de si los establecimientos respetan sus restricciones y en qué platos, opiniones y valoraciones sobre los restaurantes.

Tanto el guion seguido en estas entrevistas como todas las notas recogidas durante las mismas pueden verse en el anexo I.

2.1.1.1.2 Conclusiones

Tras la realización de las entrevistas, hemos llegado a las siguientes conclusiones:

- La utilización del móvil suele darse en las viviendas y en la calle principalmente.
- Observamos una distinción en la utilización de las aplicaciones móviles o páginas webs: algunos usuarios son generalmente más activos realizando aportaciones y utilizando las valoraciones del resto para decidir mientras que otros se limitan únicamente a conocer la opinión del resto de usuarios para tomar una decisión.
- Existe una problemática real en torno a este tema que limita y condiciona la vida de las personas, provocando generalmente malestar e inquietud.
- Estas personas no suelen comer fuera con frecuencia debido a sus restricciones y a los pocos medios de los que disponen para encontrar establecimientos adaptados a sus necesidades, puesto que muchas de sus búsquedas no tienen éxito.
- No conocen ninguna aplicación adaptada a sus necesidades, pero sí reconocen a *Tripadvisor*, de la cual mejorarían toda la gestión de las restricciones alimentarias en torno a las búsquedas y a los establecimientos anunciados.
- Les gusta la idea del trabajo y afirman les sería de utilidad. En cuanto a las funcionalidades destacan: buscar por posición o en una ubicación, filtrar en función de las restricciones (tanto en comida como en bebida), informar de las restricciones que respeta el establecimiento, valorar y comentar los restaurantes y guardar los restaurantes favoritos.

2.1.1.2. Benchmarking

Esta técnica consiste en analizar productos similares o que compiten con el sistema interactivo que se está diseñando. Los objetivos de este método son conocer a los usuarios y sus expectativas, entender las tendencias de mercado, saber qué funcionalidades se valoran y cuáles no, estudiar las interfaces, etc.

En la introducción de este documento ya fue realizado este análisis competitivo en el que se estudiaron aplicaciones disponibles en el mercado orientadas hacia esta problemática y que disponían de funcionalidades semejantes. Toda la información obtenida nos será de gran ayuda tanto para el siguiente método de indagación como para el diseño de nuestra aplicación.

2.1.1.3 Encuestas

Este método de indagación se engloba dentro de las técnicas cuantitativas y nos permite obtener información de manera remota. Su principal ventaja se basa en que la información que se obtiene tiene una significación estadística. Sin embargo, en contraposición a los métodos cualitativos, la calidad y veracidad de la información recogida depende totalmente del diseño del cuestionario, puesto que es la herramienta que nos permite obtener la información de los usuarios.

Por ello, para definir correctamente las preguntas del cuestionario y sus posibles respuestas se han utilizado tanto los resultados obtenidos de la fase cualitativa inicial, llevada a cabo mediante las entrevistas, como la información recogida en la técnica de benchmarking o análisis competitivo. La herramienta utilizada para construir la encuesta ha sido *Google Forms* por su sencillez y potencia, dado que resulta muy fácil crear formularios y compartirlos con los participantes.

2.1.1.3.1 Diseño del cuestionario y desarrollo

El cuestionario consta de 14 preguntas divididas en cinco secciones, algunas de estas preguntas son condicionales y en función de la respuesta seleccionada se redirige hacia una sección u otra. Este cuestionario se encuentra disponible en la siguiente dirección: <https://goo.gl/forms/zCDBSR8ZiNTleela2>.

En la primera sección se pretende conocer a las personas participantes obteniendo información sobre aspectos socio-demográficos como su rango de edad y su lugar de residencia, el uso que realizan del móvil, aplicaciones y webs y sobre sus posibles restricciones alimentarias. En esta sección la pregunta 4 es clave, puesto que nos va a permitir conocer el uso que suele realizar el usuario de las aplicaciones que permiten interactuar con la comunidad de usuarios pudiendo clasificarlos en función de ello. La última pregunta también es importante para conocer si es el propio usuario quien tiene una restricción o alguien muy cercano. Además, se trata de una pregunta condicional en la que si el usuario no tiene ninguna restricción finaliza el formulario, con cualquier otra respuesta se avanza a la siguiente sección.

La segunda sección profundiza en el tipo de restricción que tiene la persona encuestada y se da la opción de elegir varias de ellas, así como añadir alguna otra que no se haya indicado.

En esta tercera sección se pregunta por la frecuencia en la que se realizan búsquedas de establecimientos para comer o tomar algo con el móvil. Esta pregunta es condicional y permite dirigir el flujo en función de la respuesta del usuario, si especifica que nunca ha realizado búsquedas se le redirige a la sección 5 y si indica alguna frecuencia entre el resto de opciones (*Rara vez, Ocasionalmente, Frecuentemente, Siempre*) continua a la siguiente sección.

Investigación: Restricciones alimentarias y establecimientos de hostelería

La presente encuesta pretende conocer más a fondo la problemática que afecta a las distintas personas que sufren restricciones alimentarias en su dieta por diferentes causas a la hora de comer fuera de casa.

***Obligatorio**

1. Por favor, introduce el rango de edad en el que te encuentras: *

Elige ▼

2. ¿En qué ciudad vives? *

Tu respuesta

3. ¿Cuánto tiempo dedicas al móvil al día aproximadamente? *

Elige ▼

4. Cuando utilizas aplicaciones y páginas webs, ¿Cuál es normalmente tu comportamiento? *

Suelo puntuar y comentar los restaurantes u hoteles que visito, los productos que compro, etc., así como utilizar las aportaciones del resto de usuarios para tomar una decisión.

No suelo puntuar o escribir comentarios, aunque generalmente sí utilizo las valoraciones que realizan otros usuarios para tomar una decisión.

Ninguna de las anteriores

5. ¿Tienes algún tipo de restricción alimentaria que condicione tu dieta por alguna causa? *

Sí

No

No, pero si un miembro de la familia o alguien muy cercano

SIGUIENTE Página 1 de 5

6. ¿Podrías indicar qué restricción alimentaria o tipo de dieta sigues? *



Sin Gluten



Sin Lactosa



Bajo en azúcares (apto para diabéticos)

Otro:

7. ¿Utilizas el móvil para buscar establecimientos que respeten restricciones alimentarias para comer o tomar algo? *

Elige ▼

Ilustración 12. Sección 1, 2 y 3 del formulario

La cuarta sección continúa con la temática de la anterior y tiene como objetivo conocer más a fondo las búsquedas que realizan los usuarios preguntando acerca del momento de la comida del día para la que buscaron y en qué lugar o entorno lo realizaron.

8. Cuando has realizado una de estas búsquedas, ¿para qué comida del día era? *

Desayuno

Tentempié o aperitivo

Comer

Merendar

Cenar

Otro:

9. ¿En qué lugar sueles realizar estas búsquedas con el móvil? *

En casa

En la calle

En el trabajo

En el transporte público

Otro:

Ilustración 13. Sección 4 del formulario

La sección 5 es la más larga de toda la encuesta y se centra en las aplicaciones existentes relacionadas y en la idea propuesta en este trabajo.

La primera parte nos servirá para conocer qué aplicaciones de esta temática conocen los usuarios, proporcionando algunas opciones obtenidas del benchmarking realizado y qué aspectos mejorarían de aquellas que usen.

La segunda parte es uno de los puntos más importantes de la entrevista y se centra en la idea de este trabajo, se comienza preguntando la opinión del participante acerca de una aplicación que lleve a cabo la idea planteada y se le pide que valore una serie de funcionalidades que hemos seleccionado como principales, a partir de nuestra definición y de las aportaciones de las personas entrevistadas. También se le da la opción de añadir nuevas funcionalidades que no han sido tenidas en cuenta. El objetivo de estas preguntas es conocer la posición del usuario sobre la idea y averiguar qué funcionalidades piensa que debería tener y cuáles son las que le aportan mayor valor.

10. ¿Conoces o usas alguna de estas aplicaciones para buscar establecimientos de hostelería que ofrezcan platos adaptados? Indica su nombre.

Tripadvisor

Celicity

Celiaquitos

Celiaco a los 30

Otro: _____

11. En caso de usar alguna aplicación para este cometido, ¿Destacarías algún aspecto a mejorar?

Tu respuesta _____

12. ¿Te parecería interesante o útil una aplicación para el móvil donde pudieras buscar restaurantes, bares y cafeterías donde comer o tomar algo de acuerdo a tus restricciones alimentarias? *

Sí

No

Tal vez

13. Valora del 1 al 5 las siguientes características de una aplicación que te permita buscar establecimientos que ofrezcan platos adecuados a tus restricciones *

	1 (Poco importante)	2	3	4	5 (Imprescindible)
Buscar establecimientos a partir de mi posición	<input type="radio"/>				
Buscar establecimientos en un lugar específico (Ej. En una ciudad)	<input type="radio"/>				
Mostrar información de los restaurantes	<input type="radio"/>				
Puntuar los restaurantes	<input type="radio"/>				
Escribir una opinión personal sobre el restaurante	<input type="radio"/>				
Subir fotos del restaurante	<input type="radio"/>				
Añadir nuevos establecimientos	<input type="radio"/>				
Guardar tus restaurantes favoritos	<input type="radio"/>				
Guardar los restaurantes a los que quieres ir	<input type="radio"/>				
Recomendar un restaurante a un amigo	<input type="radio"/>				

14. ¿Añadirías alguna otra funcionalidad a la aplicación? En caso afirmativo indícala.

Tu respuesta _____

Ilustración 14. Sección 5 del formulario II

Antes de lanzar la encuesta a los usuarios potenciales se realizó una pequeña prueba con cuatro usuarios para comprobar que las preguntas eran entendibles, que existen todas las respuestas posibles, que la duración es adecuada, etc. Esto nos permitió detectar y corregir algunos fallos como por ejemplo, en la pregunta 13 un usuario puntuó las funcionalidades solo con 1 o 2, debido a que la pantalla de su móvil solo mostraba estas puntuaciones y no se dio cuenta que deslizando existían más posibilidades. Para solucionarlo modificamos el enunciado, indicando “Valora del 1 al 5...”, para hacer consciente al usuario de la escala de puntuación.

Las personas participantes en la encuesta han sido seleccionadas, como en las entrevistas, entre las personas conocidas que presentan restricciones alimentarias y se encuentran con los problemas que ello conlleva, y, además, estos conocidos lo han difundido entre sus círculos a más personas que cumplieran estos requisitos.

Con todo ello, se ha conseguido una muestra de 20 personas que han cumplimentado la encuesta.

2.1.1.3.2 Resultados y conclusiones

En los resultados podemos observar cómo han participado personas con distintos rangos de edad, principalmente individuos entre 25-35 años, que residen en diferentes ciudades, sobre todo en Madrid y Don Benito (Badajoz).

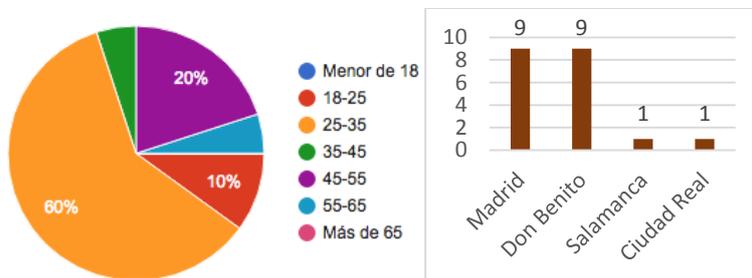


Ilustración 15. Edad y lugar de residencia de los encuestados

En lo que concierne a la utilización del dispositivo móvil, la encuesta refleja que es ampliamente utilizado por la mayoría de los usuarios. La mitad de los participantes lo utiliza entre 2 y 4 horas al día, un 30% de ellos más de 4 horas y las personas entre 45-55 años son los que menos uso hacen de él.

En las respuestas acerca del comportamiento al utilizar aplicaciones móviles y páginas webs, puede verse una clara división de los encuestados siendo el perfil predominante, como era de esperar, aquel que no suele interactuar con la comunidad pero sí utiliza las aportaciones del resto de usuarios para sus decisiones. Sin embargo, en este contexto, existe un elevado porcentaje de personas que suelen añadir valoraciones, además de utilizarlas, seguramente debido a que son conscientes de la limitación y dificultad que les provoca las restricciones alimentarias y por ello se muestran más dispuestos a ayudar a personas con los mismos problemas.



Ilustración 16. División según comportamiento en apps y webs

Entre nuestros encuestados el 90% de los usuarios tiene una restricción alimentaria y el 10% restante la tiene alguien de su familia o muy cercano. Las restricciones alimentarias se reparten de la siguiente forma:



Ilustración 17. Estadística de restricciones alimentarias

Casi todos los participantes utilizan el móvil para realizar búsquedas de establecimientos adaptados a sus restricciones con una mayor o menor frecuencia, aunque merece destacar que existe un 35% que indica que rara vez las realiza, pudiendo ser debido a los fracasos obtenidos en búsquedas anteriores y ante la poca información existente en la red como vimos en las entrevistas. Estas búsquedas son realizadas principalmente para buscar establecimientos para comer o para cenar, con un 72,2% y 66,7% de los

encuestados respectivamente, con menos asiduidad para desayunar y merendar, un 16,7%, y nunca se realizan para tomar un aperitivo o tentempié.

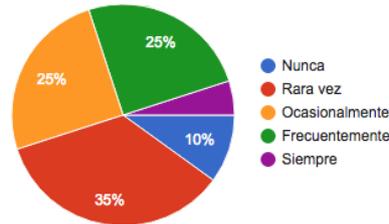


Ilustración 18. Frecuencia de búsqueda de establecimientos

La aplicación más conocida o usada es *Tripadvisor* y el aspecto a mejorar que más destacan sería poder filtrar las búsquedas en función de las restricciones, así como ofrecer mayor información del restaurante con respecto a este tema. En cuanto a aplicaciones adaptadas, algunos usuarios con dieta sin gluten han señalado que conocen o usan *Celicity* y, en menor medida, *Celiaquitos*, que son dos de las aplicaciones que fueron analizadas en el benchmarking. Así pues, parece que la mayoría de usuarios con restricciones no conoce ni utilizan una aplicación para buscar establecimientos que se adapten a sus necesidades.

La idea que se desarrolla en este trabajo ha tenido una gran aceptación con un 95% de los encuestados interesados en ella y únicamente una sola persona se mostró dubitativa. En cuanto a la valoración de las funcionalidades indicadas se han obtenido los siguientes resultados:

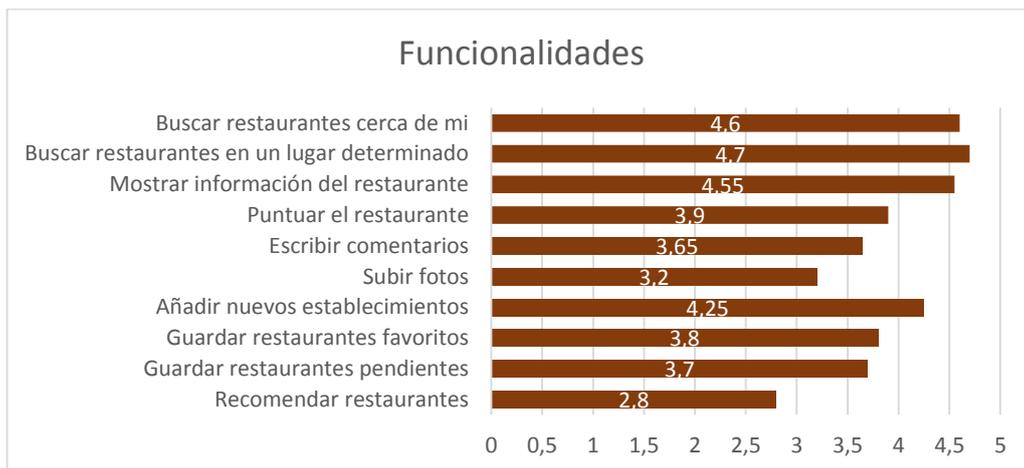


Ilustración 19. Valoración de las funcionalidades

Siendo lógico, las funcionalidades de la aplicación que la mayoría de usuarios consideran imprescindibles para una aplicación de este tipo son las búsquedas de establecimientos tanto a partir de la posición del usuario como en un lugar determinado y, por supuesto, mostrar información del restaurante. A continuación, encontramos la característica de añadir nuevos establecimientos siendo bien valorada por la mayoría de los usuarios y pudiendo destacar que todos aquellos que frecuentemente realizan aportaciones la han puntuado con la máxima nota posible.

En el resto de funcionalidades hay diversidad de opiniones, podemos observar que existe un grupo de funcionalidades valoradas por encima de 3,5 puntos, lo que demuestra su importancia, siendo ligeramente mejor valorada la funcionalidad de puntuar el establecimiento seguido de guardar los establecimientos (favoritos o pendientes) y escribir comentarios.

Por último, encontramos la funcionalidad de subir fotografías con una menor valoración (aunque se mantiene en la escala de 3 puntos) y la característica menos valorada por los encuestados ha sido la de recomendar restaurantes.

En la última pregunta acerca de funcionalidades adicionales solo el 25% de los encuestados sugirió alguna. Estas aportaciones se centran en: que se pueda filtrar por restricciones alimentarias, característica que considerábamos implícita en las búsquedas; que la información del restaurante sea completa, funcionalidad indicada en la pregunta anterior, y también que se proporcione información sobre la carta.

Por lo tanto, si analizamos con detalle estas tres últimas preguntas podemos llegar a la conclusión de que la aplicación que se plantea tendría una buena acogida entre los encuestados y de que las funcionalidades presentadas complacen las principales necesidades de los usuarios.

Tanto las preguntas de la encuesta como las respuestas de cada uno de los participantes pueden consultarse en el anexo II.

2.1.1.4 Perfiles de usuario

Los perfiles de usuario son agrupaciones de usuarios según sus características que pueden basarse en aspectos socio-demográficos, de expectativas, de uso, etc. Los diferentes perfiles de usuarios identificados en esta fase de análisis mediante las técnicas de indagación realizadas son:

Usuario Observador	
Características del perfil	<p>Este perfil estaría formado por personas que tienen algún tipo de restricción alimentaria y/o de personas que tienen algún familiar o alguien muy cercano con restricciones alimentarias, por ejemplo padres, madres o tutores de niños pequeños con restricciones.</p> <p>Estos usuarios tienen generalmente una edad comprendida entre 25 y 55 años, con experiencia en el uso de dispositivos móviles, utilizándolo entre 2 y 4 horas diarias o incluso más. Residen normalmente en ciudades que cuentan con distintas posibilidades de establecimientos donde comer y viajan con cierta frecuencia a otras ciudades.</p> <p>El objetivo fundamental de estos usuarios es encontrar un establecimiento que se adapte a sus restricciones alimentarias. La característica diferenciadora de este grupo se basa en su comportamiento al utilizar la aplicación. En general, este grupo son usuarios más pasivos que utilizan la aplicación para la consecución de su objetivo principal, buscar y elegir un establecimiento, teniendo en cuenta la información del mismo y las valoraciones del resto de usuarios para tomar una decisión. Por lo tanto, no suelen interactuar con la comunidad de usuarios compartiendo sus opiniones, subiendo fotografías, etc.</p>
Contexto de uso	<p>Los usuarios pertenecientes a este grupo utilizarán la aplicación cuando se encuentren en sus hogares y, mientras están fuera de casa, en la calle, aunque en alguna ocasión también lo harían desde el trabajo.</p> <p>Este perfil realiza búsquedas para prácticamente todas las comidas del día, de forma predominante para comer y cenar y con menos asiduidad para desayunar y merendar.</p>
Tareas	<ul style="list-style-type: none"> • Registrarse en la aplicación • Iniciar sesión en la aplicación • Buscar establecimientos por posición o en una ubicación geográfica • Acceder a la ficha de un restaurante y a las valoraciones del resto de usuarios • Guardar restaurantes favoritos o pendientes de visitar
Listado de características	<ul style="list-style-type: none"> • Poder buscar establecimientos en función de la posición del usuario o en una localización específica. • Distintos filtros de restricciones alimentarias (tanto de comidas como de bebidas) para las búsquedas. • Visualizar los resultados de las búsquedas en un mapa o lista. • Que la ficha de los restaurantes muestre información clara acerca de las restricciones alimentarias que respetan, así como la puntuación y comentarios de otros usuarios. • Añadir los establecimientos a favoritos.

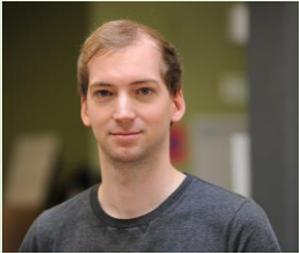
Usuario Colaborativo	
Características del perfil	<p>Igualmente, este grupo estaría formado por personas que tienen algún tipo de restricción alimentaria y/o de personas que tienen algún familiar o alguien muy cercano con restricciones alimentarias, por ejemplo padres, madres o tutores de niños pequeños con restricciones.</p> <p>Los usuarios de este perfil se ubican principalmente entre los 18 y los 55 años de edad, casi siempre con una amplia experiencia en el uso de dispositivos móviles, utilizándolo entre 2 y 4 horas diarias o incluso más de 4 horas. Residen normalmente en ciudades con variedad de establecimientos en los que poder comer y/o viajan con cierta frecuencia a otras ciudades.</p> <p>Este grupo de usuarios comparten el mismo objetivo principal que el perfil anterior, encontrar un establecimiento que respete sus restricciones alimentarias. Sin embargo, realizan un uso más extenso de las funcionalidades de la aplicación a diferencia del perfil anterior. Principalmente esta diferenciación se centra en la interacción con la comunidad de usuarios, puesto que estos usuarios, conscientes de las dificultades que les provocan estas restricciones, se muestran más dispuestos a ayudar a personas con los mismos problemas realizando distintas aportaciones como compartir nuevos establecimientos que conozcan o descubran, valorar establecimientos, escribir comentarios, etc.</p> <p>De igual modo, estos usuarios también utilizan las valoraciones del resto para elegir entre un establecimiento u otro, ya que les resulta de gran utilidad. De ahí su predisposición a realizar interacciones.</p>
Contexto de uso	<p>Este grupo de usuarios utilizarán la aplicación principalmente cuando se encuentren en casa y también, aunque en menor medida, mientras estén en la calle o viajando en el transporte público y, rara vez, en su lugar de trabajo.</p> <p>Este uso será principalmente para buscar establecimientos donde poder comer y cenar, y alguna ocasión aislada también para merendar.</p>
Tareas	<ul style="list-style-type: none"> • Registrarse en la aplicación • Iniciar sesión en la aplicación • Buscar establecimientos por posición o en una ubicación geográfica • Acceder a la ficha de un restaurante y a las valoraciones del resto de usuarios • Puntuar y comentar un restaurante • Subir fotos de su visita al restaurante • Guardar restaurantes favoritos o pendientes de visitar • Añadir un nuevo establecimiento
Listado de características	<ul style="list-style-type: none"> • Realizar búsquedas de establecimientos a partir de la posición del usuario o en una ubicación geográfica dada. • Filtros de restricciones alimentarias de comida y bebida para acotar las búsquedas. • Visualización de resultados de las búsquedas en forma de mapa o de lista. • Información de los restaurantes completa y bien organizada, sobre todo en el tema de las restricciones que respetan. • Especificar como mínimo al añadir un nuevo restaurante las restricciones alimentarias a las que se adaptan y su dirección. • Valorar los establecimientos visitados mediante una puntuación y un comentario. • Guardar establecimientos favoritos.

2.1.2 Diseño Conceptual

En esta fase se utiliza toda la información obtenida en la etapa anterior para sintetizar los datos, resumirlos y transmitirlos para, posteriormente, tomar decisiones de diseño atendiendo a los usuarios a quienes irá dirigida la aplicación móvil.

2.1.2.2 Personajes

Esta técnica consiste en describir una serie de personajes que detallan y precisan posibles usuarios de nuestra aplicación a partir de la información recogida en la fase anterior, para tener en cuenta al usuario en las diferentes etapas del proceso de diseño y desarrollo.

PERSONAJE 1	
 <p>Pedro Pérez (fuente)</p>	<p>Pedro tiene 29 años y trabaja como técnico para una empresa de fotocopiadoras en Valencia desde hace 4 años. Su trabajo diario consiste en realizar las revisiones y solucionar las incidencias de las máquinas que tiene la empresa en los diversos clientes.</p> <p>Respecto a su vida personal, vive con su pareja, Gloria, en la zona centro de Valencia, participa en una liga de pádel con su hermano y todas las semanas queda para ver los partidos del Valencia en su bar favorito con sus amigos.</p>
Restricción alimentaria	<p>Es intolerante a la lactosa desde hace año y medio cuando se lo diagnosticaron tras someterse a las pruebas, debido a que solía tener indigestiones cada vez que tomaba leche o alguno de sus derivados. Esto hace que lleve una dieta estricta sin lactosa, lo que le resulta fácil cuando come en casa debido a la gran cantidad de productos sin lactosa que existen, pero se encuentra con problemas cuando tiene que comer fuera por trabajo o planes de ocio.</p>
Experiencia digital	<p>Pedro se desenvuelve perfectamente con las nuevas tecnologías, puesto que forman parte de su trabajo diario. Le gusta estar actualizado tecnológicamente y siempre tiene el último modelo de Samsung. Utiliza bastante el móvil para el ocio, entre 2 y 4 horas al día, principalmente para jugar, realizar búsquedas por internet, utilizar las redes sociales, etc. Generalmente, al utilizar aplicaciones y webs no suele interactuar con la comunidad de usuarios puntuando productos, restaurantes u hoteles, escribiendo su opinión acerca de los mismos, etc. aunque sí le resultan de mucha utilidad las valoraciones dejadas por el resto de usuarios (perfil observador).</p>
Objetivos	<ul style="list-style-type: none"> • Buscar establecimientos adaptados a sus restricciones • Obtener información de los restaurantes y sus valoraciones • Guardar establecimientos favoritos
Necesidades	<ul style="list-style-type: none"> • Encontrar establecimientos cercanos donde poder tomarse un café o dulces sin lactosa mientras está trabajando. • Encontrar establecimientos que ofrezcan platos sin lactosa para comer o cenar con sus amigos/familia. • Conocer la información de los establecimientos buscados como, por ejemplo, su dirección para dirigirse a él, y la opinión de los usuarios para decantarse por uno. • Guardar los establecimientos que más le gustan para tenerlos a mano fácilmente.

PERSONAJE 2

Marisa Fernández ([fuente](#))

Marisa es una mujer de 42 años que trabaja como dentista en una clínica de la ciudad de Santander, está casada con un hombre de 44 años llamado Álvaro que, actualmente, trabaja en el ayuntamiento de la ciudad. El matrimonio tiene un hijo de 5 años llamado Daniel. Una las principales aficiones de la familia es viajar a conocer nuevos lugares.

Centrándonos en Marisa es una persona organizada y responsable que se ocupa de la mayoría de gestiones de la familia. También le gusta mantenerse activa y normalmente sale a correr con sus amigas dos veces en semana por el paseo marítimo.

Restricción alimentaria

El hijo de Marisa, Daniel, es celíaco desde pequeño y lleva una dieta estricta sin gluten. Cuando tenía 16 meses comenzó a presentar indigestiones, vómitos y una pérdida de peso considerable, al realizarle las pruebas oportunas los médicos le diagnosticaron celiaquía.

Actualmente, Marisa se encarga de todo lo relacionado con la alimentación de Daniel, asegurándose de que todos los productos que toma son libre de gluten.

En casa lo llevan bastante bien, ya que conocen muchos productos sin gluten y puede comprarlos en los supermercados. Sin embargo, tienen dificultades cuando viajan a la hora de encontrar restaurantes donde poder comer.

Experiencia digital

Marisa tiene un Huawei P9 Lite que le regaló su marido, el cual utiliza no utiliza mucho, aproximadamente entre 1 y 2 horas al día para las mismas aplicaciones (Gmail, Whatsapp, navegador, etc.) aunque generalmente se adapta bien a las nuevas apps. En el uso de aplicaciones móviles o webs con comunidades de usuarios, Marisa valora enormemente los comentarios y puntuaciones del resto apoyándose en ellas para tomar sus decisiones. Sin embargo, no suele realizar aportaciones propias principalmente por pereza y falta de tiempo (perfil observador).

Objetivos

- Buscar establecimientos adaptados a las restricciones de su hijo
- Obtener información de los restaurantes y sus valoraciones
- Guardar establecimientos preferidos

Necesidades

- Buscar establecimientos que ofrezcan platos libres de gluten para su hijo en las ciudades donde va a viajar próximamente.
- Conocer información de los restaurantes que cumplen sus requisitos y la opinión del resto de usuarios para decantarse por uno.
- Almacenar los establecimientos donde más ha disfrutado y la comida era exquisita.

PERSONAJE 3

Isabel López ([fuente](#))

Isabel tiene 26 años, estudió Ingeniería Química en la universidad de Granada y actualmente se encuentra trabajando en la universidad mientras realiza su tesis doctoral.

Vive en un piso compartido con tres amigas y es una chica muy extrovertida a la que le encanta realizar planes con sus amigos y con su pareja. También le gusta mucho practicar deporte y acude al gimnasio al menos dos veces en semana para mantenerse en forma.

Restricción alimentaria	Le diagnosticaron celiacía hace 3 años al realizarle pruebas para investigar las causas de unos continuos problemas digestivos. Desde ese momento sigue una dieta estricta libre de gluten mejorando de los problemas que presentaba. Poco a poco va conociendo nuevos productos sin gluten que sacan al mercado y nuevas recetas que prueba en casa, pero lo que más le cuesta es salir a comer fuera de casa.
Experiencia digital	Isabel utiliza su móvil Samsung Note 4 alrededor de 3 horas diarias para acceder a las redes sociales, realizar búsquedas por internet, consultar el email, etc. En general, se desenvuelve perfectamente en el uso del móvil y se adapta rápido a nuevas aplicaciones. En la utilización de app móviles o webs, suele leer las opiniones del resto de usuarios para informarse antes tomar una decisión y casi siempre colabora dejando sus propias valoraciones (perfil colaborativo). Estas aportaciones las realiza en mayor medida en el sector de la hostelería, dado que es consciente de cómo ha cambiado su vida la celiacía y las dificultades con las que se encuentra casi a diario. Por ello, siempre se muestra predispuesta a ayudar a los demás compartiendo sus experiencias.
Objetivos	<ul style="list-style-type: none"> • Buscar establecimientos adaptados a sus restricciones • Obtener información de los restaurantes y sus valoraciones • Valorar los establecimientos que ha visitado • Añadir nuevos restaurantes
Necesidades	<ul style="list-style-type: none"> • Buscar establecimientos que ofrezcan platos libres de gluten en función de su posición o en una ubicación geográfica. • Conocer la información asociada al restaurante y la opinión de los usuarios para decidir si ir o no. • Puntuar, comentar y subir fotos de los establecimientos que ha visitado para que otros usuarios puedan saber cómo es. • Añadir y compartir con el resto de usuarios nuevos restaurantes que conoce o ha descubierto y no están disponibles.

2.1.2.2 Escenarios de Uso

En este apartado se plantearán los escenarios de uso, que son descripciones de un personaje en una situación de uso del sistema con unos objetivos concretos. La construcción de estos escenarios nos ayudará a descubrir objetivos, funcionalidades y deseos de los usuarios en un contexto y situación determinada.

EU1 – Buscar un establecimiento adaptado a las restricciones del usuario cerca de su posición	
Perfil de usuario	Usuario Colaborativo, Usuario Observador
Personaje	Pedro Pérez
Contexto	Son las 11 de la mañana y Pedro acaba de terminar de solucionar una incidencia en una máquina fotocopidora de un cliente nuevo en Cullera. Se encuentra algo cansado ya que esta mañana no le ha dado tiempo a desayunar y lleva trabajando con la incidencia desde primera hora. Antes de volver a la oficina en Valencia, necesita tomarse un café sin lactosa para activarse, pero no conoce ningún bar en la zona que tenga leche sin lactosa.
Objetivo	Conocer los establecimientos que respeten sus restricciones alimentarias cerca de donde se encuentra.

Tareas	<ul style="list-style-type: none"> - Buscar los establecimientos cerca del usuario - Filtrar los resultados
Necesidades de información	Conocer qué filtros desea establecer
Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos a partir de la posición del usuario - Filtrar las búsquedas
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Seleccionar la opción de buscar establecimientos 4. Seleccionar su posición como dirección y los filtros para la búsqueda 5. Visualizar los resultados

EU2 – Seleccionar un establecimiento adaptado a las restricciones del usuario en una ubicación geográfica

Perfil de usuario	Usuario Colaborativo, Usuario Observador
Personaje	Marisa Fernández
Contexto	Marisa y su familia van a viajar a Bilbao el próximo sábado para conocer la ciudad. Marisa está planificando el viaje con los monumentos y museos que van a visitar, así como los establecimientos en los que van a comer. Para ello, necesita conocer qué restaurantes de Bilbao ofrecen platos sin gluten para Daniel.
Objetivo	Conocer los establecimientos que respetan las restricciones de su hijo en una determinada ubicación geográfica.
Tareas	<ul style="list-style-type: none"> - Buscar los establecimientos en una determinada ubicación geográfica - Añadir los filtros a la búsqueda
Necesidades de información	<ul style="list-style-type: none"> - Conocer la ubicación donde desea buscar - Saber qué filtros se quieren aplicar
Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos en una ubicación específica - Filtrar las búsquedas
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Seleccionar la opción de buscar establecimientos 4. Introducir el lugar donde desea buscar 5. Elegir los filtros que desea incluir en la búsqueda 6. Visualizar los resultados

EU3 – Obtener información de un establecimiento buscado

Perfil de usuario	Usuario Colaborativo, Usuario Observador
Personaje	Marisa Fernández
Contexto	Marisa ha encontrado diferentes restaurantes en Bilbao que ofrecen platos sin gluten para que Daniel pueda comer. Sin embargo, no tiene ninguna información acerca de ellos para poder seleccionar el más adecuado o contactar con ellos para preguntarles algunas dudas.
Objetivo	Obtener la información acerca del establecimiento seleccionado como su dirección, teléfono, valoración, comentarios y fotos de los usuarios, etc.
Tareas	<ul style="list-style-type: none"> - Buscar establecimientos en la app - Acceder a la ficha de un establecimiento
Necesidades de información	Saber buscar y filtrar establecimientos en la app

Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos a partir de la posición del usuario o en una ubicación específica - Filtrar las búsquedas - Mostrar información de un establecimiento
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Buscar establecimientos 4. Acceder a un establecimiento 5. Consultar la información

EU4 – Añadir un nuevo establecimiento

Perfil de usuario	Usuario Colaborativo
Personaje	Isabel López
Contexto	Isabel ha visitado un restaurante adaptado a su dieta estricta sin gluten que han abierto nuevo cerca de la universidad. Le ha gustado tanto el restaurante que entra a la aplicación para valorarlo y que otros usuarios conozcan su opinión acerca del mismo. Sin embargo, al acceder ha podido comprobar que el restaurante no existe en la aplicación.
Objetivo	Añadir a la aplicación un nuevo establecimiento que el usuario conoce o ha descubierto
Tareas	Añadir un establecimiento con su información correspondiente
Necesidades de información	Conocer el establecimiento que se desea añadir y la información complementaria que se requiere, como dirección, tipos de restricciones aceptadas, etc.
Funcionalidades	Añadir un nuevo restaurante, bar o cafetería
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Seleccionar la opción de añadir establecimiento 4. Rellenar toda la información necesaria para crear el establecimiento 5. Crear el establecimiento

EU5 – Valorar un establecimiento (puntuar y escribir un comentario)

Perfil de usuario	Usuario Colaborativo
Personaje	Isabel López
Contexto	Isabel y su pareja han celebrado su aniversario cenando en un restaurante de categoría que ofrecía platos adaptados a sus restricciones. Ante el exquisito trato por parte del servicio y los espectaculares platos degustados, Isabel quiere compartir su experiencia y valorar el establecimiento dándoles una excelente puntuación y comentando la calidad de los platos.
Objetivo	Valorar su experiencia en el restaurante mediante una puntuación y un comentario acerca de la visita.
Tareas	<ul style="list-style-type: none"> - Obtener la información de un establecimiento buscado - Puntuar el establecimiento - Escribir un comentario del establecimiento

Necesidades de información	<ul style="list-style-type: none"> - Saber buscar y filtrar establecimientos en la app - Saber acceder a la información de un establecimiento - Conocer la puntuación que se quiere dar al establecimiento - Conocer la opinión que se tiene del establecimiento
Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos a partir de la posición del usuario o en una ubicación específica - Filtrar las búsquedas - Mostrar información de un establecimiento - Puntuar un establecimiento - Escribir un comentario de un establecimiento
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Buscar establecimientos 4. Acceder a un establecimiento 5. Valorar el establecimiento 6. Escribir la opinión personal acerca del establecimiento 7. Publicar la valoración

EU6 – Subir una fotografía de la visita a un establecimiento

Perfil de usuario	Usuario Colaborativo
Personaje	Isabel López
Contexto	Isa va a merendar con unas amigas en una pastelería sin gluten que encontró en la aplicación, después de que una nutricionista famosa la compartiera. En el establecimiento el camarero les surgiere un nuevo dulce que han sacado, Isabel le hace una foto y queda encantada ante la explosión de sabor. Tras su visita al establecimiento Isabel decide compartir la fotografía del nuevo postre en una valoración a través de la aplicación.
Objetivo	Subir una fotografía de la visita del usuario al establecimiento en una valoración
Tareas	<ul style="list-style-type: none"> - Obtener la información de un establecimiento buscado - Valorar establecimiento - Subir la fotografía de la visita al establecimiento
Necesidades de información	<ul style="list-style-type: none"> - Saber buscar y filtrar establecimientos en la app - Saber acceder a la información de un establecimiento - Conocer la valoración que se quiere dar al establecimiento - Disponer de la fotografía que se quiere subir
Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos a partir de la posición del usuario o en una ubicación específica - Filtrar las búsquedas - Mostrar información de un establecimiento - Valorar un establecimiento - Añadir una fotografía de la visita al establecimiento
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Buscar establecimientos 4. Acceder a un establecimiento 5. Valorar el establecimiento 6. Puntuar y comentar el establecimiento 7. Seleccionar la fotografía que se quiere añadir 8. Publicar la valoración

EU7 – Guardar un establecimiento como favorito o como pendiente de visitar	
Perfil de usuario	Usuario Colaborativo, Usuario Observador
Personaje	Pedro Pérez
Contexto	El sábado por la tarde Pedro está tomando unas cervezas con una pareja de amigos. En un momento dado el camarero pasa con unas tortitas con chocolate en la bandeja, ante lo cual Pedro lanza un suspiro de decepción al saber que no puede comerlas. La pareja al ver esta situación recuerda una pastelería en la que tienen todo tipo de dulces para intolerantes a la lactosa y se la recomiendan a Pedro, que la busca en la aplicación viendo que tiene muy buenos comentarios y puntuación.
Objetivo	Guardar un establecimiento como favorito o como pendiente de visitar
Tareas	<ul style="list-style-type: none"> - Buscar el establecimiento - Obtener la información de un establecimiento buscado - Guardar como favorito o como pendiente de visitar
Necesidades de información	<ul style="list-style-type: none"> - Saber buscar y filtrar establecimientos en la app - Saber acceder a la información de un establecimiento - Conocer qué establecimiento se quiere guardar como favorito o como pendiente de visitar
Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos a partir de la posición del usuario o en una ubicación específica - Filtrar las búsquedas - Mostrar información de un establecimiento - Guardar establecimiento como favorito o como pendiente de visitar
Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Buscar establecimientos 4. Acceder a un establecimiento 5. Seleccionar en la ficha del establecimiento como favorito o pendiente de visitar

EU8 – Recomendar un restaurante	
Perfil de usuario	Usuario Colaborativo, Usuario Observador
Personaje	Marisa Fernández
Contexto	Marisa está tomando café con su amiga Ana contándole su viaje familiar por Bilbao y, en especial, lo bien que comieron en el restaurante que encontró a través de la aplicación. Ana que también es celíaca como Daniel, el hijo de Marisa, se interesa por el restaurante con la intención de visitarlo algún día, puesto que sus abuelos viven en un pueblo cercano a Bilbao, y le pide a Marisa si le puede enviar el restaurante al que fueron.
Objetivo	Recomendar un establecimiento a un amigo
Tareas	<ul style="list-style-type: none"> - Buscar el establecimiento - Obtener la información de un establecimiento buscado - Recomendar el establecimiento
Necesidades de información	<ul style="list-style-type: none"> - Saber buscar y filtrar establecimientos en la app - Saber acceder a la información de un establecimiento - Conocer que establecimiento se quiere recomendar y a que contacto
Funcionalidades	<ul style="list-style-type: none"> - Buscar establecimientos a partir de la posición del usuario o en una ubicación específica - Filtrar las búsquedas - Mostrar información de un establecimiento - Recomendar el establecimiento a un amigo

Desarrollo de tareas	<ol style="list-style-type: none"> 1. [Registrarse] 2. Iniciar sesión en la aplicación 3. Buscar establecimientos 4. Acceder a un establecimiento 5. Elegir al contacto destino de la recomendación 6. Enviar recomendación
-----------------------------	---

2.1.2.2 Flujo de Interacción

Los flujos de interacción nos permiten obtener una visión global de la interacción, ya que nos muestran de manera gráfica las decisiones y caminos que seguirán los usuarios para lograr sus objetivos. Su construcción también nos permite detectar necesidades funcionales o de información de los usuarios en cualquiera de los puntos.

En la siguiente imagen podemos apreciar el flujo de interacción que nos muestra la estructura general de la aplicación:

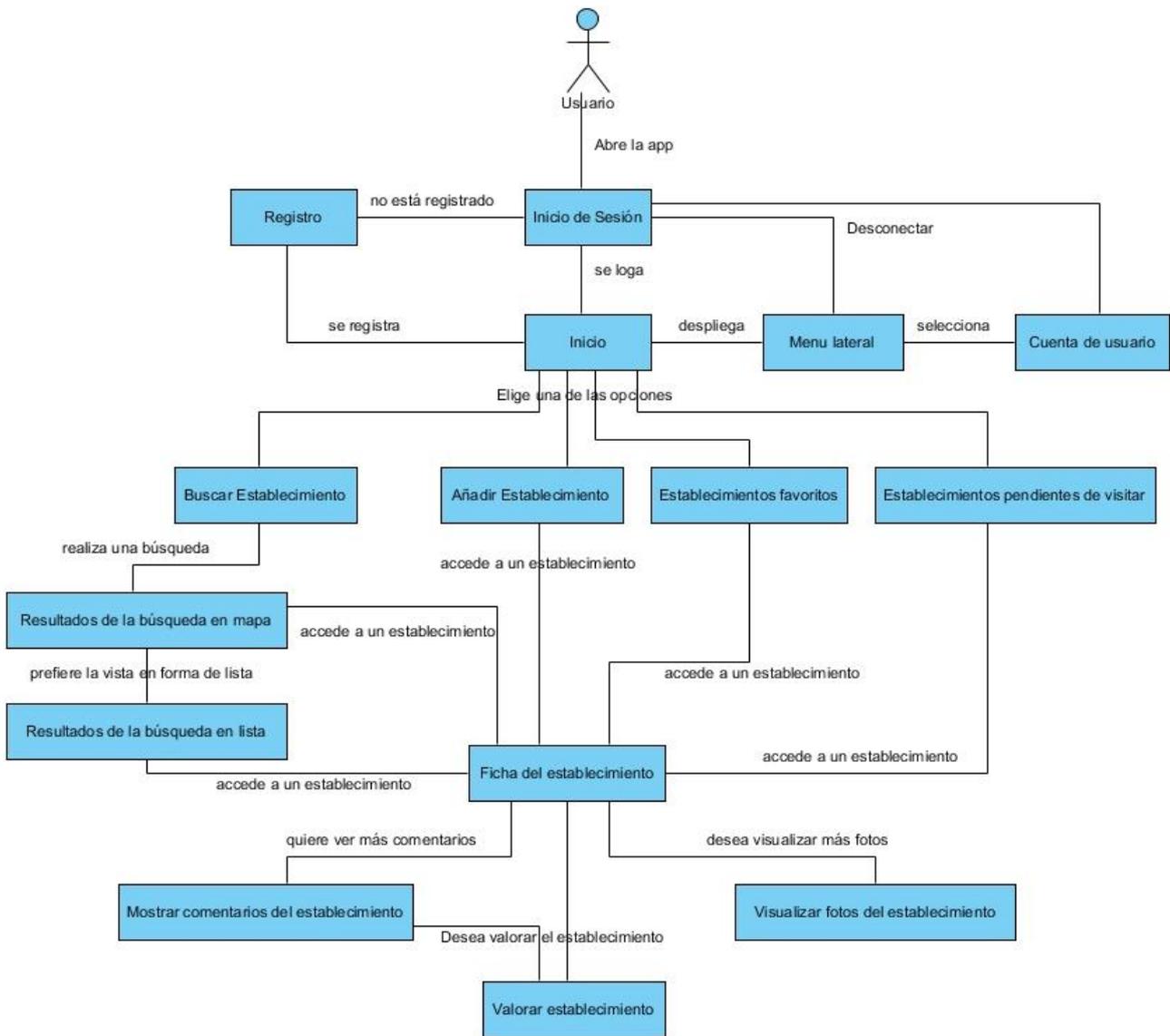


Ilustración 20. Flujo de interacción

2.1.3 Prototipado

En esta tercera fase del DCU se lleva a cabo la técnica del prototipado que consiste en la construcción de uno o diferentes modelos del sistema que se diseña. Esto nos permite comunicar decisiones de diseño y evaluar los modelos antes de desarrollar el producto final.

2.1.3.1 Sketches

Los Sketches son prototipos de baja fidelidad realizados a mano alzada que representan de forma esquemática el producto interactivo o el diseño de la interfaz. Su objetivo es proporcionar una primera idea de cómo será la interfaz, la disposición de sus elementos y la visibilidad que deben tener.

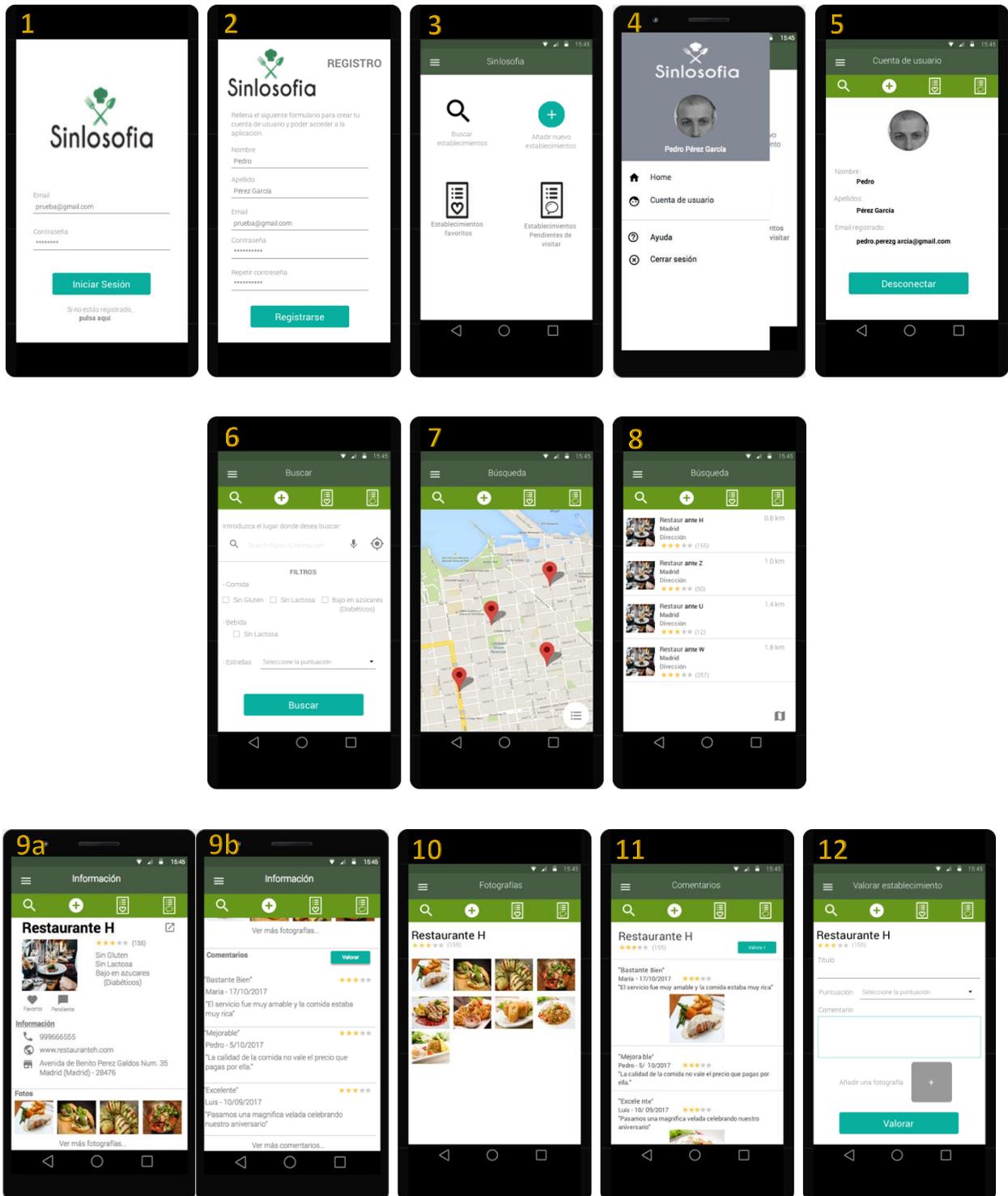


Ilustración 21. Sketches

2.1.3.2 Prototipos horizontales de alta fidelidad

En esta sección se elaboran prototipos de alta fidelidad que tratan de construir un modelo lo más próximo posible al sistema que se diseña y se desarrolla para evaluar de manera más precisa aspectos funcionales y de usabilidad. En concreto, en los prototipos horizontales este modelo abarca todas o casi todas las funcionalidades, espacios y menús del producto, sin que estén implementadas y funcionen realmente.

A continuación, se presenta el prototipo horizontal de alta fidelidad de la aplicación que hemos realizado con la herramienta *Prototyper* de *Justinmind*:



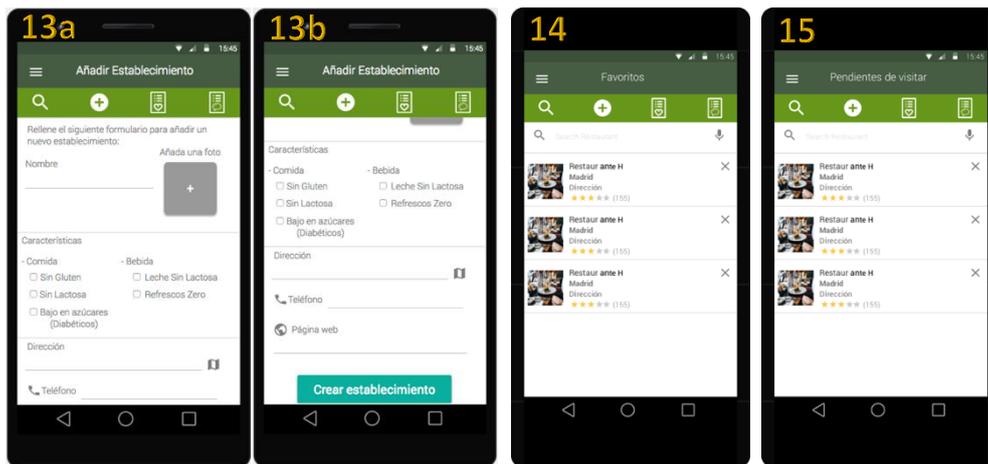


Ilustración 22. Prototipo horizontal de alta fidelidad

A continuación, se muestra una breve descripción del diseño de cada pantalla:

1. **Inicio de sesión:** se trata de la pantalla inicial donde los usuarios pueden logarse en la aplicación con sus credenciales.
2. **Registro:** cuando un usuario no se encuentra registrado, esta pantalla posibilita a los nuevos usuarios registrarse en el sistema rellenando un formulario básico.
3. **Inicio:** es la home de la app una vez el usuario ha iniciado sesión, en ella se muestran de forma clara y sencilla al usuario las opciones disponibles en la aplicación. En el resto de pantallas se incluirá en la parte superior un panel de acceso rápido a cada una de estas opciones para poder acceder desde cualquier punto de la app.
4. **Menú lateral:** está disponible en toda la aplicación desde que el usuario inicia sesión y nos ofrece un acceso rápido a la home, a los datos de la cuenta de usuario y la ayuda, así como la posibilidad de cerrar la sesión activa del usuario.
5. **Cuenta de usuario:** esta pantalla muestra al usuario los datos asociados a su cuenta, es decir, aquellos que introdujo cuando se registró.
6. **Buscar establecimiento:** esta pantalla recoge la funcionalidad principal de la aplicación. Se compone de un buscador en el que se indica la dirección donde se desea buscar establecimientos (el usuario puede especificar tanto una localización determinada como su posición actual) y un conjunto de filtros sobre restricciones alimentarias que deberán respetar los restaurantes resultado de la búsqueda.
7. **Resultados de la búsqueda en mapa:** muestra los establecimientos que cumplen los requisitos de la búsqueda del usuario en un mapa, de forma que el usuario puede visualizar de un vistazo los establecimientos disponibles en esa zona.
8. **Resultados de la búsqueda en lista:** muestra los establecimientos que cumplen los requisitos de la búsqueda del usuario en una lista, permitiendo al usuario visualizar los establecimientos de manera más ordenada y con algo más de detalle.
9. **Ficha del establecimiento:** en esta página se mostrará toda la información acerca del restaurante y se permitirá realizar diversas acciones como añadir el establecimiento a la lista de favoritos o a la lista de pendientes de visitar, recomendar a un amigo o valorar el establecimiento. También se mostrará un extracto de las últimas fotos de los usuarios y los últimos comentarios únicamente para que no tarde mucho en cargar, aunque se permite al usuario acceder a la información completa.
10. **Mostrar comentarios del establecimiento:** en esta página se mostrará en la parte superior el nombre del establecimiento y su puntuación para ubicar en la aplicación al usuario en todo momento y, a continuación, los comentarios de los usuarios acerca del establecimiento. Para reducir el tiempo de carga de la página se utilizará una lista dinámica en la que solo se cargarán las valoraciones del establecimiento visibles en la pantalla y según el usuario explore hacia abajo en la lista se cargarán nuevas valoraciones.
11. **Visualizar fotografías del establecimiento:** como en la página anterior, se mostrará en la parte superior el nombre del establecimiento y su puntuación para ubicar en la aplicación al usuario en todo momento y, posteriormente, las fotografías del establecimiento que los usuarios han subido. Se

empleará la misma técnica para reducir el tiempo de carga de la página, pero en este caso, se utilizará un *grid* dinámico en el que solo se cargarán las fotografías visibles en la pantalla y según el usuario navegue hacia abajo en el *grid* se cargarán las nuevas fotografías.

12. **Valorar establecimiento:** esta pantalla permite a los usuarios añadir valoraciones de los establecimientos que hayan visitado, indicando la puntuación del mismo, un título, el comentario y, opcionalmente, una fotografía.
13. **Añadir establecimiento:** esta pantalla permite a los usuarios añadir nuevos establecimientos introduciendo los datos del mismo a través de un formulario, una vez creado se redirige a la página del establecimiento.
14. **Establecimientos favoritos:** esta pantalla muestra la lista de los establecimientos que los usuarios han marcado como favoritos y añade un buscador en la parte superior para encontrar más rápidamente el establecimiento deseado.
15. **Establecimientos pendientes de visitar:** en esta opción la pantalla es idéntica a la anterior solo difiere en los datos que presenta, establecimientos marcados como pendiente de visitar.

Este prototipo se encuentra actualizado con las mejoras obtenidas en la siguiente fase, como parte del proceso iterativo de DCU.

2.1.4 Evaluación

Esta fase nos permite obtener información sobre los diseños realizados y así poder mejorarlos. De esta forma, podemos evaluar los diseños y corregir los errores de manera iterativa.

En el desarrollo del prototipo se ha ido evaluando tanto el diseño como la navegación de las pantallas, con el fin de obtener los mejores resultados. Para ello, hemos utilizado la técnica de test con usuarios.

2.1.4.1 Test de Usuarios

El objetivo principal de la realización del test de usuarios es evaluar el prototipo de la aplicación para poder conocer con un elevado nivel de detalle, la facilidad de uso, el nivel de eficacia, la eficiencia y la satisfacción del usuario.

2.1.4.1.1 Perfil socio-demográfico y documento de *Screening*

El perfil socio demográfico de los usuarios a los que vamos a realizar el test es el siguiente:

- Personas de cualquier género que padecen algún tipo de restricción alimentaria o son responsables de alguien que tiene una restricción.
- Edad comprendida entre los 18 y los 65 años, dado que son los usuarios potenciales de nuestra aplicación.
- Que residan en una localidad de más de 20.000 habitantes, para que exista cierta variedad de restaurantes, bares y cafeterías.
- De nacionalidad española, dado que la aplicación solo estará disponible en español.
- Con un uso del dispositivo móvil al menos básico (correo electrónico, Whatsapp, alguna red social, etc.) para que sea capaz de desenvolverse con cierta soltura.

El documento de "*Screening*" nos permitirá elegir a los usuarios cuyo perfil socio-demográfico cumpla los criterios anteriormente descritos y seleccionarlos para la realización del test.

Datos del usuario							
Género:	Hombre			Mujer			
Restricción alimentaria:							
Edad:	<18	18-25	26-35	36-45	46-55	55-65	>65
Población:							
Nacionalidad:							
Uso del móvil:	Email	Redes sociales		Whatsapp	No usa		

Ilustración 23. Documento de *Screening*

2.1.4.1.2 Cuestionario Pre-test

Previo a la realización del test, los usuarios realizarán un cuestionario para obtener datos socio-demográficos que nos ayudarán en el proceso de análisis de los resultados.

Cuestionario Pre-test	
➤	Género:
➤	Población:
➤	¿Qué tipo de restricción tiene usted o la persona de la cual es responsable?
➤	¿Qué edad tiene?
➤	¿Cuál es su profesión?
➤	¿Cuántas horas utilizas el móvil cada día para ocio? ¿Y a la semana?
➤	¿En qué lugares sueles utilizarlo? Ej. En casa, en la calle, en el transporte público, etc.
➤	¿Para qué lo utilizas principalmente?
➤	¿Con qué frecuencia utilizas el móvil para realizar gestiones por internet?
➤	¿Sueles puntuar los restaurantes u hoteles que has visitado, los productos que has comprado, etc.? ¿Y escribir comentarios acerca de ellos?
➤	¿Utilizas normalmente la puntuación y los comentarios de productos, hoteles, restaurantes, etc. que dejan otros usuarios para tomar tus decisiones?
➤	¿Te gusta salir a comer o cenar fuera de casa? ¿Con que frecuencia lo haces?
➤	¿Realizas búsquedas de establecimientos adaptados a tus restricciones a través del móvil? ¿Utilizas alguna aplicación? ¿Cuál?

Ilustración 24. Cuestionario Pre-test

2.1.4.1.3 Escenario y tareas

A continuación, describiremos el escenario y las tareas que los usuarios deberán resolver durante el test de usuarios. Estas tareas abarcan las funcionalidades principales de la aplicación.

Escenario

Ayer fuiste a cenar a tu restaurante favorito “El Museo”, el cual respeta tus restricciones alimentarias, con tu pareja para celebrar vuestro aniversario. Esta mañana te ha llamado tu amiga María, que tiene intolerancia a la lactosa, para contarte que ha encontrado una aplicación que permite buscar establecimientos que se adapten a tus restricciones alimentarias, aunque es nueva y todavía no tiene una gran cantidad de establecimientos (entre ellos tu restaurante favorito). Cuando llegas a casa, harto/a de la dificultad que tienes siempre para comer fuera debido a tu restricción alimentaria, decides descargar la aplicación para probarla puesto que el sábado por la noche quieres ir a cenar con tus amigos a algún sitio nuevo.

Este escenario consta de las siguientes tareas:

1. Añade a la aplicación tu restaurante favorito “El Museo” para que el resto de usuarios pueda conocerlo. Este restaurante ofrece platos sin gluten y sin lactosa, está ubicado en la calle Toledo número 6 de Zaragoza (28123) y su número de teléfono es 999222333.
2. Valora tu restaurante favorito que acabas de añadir, dejando una opinión que incluya la puntuación que creas que tiene, un comentario sobre tu visita de ayer y la fotografía del postre que te encanta.
3. Realiza una búsqueda de los establecimientos que ofrecen platos adaptados a tus restricciones en su menú que se encuentren en la ciudad de Barcelona con una valoración mínima de 3 estrellas y accede a la información de uno cualquiera.
4. Guarda el restaurante al que has accedido en la lista de establecimientos favoritos de la aplicación y visualiza dicha lista.

2.1.4.1.4 Cuestionario Post-test

Tras la realización del test por parte del usuario le dejaremos unos minutos para que nos exprese su experiencia con la aplicación mediante un formulario que nos permitirá evaluar el grado de satisfacción del mismo.

Cuestionario Post-test					
	1	2	3	4	5
➤ ¿Crees que ha sido sencillo utilizar la aplicación?					
➤ ¿Qué impresión te ha causado la aplicación?					
➤ ¿En algún momento no has sabido en qué lugar de la aplicación te encontrabas?					
➤ ¿La interfaz de la aplicación te ha ayudado a conseguir tus propósitos?					
➤ ¿Has conseguido realizar las tareas?					
➤ ¿Qué tarea te ha resultado más costosa? ¿Por qué?					
➤ ¿Qué te ha parecido el diseño gráfico de la aplicación?					
➤ ¿Qué es lo que más te ha gustado de la aplicación? ¿Y lo que menos?					
➤ ¿Si pudieras cambiar algo de la aplicación que sería?					
➤ ¿Añadirías o eliminarías alguna funcionalidad? ¿Cuál/es?					
➤ ¿Usarías la aplicación para buscar un establecimiento que se adapte a tus restricciones?					

Ilustración 25. Cuestionario Post-test

2.1.4.1.5 Desarrollo y conclusiones

Para la realización de este test hemos utilizado laboratorios portátiles de usabilidad, adaptando habitaciones comunes para convertirlas en nuestro laboratorio. Esto nos ha proporcionado beneficios como: una mayor autonomía, dado que es más fácil de organizar el test al no depender de una sala específica; ahorrar costes al no necesitar un laboratorio estable; comodidad para el usuario y adaptación al entorno y contexto, ya que se trata de uno de los entornos reales de uso de nuestro producto.

En el desarrollo de este método he realizado la función de facilitador conduciendo el test y acompañando al usuario en todo momento, así como el rol de evaluador obteniendo datos durante la realización del test y con el análisis de resultados.

Por último, el guion empleado para realizar el test de usuario es el siguiente:

- Al comienzo del test de usabilidad me he presentado al usuario, intentando conversar con él para tranquilizarle, explicándole que no vamos a evaluarle a él sino a la aplicación y que no se preocupe si comete cualquier error puesto que esto significaría que la app/prototipo no está correctamente diseñada.
- Posteriormente también le explico en qué consiste el test de usuario y las partes que tiene (pre-test, tareas, post-test, etc.), así como el protocolo de pensamiento manifiesto que debe realizar durante el test poniéndole algún ejemplo.
- Después comenzamos con el test, lo primero que se le pide al usuario es que responda al cuestionario pre-test explicándole que se trata de unas preguntas para obtener más información acerca de él: preferencias, comportamientos, etc.
- Una vez finalizado este cuestionario se le indica cuál es el producto que vamos a analizar y se da a conocer al usuario el escenario y la primera de las tareas. Cuando la finalice o no sea capaz de continuar se pasa a la siguiente hasta finalizar todas.
- Tras la realización de las tareas se pide al usuario que rellene el cuestionario post-test para conocer su opinión acerca de la aplicación que ha usado y cuales han sido sus percepciones y sus sensaciones.
- Por último, se le pide al usuario que firme la hoja (carta de autorización) en la que da su consentimiento para poder almacenar y tratar los datos recogidos durante el test, con el objetivo de mejorar el diseño y la usabilidad de la aplicación.

El test ha sido realizado a una persona seleccionada entre mis conocidos que tienen una restricción alimentaria y se enfrentan con la problemática sobre la que pretende ayudar la aplicación.

A partir del documento *Screening* y del cuestionario Pre-test podemos obtener el siguiente perfil del usuario:

- Se trata de una chica de 27 años, enfermera de profesión que actualmente vive en Madrid y padece intolerancia a la lactosa y celiaquía. Tiene experiencia en el uso de dispositivos móviles, utilizándolo para temas de ocio unas 3 horas diarias en cualquier sitio, aunque principalmente en casa o en la calle. Estas horas las dedica a hablar por Whatsapp, acceder a redes sociales, mirar el correo electrónico y realizar búsquedas por internet. Suele realizar gestiones por internet con frecuencia y tiene un perfil colaborativo (se fija en las valoraciones del resto de usuarios y deja las suyas propias). Es una chica activa con pasión por hacer planes que incluyen comer o cenar fuera de casa pero tiene muchas dificultades debido a sus problemas. También suele realizar búsquedas de establecimientos y no utiliza ninguna aplicación.

A continuación, se muestra la tabla cumplimentada por el evaluador en la que se especifica el porcentaje de éxito para cada tarea y los comentarios y observaciones tomados:

Tareas	% Éxito	Comentarios
1	100%	- El usuario ha dudado un poco en la pantalla de añadir establecimiento, puesto que, al ser un prototipo y realizar el test con un PC, no se había dado cuenta de que la página tenía scroll y no veía el botón de añadir. Finalmente se ha dado cuenta de esto y ha finalizado la tarea correctamente.
2	100%	- El usuario ha intentado adjuntar la foto pulsando sobre el botón de añadir fotografía pero, como es un prototipo y carece de funcionalidad, ha indicado en voz alta correctamente lo que esperaba que ocurriera.
3	100%	- El usuario ha mostrado al navegar entre algunas pantallas signos de dudas y ha indicado que se encuentra un poco perdido, pero posteriormente ha reconducido la situación al volver al inicio de buscar.
4	100%	- El usuario no sabía si pulsar en el acceso rápido a la lista de favoritos o en el botón de añadir el establecimiento a favoritos. Nos indica que el botón le confunde porque es el mismo. - El usuario también nos ha expresado que se encuentra algo desorientado. - El usuario al no ver ningún buscador en la lista de favoritos nos ha destacado que si tuviera muchos favoritos tendría que ir buscando uno a uno en la lista.

Ilustración 26. Comentarios de las tareas realizadas

*En la labor de evaluación hemos tenido en cuenta que el test ha sido realizado con un prototipo sin funcionalidad real por lo que algunas acciones no eran posibles de realizar directamente (Ej. Elegir la fotografía a subir). Por ello, hemos dado mucho valor a los comentarios que el usuario nos indicaba con el pensamiento manifiesto, además de sus acciones.



Ilustración 27. Test de usuarios

Del cuestionario post-test realizado por este usuario podemos obtener las siguientes conclusiones:

- En general, la aplicación le ha resultado sencilla y le ha causado una buena impresión. Indica que la interfaz le ha ayudado pero tiene margen de mejora (4/5), valora el diseño gráfico y destaca funcionalidades como buscar establecimientos adaptados, la información de los restaurantes y el poder añadir establecimientos.
- Considera acertadamente que ha finalizado todas sus tareas correctamente y revela que la tarea 4 es la de mayor dificultad, debido a que los botones de añadir a favoritos y acceder a la lista de favoritos la confunden. Esto seguramente sea debido a que ambas opciones tienen el mismo icono. De igual modo, le habría ocurrido si la tarea hubiera sido añadir a pendiente de visitar, dado que también comparten icono.
- Por otra parte, debido a que en ocasiones no tenía claro en qué punto de la app se encontraba, nos indica que debemos mejorar este punto para que el usuario sepa el apartado exacto en el que se encuentra.
- También nos propone una funcionalidad adicional que sería buscar un establecimiento concreto dentro de la lista de favoritos, para evitar tener que ir buscando uno a uno en la lista el establecimiento que desea ver.
- Por último, nos indica que sí utilizaría la app.

De los resultados y conclusiones obtenidos en el test de usuarios hemos podido obtener una serie de mejoras que nos han permitido corregir y perfeccionar el prototipo anterior:

- Cambiar el icono de la lista de favoritos y de la lista de pendiente de visitar por otro más intuitivo y diferenciador para que el usuario pueda distinguir de forma clara, sin confundirse, cuál es la funcionalidad del botón de añadir a favoritos y la del acceso rápido a la lista de favoritos, igualmente con añadir a pendiente de visitar y su lista.
- Cambiar el título de la aplicación ubicado en la barra superior por un título informativo de la pantalla donde se encuentra y añadir en las pantallas de valoraciones y en la de fotografías, el nombre del restaurante al que pertenecen. Todo ello, con el objetivo de mejorar la orientación y ubicación del usuario acerca de la pantalla en la que se encuentra.
- Añadir un buscador en la lista de favoritos y en la de pendiente de visitar para que el usuario pueda encontrar de forma rápida lo que desea y no tenga que ir buscando elemento a elemento en las listas.

Todas estas mejoras ya se encuentran añadidas en el prototipo de alta fidelidad mostrado en el apartado anterior, como parte del proceso iterativo que es el Diseño Centrado en el Usuario.

En el anexo III pueden encontrarse todos los documentos rellenos por el usuario durante la realización del test.

2.2 Diseño técnico

Toda la información recogida mediante el Diseño Centrado en el Usuario nos sirve de gran ayuda para definir los casos de uso del sistema, las entidades que componen la aplicación, la arquitectura que mejor encaja en nuestra aplicación, etc.

2.2.1 Casos de uso

Este apartado se centra en los casos de uso del proyecto. En primer lugar, vamos a definir los actores que participan en nuestro sistema.

ACT-001	Usuario
Descripción	Este actor representa a todos los usuarios, tanto los que tienen un perfil colaborativo como aquellos con perfil observador, que interactúan con la aplicación Sinlosofía a través de un dispositivo móvil Android

Ilustración 28. ACT Usuario

A continuación, se presenta el diagrama de paquetes de casos de uso del sistema.

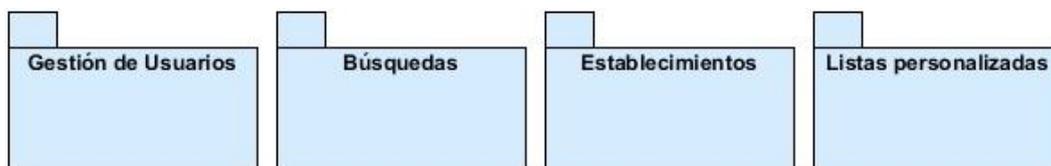


Ilustración 29. Diagrama de paquetes de casos de uso del sistema

En la siguiente imagen se despliega el diagrama de casos de uso del sistema, donde se muestra cómo se relacionan los paquetes con los actores de nuestro sistema.

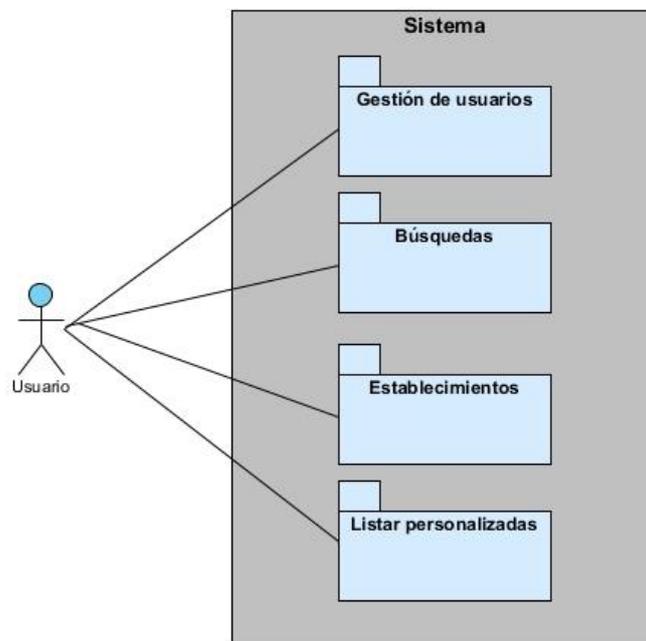


Ilustración 30. Diagrama de casos de uso del sistema

Seguidamente se representan los diagramas de casos de uso de cada paquete del sistema.

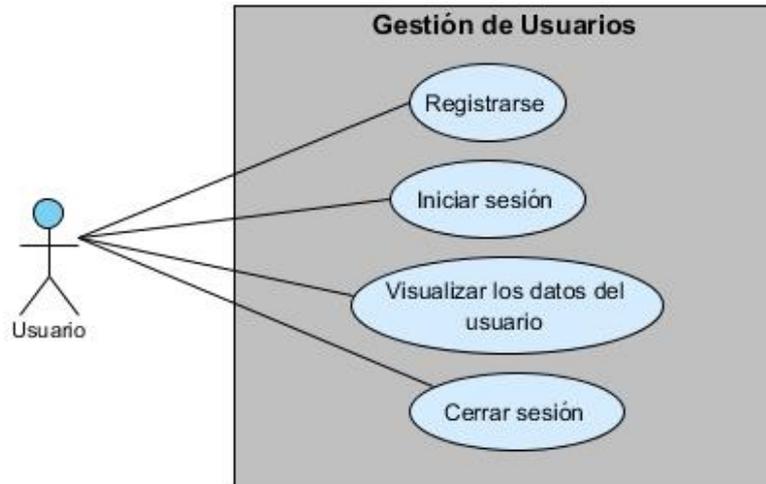


Ilustración 31. Diagrama de casos de uso - Gestión del usuario

En este diagrama pueden verse los casos de uso relacionados con la gestión de usuarios que se pueden realizar.

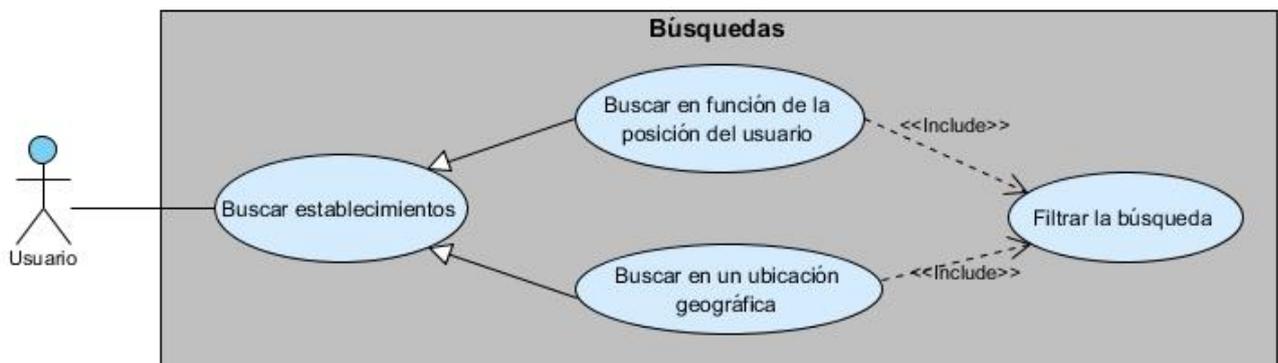


Ilustración 32. Diagrama de casos de uso - Búsquedas

En este diagrama podemos observar cómo se relaciona el usuario con la aplicación en el paquete de búsquedas. Existen algunas relaciones que requieren una especial mención:

- ❖ **Generalizaciones** entre el caso de uso general *Buscar establecimientos* y los casos de uso específicos *Buscar en función de la posición del usuario* y *Buscar en una ubicación geográfica*, dado que el caso de uso general es abstracto y son los casos de uso especializados los que proporcionan el comportamiento.
- ❖ **Includes** desde ambos *Buscar* hacia *Filtrar las búsquedas*, dado que cuando el usuario realice búsquedas se le aplicarán los filtros que haya especificado.

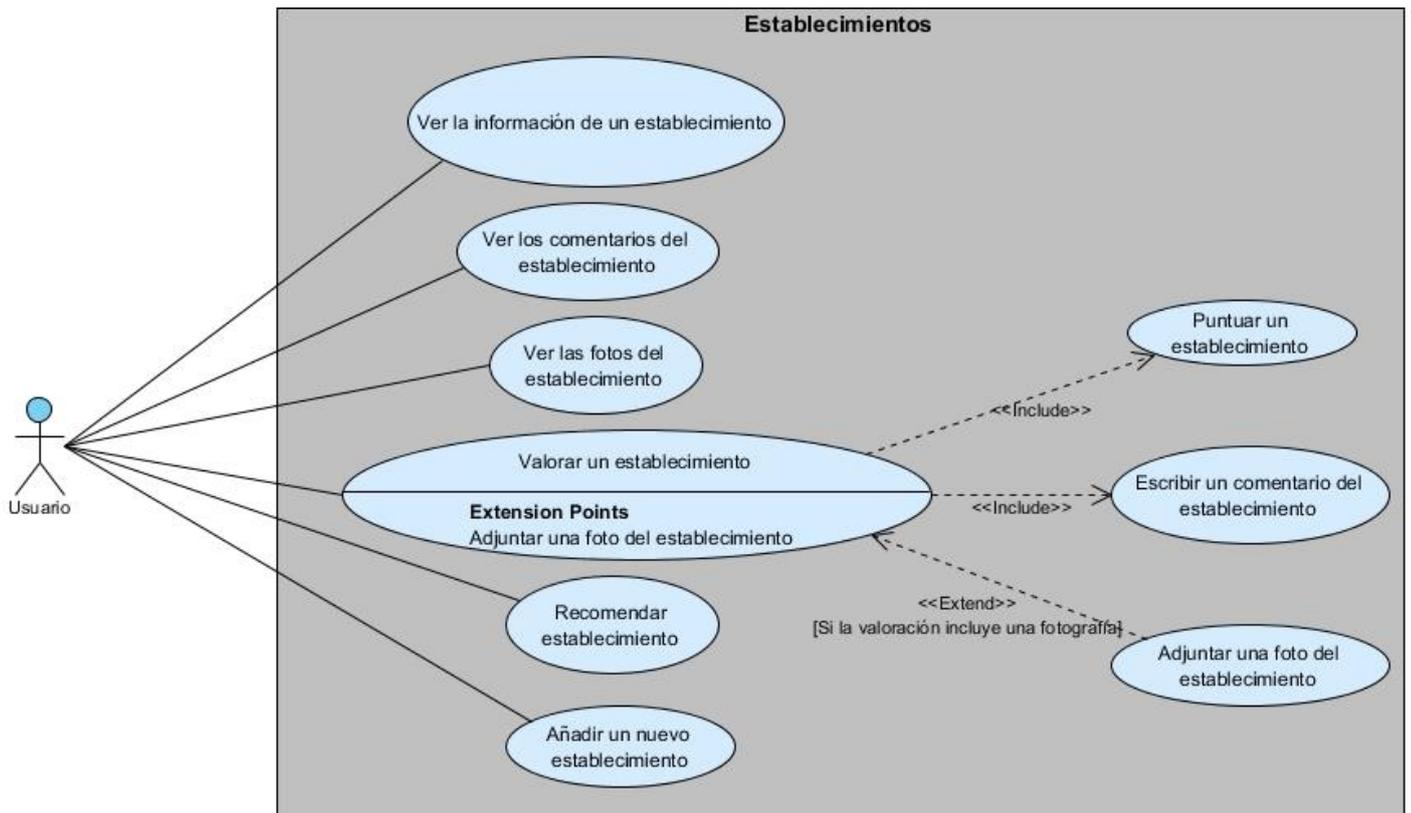


Ilustración 33. Diagrama de casos de uso – Establecimientos

Este diagrama muestra cómo el usuario se relaciona con los casos de uso del paquete Establecimientos. Entre las relaciones existentes merece la pena destacar:

- ❖ **Includes**, de *Valorar un establecimiento* a *Puntuar un establecimiento* y *Escribir un comentario del establecimiento*, puesto que cuando el usuario quiera dejar una valoración de un establecimiento deberá puntuar al mismo y escribir su opinión acerca de la visita.
- ❖ **Extend** de *Valorar un establecimiento* a *Adjuntar una foto del establecimiento*, ya que solo si el usuario desea adjuntar una fotografía de la visita al establecimiento se realiza el caso de uso.

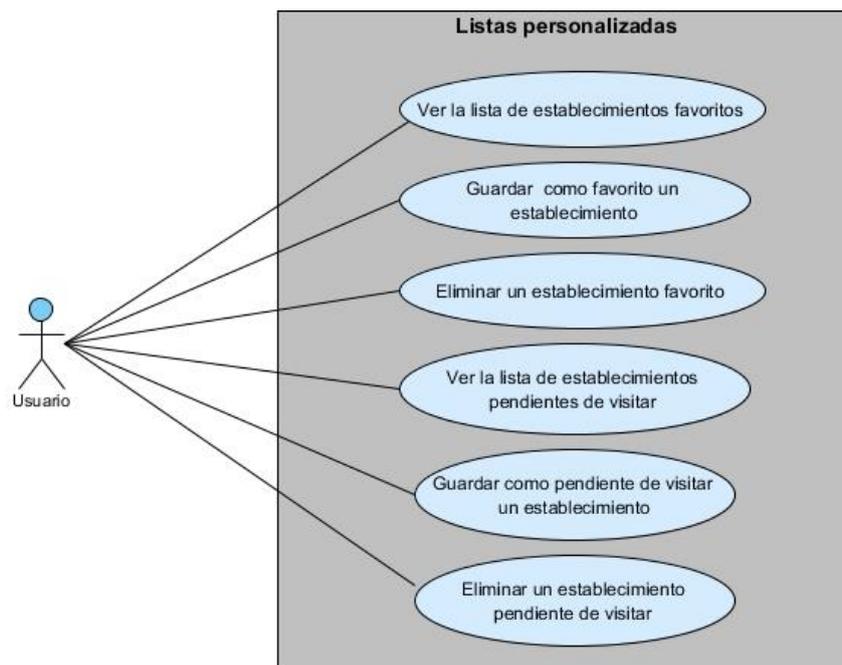


Ilustración 34. Diagrama de casos de uso - Listas personalizadas

Este último diagrama nos muestra cómo se relaciona el usuario con la aplicación mediante los casos de uso incluidos en el paquete Listas personalizadas.

A continuación, se definen cada uno de los casos de uso mostrados en los diagramas UML anteriores:

CU-001	Registrarse
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios registrarse en el sistema, creando una cuenta de usuario que les permita acceder a la aplicación
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> No estar registrado previamente con ese email o usuario
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> Iniciar la aplicación Acceder al registro Rellenar el formulario indicando nombre, apellidos, email y contraseña (2 veces para validación) Pulsar sobre la opción de registrarse El sistema comprueba los datos introducidos y valida que ese usuario no esté ya registrado El sistema crea el usuario con los datos introducidos y redirige al usuario a la home de app
Flujo de excepción	<p>5b. Si el usuario no ha rellenado todos los campos obligatorios, se muestra un error informativo en pantalla</p> <p>5c. En caso de que ya exista un usuario registrado con ese email, se muestra un error al usuario informándole del mismo</p>
Postcondiciones	El sistema creará una nueva cuenta de usuario
Notas	Al realizar el registro, también se produce el login del usuario

CU-002	Iniciar sesión
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios registrados logarse para poder acceder a la aplicación
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> Haber realizado el caso de uso <i>CU-001 Registrarse</i> Recordar tu usuario y contraseña
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> Iniciar la aplicación Introducir el usuario y contraseña de tu cuenta Pulsar sobre la opción de iniciar sesión Validar credenciales del usuario Tras una validación exitosa, se redirige el usuario a la home de la app
Flujo de excepción	<p>5b. Si el usuario no ha rellenado todos los campos obligatorios, se muestra un error informativo en pantalla</p> <p>5c. Si las credenciales del usuario no son correctas se muestra un error informando de ello</p>
Postcondiciones	El usuario inicia sesión en la aplicación

CU-003	Visualizar los datos del usuario
Prioridad	Baja
Descripción	Este caso de uso se basa en mostrar al usuario los datos de su cuenta
Actores	Usuario

Precondiciones	<ul style="list-style-type: none"> Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> Pulsar sobre la opción cuenta de usuario en el menú lateral Obtener los datos de usuario asociados a la cuenta del backend Acceder a la pantalla de cuenta de usuario y mostrar los datos obtenidos en pantalla
Postcondiciones	El usuario deberá visualizar los datos de su cuenta de usuario

CU-004	Cerrar sesión
Prioridad	Normal
Descripción	Este caso de uso permite al usuario cerrar la sesión activa en la aplicación
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> Pulsar sobre la opción de cerrar sesión en el menú lateral o en la pantalla de cuenta de usuario Cerrar sesión del usuario Redirigir a la pantalla de login
Postcondiciones	Se cerrará la sesión activa del usuario en la aplicación

CU-005	Buscar establecimientos
Prioridad	Alta
Descripción	Este caso de uso se centra en la búsqueda de establecimientos en una ubicación y en función de una serie de filtros
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> Acceder a la opción de buscar establecimientos En función de cómo desea buscar: <ol style="list-style-type: none"> Si el usuario decide buscar establecimientos en función de su posición, se realiza el caso de uso <i>CU-006</i> Si el usuario decide buscar establecimientos en una ubicación específica, se realiza el caso de uso <i>CU-007</i> Pulsar sobre la opción buscar Buscar de los establecimientos que se encuentran en la localización establecida y que cumplen los filtros fijados por el usuario Mostrar los resultados de la búsqueda en un mapa o lista
Postcondiciones	Se mostrarán todos los establecimientos que cumplen los requisitos del usuario en un mapa o en una lista

CU-006	Buscar en función de la posición del usuario
Prioridad	Alta
Descripción	Este caso de uso complementa el caso de uso <i>CU-005</i> , especificando la zona en la que se realizará la búsqueda a partir de la posición del usuario
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> Haber iniciado el caso de uso <i>CU-005 Buscar establecimientos</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> Seleccionar la opción de buscar en función de la posición actual del usuario Obtener la posición del usuario

	<ol style="list-style-type: none"> 3. Mostrarla en la pantalla 4. Se realiza el caso de uso <i>CU-008 Filtrar la búsqueda</i>
Postcondiciones	Se establecerá la posición del usuario como lugar a realizar la búsqueda y se mostrará por pantalla

CU-007	Buscar en una ubicación geográfica
Prioridad	Alta
Descripción	Este caso de uso complementa el caso de uso <i>CU-005</i> , indicando la zona en la que se realizará la búsqueda de acuerdo con lo especificado por el usuario
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber iniciado el caso de uso <i>CU-005 Buscar establecimientos</i> • Conocer la ubicación en la que se desea buscar
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Introducir la localización geográfica en la que se desea buscar establecimientos 2. Se realiza el caso de uso <i>CU-008 Filtrar la búsqueda</i>
Postcondiciones	Se establecerá la a ubicación introducida por el usuario como lugar a realizar la búsqueda

CU-008	Filtrar la búsqueda
Prioridad	Alta
Descripción	Este caso de uso continúa los casos de uso <i>CU-006</i> y <i>CU-007</i> , estableciendo los filtros que el usuario desea para realizar la búsqueda.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-006 Buscar en función de la posición del usuario</i> o el <i>CU-007 Buscar en una ubicación geográfica</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Seleccionar aquellos filtros que se desean que el establecimiento cumpla
Postcondiciones	Se establecerán los filtros seleccionados por el usuario para la búsqueda

CU-009	Ver la información de un establecimiento
Prioridad	Alta
Descripción	Este caso de uso permite que los usuarios visualicen la información de un establecimiento en la página o ficha del mismo.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Debe existir el establecimiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la página de un establecimiento, ya sea mediante una búsqueda realizada previamente o a través de las listas personalizadas de favoritos o pendiente de visitar 2. Obtener toda la información asociada al establecimiento del backend 3. Mostrar la página del establecimiento con la información obtenida
Postcondiciones	Se mostrará en pantalla toda la información asociada del establecimiento al que el usuario ha accedido

CU-010	Ver los comentarios del establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios visualizar las valoraciones del establecimiento que han dejado el resto de usuarios
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Estar en la página de un establecimiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Pulsar sobre la opción para ver comentarios del establecimiento 2. Acceder a la pantalla de comentarios del establecimiento 3. Obtener del backend las valoraciones de los usuarios acerca del establecimiento 4. Mostrar los comentarios obtenidos en pantalla
Postcondiciones	Se mostrarán por pantalla las valoraciones del establecimiento que han compartido el conjunto de usuarios

CU-011	Ver las fotos del establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios explorar las fotografías del establecimiento que han subido el resto de usuarios en sus valoraciones
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Estar en la página de un establecimiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Pulsar sobre la opción para ver fotografías del establecimiento 2. Acceder a la pantalla de fotografías del establecimiento 3. Obtener del backend las fotografías del establecimiento subidas por los usuarios en sus valoraciones 4. Mostrar las fotografías obtenidas en pantalla
Postcondiciones	Se mostrarán por pantalla todas las fotografías del establecimiento que han compartido los usuarios en sus valoraciones

CU-012	Valorar un establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios valorar los establecimientos que han visitado mediante la puntuación del establecimiento, la opinión personal del usuario y, opcionalmente, una fotografía.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Estar en la página de un establecimiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Pulsar sobre la opción de valorar el establecimiento 2. Acceder a la pantalla de valorar establecimiento 3. Realizar el caso de uso <i>CU-013 Puntuar un establecimiento</i> 4. Realizar el caso de uso <i>CU-014 Escribir un comentario del establecimiento</i> 5. Si el usuario desea incluir una fotografía en su valoración, se realiza el caso de uso <i>CU-015 Adjuntar una foto del establecimiento</i> 6. Pulsa sobre la opción de valorar 7. Se comprueban los datos introducidos 8. Se registra y almacena la valoración realizada por el usuario 9. Se redirige a la página del establecimiento

Excepción	<p>7b. Si la comprobación de los datos introducidos falla, ya sea porque:</p> <ul style="list-style-type: none"> • No se han rellenado todos los campos obligatorios • Se produce un error al registrar y almacenar la valoración realizada por el usuario en el backend <p>Se muestra un mensaje de error informando de lo ocurrido al usuario como resultado</p>
Postcondiciones	La valoración del usuario se registrará en el sistema y estará disponible para todos los usuarios

CU-013	Puntuar un establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios puntuar al establecimiento cuando están realizando una valoración
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Haber iniciado el caso de uso <i>CU-012 Valorar un establecimiento</i> • Conocer la puntuación que se quiere dar al establecimiento
Iniciado por	Usuario
Flujo	1. Seleccionar la puntuación que se quiere otorgar al establecimiento
Postcondiciones	La puntuación indicada por el usuario debe quedar seleccionada.

CU-014	Escribir un comentario del establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios escribir su opinión personal acerca de su visita al establecimiento a través de un comentario
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Haber iniciado el caso de uso <i>CU-012 Valorar un establecimiento</i> • Conocer la opinión que se quiere dar al establecimiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Introducir el título del comentario 2. Escribir la opinión en el campo comentario
Postcondiciones	La opinión introducida por el usuario debe mostrarse por pantalla

CU-015	Adjuntar una foto del establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios incluir una fotografía de su visita al establecimiento en su valoración
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Haber iniciado el caso de uso <i>CU-012 Valorar un establecimiento</i> • Disponer de la fotografía que se quiere subir en el dispositivo móvil
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Pulsar sobre la opción de adjuntar fotografía 2. Seleccionar la fotografía que se quiere subir del medio correspondiente 3. Cargar la fotografía seleccionada en el cuadro de imagen de la pantalla
Postcondiciones	La fotografía debe quedar seleccionada y mostrarse en el campo correspondiente de la pantalla

CU-016	Recomendar establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios recomendar un establecimiento a través de algunas de las aplicaciones instaladas
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • Estar en la página de un establecimiento
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Pulsar sobre la opción de recomendar establecimiento 2. Seleccionar el medio a través del que se quiere recomendar 3. Seleccionar el destino de la recomendación en el medio elegido 4. Enviar la recomendación del establecimiento al destino
Postcondiciones	La recomendación del usuario debe enviarse por el medio elegido a su destino

CU-017	Añadir un nuevo establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios añadir nuevos establecimientos al sistema introduciendo la información asociada al mismo.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • El establecimiento no debe existir en la aplicación
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Pulsar sobre la opción de añadir nuevo establecimiento 2. Acceder a la pantalla de añadir nuevo establecimiento 3. Introducir los datos obligatorios para crear un nuevo establecimiento 4. Pulsar sobre la opción crear establecimiento 5. Comprobar la información introducida por el usuario 6. Se crea el establecimiento en el sistema 7. Se redirige a la página del nuevo establecimiento
Excepción	<p>6b. Si la comprobación de los datos introducidos fracasa, ya sea porque:</p> <ul style="list-style-type: none"> • No se han rellenado todos los campos obligatorios • Se produce un error al registrar y almacenar el nuevo establecimiento introducido por el usuario en el backend <p>Se muestra un mensaje de error informando de lo ocurrido al usuario como resultado</p>
Postcondiciones	Se creará el establecimiento introducido por el usuario y estará disponible para todos los usuarios

CU-0018	Ver la lista de establecimientos favoritos
Prioridad	Normal
Descripción	Este caso de uso permite que los usuarios visualicen una lista con los establecimientos que han marcado como favoritos
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la opción de establecimientos favoritos 2. Cargar del backend los establecimientos favoritos de la cuenta del usuario 3. Explorar la lista de los establecimientos favoritos
Postcondiciones	Se mostrará la lista de los establecimientos que el usuario ha marcado como favorito

CU-0019	Guardar como favorito un establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios marcar un establecimiento como favorito.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • El establecimiento debe existir en la aplicación
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la página de información del establecimiento 2. Pulsar sobre el icono de favorito 3. Añadir el establecimiento a la lista de favoritos del usuario
Postcondiciones	El establecimiento debe añadirse a la lista de favoritos del usuario

CU-0020	Eliminar un establecimiento favorito
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios eliminar un establecimiento marcado como favorito.
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • El establecimiento debe estar marcado como favorito
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la página de información del establecimiento 2. Pulsar sobre el icono de favorito que se encuentra marcado 3. Eliminar el establecimiento de la lista de favoritos
Postcondiciones	El establecimiento debe eliminarse de la lista de favoritos asociada a la cuenta del usuario

CU-0021	Ver la lista de establecimientos pendientes de visitar
Prioridad	Normal
Descripción	Este caso de uso permite que los usuarios visualicen una lista con los establecimientos que han marcado como pendiente de visitar
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i>
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la opción de establecimientos pendientes de visitar 2. Cargar del backend los establecimientos pendientes de visitar de la cuenta del usuario 3. Explorar la lista de los establecimientos pendientes de visitar
Postcondiciones	Se mostrará la lista de los establecimientos que el usuario ha marcado como pendiente de visitar

CU-0022	Guardar como pendiente de visitar un establecimiento
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios marcar un establecimiento como pendiente de visitar
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • El establecimiento debe existir en la aplicación

Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la página de información del establecimiento 2. Pulsar sobre el icono de pendiente de visitar 3. Añadir el establecimiento a la lista de pendientes de visitar del usuario
Postcondiciones	El establecimiento debe añadirse a la lista de pendientes de visitar del usuario

CU-0023	Eliminar un establecimiento pendiente de visitar
Prioridad	Normal
Descripción	Este caso de uso permite a los usuarios eliminar un establecimiento marcado como pendiente de visitar
Actores	Usuario
Precondiciones	<ul style="list-style-type: none"> • Haber realizado el caso de uso <i>CU-002 Iniciar sesión</i> • El establecimiento debe estar marcado como pendiente de visitar
Iniciado por	Usuario
Flujo	<ol style="list-style-type: none"> 1. Acceder a la página de información del establecimiento 2. Pulsar sobre el icono de pendiente de visitar que se encuentra marcado 3. Eliminar el establecimiento de la lista de pendientes de visitar
Postcondiciones	El establecimiento debe eliminarse de la lista de pendientes de visitar asociada a la cuenta del usuario

2.2.2 Diseño de datos

Esta sección hace referencia al cometido del diseño de datos, que es el de seleccionar representaciones lógicas de los objetos de datos identificados en la fase de análisis.

A continuación, se presenta el diagrama de clases del sistema:

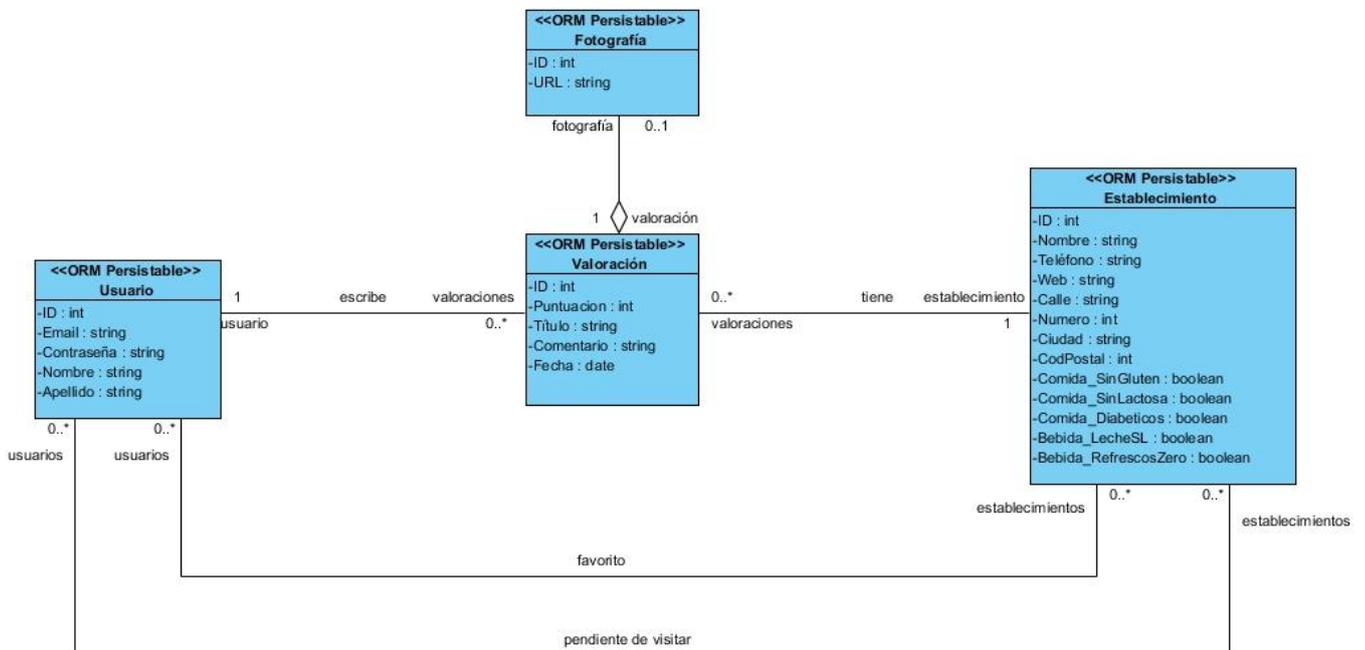


Ilustración 35. Diagrama de Clases del Sistema

En la siguiente imagen se presenta el diagrama de entidad-relación de la base de datos en el que se pueden observar las distintas entidades y las relaciones entre ellas.



Ilustración 36. Diagrama de Entidad-Relación

A continuación, se detallan los atributos de cada entidad mostrada en el diagrama anterior:

- Entidad Usuario: representa a los usuarios registrados en la aplicación y almacena todos sus datos.

Usuario	
Nombre	Descripción
ID	Identificador único de cada usuario
Email	Correo electrónico del usuario, utilizado para realizar la autenticación
Contraseña	Contraseña de la cuenta de usuario
Nombre	Nombre del usuario
Apellidos	Apellidos del usuario

- Entidad Establecimiento: esta entidad representa a cada establecimiento existente en la aplicación y recoge toda la información que definen sus características.

Establecimiento	
Nombre	Descripción
ID	Identificador único del establecimiento
Nombre	Nombre del establecimiento
Telefono	Número de contacto del establecimiento
Web	URL de la página web del establecimiento
Calle	Calle en la que se ubica el establecimiento
Número	Número de la calle en el que se encuentra el establecimiento
Ciudad	Localidad en la que está el establecimiento
CodPostal	Código postal de la zona en la que se localiza el establecimiento
Comida_SinGluten	Bandera que nos indica si el establecimiento ofrece comida sin gluten
Comida_SinLactosa	Bandera que nos indica si el establecimiento ofrece comida sin lactosa
Comida_Diabeticos	Bandera que nos indica si el establecimiento ofrece comida baja en azúcares (apta para diabéticos)
Bebida_LecheSL	Bandera que nos indica si el establecimiento ofrece leche sin lactosa
Bebida_RefrescosZero	Bandera que nos indica si el establecimiento ofrece refrescos con cero azúcar

- Entidad Valoración: representa las valoraciones realizadas por cada usuario almacenando toda su información.

Valoración	
Nombre	Descripción
ID	Identificador único de cada valoración
EstablecimientoID	Identificador único del establecimiento sobre el que se realizó la valoración

UsuarioID	Identificador único del usuario que escribió la valoración
Puntuación	Puntuación dada por el usuario al establecimiento
Título	Título de la valoración del usuario
Comentario	Opinión personal del usuario acerca de su visita al establecimiento
Fecha	Fecha en la que se realizó la valoración

- Entidad Fotografía: esta entidad hace referencia a las fotografías que los usuarios pueden subir junto con su valoración.

Fotografía	
Nombre	Descripción
ID	Identificador único de cada fotografía subida
ValoracionID	Identificador único de la valoración a la que pertenece la fotografía
URL	Dirección web en la que se encuentra la imagen para poder descargarla

- Entidad Lista Favoritos: esta entidad conforma una lista con todos los establecimientos favoritos de cada usuario.

Lista Favoritos	
Nombre	Descripción
EstablecimientoID	Identificador único del establecimiento que ha marcado como favorito
UsuarioID	Usuario que marcó como favorito el establecimiento

- Entidad Lista Pendiente de visitar: esta entidad conforma una lista con todos los establecimientos pendiente de visitar de cada usuario.

Lista Pendiente de visitar	
Nombre	Descripción
EstablecimientoID	Identificador único del establecimiento que ha marcado como pendiente de visitar
UsuarioID	Usuario que marcó como pendiente de visitar el establecimiento

2.2.3 Diseño de la arquitectura

En este apartado se llevará a cabo el proceso de transformación de ideas y modelos que se han obtenido en el análisis, convirtiéndolos en modelos más cercanos a la solución que se pretende obtener. El objetivo es desarrollar la estructura modular, representando las relaciones de control de los módulos, combinando la estructura del programa con la estructura de datos y definiendo las interfaces que permiten el flujo de datos a través del programa.

El sistema del proyecto utiliza una arquitectura **Cliente-Servidor**, que es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados Servidores, y los demandantes, llamados Clientes. Los clientes realizan peticiones al Servidor que le da respuesta.

En este tipo de arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

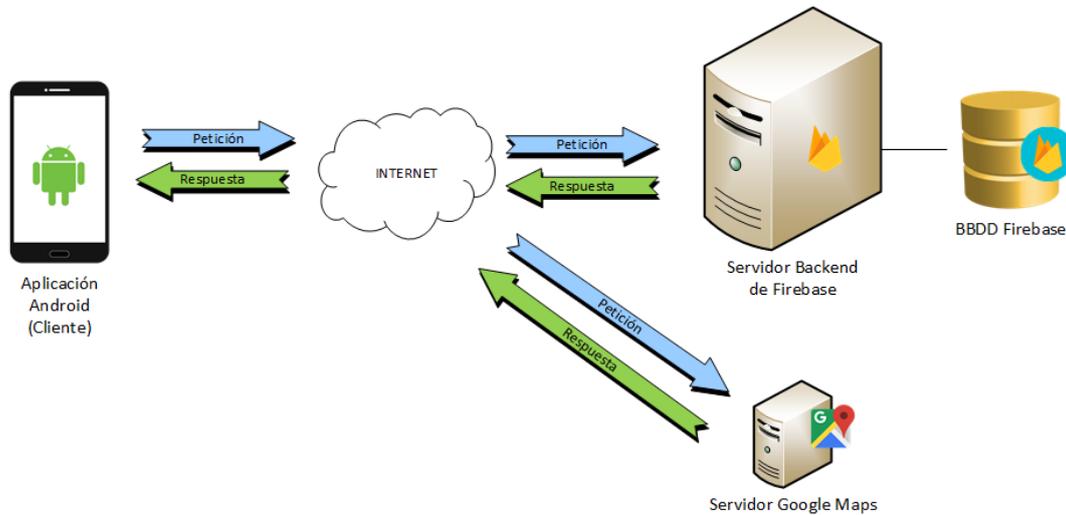


Ilustración 37. Arquitectura del sistema

Como puede verse en diagrama anterior, nuestro sistema se divide principalmente en dos componentes:

- Una aplicación Android nativa que obtiene los datos del Servidor backend para proporcionar las distintas funcionalidades a los usuarios. Este nodo realiza el papel de cliente interactuando con los usuarios finales a través de la interfaz gráfica, realizando solicitudes al servidor, esperando y recibiendo las respuestas, conectándose a varios servidores, etc.
- Servidor backend de Firebase que se encargará de responder a las peticiones de las aplicaciones para autenticar a los usuarios y proporcionar los datos del sistema almacenados como establecimientos, valoraciones, etc. Como vemos, este nodo ejerce el papel de servidor esperando que lleguen las solicitudes de los clientes, procesando las peticiones que recibe y enviando las respuestas al cliente, aceptando conexiones de varios clientes, etc.

Adicionalmente, nuestro sistema cuenta con un tercer nodo que también realiza el rol de Servidor, la aplicación Android se comunicará con los servidores de Google Maps para hacer uso de su servicio de mapas y *places* mediante sus APIs.

2.2.3.1 Vista Estática

2.2.3.1.1 Arquitectura de la aplicación

En cuanto a la arquitectura de la aplicación Android, se va a utilizar el patrón de diseño de software MVC (Modelo-Vista-Controlador). Este patrón arquitectónico consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos:

- **Modelo:** este componente encapsula los datos y la funcionalidad central y es independiente de la entrada y la salida.
- **Vista:** presenta la información al usuario. Una vista obtiene los datos del modelo, pudiendo haber múltiples vistas del modelo.
- **Controlador:** los controladores reciben la entrada, normalmente eventos que son trasladados para servir las peticiones del modelo o de la vista. El usuario interactúa con el sistema sólo a través de los controladores.

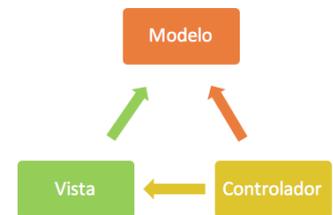


Ilustración 38. Patrón MVC

Algunos de los principales problemas a los que se hace frente con este patrón son la propensión que tienen las interfaces de usuario para cambiar, extender funcionalidad de una aplicación o portarla a un entorno gráfico diferente. Además, construir un sistema flexible es caro y propenso a errores si la interfaz de usuario está altamente entremezclada con el núcleo funcional.

Por ello, su aplicación nos permite realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. Así las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

El siguiente diagrama representa la arquitectura de la aplicación Android, permitiéndonos observar cómo se organizan las clases de la aplicación según el patrón MVC. Para simplificar el diagrama se han omitido las vistas xml de la aplicación y las relaciones específicas entre estas y los controladores.

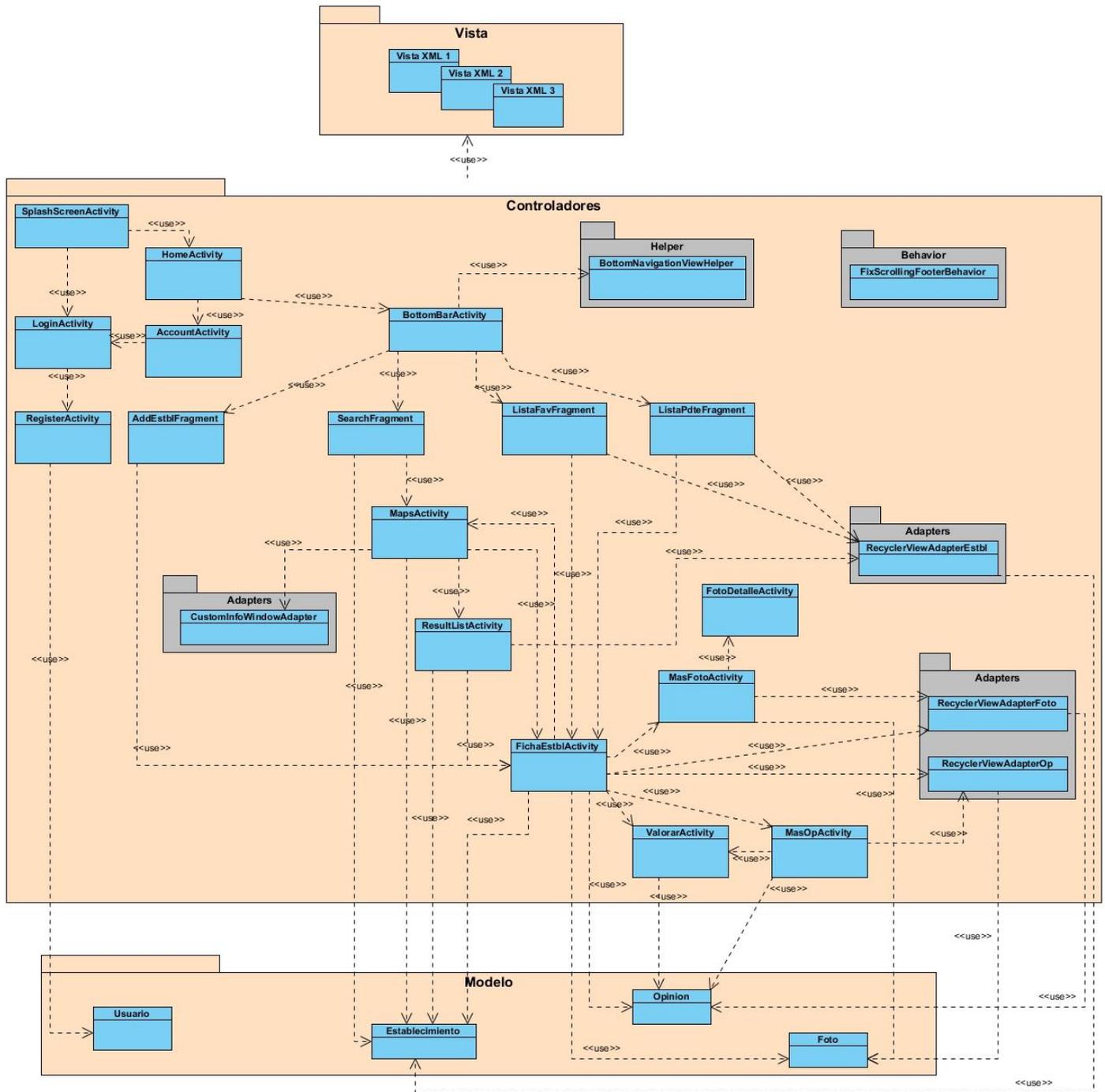


Ilustración 39. Arquitectura aplicación MVC

2.3.3.1.2 Vista Física

Por último, en esta vista se pretende reflejar cómo estará distribuida la aplicación entre los distintos componentes hardware que conformen el sistema. En este caso la principal aportación de esta vista será el mostrar cómo está implementada la arquitectura Cliente-Servidor.

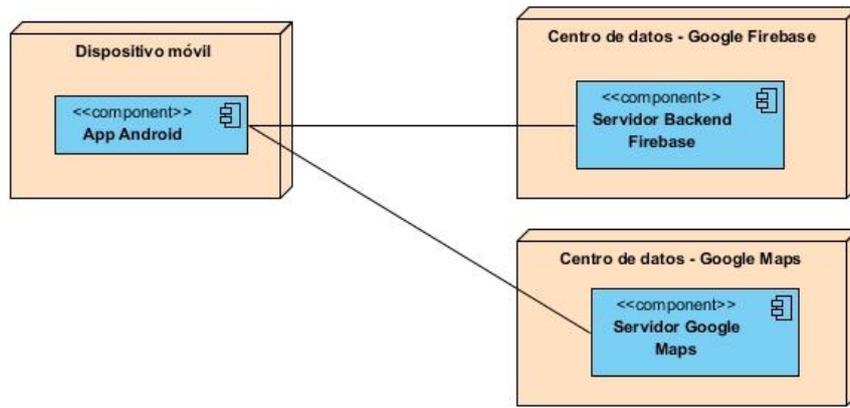


Ilustración 40. Diagrama de despliegue

3. Implementación

3.1 Herramientas y APIs utilizadas

Durante esta fase se han utilizado diferentes herramientas (software y hardware) y APIs para llevar a cabo el desarrollo del sistema que nos han ayudado a resolver diferentes problemas. La siguiente imagen nos muestra un resumen y en los siguientes apartados profundizaremos en algunas de ellas.



Ilustración 41. Herramientas y APIs utilizadas

3.1.1 Android

Android es un sistema operativo (SO) basado en el núcleo Linux diseñado principalmente para dispositivos móviles con pantalla táctil y también para relojes inteligentes, televisores y automóviles.

La arquitectura del SO se divide en una serie de componentes principales:

- **Apps del sistema:** conjunto de aplicaciones base (email, contactos, navegador, mapas, etc.) escritas en el lenguaje de programación Java.
- **Java API Framework:** el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java, que son los cimientos para la creación de apps Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares.
- **Librerías C/C++ Nativas:** conjunto de bibliotecas de C/C++ usadas por varios componentes y servicios del sistema Android.
- **Android Runtime:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada app Android ejecuta sus propios procesos con sus propias instancias de la máquina virtual (ART en API 21 y posteriores / Dalvik en anteriores).
- **HAL:** la capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al framework de la Java API de nivel más alto.
- **Núcleo de Linux:** el uso del *kernel* de Linux se utiliza para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores.

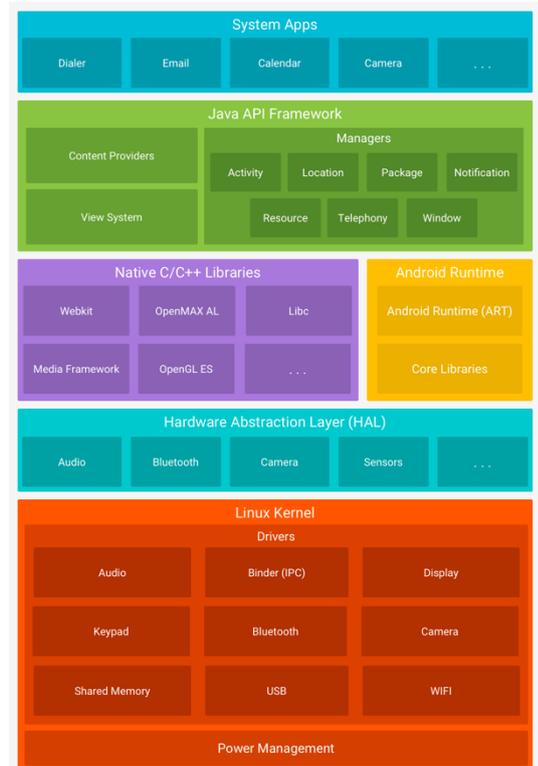


Ilustración 42. Arquitectura Android

3.1.1.1 Android Studio

Android Studio es el entorno de desarrollo integrado (*IDE -Integrated Development Environment*) para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos inteligente y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de apps para Android, como un sistema de compilación basado en Gradle, un emulador rápido con diversas funciones, un entorno unificado para cualquier dispositivo, *Instant Run*, etc.

3.1.2 Firebase

Firebase es una plataforma de Google que te permite desarrollar aplicaciones web y móviles de forma rápida, sencilla y sin lenguaje de programación del lado del servidor (*BaaS, Backend-as-a-service*). Firebase permite a los desarrolladores centrarse en crear fantásticas experiencias de usuario sin tener que preocuparse por la administración de servidores, las bases de datos o las APIs. Esta plataforma ofrece diferentes productos integrados (*analytics*, bases de datos, autenticación, almacenamiento, hosting, informe de fallos, AdMob, etc.) que pueden ser utilizados por las aplicaciones desarrolladas, de los cuales nuestra aplicación utiliza:

- **Firebase Authentication**

Este módulo facilita la creación de un sistema de autenticación seguro, a la vez que mejora la experiencia de incorporación y acceso para los usuarios finales. Para ello, proporciona una solución de identidad de extremo a extremo, mediante servicios de backend, SDK fáciles de usar y bibliotecas de UI (*User Interface*, Interfaz de Usuario) ya elaboradas para autenticar a los usuarios y, además, admite la autenticación mediante correo/contraseñas, números de teléfono y proveedores de identidad federados populares, como Google, Facebook y Twitter.

- **Firebase Realtime Database**

Este producto permite almacenar y sincronizar datos entre usuarios y dispositivos en tiempo real a través de una base de datos NoSQL alojada en la nube. Los datos se almacenan en la base de datos en formato JSON, se sincronizan en tiempo real con cada cliente conectado y permanecen disponibles si la app pierde la conexión a la red, lo que brinda una experiencia del usuario de alta calidad sin importar el estado de la conectividad. La API de Realtime Database está diseñada para permitir solo operaciones que se pueden ejecutar rápidamente, permitiendo crear una excelente experiencia en tiempo real que puede servir a millones de usuarios sin afectar la capacidad de respuesta.

- **Firebase Storage**

Cloud Storage para Firebase es un servicio de almacenamiento de objetos potente y simple que permite subir y compartir contenido generado por los usuarios, como imágenes y vídeos, integrando así contenido multimedia en las apps. Los datos se almacenan en un depósito de Google Cloud Storage, una solución de almacenamiento de objetos a escala de exabytes con alta disponibilidad y redundancia global. Además, los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para las apps de Firebase, sin importar la calidad de la red. En relación con Firebase Storage también se utiliza *Firestore* que es una librería *open source* para Android que nos permite conectar de forma fácil y rápida los elementos comunes de la UI con la API de Firebase.

3.1.3 Google Maps API

El servidor de aplicaciones de mapas de Google nos permite utilizar su funcionalidad mediante APIs para Android que están disponibles a través de los servicios de Google Play, de modo que las apps puedan conocer la ubicación, incluir mapas y sus datos, encontrar sitios cercanos relevantes, etc.

- **Google Maps Android API**

Esta API permite agregar mapas basados en datos de Google Maps a las apps. Para ello, la API administra de forma automática el acceso a servidores, descargas de datos, visualización de

mapas y respuesta a gestos de mapas de Google Maps. También ofrece métodos para agregar marcadores, polígonos y superposiciones a un mapa básico, y para cambiar la vista del usuario de modo que se muestre un área del mapa en particular.

- **Google Place API for Android**

Este API obtiene información de los *places* o sitios almacenados en la base de datos de Google Maps y Google+ Local, permitiéndonos utilizar: un seleccionador de *places* que permite que los usuarios puedan elegir un conjunto de sitios cercanos que se muestren en un mapa, el autocompletado de sitios que nos predice el nombre o la dirección de un sitio mientras se escribe y los detalles de un sitio pudiendo recuperar información sobre millones de ubicaciones.

3.1.4 Glide

Es un rápido y eficiente framework de carga de imágenes y gestión de recursos multimedia *open source* para Android que integra la decodificación de archivos multimedia, el cacheo en memoria y disco y la agrupación de recursos en una interfaz simple y fácil de usar. Glide admite la búsqueda, decodificación y la visualización de imágenes fijas, videos y GIF animados. También incluye una API flexible que permite a los desarrolladores conectarse a casi cualquier pila de red.

El objetivo principal de Glide es hacer que el desplazamiento de cualquier tipo de lista de imágenes sea lo más fácil y rápido posible, pero Glide también es efectivo para casi cualquier caso en el que se necesite buscar, cambiar el tamaño y mostrar una imagen remota.

3.1.5 Espresso

Se trata de un framework de testing *open source* de Google que provee una API que permite crear pruebas de interfaz de usuario simulando interacciones en una aplicación Android. El *core* de este API es pequeño, predecible y fácil de aprender, además de ser personalizable. La principal ventaja de este framework es que nos permite la sincronización automática de las acciones de las pruebas con la interfaz de usuario de nuestra aplicación. Espresso está disponible a partir de la versión Android 2.2 y permite realizar pruebas tanto en dispositivos físicos como virtuales (emuladores).

3.2. Aspectos relevantes del desarrollo

3.2.1 Proyecto Android

La aplicación ha sido desarrollada de forma nativa con el IDE Android Studio y es compatible con dispositivos que tengan una versión Android 4.1 – *JellyBean (API 16)* o superior. Esta elección permite que nuestra que aplicación sea compatible con el 99,2% de los dispositivos Android activos de acuerdo con la información de Google.

El proyecto se encuentra estructurado en diferentes paquetes, dentro de la carpeta Java se encuentra el paquete de *com.imac.josecrespot.sinlosofia* que alberga todas las clases de las actividades y de los fragment y contiene a su vez cuatro paquetes: *adapters* (con las clases que extienden de *adapters*), *behavior* (con una clase que implementa un comportamiento *custom* del *AppBar*), *helper* (con una clase auxiliar para el *BottomNavigationView*) y *datos* (con las clases que representan el modelo). Dentro de esta carpeta, también tenemos el paquete *com.imac.josecrespot.sinlosofia (AndroidTest)* que contiene todas las clases de los test automatizados con Espresso. Por último, en la carpeta *res* encontramos todos los recursos del proyecto (imágenes, layout, *string*, etc.).

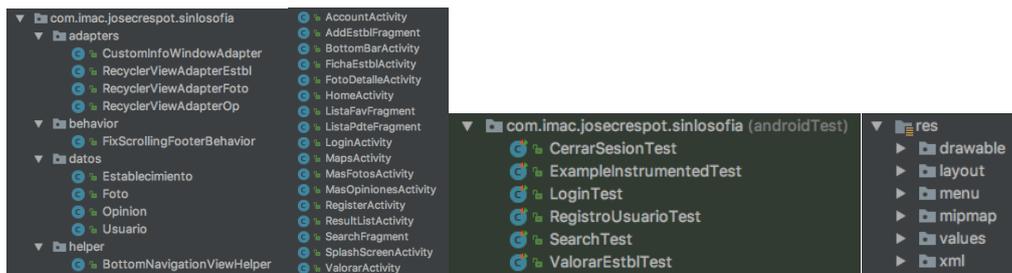


Ilustración 43. Estructura del proyecto

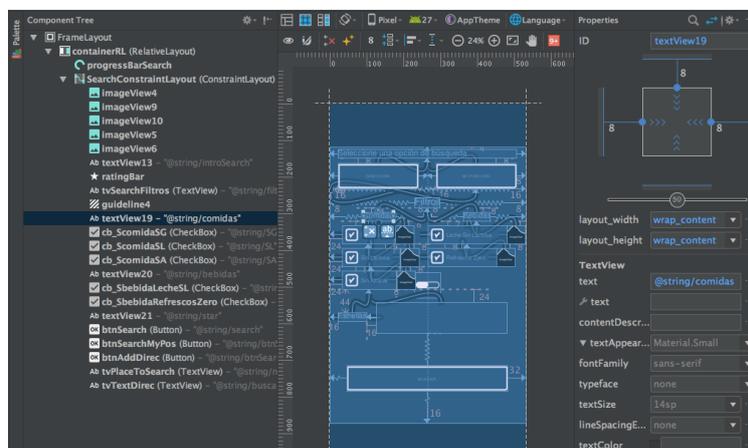
3.2.2 Diseños *responsive* para diferentes tamaños de pantalla

Ante la gran cantidad de tamaños y densidades de pantallas diferentes que existen en el ecosistema Android y con el objetivo de maximizar el número de dispositivos compatibles con la aplicación desarrollada, se han llevado a cabo una serie de buenas prácticas recomendadas por Google en la documentación de Android Developer.

Toda la interfaz gráfica de la aplicación está diseñada con layouts relativos que posicionan las vistas en función del tamaño de la pantalla. La IU de la app ha sido construida utilizando principalmente *ConstraintLayout*, aunque también se usan otros layout para situaciones específicas como *RelativeLayout*. *ConstraintLayout* es un novedoso layout que permite crear grandes y complejos layouts con una jerarquía de vista plana a partir de unas relaciones llamadas *constraint*, que se establecen entre las *view* que componen el layout y con el *parent* del mismo. En general, este layout es similar a *RelativeLayout*, pero es más flexible y fácil de usar.

Además, para definir los tamaños de alto y ancho de las *views* de la IU se ha utilizado:

- *wrap_content* y *match parent*. Estos valores de tamaño permiten que las vistas ajusten el tamaño al contenido de la misma o que se adapten al tamaño de la vista padre respectivamente.
- Píxeles independientes de la densidad (*dp*) para definir los tamaños de algunas vistas y píxeles independientes de la escala (*sp*) en el caso de texto.

Ilustración 44. UI de *SearchFragment* diseñada con *ConstraintLayout*

3.2.3 Diseño de la *Navigation Drawer* Activity

El *Navigation Drawer* está controlado y gestionado enteramente por la actividad *HomeActivity*, siendo esta quien define las respuestas a cada una de las opciones disponibles en el menú lateral (Mi cuenta de usuario, Información, Compartir app y Cerrar sesión).

Por otra parte, el diseño de la interfaz está compuesto por varios layout combinados de la siguiente forma:

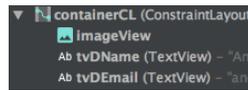
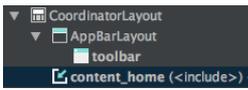
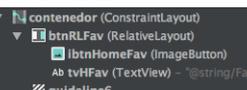
activity_home.xml	nav_header_home.xml	app_bar_home.xml	content_home.xml
Estructura: 	Estructura: 	Estructura: 	Estructura: 
<p>Se trata del layout principal. Incluye el layout <i>app_bar_home</i> y carga en el <i>NavigationView</i> el layout <i>nav_header_home</i> como <i>header</i> y el menú <i>activity_home_drawer</i></p>	<p>Contiene el diseño del <i>header</i> del menú lateral</p>	<p>Se compone de un <i>CoordinatorLayout</i> que contiene el <i>AppBarLayout</i> de la pantalla e incluye el layout <i>content_home</i></p>	<p>Contiene el diseño propio de la pantalla de la <i>HomeActivity</i></p>

Ilustración 45. Diseño UI *Navigation Drawer Activity*

3.2.4 Diseño de la *Bottom Navigation Activity*

El *BottomNavigationView* se controla y se gestiona desde la *BottomBarActivity*, que es quien define cada una de las respuestas a las opciones del menú inferior (Buscar, Añadir, Favoritos y Recomendados). Cuando un usuario selecciona cualquiera de las diferentes opciones de la aplicación disponibles en la *HomeActivity*, se redirige al usuario a la *BottomBarActivity*, esta recibe la petición y carga en un *FrameLayout* el fragment de la opción seleccionada. Del mismo modo, cuando el usuario selecciona otra opción en el menú inferior, la actividad carga el fragment elegido.

El diseño de la interfaz es bastante sencillo: tenemos un layout principal *activity_bottom_bar.xml*, que se compone de un *FrameLayout* donde se cargan los fragment y del *BottomNavigationView*, y 4 fragment, uno para cada una de las opciones del menú inferior (*fragment_search.xml*, *fragment_add_estbl.xml*, *fragment_lista_fav.xml* y *fragment_lista_pdte.xml*).

3.2.5 Estructura de Realtime Database Firebase

Como la base de datos que nos proporciona Firebase en el producto RealTime Database es de tipo NoSQL y almacena los datos en formato JSON, ha sido necesario transformar el diseño de datos realizado en el apartado de arquitectura a un diseño adaptado a este tipo de bases de datos y a nuestro contexto.

Nuestra estructura de la base de datos NoSQL está modelada en función de cómo nuestra aplicación va a leer los datos principalmente, aunque también se han tenido en cuenta la forma de escribirlos, de manera que el proceso sea lo más simple posible consiguiendo un mejor rendimiento y optimización. En la creación de esta estructura también se siguieron las recomendaciones dadas por Firebase como evitar la anidación excesiva de datos por seguridad y para evitar recuperar datos innecesarios en una consulta, compactar la estructura de datos y crear datos escalables, desnormalizando la estructura mediante la duplicación de datos para facilitar las operaciones de lectura.

Finalmente, tras un exhaustivo análisis de las consultas que se realizan desde la aplicación y los conceptos que se deben tener en cuenta a la hora de trabajar con una base de datos NoSQL, la estructura JSON definida para la misma se divide en tres grandes grupos:

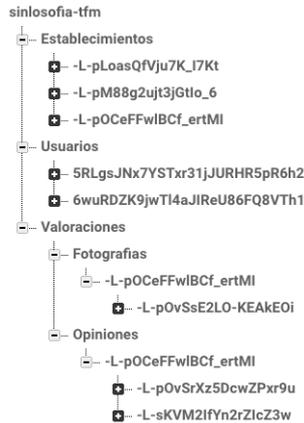


Ilustración 46. Estructura general de la base de datos NoSQL

- El nodo de **Establecimientos** alberga un nodo por cada uno de los establecimientos creados en la aplicación. Cada uno de estos nodos se identifica mediante un ID unívoco, contiene toda la información de los establecimientos (características, dirección, puntuación, etc.) y puede albergar (si existe) dos subnodos que identifican por ID a los usuarios que han guardado el establecimiento como favorito o recomendado (*pdte*). Así, podremos obtener todos los establecimientos favoritos o recomendados de un determinado usuario.
- El nodo de **Usuarios** contiene un nodo por cada usuario que se ha registrado en la aplicación, identificándolos por un ID unívoco.
- El nodo **Valoraciones** almacena, por un lado, en el nodo **Opiniones** todos los comentarios realizados en la aplicación organizados en subnodos con el ID del establecimiento al que pertenecen los comentarios y, por otro lado, el nodo **Fotografías** con todas las fotos subidas dentro de las valoraciones de los usuarios organizadas del mismo modo. Así, si queremos obtener todos los comentarios o fotografías de un determinado establecimiento simplemente debemos recuperar el subnodo dentro de Opiniones o Fotografías que tiene su ID.

Cada nodo dentro de opiniones ubicado dentro de un nodo ID de establecimiento almacena toda la información relativa a la valoración (título, puntuación, fecha, etc.) y, del mismo modo, cada nodo dentro de fotografías que se ubica dentro de un nodo ID de establecimiento guarda la información relativa a la fotografía.



Ilustración 47. Estructura en detalle de la base de datos NoSQL

3.2.6 Almacenamiento y recuperación de datos en Firebase RealTime Database

En primer lugar, es necesario especificar las reglas de acceso a la base de datos de Firebase. En nuestro caso, se ha decidido por seguridad que solo los usuarios registrados pueden leer o escribir datos. Esto se hace efectivo mediante la siguiente regla:

```
"rules": {
  ".read": "auth != null",
  ".write": "auth != null"
```

Ilustración 48. Reglas de acceso a Firebase Database

Hecho esto, lo siguiente que se necesita para poder trabajar con la base de datos es obtener una instancia de *DatabaseReference* mediante la función *getInstance().getReference()* de la clase *FirebaseDatabase*.

Las escrituras en Firebase trabajan con los tipos de datos disponibles en JSON (*String, Long, Double, Boolean, Map<String, Object>* o *List <Object>*) o con objetos Java personalizados, siendo este último tipo el utilizado principalmente en la app desarrollada. Para ejecutar operaciones de escrituras se utiliza la función *setValue()* de *DatabaseReference*, que guarda datos en la referencia que se especifique o reemplaza los datos existentes en esa ruta de acceso. La ruta o ubicación donde se quieren almacenar los datos puede establecerse mediante la función *child()* de *DatabaseReference* que nos permitirá recorrer los diferentes nodos del árbol JSON hasta llegar al nodo donde se quiere almacenar los datos.

Un ejemplo de escritura realizada en la aplicación, en concreto el registro de los nuevos usuarios, sería el siguiente:

```
//Creamos una instancia de la clase Usuario con los datos introducidos
Usuario newuser = new Usuario(etAddress.getText().toString().trim(), etApellidos.getText().toString().trim(),
    etEmail.getText().toString().trim());
//Guardamos en la base de datos la instancia de usuario dentro de /Usuarios/Id/
mDatabase.child("Usuarios").child(userId).setValue(newuser);
```

Ilustración 49. Inserción en Firebase Database a través de objeto java

En cuanto a las operaciones de lectura, para recuperar datos en una determinada ruta o *path* y escuchar para detectar cambios, se utiliza el método *addValueEventListener()* o el método *addListenerForSingleValueEvent()* para agregar un *ValueEventListener* a *DatabaseReference*. Este *listener* llamará al método *onDataChange()* cuando obtenga los resultados, lo que nos permitirá leer una instantánea estática que contiene todos los datos de dicha ruta, incluidos los datos secundarios. En la aplicación desarrollada se utiliza *addListenerForSingleValueEvent()* para que el método *onDataChange()* se llame una única vez obteniendo la instantánea de ese momento, en lugar de cada vez que se produzcan cambios en los datos.

Para ordenar y filtrar las consultas sobre la base de datos de Firebase se utiliza la clase *Query*. En estas consultas, para recuperar datos ordenados debemos seleccionar un único método de ordenamiento (*orderByChild()*, *orderByKey()* o *orderByValue()*) y para filtrar los datos podemos utilizar cualquiera de los métodos de filtrado disponibles (*limitToFirst()*, *limitToLast()*, *startAt()*, *endAt()* o *equalTo()*), pudiendo combinarlos con un único método de ordenamiento o entre sí varios de los métodos de límite o rango.

A continuación, se muestra un ejemplo de una de consulta realizada en la aplicación, en concreto la consulta de los tres últimos comentarios de un establecimiento:

```
//Obtenemos los últimos tres comentarios por fecha del establecimiento
Query query4 = mDatabase.child("Valoraciones").child("Opiniones").child(estblActual.getEstablecimientoID())
    .orderByChild("timestamp").limitToLast(3);
query4.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        //Vaciamos el array
        LastComments.clear();

        //Recorremos la instantánea devuelta y creamos un objeto Opinion por cada uno
        // de los devueltos y lo añadimos en el array
        for(DataSnapshot OpSnapshot : dataSnapshot.getChildren()) {
            Opinion op = OpSnapshot.getValue(Opinion.class);
            LastComments.add(op);
        }
        //...
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
        // Se ha producido un error en la consulta
        //...
    }
});
```

Ilustración 50. Consulta a Firebase Database

3.2.7 Gestión de usuarios con Firebase Authentication

La app de Sinlosofía utiliza el módulo de Authentication de Firebase para proporcionar las siguientes funcionalidades: autenticación mediante correo electrónico, recuperación de contraseñas, login automático, registro de nuevos usuarios y cierre de sesión.

En cuanto al login, la validación de las credenciales introducidas por el usuario se realiza en *LoginActivity* mediante la función *signInWithEmailAndPassword(email, pass)* de la clase *FirebaseAuth* y se define un *listener* para comprobar si el login tuvo éxito o no en función de la *Task<AuthResult>* devuelta. Para evitar que los usuarios tengan que logarse cada vez que usen la app, en la *SplashScreenActivity()* se comprueba si el usuario se encuentra logado mediante *getCurrentUser()* de la clase *FirebaseAuth* que nos devuelve el *FirebaseUser* actual. En caso éxito se valida el *token* del usuario con el servidor mediante el método *getIdToken()* de *FirebaseUser* y se define un *listener* para manejar el resultado. Si el *token* es válido se redirige al usuario a la home de app y en cualquier otro caso a la pantalla de login.

Por otra parte, *LoginActivity* también incorpora una funcionalidad para recordar la contraseña del email introducido en el campo de texto. Para ello, utilizamos la función *sendPasswordResetEmail(email)* de la clase *FirebaseAuth* y de igual modo se define un *listener* para obtener el resultado.

Respecto al registro, la creación de nuevos usuarios a partir de los datos introducidos se realiza en *RegisterActivity* mediante la función *createUserWithEmailAndPassword(email, pass)* de la clase *FirebaseAuth* y, como anteriormente, se define un *listener* para conocer el resultado. En caso de éxito, se obtiene el *FirebaseUser* que se ha creado y se actualiza su perfil con el nombre y apellidos del usuarios utilizando la función *updateProfile()* de *FirebaseUser* y pasándole un *UserProfileChangeRequest* con el nuevo nombre. También se registra el nuevo usuario creado en la Database de Firebase en */Usuarios/UserId*. Posteriormente se redirige al usuario hacia *HomeActivity*, ya que el login está implícito en el registro dado que obtenemos el *FirebaseUser*.

Por último, cuando un usuario desea cerrar su sesión dentro de la aplicación, ya sea desde *AccountActivity* o desde el menú lateral ubicado en la *HomeActivity*, se utiliza la función *signOut()* la clase *FirebaseAuth*.

3.2.8 Obtención y compresión de imágenes y almacenamiento en Firebase Storage

Dentro de la app desarrollada existen dos puntos en los que se trabaja con imágenes: cuando se añade un establecimiento para seleccionar la imagen del mismo y cuando se realiza una valoración con foto de un establecimiento.

En ambos casos se realiza el mismo procedimiento para obtener la imagen y posteriormente subirla a Firebase Storage. Este procedimiento comienza con un *AlertDialog* que le pregunta al usuario desde dónde desea obtener la imagen:

- Realizar una fotografía con la cámara

En este caso primero se comprueba si la aplicación tiene concedidos los permisos de *CAMERA* y *WRITE_EXTERNAL_STORAGE* (para realizar fotografías y almacenarlas en el teléfono) mediante la función *checkSelfPermission()* y, en caso de ser necesario, se solicitan estos permisos al usuario mediante *requestPermissions()*.

Una vez se han concedido los permisos se llama al método *dispatchTakePictureIntent()* que obtiene un *File* con una ruta pública donde se almacenará la imagen mediante la función *getOutputMediaFile()* y crea un *Intent* que abre la cámara para realizar la fotografía mediante *MediaStore.ACTION_IMAGE_CAPTURE*, al cual se le pasa una *Uri* obtenida a partir del *File* creado mediante un *FileProvider*.

```

// Esta función crea un intent para abrir la cámara del dispositivo y crea un // Creamos un File para guardar una imagen o un video
// archivo donde guardar la imagen capturada. private static File getOutputMediaFile() throws IOException {
private void dispatchTakePictureIntent(int actionCode) {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    // Creamos el File donde la fotografía irá
    File photoFile = null;
    try {
        photoFile = getOutputMediaFile();
        mCurrentPhotoPath = photoFile.getAbsolutePath();
    } catch (IOException ex) {
        // Error occurred while creating the File
    }

    // Continúa solo si el File fue creado correctamente
    if (photoFile != null) {
        uriFile = FileProvider.getUriForFile(this,
            "com.inac.iosscrespot.sinlosofia.fileprovider",
            photoFile);
        takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, uriFile);
        startActivityForResult(takePictureIntent, actionCode);
    }
}

// Obtenemos un File con el directorio público donde se almacenarán las fotografías
File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_PICTURES), "Sinlosofia");

// Creamos el directorio si no existe
if (!mediaStorageDir.exists()){
    if (!mediaStorageDir.mkdirs()){
        Log.d("Sinlosofia", "failed to create directory");
        return null;
    }
}

// Creamos un nombre para el archivo multimedia
String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
File mediaFile = new File(mediaStorageDir.getPath() + File.separator +
    "IMG_" + timeStamp + ".jpg");

return mediaFile;
}

```

Ilustración 51. Funciones para obtener la imagen capturada con la cámara

La imagen capturada en forma de *Uri* se obtiene en *onActivityResult()* y se le pasa a la función *handleImage()* para continuar con el proceso. Esta imagen, al finalizar dicho proceso, se guarda en la galería mediante el método *galleryAddPic()* que crea un *Intent* con *ACTION_MEDIA_SCANNER_SCAN_FILE*, al que se le pasa la *Uri* obtenida a partir del *File* donde está almacenado la imagen.

- **Seleccionar una imagen de la galería**

Para esta opción se repite el proceso de comprobación y solicitud de permisos anterior, pero en este caso se solicita únicamente el permiso *READ_EXTERNAL_STORAGE* para poder obtener las imágenes de la memoria del teléfono. Una vez los permisos se han concedido se llama al método *dispatchSelectPictureGalleryIntent()* que simplemente crea un *Intent* que permite al usuario seleccionar la fuente desde la que quiere obtener la imagen.

La imagen capturada en forma de *Uri* de este *Intent* se obtiene en *onActivityResult()* y se le pasa a la función *handleImage()* para continuar con el proceso.

A continuación la función *handleImage()* se encarga de:

- Reducir la imagen recibida a un tamaño aproximado de 480x360 píxeles mediante la función *setPic()* para evitar problemas de memoria y para que las imágenes almacenadas en el backend no sean muy pesadas. Para esto, transforma la *Uri* de la imagen en un *InputStream*, calcula el factor de escala en función del tamaño real de la imagen y la escala convirtiéndola en un *Bitmap* mediante la función *decodeStream()* de *BitmapFactory*.

```

// Escala/Reduce la imagen para que tenga un tamaño más pequeño y adecuado para evitar problemas de memoria y que
// las imágenes no sean muy pesadas de cara a su posterior almacenamiento en Firebase.
private Bitmap setPic(Uri uriImg) {
    // Dimensiones objetivo
    int targetW = 480;
    int targetH = 360;

    try {
        // Obtenemos la dimensiones de la imagen
        BitmapFactory.Options bmOptions = new BitmapFactory.Options();
        bmOptions.inJustDecodeBounds = true;
        BitmapFactory.decodeStream(getContentResolver().openInputStream(uriImg), null, bmOptions);
        int photoW = bmOptions.outWidth;
        int photoH = bmOptions.outHeight;

        // Calculamos cuando es necesario reducir la imagen
        int scaleFactor = Math.min(photoW/targetW, photoH/targetH);

        // Decodificamos el inputStream en un bitmap de menor tamaño escalando o reduciendo
        // la imagen de acuerdo con el factor calculado
        bmOptions.inJustDecodeBounds = false;
        bmOptions.inSampleSize = scaleFactor;
        bmOptions.inPurgeable = true;
        Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(uriImg), null, bmOptions);

        return bitmap;
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    return null;
}

```

Ilustración 52. Función que reduce/escala la imagen obtenida

- Comprobar mediante el método *rotateImageIfRequired()* si la imagen tiene la orientación correcta, debido a que en algunos dispositivos como el Galaxy S3 devuelven la fotografía seleccionada girada y, en caso de que suceda, rota la imagen los grados necesarios para corregirla mediante la función *rotateImage()*.
- Establece la imagen en la pantalla para que el usuario pueda visualizarla.

Por último, cuando el usuario pulse en añadir o en valorar el establecimiento y se ejecute el *listener* de del botón, tras realizar todas las comprobaciones necesarias y realizar las inserciones correspondientes en Firebase Database, se procede a subir la imagen seleccionada a Firebase Storage. Para ello, en primer lugar, se genera una clave unívoca para la imagen mediante las funciones *push().getKey()* de *DatabaseReference* y se crea un *StorageReference* que define la ruta en la que se almacenará la imagen en Firebase Storage (para los establecimientos */imgEstbl/photold.jpg* y para los comentarios */imgOp/opld.jpg*). Posteriormente, se convierte el *Bitmap* de la imagen en un *array* de bytes y sube la imagen mediante un *UploadTask* al que se le asigna dicho *array* mediante el método *putBytes()* de *StorageReference*. Una vez finalizado el proceso se llama a las funciones *callback* definidas para *UploadTask* en función del resultado (*onSuccess()* o *onFailure()*).

```
// Obtenemos un Id para la nueva fotografía y la ruta donde se va a guardar la foto
String idPhoto = mDatabase.push().getKey();
StorageReference filepath = mStorage.child("imgOp").child(estblID).child(idPhoto+".jpg");

// Transformamos la imagen seleccionada a un array de bytes
ByteArrayOutputStream baos = new ByteArrayOutputStream();
mImageBitmap.compress(Bitmap.CompressFormat.JPEG,100,baos);
byte[] data = baos.toByteArray();
// Subimos la fotografía a Firebase Storage
UploadTask uploadTask = filepath.putBytes(data);
uploadTask.addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
        // La foto se subió con éxito al Storage de Firebase
        // ...
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {
        // Se produjo un error al subir la fotografía
        // ...
    }
});
```

Ilustración 53. Código que sube la fotografía a Firebase Storage

Por último, me gustaría comentar que para relacionar Firebase Database y Firebase Storage en el sistema desarrollado, se guarda en las entidades o nodos relacionados con las imágenes (Establecimientos, Valoraciones y Fotografías) la ruta donde se encuentra la imagen en Storage, de forma que pueda descargarse mediante dicha ruta.



Ilustración 54. Sincronización entre Database y Storage de Firebase

3.2.9 Descarga de imágenes de Firebase Storage con Glide y FirebaseUI

En la aplicación desarrollada se utiliza la librería de Glide combinada con FirebaseUI para descargar, almacenar en caché y mostrar imágenes en una *view* o vista de forma fácil desde Cloud Storage.

El proceso es sencillo, en primer lugar se obtiene un *StorageReference* con la ruta donde se encuentra la imagen que se quiere descargar y, posteriormente, se llama a Glide especificando en *using()* el cargador *FirebaseImageLoader()*, en *load()* la *StorageReference* a la imagen y en *into()* la *view* donde se quiere mostrar la imagen.

Aparte de estas opciones existen muchas más, algunas empleadas han sido: *placeholder()* para fijar una imagen mientras carga, *error()* para establecer una imagen en caso de fallo, *centerCrop()* para escalar la imagen, *listener()* para establecer un *listener* que nos avisará cuando el recurso esté disponible [*onResourceReady()*], etc.

```
StorageReference gsReference = FirebaseStorage.getInstance().getReferenceFromUrl(mDataFiltered.get(position).getPicturePath());
Glide.with(mContext)
    .using(new FirebaseImageLoader())
    .load(gsReference)
    .fitCenter()
    .centerCrop()
    .placeholder(R.drawable.ic_store_mail_directory_green_square_128dp)
    .into(holder.imgEstbl);
```

Ilustración 55. Descarga de imagen de Firebase Storage

Estos métodos se utilizan en diversos puntos de la aplicación como en la ficha del establecimiento, los *custom adapter* de los *RecyclerView* y de *InfoWindow* y cuando se selecciona el detalle de una foto en *FotoDetalleActivity*.

3.2.10 Obtención de direcciones con Google Maps

Para introducir direcciones en la aplicación se hace uso del *Intent* de Google Maps *PlaceAutocomplete* que nos muestra una nueva actividad donde introducir una dirección, prediciendo la dirección deseada a medida que vamos escribiendo.

Este *Intent* se utiliza en la aplicación para introducir la dirección donde deseamos buscar establecimiento y para introducir la dirección al crear un nuevo establecimiento. Además, los *Intents* utilizados están configurados con filtros de la clase *AutocompleteFilter* que limitan los resultados a localizaciones españolas y, en el caso de añadir establecimiento, también a direcciones únicamente.



Ilustración 56. Intent de Google Maps PlaceAutocomplete

El resultado de este *Intent* es un objeto *Place* del que se extrae tanto la dirección como las coordenadas y se obtiene en *onActivityResult()* mediante la función *getPlace()* de la clase *PlaceAutocomplete*.

La decisión de utilizar este *Intent* viene dada principalmente por derivar la responsabilidad de la validación de la dirección a Google, puesto que es un proceso complejo debido a los atributos que la forman (calle, ciudad, provincia, código postal, etc.) y al abanico lingüístico existente en España. Otra ventaja de la utilización de este *Intent* es que facilita a los usuarios la introducción de las direcciones mediante el autocompletado.

3.2.11 Geocoder

La clase *Geocoder* se utiliza para manejar la geocodificación, que es el proceso de transformar una dirección de calle u otra descripción de una ubicación en una coordenada (latitud, longitud), y la geocodificación inversa, que es el proceso contrario. Para realizar este proceso se requiere la conexión a internet, dado que la clase *Geocoder* requiere un servicio de back-end que no está incluido en el *core Android framework*.

En la aplicación desarrollada se hace uso de esta clase en diferentes puntos:

- En la pantalla de búsqueda en:
 - El *listener* del botón buscar para traducir las coordenadas seleccionadas para la búsqueda (ya provengan de una dirección introducida o de la ubicación del usuario) en la dirección de calle con el objetivo de obtener la ciudad para la búsqueda.
 - Cuando se conocen las coordenadas de la ubicación del usuario para obtener la dirección de calle con el objetivo de mostrarla al usuario.

Esto se realiza mediante el método *getFromLocation()*, al que se le pasan las coordenadas a traducir y se obtiene un *array* de *addresses* con la dirección de calle.

- A la hora de añadir un nuevo establecimiento para obtener los diferentes parámetros de la dirección (calle, número, ciudad, etc.) a partir de la dirección completa del *Place*, que nos devuelve el *Intent PlaceAutocomplete* visto anteriormente. En este caso se utiliza el método *getFromLocationName()* de la clase *Geocoder* que recibe como parámetro la dirección del *Place* y, de igual modo, nos devuelve un *array* de *addresses* con los diferentes parámetros de la dirección.

3.2.12 Obtener la localización del usuario

La ubicación del usuario se utiliza para realizar una búsqueda de los establecimientos que se encuentran cerca de su posición. El proceso sucede en *SearchFragment* y comienza con el *listener* del botón *Mi posición*, tras la pulsación del usuario se comprueba si se tienen los permisos necesarios para acceder a la localización (*ACCESS_FINE_LOCATION*), de igual modo que para la cámara y la galería.

Una vez concedidos los permisos, se llama al método *startLocation()* que comienza creando una instancia de *LocationManager* y obteniendo el estado del GPS y la red (*NETWORK_PROVIDER*). En caso de que no estén habilitados los servicios de localización, se muestra un *AlertDialog* solicitando al usuario que los active. En caso contrario, si el GPS está disponible:

- Se solicitan las actualizaciones de ubicación del mismo mediante el método *requestLocationUpdates()* de la clase *LocationManager*, a la cual se le pasa un *LocationListener* que manejará los eventos de la interfaz *LocationListener* a través de las cuatro funciones que define
- Se intenta obtener la última ubicación conocida mediante *getLastKnownLocation()* de *LocationManager* para establecerla como ubicación inicial llamando a la función creada *getLocation()*.

A continuación se comprueba si el *Network Provider* está disponible, en cuyo caso solicitamos también las actualizaciones de ubicación a través de la red de forma similar al GPS, pero se pasa como atributo un nuevo *LocationListener* que manejará los eventos de este proveedor, y si no se ha podido obtener la última ubicación conocida con el GPS se intenta obtener de este proveedor.

En resumen, tenemos dos *LocationListener* que obtienen por separado las ubicaciones del GPS y de la red. Cuando se detecte a través del GPS que la posición del usuario ha cambiado se invocará al método *onLocationChanged()* de su *LocationListener* con esta nueva posición. En este método se procesará la posición llamando al método *getLocation()* y se cancelarán las actualizaciones del *Network Provider*, puesto que ya estamos obteniendo la ubicación del GPS. Por el contrario, si se obtiene una nueva ubicación de la posición del usuario a través de la red, se realiza el mismo proceso pero sin cancelar ninguna de las actualizaciones.

Este planteamiento nos proporciona una serie de ventajas como: una mayor precisión de la posición del usuario e incluso ir actualizándola si el usuario se está moviendo; se utiliza como *provider* principal al GPS (es más preciso) puesto que si se reciben actualizaciones del GPS se cancela las actualizaciones del *Network Provider*, y obtiene la ubicación a través de la red en situaciones complicadas para el GPS como por ejemplo en edificios, túneles, etc.

```

// GPS
//Obtenemos la localización
//Definimos un location listener para recibir las actualizaciones del GPS
locationListenerGps = new LocationListener() {
    //LOCATION LISTENER METHODS
    //Esta función se llama cada vez que la posición del usuario cambia
    public void onLocationChanged(Location location) {
        //Actualizamos la interfaz con la nueva posición del usuario y almacenamos la nueva ubicación
        getLocation(location);
        //Cancelamos la actualización de localización del NETWORK_PROVIDER
        locationManager.removeUpdates(locationListenerNetwork);
    }

    public void onProviderDisabled(String provider) {
        //Log.d("Sinlosofia","GPS - onProviderDisabled");
    }

    public void onProviderEnabled(String provider) {
        //Log.d("Sinlosofia","GPS - onProviderEnabled");
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
        //Log.d("Sinlosofia","GPS -onStatusChanged");
    }
};

// Solicitamos que queremos recibir las actualizaciones de localización periódicamente para actualizar la posición del usuario
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, MIN_TIME_FOR_UPDATE, MIN_DISTANCE_CHANGE_FOR_UPDATE, locationListenerGps);

//Intentamos obtener la última posición conocida para establecerla como posición inicial
Location lastLocation = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
if (lastLocation != null) {
    getLocation(lastLocation);
}

```

Ilustración 57. Obtención de la ubicación del usuario del GPS

También se define una función que transcurridos 45sg comprueba si se ha obtenido alguna localización, en caso negativo se informa al usuario de un error y se cancela las actualizaciones de ambos *provider*.

El método *getLocation()* recibe las coordenadas de la ubicación del usuario, utiliza la clase *Geocoder* para traducir las coordenadas obtenidas a la dirección de calle correspondiente y muestra por pantalla la localización en texto al usuario. Por último, las actualizaciones de ambos *provider* son deshabilitadas cuando la pantalla pasa a un segundo plano.

3.2.13 Búsqueda de establecimientos y visualización de resultados en mapa

La búsqueda de establecimientos en la aplicación se realiza en función de tres parámetros que el usuario debe especificar:

- La dirección donde desea buscar: el usuario puede introducir una dirección mediante el *Intent PlaceAutocomplete* de Google o puede elegir su ubicación como dirección para realizar la búsqueda
- Las características de restricciones que debe cumplir el establecimiento
- La puntuación mínima que debe tener el establecimiento

Este proceso de búsqueda y filtrado de establecimiento se realiza de forma compartida entre el backend de Firebase y la aplicación Android. Una vez el usuario ha introducido los parámetros necesarios para realizar la búsqueda y pulsa sobre buscar, el proceso comienza obteniendo la ciudad de la dirección (introducida u obtenida) mediante la clase *Geocoder* y se construye un índice mínimo y máximo, compuesto por la *ciudad_puntuaciónSeleccionada* y *ciudad_5.01* (dado que la puntuación máxima de un establecimiento es 5), para realizar la búsqueda en el backend. Acto seguido, se obtienen de la base de datos todos los establecimientos que se encuentran en la ciudad especificada en la dirección y tienen como mínimo la puntuación seleccionada por el usuario, es decir, aquellos cuyo índice de búsqueda se encuentran dentro del intervalo compuesto por el índice máximo y mínimo. Cuando el cliente obtiene la respuesta del backend realiza un nuevo filtrado, guardando los establecimientos que cumplen los filtros introducidos y descartando el resto.

```

// PROCESO DE BÚSQUEDA
// Construimos el índice de búsqueda mínimo y máximo usando la ciudad y
// la puntuación que ha introducido el usuario
String indexMin = ciudadBusqueda + " " + ratingBar.getRating();
String indexMax = ciudadBusqueda + "_5.01";
//Realizamos la búsqueda en la Database de Firebase
Query query = mDatabase.child("Establecimientos")
    .orderByChild("indice_búsqueda").startAt(indexMin).endAt(indexMax);
query.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        //Borramos la lista de establecimientos encontrados anteriormente
        establecimientosList.clear();

        //Recorremos cada uno de los establecimientos encontrados en la base de datos
        // y los introducimos en la lista
        for(DataSnapshot estblSnapshot : dataSnapshot.getChildren()) {
            Establecimiento estbl = estblSnapshot.getValue(Establecimiento.class);
        }
    }
});

```

```

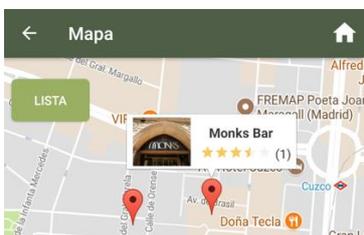
// Filtramos los establecimientos en función de los filtros seleccionados
// por el usuario
boolean cumpleReq = true;
if(cb_comidaSG.isChecked())
    if(!estbl.isComida_SinGluten())
        cumpleReq = false;
if(cb_comidaSL.isChecked())
    if(!estbl.isComida_SinLactosa())
        cumpleReq = false;
if(cb_comidaSA.isChecked())
    if(!estbl.isComida_diabeticos())
        cumpleReq = false;
if(cb_bebidaLecheSL.isChecked())
    if(!estbl.isBebida_lecheSL())
        cumpleReq = false;
if(cb_bebidaRefrescosZero.isChecked())
    if(!estbl.isBebida_refrescosZero())
        cumpleReq = false;

if(cumpleReq)
    establecimientosList.add(estbl);

```

Ilustración 58. Proceso de búsqueda

Una vez se han obtenido los establecimientos que cumplen los requisitos del usuario estos se envían a *MapsActivity*, que utiliza el fragment *SupportMapFragment* de Google Maps para mostrar un mapa en el que crea un *Marker* con una *InfoWindow* personalizada por cada establecimiento mediante el método *addMarker()* y un *adapter* personalizado *CustomInfoWindowAdapter*.

Ilustración 59. Creación de *markers* en el mapa

La decisión de realizar este proceso de forma compartida y con el uso de índices es porque Firebase solo permite indicar un único método de ordenación en las consultas, no permitiendo filtrar por varios nodos (de ahí que se haya construido un índice con la ciudad y la puntuación), y porque no era factible construir un índice que implicara a los tres parámetros, debido a la variabilidad de los filtros.

3.2.14 Valoración y actualización de puntuación media

El cálculo de la valoración media de cada establecimiento se realiza con cada nuevo comentario. Tras añadir el nuevo comentario del establecimiento a la base de datos y subir la fotografía de la valoración a Firebase Storage (en caso de que tenga), se realiza una transacción con la base de datos de Firebase para actualizar la puntuación media. Las transacciones aceptan dos argumentos, una función de actualización y una función opcional de finalización. La función de actualización obtiene el estado actual de los datos como argumento y genera el nuevo estado que deseas escribir. Si otro cliente escribe en la ubicación antes de que el nuevo valor se escriba correctamente, se llama de nuevo a la función de actualización con el nuevo valor actual y se vuelve a intentar la operación de escritura.

Para llevar a cabo esta transacción, se obtiene una *DatabaseReference* de la ruta del establecimiento (*/Establecimientos/estblIID/*) y se llama a la función de *runTransaction()* definiendo las funciones de actualización *doTransaction()* y de finalización *onComplete()*. En la función de actualización se recibe el valor actual del establecimiento, se obtiene el número de puntuación y el número de opiniones actual y se calculan de nuevo con los datos de nueva valoración. Una vez calculados se forma el nuevo índice de búsqueda y se actualizan los tres valores en el establecimiento que se devuelve para que se escriba correctamente.

```

//Actualizamos la puntuación media del establecimiento mediante una transacción
DatabaseReference postRef = mDatabase.child("Establecimientos").child(estblID);
postRef.runTransaction(new Transaction.Handler() {
    @Override
    public Transaction.Result doTransaction(MutableData mutableData) {
        //La transacción
        Establecimiento estbl = mutableData.getValue(Establecimiento.class);
        //Si la transacción no puede obtener los datos puede devolvernos valores nulos
        if (estbl == null) {
            return Transaction.success(mutableData);
        }

        //Calculamos la suma total de todas las puntuaciones
        double sumaPuntNew = estbl.getPuntMedia()*estbl.getNumOp() + ratingOp.getRating();
        //Calculamos el número de comentarios
        double numOpNew = estbl.getNumOp() + 1;
        //Obtenemos la nueva puntuación media
        double puntMediaNew = sumaPuntNew / numOpNew;
        //Redondeamos a dos decimales la nueva puntuación media
        double puntMediaNewRound = Math.round(puntMediaNew * 100.0) / 100.0;

        //Construimos el nuevo índice
        String indexSplit [] = estbl.getIndice_busqueda().split("#");
        String indiceNew = indexSplit[0]+"#"+puntMediaNewRound;

        //Actualizamos los nodos del establecimiento
        mutableData.child("indice_busqueda").setValue(indiceNew);
        mutableData.child("puntMedia").setValue((float)puntMediaNew);
        mutableData.child("numOp").setValue((int)numOpNew);
        return Transaction.success(mutableData);
    }

    @Override
    public void onComplete(DatabaseError databaseError, boolean b,
        DataSnapshot dataSnapshot) {
        // Transacción completa
        //Log.d("sinlosofia", "postTransaction:onComplete:" + databaseError);
        //...
    }
});

```

Ilustración 60. Actualización de la puntuación mediante una transacción

Este procedimiento nos permite actualizar la puntuación media de los establecimientos y el resto de valores asociados de manera atómica, de forma que los datos siempre sean coherentes y no permitiendo que se produzcan cruces entre los clientes en la actualización de los datos.

La solución inicial a este problema se planteó mediante el método *updateChildren()* que nos permite escribir en diferentes nodos de forma atómica. Sin embargo, presentaba problemas de escalabilidad debido a que se calculaba la puntuación media obteniendo todos los comentarios del establecimiento y no permitía realizar la lectura y escritura de forma atómica.

3.2.15 Custom Adapters y filter

Todos los *RecyclerView* que se utilizan en la aplicación son manejados por *custom adapters* creados para utilizar *layout* personalizados como *row* o celdas. Estos *adapters* han sido creados para manejar los *RecyclerView* que muestran Establecimientos, Valoraciones de los usuarios y Fotografías de las valoraciones.

Además, el *adapter RecyclerViewAdapterEstbl* implementa la interfaz *Filterable* que nos permite filtrar en función del nombre, dirección o ciudad los establecimientos mostrados en el *RecyclerView* del mediante la función *getFilter()* de la interfaz. Esta función se llama desde un *SearchView* ubicado en la *ActionBar* cada vez que se escribe un carácter. Las pantallas que presentan esta funcionalidad son la lista de favoritos y de recomendados o pendientes y la pantalla de resultados de búsqueda en forma de lista.

La razón de utilizar *RecyclerView* en la aplicación, en lugar de *ListView* y *GridView*, es porque recicla las vistas de los elementos anteriores que no se están visualizando en la pantalla para utilizarlo en los próximos elementos, lo que permite una mejora y optimización de su rendimiento sobre todo ante grandes conjuntos de datos.

3.2.16 Parcelable

Android nos permite enviar datos de tipos básicos o primitivos de una activity a otra por medio de los *Intent*. Sin embargo, en nuestra aplicación es necesario ampliar esta funcionalidad para poder enviar un objeto de la clase *Establecimiento* de una activity a otra, puesto que se trata de una clase de datos clave en el desarrollo y es utilizada por varias pantallas.

Para conseguir esto, hemos hecho que la clase *Establecimiento* implemente la interfaz *Parcelable* que permite que las instancias de dicha clase se puedan escribir (*writeToParcel()*) y restaurar

(*readFromParcel()*) desde un *Parcel*. Además de los dos métodos anteriores se han implementado: un constructor que recibe un *Parcel*, *describeContents()* y un *Parcelable.Creator* con los métodos *createFromParcel()* y *newArray()*.

Una vez hecho esto ya podemos enviar objetos de Establecimiento mediante *putExtra()* o *putParcelableArrayListExtra()* de la clase *Intent* y recibirlos mediante *getParcelableExtra()* o *getParcelableArrayListExtra()* de la misma clase.

3.3 Pruebas

Para garantizar el correcto funcionamiento del sistema se llevarán a cabo un conjunto de pruebas sobre la aplicación a lo largo de toda la fase de desarrollo. Las pruebas a realizar serán de los siguientes tipos:

- Pruebas unitarias: para comprobar que cada unidad de código funciona correctamente y eficientemente por separado.
- Pruebas de integración: para verificar que los elementos unitarios que componen el software funcionan juntos correctamente.
- Pruebas finales de aceptación: para asegurar que el sistema desarrollado cumple con todo lo especificado.

Estas pruebas serán realizadas sobre el simulador que proporciona Android Studio con una imagen de un Nexus 5X con Android 6.0. (*Marshmallow*) y sobre tres dispositivos físicos para garantizar el correcto funcionamiento de la aplicación en un entorno real. Estos dispositivos son: Samsung Galaxy S6 con Android 7.0 (*Nougat*), Nexus 5 con Android 6.0.1 (*Marshmallow*) y Samsung Galaxy SIII con Android 4.4.4 (*KitKat*). Por último, las pruebas serán realizadas mayoritariamente de forma manual utilizando un *checklist* con todas las funcionalidades a testear, aunque se automatizarán algunos casos de prueba unitaria o integración mediante el framework de testing Espresso.

3.3.1 Pruebas manuales

A lo largo de la fase de implementación se ha ido definiendo un *checklist* de pruebas (unitarias, de integración y de aceptación) que cubre los aspectos fundamentales de la aplicación. Estas pruebas han sido realizadas manualmente, de forma unitaria y en conjunto, a lo largo del desarrollo de la aplicación para verificar el correcto funcionamiento de sus funcionalidades. El *checklist* de pruebas resultante ha sido el siguiente:

Pantalla	Prueba
SplashScreen	Muestra la <i>SplashScreen</i>
	Verifica el <i>token</i> del usuario pre-logado con Firebase Database
Login	Validación de credenciales
	Recuperación de contraseña
	Validación de campos
Registro	Crea un nuevo usuario
	Guarda el usuario en Firebase Database correctamente
	Registrarse con un email ya registrado
	Validación de campos
Home	Recomienda la app
	Muestra información
	Cierra la sesión activa del usuario
Mi cuenta de usuario	Muestra la información del usuario
	Cierra la sesión activa del usuario
Añadir establecimiento	Selecciona una foto desde la cámara
	Selecciona una foto desde una app (galería u otros)
	Selecciona una dirección para el establecimiento
	Validación de campos
	Guarda el establecimiento en Firebase Database correctamente
	Sube la fotografía a Firebase Storage correctamente con un tamaño, peso y resolución adecuada

Buscar establecimiento	Obtiene la ubicación del usuario
	Selecciona una dirección de búsqueda
	Realiza la búsqueda con los filtros seleccionados
Mapa	Muestra en un mapa los resultados de la búsqueda
	Muestra la información y la fotografía del establecimiento seleccionado en el mapa
	Accede a la ficha del establecimiento seleccionado
	Muestra la localización del establecimiento que recibe
Resultados en lista	Muestra los establecimientos resultados de la búsqueda y sus fotografías
	Calcula y muestra la distancia desde la dirección de búsqueda a los establecimientos
	Filtra los establecimientos de la búsqueda
	Accede a la ficha del establecimiento seleccionado
Favoritos	Muestra los establecimientos favoritos y sus fotografías
	Filtra los establecimientos favoritos
	Accede a la ficha del establecimiento seleccionado
Recomendados	Muestra los establecimientos recomendados y sus fotografías
	Filtra los establecimientos favoritos
	Accede a la ficha del establecimiento seleccionado
Ficha establecimiento	Carga la imagen del establecimiento
	Obtiene y establece correctamente si es favorito o pendiente
	El <i>AppBar</i> se expande y se contrae ocultando y mostrando la imagen
	Permite compartir el establecimiento
	Muestra los datos del establecimiento
	Redirige a la vista de mapas para mostrar la dirección
	Redirige al dial con el número premarcado
	Abre un navegador para acceder a la página web
	Obtiene y muestra las 10 últimas fotografías ordenadas o muestra un mensaje si no hay ninguna
	Obtiene y muestra los 3 últimos comentarios ordenados o muestra un mensaje si no hay ninguno
Actualiza la puntuación y el número de opiniones tras valorar el establecimiento	
Más Fotografías	Muestra el nombre, la valoración y el número de opiniones del restaurante
	Muestra un <i>placeholder</i> en la imagen
	Carga todas las fotografías de las valoraciones del establecimiento
Foto Detalle	Establece el nombre del establecimiento en el título
	Muestra un <i>placeholder</i> en la imagen
	Carga la imagen seleccionada
Mas opiniones	Muestra el nombre, la valoración y el número de opiniones del establecimiento
	Carga todos los comentarios del establecimiento
	Carga las fotografías de las valoraciones que las tengan
Valorar establecimiento	Muestra el nombre, la valoración y el número de opiniones del establecimiento
	Selecciona una foto desde la cámara
	Selecciona una foto desde una app (galería u otros)
	Validación de campos
	Guarda el comentario en Firebase Database correctamente
	Guarda la fotografía en Firebase Database correctamente
	Sube la fotografía a Firebase Storage correctamente con un tamaño, peso y resolución adecuada
Actualiza la puntuación y el número de opiniones del establecimiento de forma atómica en Firebase	

Ilustración 61. Checklist de pruebas

Algunos de los muchos errores detectados al realizar estas pruebas durante el desarrollo han sido:

- No funcionaba la opción de realizar una fotografía en Android 7.0 debido a que esta versión requiere utilizar un *FileProvider*.
- En las versiones superiores de Android 6.0 no obtenía la localización al no solicitar los permisos correspondientes.
- La aplicación hacía el autologin siempre que tuviera algún *FirebaseUser* en la caché, sin comprobar si el *token* de dicho usuario era válido y provocando que los accesos a base de datos no tuvieran éxito.

- El *AppBar* expandible que contiene la imagen del establecimiento provocaba un fallo al hacer scroll. Esto hacía que se cerrara la aplicación en las versiones de Android 6.0 debido a un bug de Android y se solucionó añadiendo un atributo al *AppBar* en el *xml*.
- La imagen de los establecimientos en las *InfoWindow* del mapa no siempre se cargaba debido a que no le daba tiempo.



Ilustración 62. Pruebas en S7, Nexus 5 y SIII

3.3.2 Pruebas automáticas

El primer paso para abordar el tema de las pruebas automáticas es decidir el framework de testing a utilizar. Para ello realizamos un análisis de las diferentes posibilidades tecnológicas para automatizar pruebas en Android, obteniendo como resultados cinco posibilidades diferentes que podemos ver en la siguiente imagen:

	Robotium	uiautomator	Espresso	Appium	Calabash
Android	Yes	Yes	Yes	Yes	Yes
iOS	No	No	No	Yes	Yes
Mobile web	Yes (Android)	Limited to x.y clicks	No	Yes (Android & iOS)	Yes (Android)
Scripting Language	Java	Java	Java	Almost any	Ruby
Test creation tools	Testdroid Recorder	UI Automator viewer	Hierarchy Viewer	Appium.app	CLI
Supported API levels	All	16 =>	8, 10, 15 =>	All	All
Community	Contributors	Google	Google	Active	Pretty quiet

Ilustración 63. Comparativa Framework de testing Android
(Fuente: Wikipedia)

En un primer momento se intentó probar Appium, que es un framework muy completo y potente, pero se acabó descartando por problemas en la instalación y configuración. Finalmente, tras analizar el resto de opciones se decidió utilizar Espresso por las siguientes razones:

- Está totalmente integrado con el IDE Android Studio, sin ser necesaria realizar ninguna instalación o configuración adicional
- El framework se encarga por sí mismo de administrar el *thread* dedicado a la UI para hacer que las interacciones esperen el tiempo adecuado para seguir los pasos indicados en el test
- Su sencillez de uso

El framework de Espresso nos permite crear pruebas de interfaz de usuario que se basan en simular las interacciones que los usuarios realizarían sobre una aplicación Android (pulsar sobre un botón, introducir un texto en un campo determinado, etc.) y evaluar las respuestas de la aplicación (comprobar que un texto ha cambiado, que un widget está visible en la pantalla, etc.). Para ello, los test realizados con este framework se basan en una secuencia de sentencias del método *onView()*:

```
onView(ViewMatcher)
    .perform(ViewAction)
    .check(ViewAssertion);
```

Ilustración 64. Método *onView* de Espresso

- **ViewMatcher:** nos permite encontrar la vista de la UI sobre la que queremos actuar en función de diferentes parámetros como propiedades de usuario (como el ID o el texto que contiene), propiedades de UI (como si está visible o habilitado), la jerarquía (como si es hijo o padre de), etc.
- **ViewAction:** nos permite realizar acciones con las vistas encontradas como hacer click, doble click, scroll hasta su posición, etc.
- **ViewAssertion:** nos permite comprobar el estado de una vista principalmente mediante el método *matches()* que nos permite usar los *ViewMatcher* existentes.

Los test creados en Android Studio con este framework son ejecutados por el JUnit4 de Android. Para crear un test de Espresso, una vez tengamos la clase java de test, lo primero que tenemos que hacer es crear una regla a través de la etiqueta *@rule* que especifica la activity a partir de la cual comienza a ejecutarse y realiza todos los test de la clase. A continuación, se escriben todos los test de prueba que se quieren realizar sobre la UI mediante funciones etiquetadas con *@test*.

```
@RunWith(AndroidJUnit4.class)
@SmallTest
public class LoginTest {

    @Rule
    public ActivityTestRule<LoginActivity> mActivityRule = new ActivityTestRule<>(LoginActivity.class);

    @Test
    public void login() {
        // Este test prueba el login con el usuario registrado en el test RegistroUsuariosTest
        // Introducimos las credenciales de usuario
        onView(withId(R.id.etEmail)).perform(replaceText("espresso@gmail.com"), closeSoftKeyboard());
        onView(withId(R.id.etPass)).perform(replaceText("87654321"), closeSoftKeyboard());

        // Pulsamos sobre el botón de iniciar sesión
        onView(withId(R.id.btnLogin)).perform(click());

        // Comprobamos que hemos accedido a la home de la App
        onView(withId(R.id.tvSearch)).check(matches(isDisplayed()));
    }
}
```

Ilustración 65. LoginTest.java

Con este framework hemos automatizado tres pruebas unitarias sencillas que, además de probar la funcionalidad propiamente, nos ha servido para tomar contacto con el funcionamiento y dinámica del framework:

- **RegistroUsuarioTest:** este test prueba la funcionalidad del registro de usuarios creando uno nuevo en la aplicación. Para ello, accede desde login a la pantalla de registro, rellena el formulario con datos válidos y pulsa sobre crear usuario, validando que se acceda a la home de la App tras crear el usuario. La correcta ejecución de este test requiere que no haya ningún usuario ya logado para acceder directamente a la pantalla de login.
- **LoginTest:** se trata de un test muy sencillo que verifica el inicio de sesión con un usuario correcto. Simplemente se limita a introducir las credenciales de usuario, pulsar sobre el botón de iniciar sesión y verificar el login mediante el acceso a la home de la App. Como en el caso anterior, es necesario que no haya ningún usuario pre-logado y que el usuario con el que se intenta acceder (el creado en el test anterior) exista.
- **CerrarSesionTest:** este test comprueba la funcionalidad de cerrar sesión desde el menú lateral. Por tanto, como premisa debe haber un usuario ya logado en la aplicación. El test simplemente consiste en abrir el menú lateral, pulsar sobre la opción de cerrar sesión presente y validar el cierre de la sesión mediante la redirección a la pantalla de login.

Una vez conocimos la dinámica base de este framework, se desarrollaron dos pruebas de integración en las que se verifican varias funcionalidades de la aplicación al mismo tiempo y poseen una mayor complejidad:

- **SearchTest:** siguiendo la dinámica comentada en las pruebas anteriores, este test simula todas las interacciones de usuario necesarias para probar la funcionalidad de buscar establecimientos a través de la ubicación del usuario y va evaluando que las respuestas de la aplicación sean las

correctas. Para una adecuada ejecución del test es necesario estar logado, tener los servicios de localización activos y que existan establecimientos en la ciudad donde se encuentra el usuario. Los aspectos más relevantes de este test son:

- Se conceden los permisos necesarios para acceder a la localización de forma previa en una función etiquetada con *@before*.
- Se ha ampliado la funcionalidad de Espresso para poder seleccionar la puntuación en el *RatingBar*, creando una clase *SetRating* que implementa la interfaz *ViewAction* y define la acción a realizar cuando se llama al método *perform*, que en este caso es seleccionar una estrella.
- **ValorarEstblTest:** como en el caso anterior este test simula las interacciones necesarias para probar principalmente la funcionalidad de valorar un establecimiento favorito y comprueba que las respuestas son correctas. Las precondiciones que necesita este test para ejecutarse correctamente son estar logado y tener al menos un establecimiento guardado en favoritos. Las peculiaridades de este test en comparación con el resto son:
 - Se ha creado un *ViewAction* personalizado *waitId()*, que nos permite esperar hasta que exista un determinado elemento o se cumpla un *timeout*. Esto se usará para esperar a que termine la consulta de favoritos y cargue la lista en la pantalla.
 - Se ha creado otro *ViewAction* *CustomScrollTo()* que nos permite mover el scroll hasta un elemento en un *NestedScrollView*, debido a que *ScrollTo()* no es compatible con este tipo de vista.

Todos estos test automatizados han sido probados sobre la aplicación desarrollada con resultado satisfactorio en diversas ocasiones.

Test Results		7s 329ms
com.imac.josecrespot.sinlosofia.RegistroUsuarioTest	7s 329ms	
registrarUsuario	7s 329ms	
com.imac.josecrespot.sinlosofia.LoginTest	1s 369ms	
login	1s 369ms	
com.imac.josecrespot.sinlosofia.SearchTest	5s 290ms	
realizarBusqueda	5s 290ms	
com.imac.josecrespot.sinlosofia.ValorarEstblTest	15s 880ms	
valorarEstbl	15s 880ms	
com.imac.josecrespot.sinlosofia.CerrarSesionTest	2s 233ms	
cerrarSesion	2s 233ms	

Ilustración 66. Ejecución de los test automáticos

3.4. Funcionalidad de la aplicación

A continuación, se muestran las capturas de la ejecución de la aplicación en las que se observan cada una de las funcionalidades definidas en los casos de uso del proyecto.

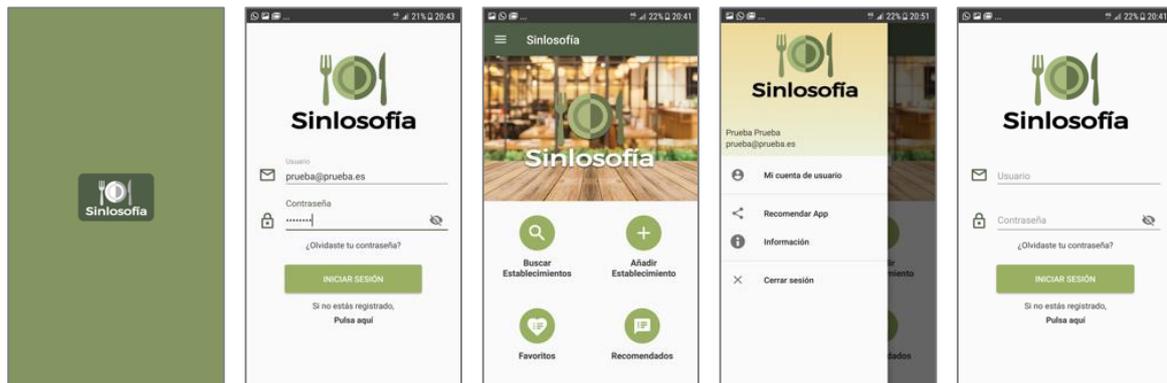
3.4.1 Gestión de usuarios

3.4.1.1 Registro de usuario y visualización de la cuenta del usuario



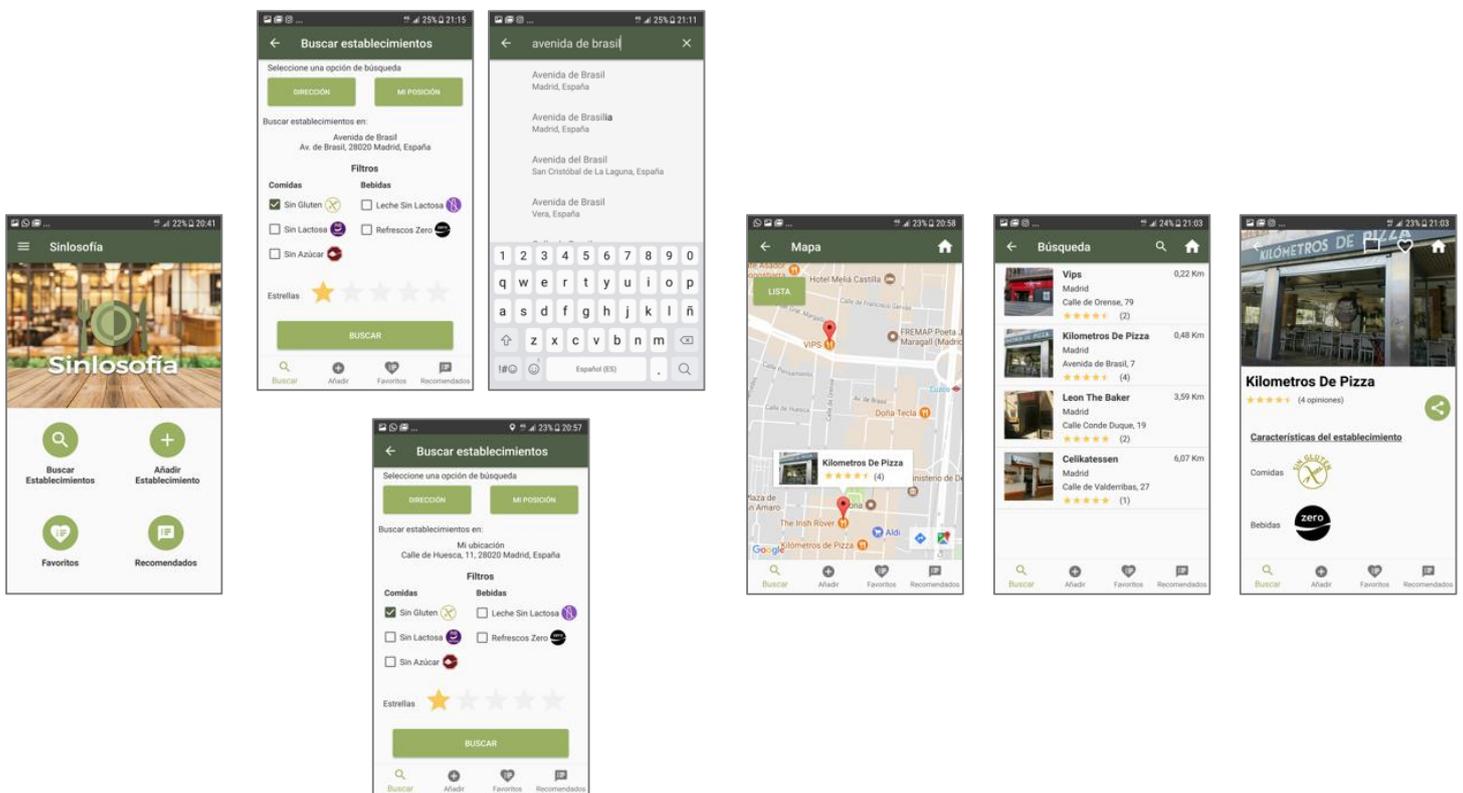
Comentarios: al realizarse el registro se realiza automáticamente el login.

3.4.1.1 Login y cierre de sesión



Comentarios: el cierre de sesión también puede realizarse desde la pantalla de cuenta de usuario

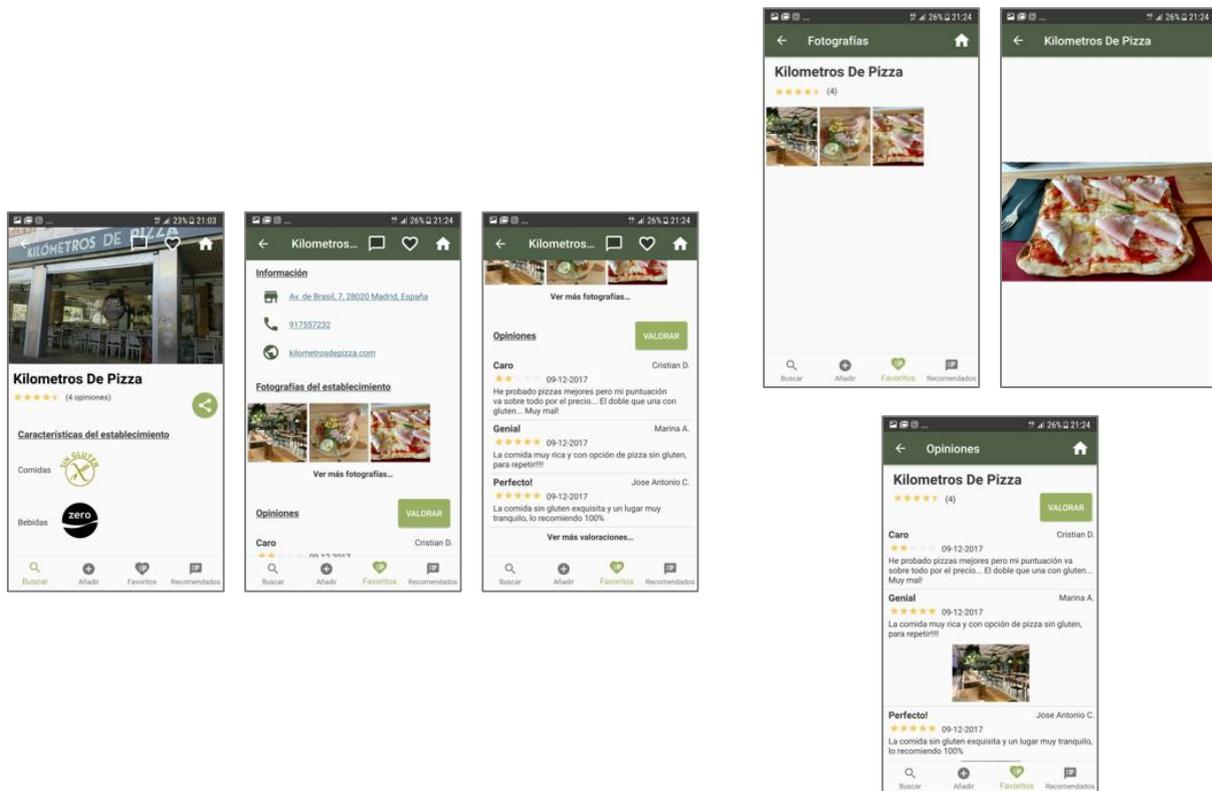
3.4.2 Búsquedas



Comentarios: la secuencia anterior muestra tanto la búsqueda por dirección como la búsqueda por localización y finalmente se accede a uno de los establecimientos buscados.

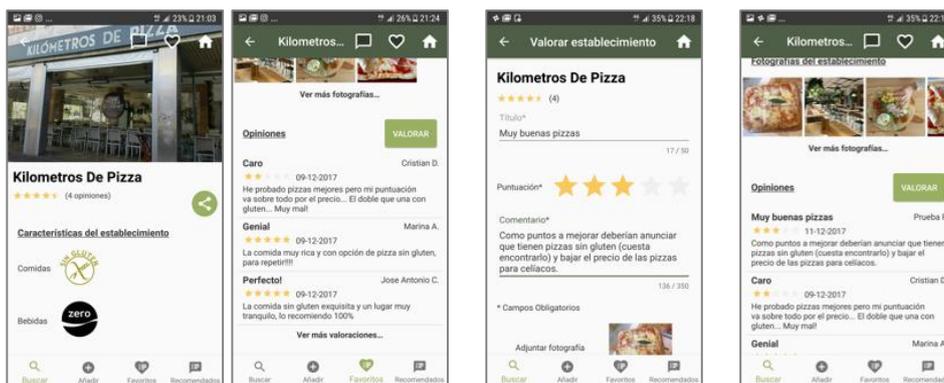
3.4.3 Establecimientos

3.4.3.1 Visualización de los datos, comentarios y fotografías del establecimiento



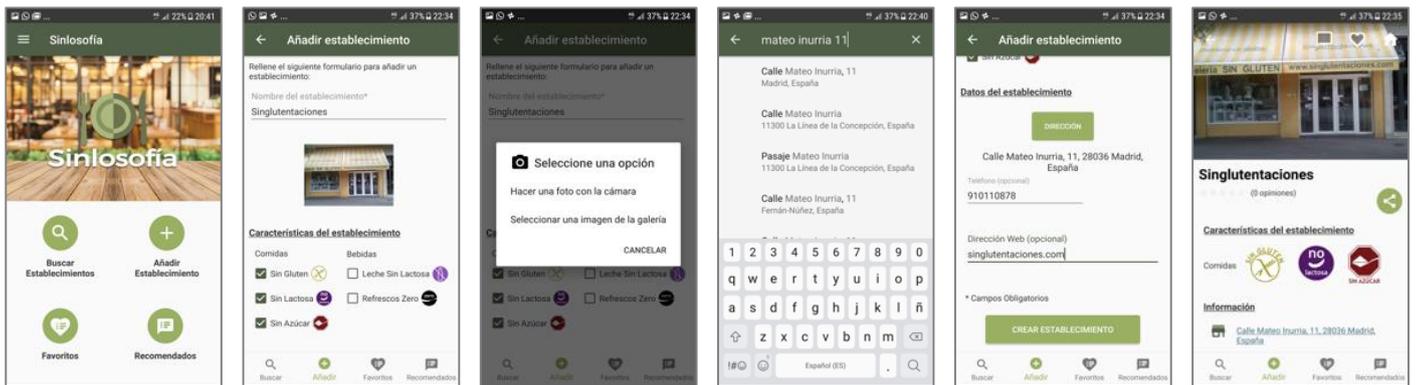
Comentarios: en la ficha del establecimiento puede verse sus características, sus datos (dirección, teléfono y web), las últimas 10 fotografías de las valoraciones y las tres últimas valoraciones y nos permite acceder a todas las fotografías u opiniones del establecimiento.

3.4.3.2 Valorar un establecimiento



Comentarios: para valorar un establecimiento podemos hacerlo desde el botón valorar ubicado en la ficha del mismo o en la pantalla de opiniones del establecimiento. Los comentarios pueden incorporar fotografías del usuario y tras publicar un comentario se actualiza la puntuación y número de opiniones del establecimiento, además de las últimas 10 fotografías y los 3 últimos comentarios mostrados en la ficha.

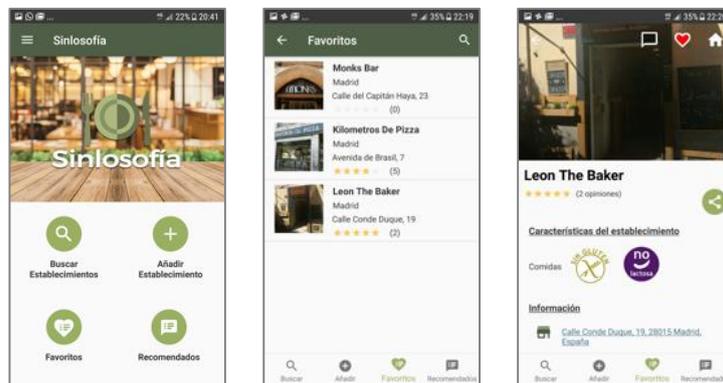
3.4.3.3 Añadir establecimiento



Comentarios: es necesario incluir una imagen del establecimiento para poder crearlo.

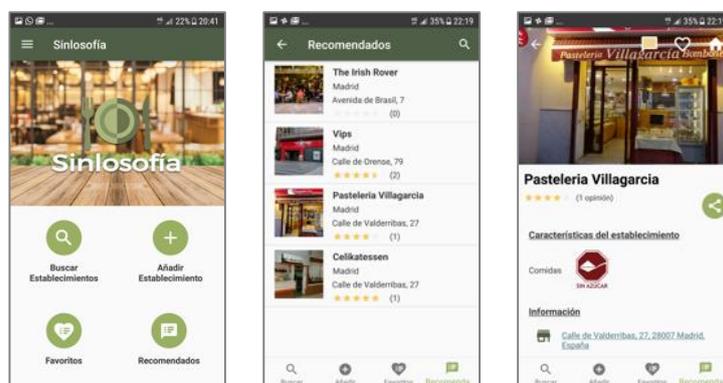
3.4.4 Listas personalizadas

3.4.4.1 Ver tus establecimientos favoritos y añadir o eliminar un establecimiento favorito



Comentarios: para poder añadir o eliminar un establecimiento debemos acceder a la ficha del establecimiento y pulsar sobre el botón en forma de corazón ubicado en la parte superior.

3.4.4.2 Ver tus establecimientos recomendados y añadir o eliminar un establecimiento recomendado



Comentarios: se realiza de igual forma que en favoritos, pero sobre el botón en forma de bocadillo.

4. Conclusiones

Tras la realización del Trabajo Fin de Máster se han obtenido una serie de conclusiones que, a continuación, me dispongo a valorar.

Personalmente, la realización de este proyecto me ha permitido afianzar, practicar y aumentar los conocimientos adquiridos en el máster, carreras anteriores y en mi carrera profesional. En concreto, he podido profundizar en el desarrollo de aplicaciones Android, en la utilización de un backend como Firebase y en la construcción de un sistema compuesto por ambos elementos (frontend – app Android y backend – Firebase).

El desarrollo del proyecto ha sido un gran reto personal al que le he dedicado bastante esfuerzo y que, en mi opinión, he conseguido resolver satisfactoriamente obteniendo un producto funcional, agradable y depurado del que me siento orgulloso. Por otra parte, en la realización de la memoria he intentado justificar todas las decisiones tomadas a lo largo del proyecto, pudiendo comprobar lo importante que es tomar decisiones adecuadas en las fases tempranas para evitar posibles desviaciones en la planificación.

La tarea más costosa y que más dudas ha planteado ha sido la definición de la estructura de la base de datos NoSQL de Firebase, puesto que no disponía de ninguna experiencia previa en este ámbito y porque para trabajar con esta nueva tecnología es necesario abstraerse y dejar a un lado el conocimiento que se tiene sobre las bases de datos relacionales, donde imperan conceptos como las reglas ACID o la normalización.

Al analizar los objetivos y requisitos definidos al inicio del proyecto, se puede comprobar como estos han sido superados en su totalidad. El sistema desarrollado cumple todos los requisitos funcionales y de diseño definidos ofreciendo las funcionalidades principales para una aplicación de este tipo. Además, resaltar también que he intentado trabajar mucho la interfaz gráfica, cuidando multitud de aspectos y creando nuestros propios recursos gráficos a medida, puesto que es un aspecto muy importante en la usabilidad de la aplicación y en la adopción por parte de los usuarios.

A nivel de gestión de proyectos, considero que realizar una buena planificación del tiempo para desarrollar el proyecto es esencial para el éxito del mismo, dicho de otro modo, se deben marcar unos objetivos a cumplir dentro de unas fechas concretas y tener ciertos aspectos implementados, así como partes de la memoria realizadas.

En nuestro proyecto la planificación nos ha permitido cumplir con las fechas de entrega fijadas y realizar seguimiento del mismo para detectar desviaciones y corregirlas lo antes posible. En general, la realización del proyecto se ajusta bastante a la planificación temporal mostrada en el diagrama de Gantt, permitiéndonos alcanzar los objetivos definidos para cada etapa del proyecto sin retrasos en las fechas estipuladas.

Sin embargo, en la fase de implementación se han producido pequeños desvíos que han implicado medidas correctivas como aumentar el número de horas de dedicación diaria hasta las 6 horas en día laboral y hasta las 12 horas en día festivo o fin de semana, debido a la gran carga de trabajo de esta fase del proyecto, a una mayor carga laboral en este período y a un viaje realizado que recortó algunos días a la planificación. Esto también ha permitido añadir algunos puntos extra a la aplicación que no estaban previstos, como la opción de restaurar la contraseña o permitir elegir al usuario entre realizar una fotografía o seleccionar una imagen de una app como la galería.

Respecto a la planificación de las tareas, únicamente comentar que la tarea de *“Desarrollo de backend”* de esta fase implicó una menor dedicación a la estipulada, siendo finalizada antes de lo previsto gracias a la potencia de Firebase que realiza de forma transparente para nosotros una gran cantidad de tareas del backend. Estas horas fueron aprovechadas para avanzar en la tarea *“Desarrollo App Cliente Android”* que tuvo una mayor carga de lo previsto.

En cuanto a la metodología, considero que el modelo en cascada ha sido una buena elección para un proyecto de estas características, ya que me ha permitido:

- Desarrollar un producto completo en el tiempo estipulado
- Afianzar de nuevo las etapas que deben seguirse a la hora de desarrollar un producto software

- Comprender la importancia que tienen las fases previas de análisis y diseño, con especial mención al DCU que nos sitúa las necesidades y características del usuario en el centro de cada una de las etapas de diseño, ya que van a marcar de manera muy determinante la calidad del producto final y nos va a permitir construir un producto centrado en las necesidades del usuario
- Ayudar a reducir el tiempo total de desarrollo

Todo ello resulta importante para demostrar que el desarrollo de software no sólo consiste en escribir código.

Como conclusión final, me gustaría comentar que el desarrollo del proyecto me ha resultado fascinante e interesante y que cada vez que pruebo el sistema construido me resulta gratificante ver lo que he conseguido realizar. Sin duda, acerté al elegir el área “Desarrollo de aplicaciones sobre dispositivos móviles” para realizar el TFM, puesto que es un ámbito que me apasiona y en pleno auge.

4.1 Líneas de trabajo futuro

Como el Trabajo Fin de Máster tiene una duración estipulada con una fecha límite, en la definición del proyecto hubo que fijar una serie de funcionalidades concretas que ofrecería nuestra aplicación.

Por ello, existen diferentes aspectos y funcionalidades que podríamos añadir para complementar y enriquecer una aplicación de este tipo consiguiendo así un salto de calidad:

- ❖ Incorporar nuevos filtros de restricciones alimentarias como, por ejemplo, cerveza sin gluten.
- ❖ Agregar más opciones en la gestión de usuarios como añadir una fotografía de perfil, modificar tus datos, borrar tu cuenta, etc.
- ❖ Implementar un mecanismo que permita a los usuarios: avisar de errores en la información de los establecimientos (como que la dirección o teléfono de un establecimiento es erróneo) o actualizar los datos de estos (como que un establecimiento respeta un nuevo filtro de comida o bebida).
- ❖ Afianzar el control de errores de la aplicación para evitar excepciones no controladas.
- ❖ Mejorar la escalabilidad del sistema, simplificando los procesos de consulta y actualización de datos lo máximo posible.
- ❖ Añadir más información de los establecimientos, como por ejemplo el horario.
- ❖ Optimizar la aplicación para consumir el menor número de recursos de RAM, CPU y batería posible. Estos tres componentes son críticos en el rendimiento de cualquier sistema móvil y es importante cuidarlos.
- ❖ Internacionalización de la aplicación para que pudiera utilizarse en cualquier ciudad del mundo y soporte diferentes lenguajes.
- ❖ Estudiar un plan de monetización de la aplicación principalmente mediante la inserción de un banner de publicidad (*Publicidad in-app*), aunque también puede analizarse la opción *Fremium* con una app gratuita y otra de pago con mayor funcionalidad.
- ❖ Desarrollar la aplicación para el resto de entornos móviles, principalmente en iOS.

5. Glosario

- **App (*Application*)**. Término utilizado para referirse a las aplicaciones de dispositivos móviles.
- **TFM**. Trabajo Fin de Máster.
- **Android**. Sistema operativo de Google para dispositivos móviles basado en el núcleo de Linux. Se trata de un proyecto de código abierto.
- **Diseño Centrado en el Usuario – DCU**. Filosofía de diseño de productos y aplicaciones que sitúa al usuario en el centro de todo el proceso.
- **Android Studio**. Entorno de desarrollo integrado de Android.
- **Google Play**. Tienda oficial de aplicaciones de la plataforma Android.
- **Backend**. Parte de un sistema que se encarga de procesar la entrada desde la parte que interactúa con los usuarios y devolverle la información que se requiere.
- **Firebase**. Servicio de backend para aplicaciones móviles de Google.
- **Diagrama de Gantt**. Herramienta gráfica que muestra el tiempo de dedicación previsto para cada tarea a lo largo del proyecto.
- **MVC – Modelo Vista Controlador**. Se trata de un patrón arquitectónico de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **Open source**. Modelo de desarrollo de software basado en la colaboración abierta. La idea del código abierto se centra en la premisa de que al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario, es una visión técnica.
- **API - *Application Programming Interface* (Interfaz de Programación de Aplicaciones)**. Conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción
- **Framework**. Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

6. Bibliografía

6.1 Recursos de texto

- ❖ **Documentación Asignatura Ingeniería de la Usabilidad – UOC**
Autores: Amaia Calvo-Fernández, Rodríguez Sergio Ortega Santamaría, Alicia Valls Saez, Mònica Zapata Lluçh, Joan Vinyets i Rejón
Universidad: Universitat Oberta de Catalunya, Máster Universitario en Ingeniería Informática
Fecha publicación: 2015
Licencia: No especificada por lo que asumimos que tiene Copyright
- ❖ **Documentación Asignatura Tecnología y desarrollo en dispositivos móviles – UOC**
Autores: Josep Prieto Blázquez, Julián David Morillo Pozo, Robert Ramírez Vique, Marc Domingo Prieto
Universidad: Universitat Oberta de Catalunya, Máster Universitario en Ingeniería Informática
Fecha publicación: 2016
Licencia: No especificada por lo que asumimos que tiene Copyright
- ❖ **Documentación Asignatura Tecnología y desarrollo en dispositivos móviles – UOC**
Autores: José Ramón Rodríguez, Pere Mariné Jové, David López
Universidad: Universitat Oberta de Catalunya, Máster Universitario en Ingeniería Informática
Fecha publicación: 2016
Licencia: No especificada por lo que asumimos que tiene Copyright
- ❖ **Documentación Ingeniería del Software – USAL**
Autores: Dr. Francisco José García Peñalvo
Universidad: Universidad de Salamanca, Ingeniería Técnica en Informática de Sistemas
Fecha publicación: 2011
Licencia: No especificada por lo que asumimos que tiene Copyright
- ❖ **Documentación Diseño Centrado en el Usuario – UOC**
Autores: Muriel Garreta Domingo, Enric Mor Pera
Universidad: Universitat Oberta de Catalunya, Máster Universitario en Ingeniería Informática
Fecha publicación: 2017
Licencia: No especificada por lo que asumimos que tiene Copyright
- ❖ **Documentación DCU para dispositivos móviles – UOC**
Autores: Jordi Almirall López
Universidad: Universitat Oberta de Catalunya, Máster Universitario en Ingeniería Informática
Fecha publicación: 2013
Licencia: No especificada por lo que asumimos que tiene Copyright
- ❖ **Desarrollo de aplicaciones para Android. Edición 2017**
Autor: Joan Ribas Lequerica
Licencia: © EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2016
- ❖ **Android 4. Desarrollo de aplicaciones**
Autor: Wei-Meng Lee
Licencia: © EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2011

6.2 Recursos web

- ❖ **Celiacos.org** - Licencia: Copyright © 2017 FACE. Todos los derechos reservados. <https://www.celiacos.org/> [Sept. 2017]
- ❖ **Wikipedia** - Licencia: Creative Commons Atribución Compartir Igual 3.0. <https://es.wikipedia.org> [Sept., Oct, Nov. y Dic. 2017]
- ❖ **Lactosa.org** - Autor: Adilac. Licencia: © 2016 – Adilac. <https://lactosa.org/> [Sept. y Nov. 2017]
- ❖ **Fundación para la diabetes** - Licencia: © 2015 Fundación para la Diabetes. <http://www.fundaciondiabetes.org/> [Sept. 2017]
- ❖ **Consumer Barometer** - Autor: Google. Licencia: No especificada por lo que asumimos que tiene copyright. <https://www.consumerbarometer.com> [Sept. 2017]
- ❖ **Amic** - Autor: ditendia. Licencia: No especificada por lo que asumimos que tiene copyright. <https://www.amic.media/> [Sept. 2017]

- ❖ **OpenClassRooms** - Autor: Pablo Dominguez. Licencia: No especificada por lo que asumimos que tiene copyright. <https://openclassrooms.com/> [Oct. 17]
- ❖ **Visual Paradigm** - Licencia: © Copyright by Visual Paradigm. All rights reserved. <https://knowhow.visual-paradigm.com/> [Oct. 17]
- ❖ **Pixabay** - Licencia: CC0 Creative Commons. <https://pixabay.com/> [Oct. 17]
- ❖ **Blog Usabilla** - Autor: Sabina Idler. Licencia: No especificada por lo que asumimos que tiene copyright. <http://blog.usabilla.com/how-user-scenarios-help-to-improve-your-ux/> [Oct. 17]
- ❖ **UOC xWiki** - Autor: UOC, Robert Clarisó, Marc García Guerrero. Licencia: Creative commons 2.0. <http://cv.uoc.edu/webapps/xwiki/wiki/matb1916es> [Nov. 17]
- ❖ **Android Developers** - Licencia: Creative commons attribution 2.5. <https://developer.android.com/index.html?hl=es-419> [Sept., Nov. y Dic 17]
- ❖ **Firebase** - Licencia: Contenido: Creative Commons Attribution 3.0 License; Código de ejemplo: Apache 2.0 License. <https://firebase.google.com/docs/?authuser=0> [Nov. y Dic. 17]
- ❖ **Google Maps API** - Licencia: Contenido: Creative Commons Attribution 3.0 License; Código de ejemplo: Apache 2.0 License. <https://developers.google.com/> [Nov. y Dic. 17]
- ❖ **Hermosa Programación** - Autor: James Revelo. Licencia: Hermosa Programación: +50 Tutoriales Desarrollo Android Copyright © 2017. <http://www.hermosaprogramacion.com/> [Nov. y Dic. 17]
- ❖ **Androidhive** - Autor: Ravi Tamada. Licencia: copyright © 2017 Droid5 Informatics Pvt Ltd. <https://www.androidhive.info/> [Nov. y Dic. 17]
- ❖ **StackOverFlow** - Autor: Users. Licencia: © 2017 Stack Exchange Inc; User contributions licensed under cc by-sa 3.0 with attribution required. <https://stackoverflow.com/> [Nov. y Dic. 17]
- ❖ **YouTube** - Autor: Users. Licencia: No especificada por lo que asumimos que tiene copyright. <https://www.youtube.com/> [Nov. y Dic. 17]
- ❖ **GitHub** - Autor: Users. Licencia: © 2017 GitHub, Inc. <https://github.com/> [Nov y Dic. 17]
- ❖ **Vogella** - Autor: Lars Vogel. Licencia: Creative Commons Attribution-NonCommercial-ShareAlike 3.0. <http://www.vogella.com> [Nov. 17]
- ❖ **Envatotuts+** - Autor: Chinedu Izuchukwu, Paul Trebilcox-Ruiz, Chike Mgbemena, Matthew Kim. Licencia: © 2017 Envato Pty Ltd. <https://code.tutsplus.com> [Nov. y Dic. 17]
- ❖ **Material Pallette** - Licencia: No especificada por lo que asumimos que tiene copyright. <https://www.materialpalette.com> [Nov. 17]
- ❖ **Color Hunt** - Licencia: No especificada por lo que asumimos que tiene copyright. <http://colorhunt.co/c/97616> [Nov. 17]
- ❖ **Blog Ingeniería del Software** - Autor: Jorge Hernan Abad Londoño. Licencia: No especificada por lo que asumimos que tiene copyright. <http://ing-sw.blogspot.com.es/2005/04/tipos-de-pruebas-de-software.html> [Dic. 17]
- ❖ **Android Studio Faqs** - Autor: Andrea Ardions, AS. Licencia: No especificada por lo que asumimos que tiene copyright. <https://androidstudiofaqs.com> [Nov. y Dic. 17]
- ❖ **Java Sample Approach** - Licencia: © 2016–2017 JavaSampleApproach. All rights reserved. <http://javasampleapproach.com> [Nov. y Dic. 17]

6. Anexos

En este capítulo se indican los apartados que son demasiado extensos para incluir dentro de la memoria, que tienen un carácter autocontenido y que complementan el presente documento.

- ❖ **Anexo I – Documentos Entrevistas en profundidad.** En este documento se muestra el guion utilizado y las notas tomadas en las entrevistas realizadas durante la fase de análisis del DCU.
- ❖ **Anexo II – Encuesta.** Este archivo contiene el cuestionario realizado en la etapa de análisis del DCU, así como sus resultados.
- ❖ **Anexo III – Documentos Test de Usuarios.** Este anexo muestra los documentos que el usuario cumplimentó durante el test de usuario en la fase de evaluación del DCU.
- ❖ **Anexo IV – Manual de Usuario.** Este anexo contiene una guía de instalación y configuración de la aplicación y el manual de usuario, dirigidos al usuario del sistema.
- ❖ **Anexo V – Bibliografía web detallada.** Muestra la bibliografía web consultada de forma específica con cada uno de los enlaces utilizados.