

Feedin'

Óscar Cobo Gómez

Ingeniería Informática

Desarrollo de Aplicaciones para Dispositivos Móviles

David Escuer Latorre

Carles Garrigues Olivella

Enero 2024



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Feedin'</i>
Nombre del autor:	<i>Óscar Cobo Gómez</i>
Nombre del consultor/a:	<i>David Escuer Latorre</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2024
Titulación:	<i>Ingeniería Informática</i>
Área del Trabajo Final:	<i>Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Valorar, colaborar y descubrir.</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>La aplicación Feedin' nace del deseo de disfrutar al máximo posible del ocio gastronómico.</p> <p>Actualmente, no hay ninguna forma cómoda que permita a la gente: tener una lista abierta donde registrar las múltiples visitas a establecimientos, opinión individual de los platos, tipos de pedido, etc.</p> <p>Esto, es lo que intenta cambiar Feedin'. La aplicación, ayuda a sus usuarios a llevar un control de los restaurantes que visitan, platos que consumen, permite ver la valoración global de cada plato, ayuda a gestionar los gastos y permite descubrir nuevos restaurantes y platos en base los gustos del usuario.</p> <p>Para que la aplicación sea lo más atractiva posible para el usuario, se ha seguido un diseño centrado en el usuario. En el cual, mediante encuestas y entrevistas se ha ido puliendo el diseño hasta conseguir una interfaz que invita al usuario a utilizarla.</p> <p>Con la primera versión de Feedin' disponible y probada por algunos usuarios, se ve una clara línea de futuras versiones que seguir desarrollando, puesto que la gente muestra un interés activo en utilizar la aplicación. A estas sugerencias, también se le añadirán todos los objetivos no alcanzados en esta primera fase.</p> <p>Aun así, la conclusión es que actualmente la aplicación no es rentable. Se han detectado algunas ideas que permitirían que la aplicación lo fuera monetizando alguna funcionalidad, pero queda realizar un estudio exhaustivo para decidir si el proyecto puede continuar.</p>	
Abstract (in English, 250 words or less):	
Feedin' application was born from the desire to enjoy gastronomic leisure as	

much as possible.

Currently, there is no easy way that allows people to: have an open list where they can write their visits to restaurants, individual review from different dishes, order types, etc.

This is what Feedin' is trying to change. The app helps its users allowing them to have control on which restaurants they visit, dishes they consume, allows them to see the overall review of each dish, helps manage expenses and allows them to discover new restaurants and dishes based on the user's taste.

To make the app as attractive as possible, we have followed the user-centered design. In which, through surveys and interviews, the interface has been polished until archiving one which invites the user to use it.

With Feedin's first version available and tested by some users, there is a clear line of future versions to keep developing, since every user showed a lot of interest in using it. To these suggestions, all the objectives not achieved in this first phase will be added.

Even so, the bottom line is that the app is currently not profitable. A few ideas have been detected that would allow some features of the app to be monetized, but for deciding the future of the app, an exhaustive study remains to be carried out.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	4
1.3 Enfoque y método seguido.....	5
1.4 Planificación del Trabajo.....	6
1.5 Breve resumen de productos obtenidos.....	7
1.6 Breve descripción de los otros capítulos de la memoria.....	7
2. Diseño Centrado en el Usuario.....	8
2.1 Métodos de indagación.....	9
2.2 Perfiles de usuario.....	18
2.3 Point of view statements.....	20
2.4 Prototipo.....	21
2.5 Resultados del test con usuarios.....	34
3. Diseño técnico.....	41
3.1 Definición de los casos de uso.....	41
3.2 Diagrama UML de la base de datos.....	44
3.3 Arquitectura del sistema.....	45
4. Herramientas utilizadas durante el desarrollo.....	47
5. Pruebas de la aplicación.....	49
6. Estado actual del proyecto.....	50
6.1 Objetivos del proyecto.....	50
6.2 Mejoras implementadas y pendientes de implementar.....	52
6.3 Planificación del proyecto.....	54
7. Valoración económica del trabajo.....	55
8. Conclusiones.....	56
9. Glosario.....	58
10. Bibliografía.....	60
11. Anexos.....	63

Lista de figuras

Ilustración 1 - Diagrama de Gantt	6
Ilustración 2 - Cuestionario: Edad	9
Ilustración 3 - Cuestionario: Situación	9
Ilustración 4 - Cuestionario: Gustos	9
Ilustración 5 - Cuestionario: Control de gastos	10
Ilustración 6 - Cuestionario: Gastos	10
Ilustración 7 - Cuestionario: Gestión de gastos	10
Ilustración 8 - Cuestionario: Tipo de pago	10
Ilustración 9 - Cuestionario: Efectivo	11
Ilustración 10 - Cuestionario: Propina	11
Ilustración 11 - Cuestionario: Registro	11
Ilustración 12 - Cuestionario: Acompañantes	12
Ilustración 13 - Cuestionario: Pago en grupos	12
Ilustración 14 - Cuestionario: Cantidad de pedidos	13
Ilustración 15 - Cuestionario: Repetir	13
Ilustración 16 - Cuestionario: Tipos de comida	13
Ilustración 17 - Cuestionario: Gustos	13
Ilustración 18 - Cuestionario: Decisiones	14
Ilustración 19 - Cuestionario: Olvidar	14
Ilustración 20 - Cuestionario: Descubrimiento	14
Ilustración 21 - Cuestionario: Repetir platos	15
Ilustración 22 - Cuestionario: Comparar platos	15
Ilustración 23 - Cuestionario: Valorar platos	15
Ilustración 24 - Cuestionario: Calidad de platos	15
Ilustración 25 - Cuestionario: Aplicación con recordatorios	16
Ilustración 26 - Cuestionario: Aplicación con recomendaciones	16
Ilustración 27 - Cuestionario: Aplicación con control de gastos	16
Ilustración 28 - Cuestionario: Preferencias de funcionalidades	16
Ilustración 29 - Cuestionario: Privacidad	17
Ilustración 30 - Boceto 1	21
Ilustración 31 - Boceto 2	21
Ilustración 32 - Diseño: Home	22
Ilustración 33 - Diseño: Buscador	23
Ilustración 34 - Diseño: Valoraciones del restaurante	24
Ilustración 35 - Diseño: Recomendaciones	25
Ilustración 36 - Diseño: Histórico de valoraciones	26
Ilustración 37 - Diseño: Valoración de platos	27
Ilustración 38 - Diseño: Control de gastos	28
Ilustración 39 - Diseño: Configuración del usuario	29
Ilustración 40 - Diseño: Nuevo tique, restaurante	30
Ilustración 41 - Diseño: Nuevo tique, platos	31
Ilustración 42 - Diseño: Nuevo tique, valoración	32
Ilustración 43 - Diseño: Nuevo tique, precio	32
Ilustración 44 - Diseño: Nuevo tique, resumen	33
Ilustración 45 - Diseño técnico: Diagrama UML	44
Ilustración 46 - Diseño técnico: Arquitectura de la aplicación v1	45

1. Introducción

1.1 Contexto y justificación del Trabajo

Actualmente, si alguien desea llevar un control de los platos favoritos de diversos restaurantes, no queda otra opción que acordarse de ellos o ir apuntándolos en algún listado de forma manual.

Esta situación, se agrava cuando un consumidor disfruta de ir a comer fuera. Puede ocurrir que pierda la cuenta de todos aquellos restaurantes que ha visitado. Incluso, que se vea abrumado de la cantidad de opciones que tiene y no sepa cuál de ellas elegir.

Debido a ello, muchas veces ocurre que la gente se olvida de cuál era su plato favorito de un restaurante. Además, otra cosa que puede pasar es que pidan un plato que les encantaba y noten que la calidad ha disminuido o perciban que su precio ha incrementado.

Si se sigue profundizando en el tema, encontramos que no existe una forma fácil y simple de controlar los gastos de ocio (restaurantes). Desde algunas aplicaciones bancarias, existe la posibilidad de crear un presupuesto que vaya enlazado a esa categoría de gastos. Sin embargo, en la actualidad, cada vez más son los grupos que hacen pagar toda la cuenta a uno de ellos y el resto le hace una transferencia o **Bizum** ^[1] para facilitar la operativa del restaurante. Debido a esto, la funcionalidad que proporcionan las entidades bancarias queda prácticamente inutilizada, puesto que es incapaz de vincular correctamente cual ha sido el gasto de cada individuo. El pagador sufrirá una alta subida en su presupuesto, mientras que el resto no verán afectado su presupuesto.

Con las premisas expuestas, se desea desarrollar una aplicación que permita valorar individualmente cada uno de los platos que se consumen en un restaurante. Para ello, el usuario introducirá la cuenta del restaurante y la valoración de cada uno de los platos que él ha consumido. De esta forma, se podrá tener un control tanto de los gustos como de los presupuestos del usuario.

En futuras versiones, la aplicación también será capaz de ayudar al usuario a elegir qué restaurantes visitar en base a sus gustos. A su vez, enlazará las valoraciones de otros usuarios para hacer recomendaciones de potenciales restaurantes que visitar.

Si se buscan aplicaciones similares no existe ninguna en el mercado que cumpla exactamente con estas especificaciones. De una forma u otra, se podría utilizar un conjunto de aplicaciones que dotaría de la mayoría de ellas.

A continuación, se muestra un listado comparativo de las principales funcionalidades de algunas aplicaciones disponibles en el mercado, comparándolas con las funcionalidades que se desean implementar en esta aplicación:

	Puntos comunes	Puntos fuertes	Diferencias
Just Eat ^[2]	<ul style="list-style-type: none"> - Almacena los pedidos realizados por el usuario. 	<ul style="list-style-type: none"> - Aparecen los platos más populares del restaurante. - Sección de restaurantes más valorados cerca de tu posición. 	<ul style="list-style-type: none"> - Sólo se puede valorar el conjunto del pedido. - Sólo funciona para pedidos a recoger o a domicilio.
Uber Eats ^[3]	<ul style="list-style-type: none"> - Almacena los pedidos realizados por el usuario. 	<ul style="list-style-type: none"> - Permite valorar individualmente cada elemento del pedido. - Aparecen los platos más populares del restaurante. - Sección de restaurantes populares cerca de tu posición. - A veces realiza sugerencias acerca de los lugares a los que más pides. 	<ul style="list-style-type: none"> - Sólo funciona para pedidos a recoger o a domicilio.
Google Maps ^[4]	<ul style="list-style-type: none"> - Filtrar por tipos de comida deseada. - Buscar restaurantes afines por la zona del usuario. 	<ul style="list-style-type: none"> - Es la aplicación con más valoraciones por parte de los usuarios. Por tanto, se encuentra una gran variedad de opiniones. - Aparecen todos los restaurantes, tengan o no valoraciones asociadas. 	<ul style="list-style-type: none"> - No se puede realizar más de una reseña pública por restaurante. - La reseña es general, por lo que la única manera de valorar individualmente cada plato es haciéndolo de manera textual en la reseña. - No suele incluir valoraciones de pedidos a domicilio.
TripAdvisor ^[5]	<ul style="list-style-type: none"> - Filtrar por tipos de comida deseada. - Buscar restaurantes afines por la zona del usuario. 	<ul style="list-style-type: none"> - Suele ser frecuentada por usuarios con mayor nivel de conocimiento gastronómico. Debido a esto, aunque tenga menor cantidad de reseñas, suelen ser más acertadas. - Gran cantidad de filtros de búsqueda. 	<ul style="list-style-type: none"> - No se puede realizar más de una reseña pública por restaurante. - La reseña es general, por lo que la única manera de valorar individualmente cada plato es haciéndolo de manera textual en la reseña. - No suele incluir valoraciones de pedidos a domicilio.

Tal y como se había mencionado anteriormente, utilizando tan sólo una de ellas no se conseguirían las funcionalidades deseadas. Lo ideal, sería utilizar, como mínimo, dos de ellas. Por ejemplo: *Just Eats* y *Google Maps*.

Con esta combinación, mediante *Google Maps* los usuarios buscan restaurantes afines a sus gustos y con *Just Eats* realizarían los pedidos a domicilio/recoger. Pudiendo después dejar reseñas en ambas aplicaciones.

1.2 Objetivos del Trabajo

Requisitos funcionales:

- **RF1:** Permitir valorar todos los platos consumidos.
- **RF2:** Los platos deben poder valorarse un número infinito de veces. Así, se podrá hacer un seguimiento de la calidad y/o cambio en los gustos de los usuarios.
- **RF3:** Se debe introducir el precio al valorar un plato. De este modo, se podrá hacer un seguimiento de precio por visita y de si hay incremento o disminución de precios.
- **RF4:** Debe verse reflejado el gasto del mes actual.
- **RF5:** Debe verse reflejado el tipo de comida que más consume el usuario de forma mensual. Por ejemplo, ha ido tres veces a hamburgueserías y una vez a sushi.
- **RF6:** Debe verse reflejada la cantidad de pedidos en restaurante, a recoger y a domicilio que ha realizado el usuario al mes.
- **RF7:** Se deben visualizar los restaurantes que más frecuenta el usuario.
- **RF8:** Diferenciar las valoraciones en base al tipo de pedido (mesa, a recoger, a domicilio, *riders* [6], etc.) para controlar si la calidad del producto disminuye.
- **RF9:** Apartado de sugerencias donde el usuario pueda acudir cuando no sepa qué comer. Irán apareciendo sugerencias de pedidos anteriores y el usuario podrá aceptar la sugerencia o solicitar otra nueva. Sólo incluirá sugerencias en base a sus gustos. Es decir, mediante lo que ha ido introduciendo en la aplicación.
- **RF10:** Apartado de descubrimiento donde el usuario pueda encontrar restaurantes con gustos afines. Es decir, platos valorados por otros usuarios cerca del rango establecido por el usuario y que coincidan en gustos. Por ejemplo, hamburgueserías con estilo *smash*.
- **RF11:** El acceso y registro a la aplicación debe ser mediante cuenta de *Google*.
- **RF12:** El usuario debe poder hacer una foto con su teléfono móvil y automáticamente se deberían precargar todos los platos dándole la opción al usuario de revisarlo, así podrá: revisar que

los platos son correctos, eliminar los que no ha consumido y dar su valoración de ellos.

Requisitos no funcionales:

- El sistema debe estar preparado para explotar la información de los usuarios.
- El sistema debe permitir el uso de funcionalidades nativas de los dispositivos móviles como la cámara o la geolocalización.
- El sistema debe permitir un gran volumen de datos.
- El sistema debe asegurar que no guarda las contraseñas de los usuarios y que la seguridad es robusta.
- El sistema debe ser ágil y con unos tiempos de respuesta razonables para que la experiencia de usuario sea cómoda.

Plataformas:

- Teléfonos móviles iOS
- Teléfonos móviles Android

1.3 Enfoque y método seguido

Para el proyecto se va a desarrollar un producto nuevo desde cero apoyándose en nuevas tecnologías y librerías populares con soporte de la comunidad.

Este enfoque se debe a que como la aplicación es “sencilla”, se cree conveniente desarrollarla desde 0 con un código que se entienda antes que desperdiciar tiempo investigando productos existentes, entendiendo cómo funcionan, cómo están diseñados, ver que son de libre uso, etc.

Así pues, el trabajo constará de las siguientes partes:

1. Aplicación nativa Android realizada en Flutter.
2. N aplicaciones (microservicios o un monolito) que ejecutarán la lógica de negocio mediante el *framework* Spring Boot de Java.

Donde se quiere centrar el trabajo es en la aplicación móvil. Es decir, en el primer punto. El punto 2 tan sólo dotará de la lógica de negocio pertinente a la aplicación. Por ejemplo: almacenar los datos en base de datos, listar los restaurantes, gestionar el acceso a la aplicación, etc.

Esto, se ha pensado de esta forma para separar lo que sería el Front-end (aplicación móvil) del Back-end (capas de código que se encargarán de la lógica para alimentar al Front-end) puesto que no se tiene mucha experiencia en Flutter y se considera más sencillo realizar la lógica de negocio en un lenguaje conocido (Java) y consumirlo mediante API REST desde el Front-end (Flutter).

1.4 Planificación del Trabajo

De lunes a jueves se espera dedicar alrededor de 2h 30m diarias, lo que resultaría en 10 horas en los días laborables de una semana.

De viernes a domingo se espera dedicar alrededor de 6h diarias. Es decir, 18 horas en los días festivos de una semana.

Con lo cual, estaríamos hablando de unas 28 horas semanales.

A nivel de recursos, se hará uso de lo siguiente:

- Ordenador de sobremesa para la documentación y el desarrollo.
- Apple MacBook para el desarrollo y las pruebas con iOS.
- Teléfono móvil iPhone 11 Pro para las pruebas con iOS conectado al MacBook.
- Varios dispositivos móviles Android (según disponibilidad de voluntarios) para ir probando la versión entregable.
- Cuentas de diversos restaurantes acumulados durante este año.

Si lo traspasamos a un diagrama de Gantt, quedaría la siguiente planificación asumiendo empezar a día de la PEC2:

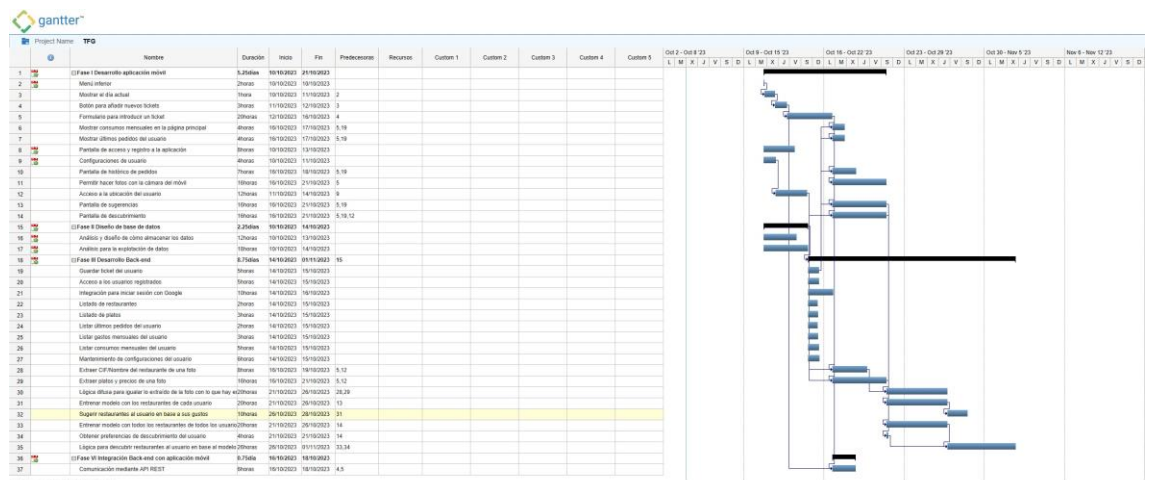


Ilustración 1 - Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

- Fichero con extensión “.APK” para instalar la aplicación móvil en dispositivos Android.
- Memoria de trabajo.
- Manual de compilación.

1.6 Breve descripción de los otros capítulos de la memoria

- **Diseño centrado en el usuario:** Engloba el proceso creativo seguido para decidir cómo se estructuraría la aplicación y cual ha sido la opinión de los usuarios.
- **Diseño técnico:** Define cómo se va a comportar la aplicación y dónde se va a almacenar la información.
- **Herramientas utilizadas durante el desarrollo:** Listado de herramientas que se han aprovechado para facilitar el desarrollo.
- **Pruebas de la aplicación:** Dispositivos con los que se ha probado la aplicación.
- **Estado actual del proyecto:** Compara los requisitos y los tiempos planificados con los reales.
- **Valoración económica del trabajo:** Revisa si el proyecto es o será viable económicamente.
- **Conclusiones:** Qué se ha aprendido con el proyecto.

2. Diseño Centrado en el Usuario

La idea principal que se quiere plantear con el diseño de la aplicación es que sea muy intuitiva de utilizar y con el contenido muy limpio. Es decir, evitar sobrecargar la pantalla del dispositivo con sombras, recuadros, colores, imágenes...

En un principio, la aplicación no mostrará ninguna imagen al usuario, serán todo textos e iconos. Aunque en futuras versiones si que es probable que aparezcan imágenes, se quiere empezar sin estas para facilitar tanto la implementación del código como el diseño de la interfaz de usuario.

Con todo esto, se quiere lograr dar todo el protagonismo posible a las acciones que realiza el usuario y que este disfrute de utilizar la aplicación de lo sencilla e intuitiva que resulta.

En la pantalla principal se pretende mostrar un pequeño resumen de las últimas acciones que ha realizado el usuario y potenciar el apartado colaborativo de la aplicación mostrando el buscador de restaurantes y las opiniones más recientes de los usuarios. El resto de las pantallas, serán dedicadas única y exclusivamente al usuario actual.

Por último, a nivel de colores se va a optar por tonos aturquesados. Esta decisión, viene dada a que es uno de los pocos colores no asociados a una marca similar y que transmiten tranquilidad. Por ejemplo, los siguientes colores quedaban descartados debido a que marcas conocidas ya los utilizan:

- Tonos anaranjados → *Just Eat*.
- Tonos verdosos → *Uber Eats*.
- Tomos amarillentos → *Glovo*.

A estos, se le suman los colores que no son del todo atractivos para el usuario:

- Tonos rojizos → Suelen ser utilizados para acciones delicadas.
- Tonos azulados → Demasiado intensos y dañan la vista.
- Tonos marrones → Dificulta la lectura y no son atractivos.

Con lo visto, el tono elegido era uno de los pocos que veía disponibles.

2.1 Métodos de indagación

Se va a comenzar utilizando una encuesta de Google Forms y, una vez haya liberadas versiones de la aplicación se pasará a la técnica de *shadowing*.

Este último, se utilizará para ver cómo se comporta la aplicación en diferentes dispositivos móviles y cómo interactúa el usuario con ella.

Por otro lado, con la encuesta, como mínimo, se espera aclarar los siguientes puntos:

- Valorar cual es la funcionalidad más deseada por los usuarios.
- Valorar si los usuarios suelen controlar sus gastos.

Los resultados de la encuesta fueron los siguientes:

Apartado “demográfico”:

¿Cuántos años tienes?
45 responses

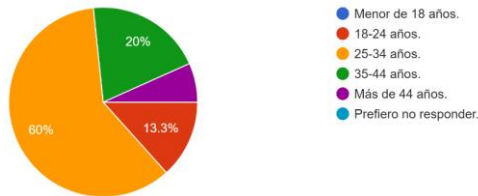


Ilustración 2 - Cuestionario: Edad

¿Cuál es tu situación actual?
45 responses

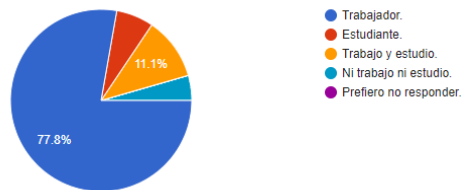


Ilustración 3 - Cuestionario: Situación

¿Te gusta comer en restaurantes?
45 responses

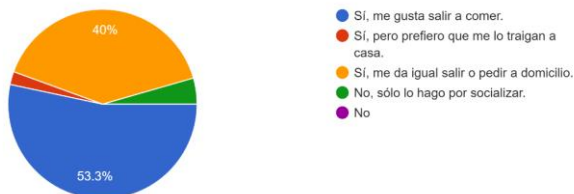


Ilustración 4 - Cuestionario: Gustos

Apartado “Control de gastos”:

¿Llevas un control de lo que gastas en restaurantes?
45 respuestas

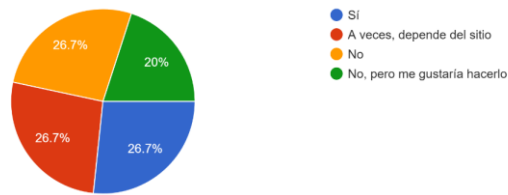


Ilustración 5 - Cuestionario: Control de gastos

Apartado “Gestión de gastos”: (sólo si han respondido positivamente la pregunta anterior)

De media, ¿Cuánto sueles gastar en cada visita?
24 respuestas

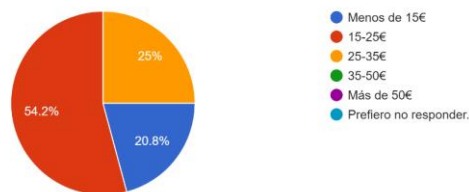


Ilustración 6 - Cuestionario: Gastos

¿Cómo gestionas el gasto cuando es otra persona quien paga?
24 respuestas



Ilustración 7 - Cuestionario: Gestión de gastos

¿Prefieres pagar en efectivo o en tarjeta?
25 respuestas

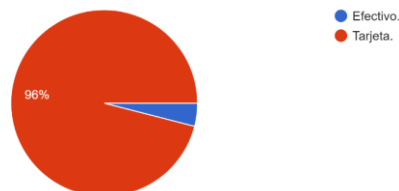


Ilustración 8 - Cuestionario: Tipo de pago

¿Te supone un problema en el control de gastos pagar en efectivo?
24 responses

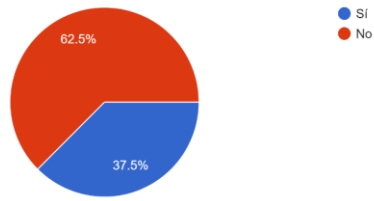


Ilustración 9 - Cuestionario: Efectivo

¿Sueles dejar propina?
25 responses

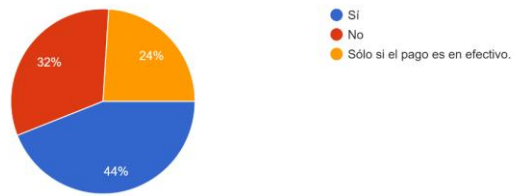


Ilustración 10 - Cuestionario: Propina

¿También registras las propinas y/o los gastos de gestión?
24 responses

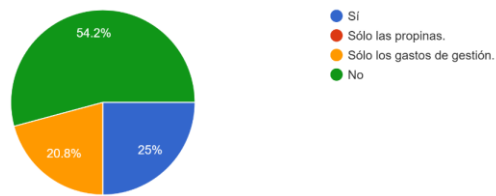


Ilustración 11 - Cuestionario: Registro

Apartado “Acompañantes”:

¿Con quien sueles hacer este tipo de actividades?
45 responses

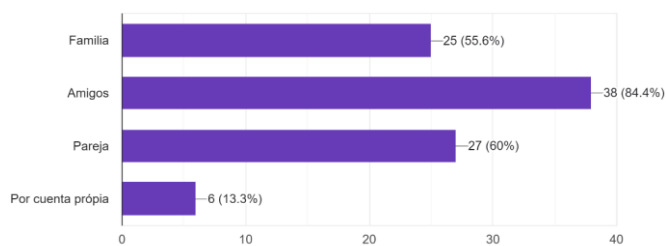


Ilustración 12 - Cuestionario: Acompañantes

Quando vas acompañado, ¿Cómo se realiza el pago?
45 responses



Ilustración 13 - Cuestionario: Pago en grupos

Apartado “Lugares”:

¿Cuántas veces a la semana sueles comer fuera o pedir a domicilio?
45 responses

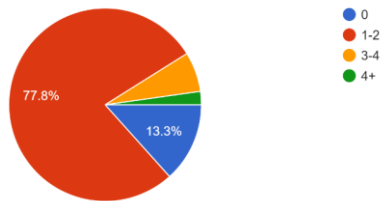


Ilustración 14 - Cuestionario: Cantidad de pedidos

¿Sueles visitar los mismos restaurantes?
45 responses

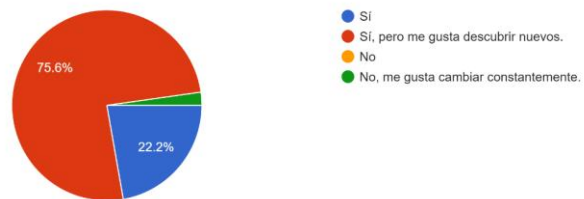


Ilustración 15 - Cuestionario: Repetir

¿Sueles pedir diferentes tipos de comida?
45 responses

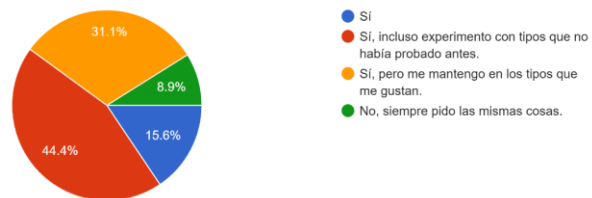


Ilustración 16 - Cuestionario: Tipos de comida

¿Sueles buscar cosas diferentes cuando sales que cuando pides a domicilio?
45 responses

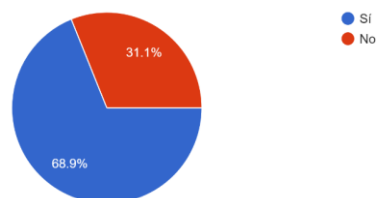


Ilustración 17 - Cuestionario: Gustos

¿Te suele costar elegir dónde ir a comer?
45 responses

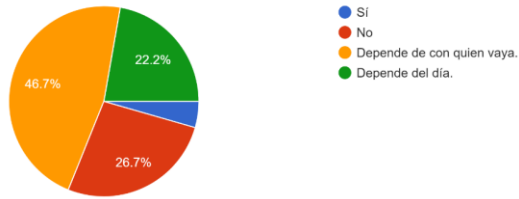


Ilustración 18 - Cuestionario: Decisiones

¿Te sueles olvidar de sitios que te han gustado?
45 responses

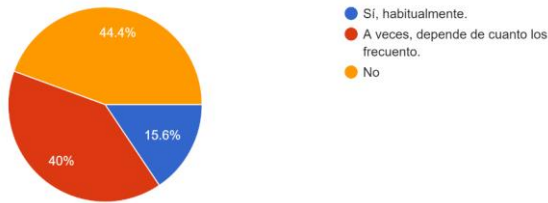


Ilustración 19 - Cuestionario: Olvidar

¿Te interesaría descubrir nuevos restaurantes?
45 responses

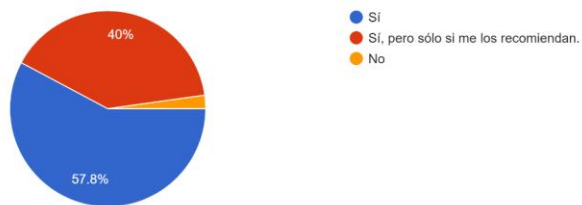
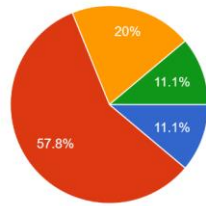


Ilustración 20 - Cuestionario: Descubrimiento

Apartado “Platos”:

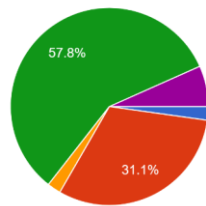
¿Sueles pedir los mismos platos?
45 responses



- Si
- Voy rotando con algunos que sé que me gustan.
- Si, a no ser que me recomienden uno en específico.
- No, me gusta probar toda la carta.

Ilustración 21 - Cuestionario: Repetir platos

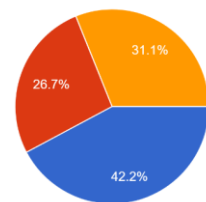
¿Sueles pedir el mismo plato en diferentes sitios?
45 responses



- Si
- Si, para encontrar el sitio que más me gusta.
- Sólo de mi plato favorito.
- A veces.
- No

Ilustración 22 - Cuestionario: Comparar platos

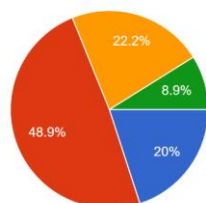
¿Te gustaría poder valorar cada plato individualmente?
45 responses



- Si
- Si, a veces no recuerdo si el plato me gustaba.
- No lo veo necesario.

Ilustración 23 - Cuestionario: Valorar platos

¿Te ha pasado alguna vez que notas que la calidad de un plato ha bajado?
45 responses



- Si, muchas veces.
- Si, ocasionalmente.
- Si, en contadas ocasiones.
- No

Ilustración 24 - Cuestionario: Calidad de platos

Apartado “Aplicación”:

¿Utilizarías una aplicación que te recomienda sitios que has visitado anteriormente?
45 responses

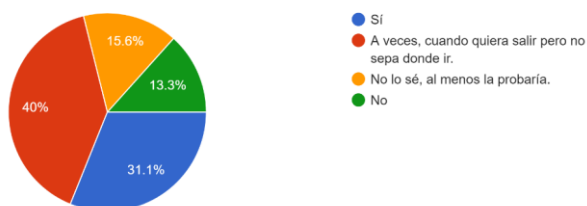


Ilustración 25 - Cuestionario: Aplicación con recordatorios

¿Utilizarías una aplicación que te sugiera restaurantes en base a tus gustos y los del resto de usuarios?
45 responses

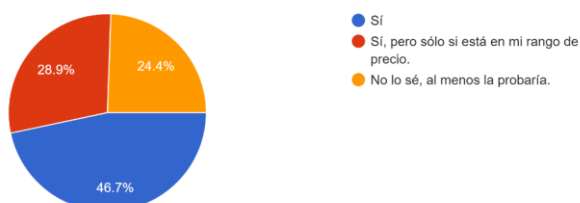


Ilustración 26 - Cuestionario: Aplicación con recomendaciones

¿Utilizarías una aplicación que te ayude a gestionar los gastos mensuales en restaurantes?
45 responses

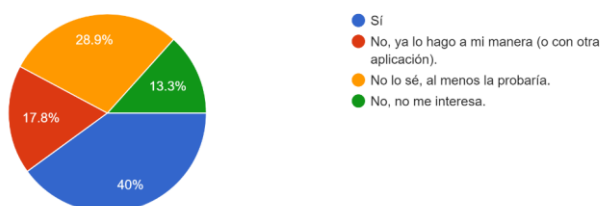


Ilustración 27 - Cuestionario: Aplicación con control de gastos

¿Cuál de estas funcionalidades te parece más interesante?
45 responses



Ilustración 28 - Cuestionario: Preferencias de funcionalidades

¿Te daría vergüenza utilizar la aplicación en lugares concurridos si esta muestra tus gastos?
45 responses

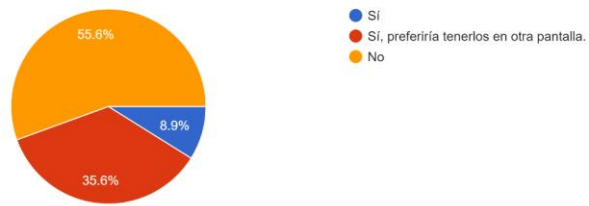


Ilustración 29 - Cuestionario: Privacidad

Gracias a la encuesta, se extrañen las conclusiones:

- Los usuarios ya utilizan aplicaciones para gestionar sus gastos. No hay que darle mucho peso en la aplicación. Se deben permitir registrar propinas y división de cuentas por Bizum.
- Los usuarios no suelen olvidarse de los sitios que visitan. Por tanto, no es necesario darle mucho peso en la aplicación.
- Lo que más valoran los usuarios es conocer nuevos restaurantes y ver las diferentes opiniones de los platos.
- Como mejora, algunos usuarios consideran muy útil diferenciar qué platos son vegetarianos, veganos, no contienen gluten, conocer sus alérgenos, etc.

2.2 Perfiles de usuario

Basándonos en las características demográficas de las respuestas del cuestionario, extraemos 5 perfiles de usuario:

Las fotografías han sido generadas mediante herramientas de IA (<https://thispersondoesnotexist.com/>).

Alicia Pino



Comportamientos

- Utiliza constantemente aparatos electrónicos.
- Estudiosa.
- Realiza todas las compras por Internet.
- No le gusta esperar.

Demografía

- 18 a 24 años (20 años).
- Soltera.
- Estudiante.
- Aspira a trabajar como *project manager*.

Necesidades y objetivos

- Que la valoren.
- Aportar valor a sus seres queridos.
- Pasarlo bien con sus amigos.
- Seguir formándose.

Juan Cobo



Comportamientos

- Le gusta la tecnología.
- Se expresa con facilidad.
- No le gustan las reuniones.
- Sigue una rutina.

Demografía

- 25 a 34 años (28 años).
- Con pareja.
- Trabajador.
- Independizado.

Necesidades y objetivos

- Disfrutar del trabajo.
- Controlar sus gastos mensuales.
- Salir con su pareja.
- Sacarse certificaciones.

Claudia López



Comportamientos

- Le encanta salir de fiesta.
- Entre semana pasa el tiempo estudiando y trabajando.
- Los fines de semana siempre está con sus amigos.

Demografía

- 25 a 34 años (26 años).
- Soltero.
- Trabaja y estudia.
- Vive con sus padres.

Necesidades y objetivos

- Desconectar del trabajo.
- Vivir la vida.
- Aprobar la carrera.

Maya Leal



Comportamientos

- Sus hijos son su prioridad.
- Se centra tanto en las cuentas de la empresa que se olvida de las suyas propias.
- Tiene una agenda muy apretada.

Demografía

- 35 a 44 años (40 años).
- Casada.
- 2 hijos.
- Jefa de una empresa mediana.

Necesidades y objetivos

- Confía en su pareja para encargarse de sus hijos mientras ella trabaja.
- Dedicarle el máximo tiempo posible a su familia.
- Que sus hijos sean felices.
- Que la empresa siga adelante.

Elías Gómez



Comportamientos

- Confía en su equipo.
- Le encanta salir a cenar.
- No le gusta cocinar.

Demografía

- Más de 44 años (52 años).
- Casado.
- 3 hijos.
- Gerente de un equipo de programación.

Necesidades y objetivos

- Llevar un control del gasto en sus cenas.
- Que el equipo esté contento.
- Que sus hijos estudien y saquen buenas notas.
- Cuidar de su salud.

2.3 Point of view statements

POV-1: Las personas que disfrutan constantemente del ocio gastronómico en grupo necesitan una forma de descubrir nuevos restaurantes para disfrutarlos con sus amigos, porque cada mes alguien propone un nuevo sitio que visitar.

POV-2: Los estudiantes con poco capital necesitan mantener un control total sobre su economía y visitas a restaurantes. Por ello, necesitan saber cuánto llevan consumido mensualmente y saber cuánto se suelen gastar en los sitios que frecuentan.

POV-3: Las familias que piden a domicilio para disfrutar con sus hijos, pero también salen a cenar en pareja necesitan diferenciar qué tipos de comida les gusta más pedir y cuales disfrutan más al salir.

POV-4: Los usuarios que quieren probar un nuevo plato agradecen saber la valoración media del resto de usuarios, porque eso les ayuda a elegir cual pedir.

POV-5: Los usuarios que no han disfrutado de un plato necesitan poder recordarse de que no les ha gustado por si en un futuro vuelven a ese restaurante, porque muchas veces pasa el tiempo y no se acuerdan si les gustó o no.

2.4 Prototipo

Los primeros diseños se hicieron a mano alzada a modo de bocetos:

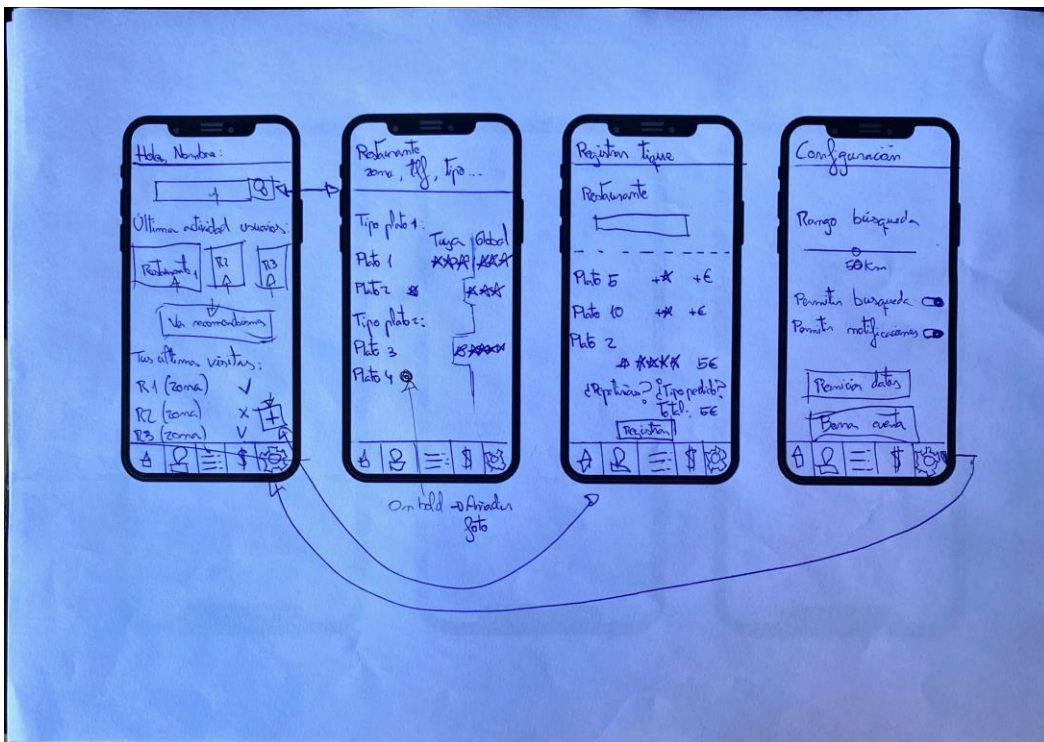


Ilustración 30 - Boceto 1

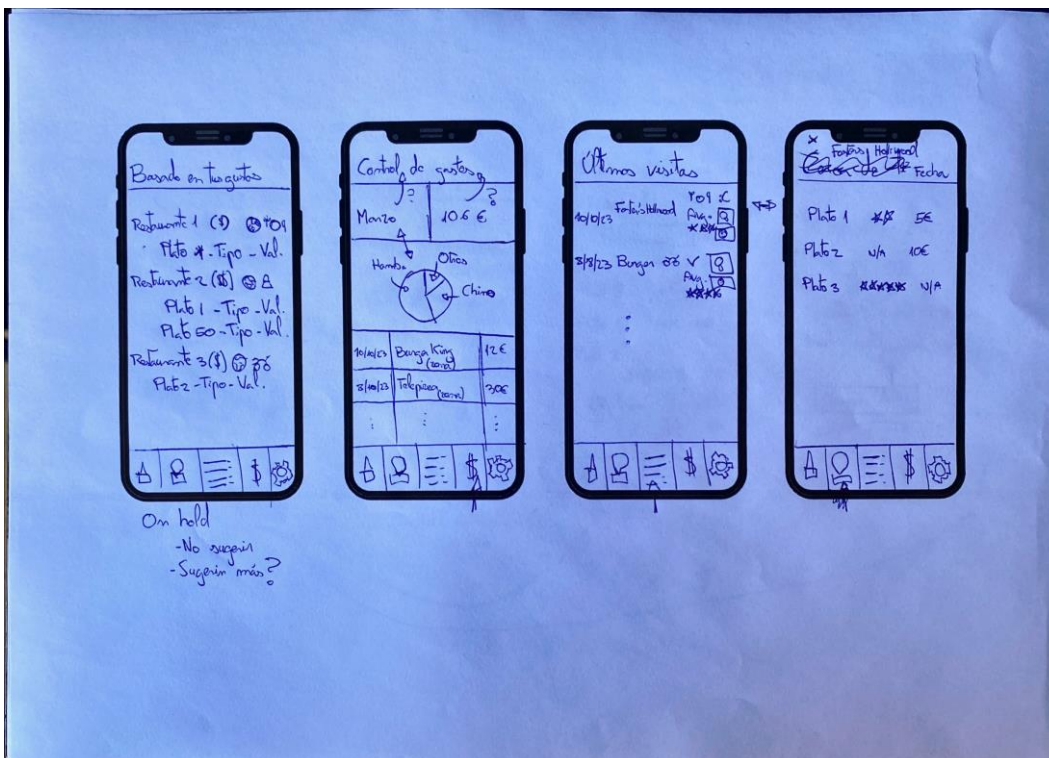


Ilustración 31 - Boceto 2

Tras realizar los bocetos y tener una idea clara de cómo se quería diseñar la aplicación, se realizó un prototipo de alta fidelidad mediante Figma. El cual, es accesible desde el enlace:

<https://www.figma.com/proto/HyKxdAgbfxM6VV6I2qV8MM/Feedin?type=design&node-id=1-3&t=k43FkuXyjYbna1sE-1&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A3&mode=design>

La idea de la página de inicio es que los usuarios tengan facilidad para las siguientes acciones:

- Añadir una nueva valoración
- Buscar restaurantes y sus valoraciones.
- Revisar recomendaciones de sitios basado en sus gustos.
- Última actividad.



Ilustración 32 - Diseño: Home

Cuando el usuario realice la búsqueda de un restaurante, le aparecerán las coincidencias ordenadas por distancia. Es decir, verá primero las más cercanas a su posición.

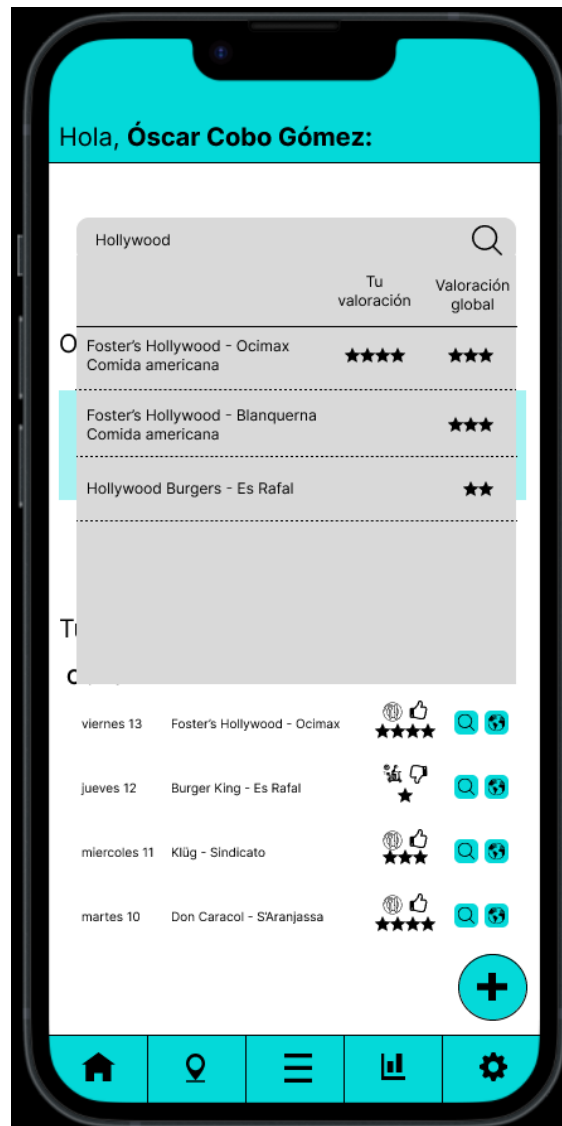


Ilustración 33 - Diseño: Buscador

Una vez el usuario escoge un restaurante del listado, se le mostrarán todos sus platos, incluyendo las valoraciones que el propio usuario ha realizado y las valoraciones globales de estos platos. Así, podrá comparar sus gustos con los de los otros usuarios y atreverse a pedir platos nuevos que los usuarios valoran positivamente.



Ilustración 34 - Diseño: Valoraciones del restaurante

Por defecto, se mostrarán las valoraciones de todos los tipos de pedido. En caso de que el usuario desee revisar las valoraciones para un tipo de pedido en concreto, dispondrá de botones flotantes para navegar entre las valoraciones.

En la parte superior se dispondrá de un enlace con la ubicación en *Google Maps* y otro para llamar al establecimiento.

Si en la página principal se pulsa sobre “Ver recomendaciones” o en el menú inferior se pulsa sobre el segundo icono, se mostrará la página de restaurantes recomendados con la recomendación de cada uno de los platos:

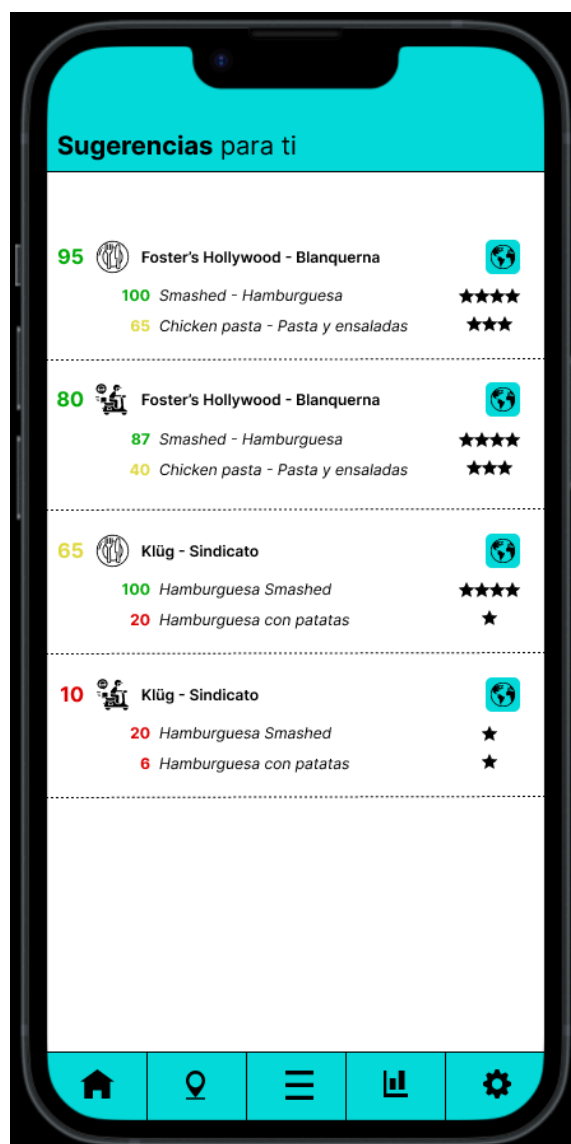


Ilustración 35 - Diseño: Recomendaciones

Como se puede observar, se muestra una puntuación en base a la afinidad con el restaurante y con cada uno de los platos, donde se separa cada restaurante por su tipo de pedido. De este modo, los usuarios tendrán más claro si hacer caso o no a la recomendación, en base al tipo de pedido que quieran realizar. Esta puntuación vendrá dada por:

- Distancia
- Gustos similares (en base a las valoraciones previas del usuario)
- Configuración del usuario.

Esta página es una de las que más dudas genera en cuanto a diseño. Se ha planteado modificarla para que sea más atractiva visualmente y tenga más funcionalidades. Por ejemplo:

- Mostrar un mapa donde los usuarios puedan ir acercando y alejando las ubicaciones que quieren explorar.
- Añadir botones para filtrar por tipo de pedido en lugar de mostrar múltiples veces el mismo establecimiento.
- Mostrar rango de precios.
- Permitir filtros dinámicos como tipo de comida, rango de precio, etc.

Por el momento, se ha optado por este diseño minimalista, donde sólo se mostrarán platos de restaurantes que el usuario aún no ha probado. En otras palabras, si el usuario ha probado todos los platos de un restaurante este restaurante no se le aparecerá en el listado.

Si un usuario desea ver su histórico completo de valoraciones, lo podrá visualizar accediendo desde el tercer botón del menú:



Ilustración 36 - Diseño: Histórico de valoraciones

Desde esta pantalla el usuario encontrará todas sus visitas registradas. Para facilitar su búsqueda, también dispondrá de un buscador en la parte superior de la pantalla.

Para cada visita, se mostrará:

- Tipo de pedido.
- Si le gustó la experiencia.
- Valoración media.

También se le ofrece la opción al usuario de revisar la opinión de cada uno de sus platos:



Ilustración 37 - Diseño: Valoración de platos

Sin embargo, en esta pantalla **no** aparecerá ningún precio. De este modo, al usuario se le mostrará únicamente la información que considera relevante para la valoración.

La siguiente pantalla es la contraria a la anterior, es la que gestiona los gastos y muestra estadísticas del usuario:



Ilustración 38 - Diseño: Control de gastos

En la parte superior se le mostrará al usuario los tipos de comida que ha consumido durante el mes actual y el desglose por tipo de pedido.

En el apartado de estadísticas se podrían añadir leyendas indicando cada tipo de comida con su correspondiente % y número entero de visitas o bien, que al pulsar sobre cada parte aparezca dicha información.

Por otro lado, en la parte inferior se mostrará el precio total registrado en la aplicación de forma mensual, indicando únicamente el tipo de pedido.

Una preocupación con este diseño es que puede pasar que el usuario vea el pedido de 35€ pero no recuerde si le gustó, por lo que tendrá que ir a la página de valoraciones y buscar el pedido por la fecha que se realizó. Quizás se podría añadir algún tipo de enlace para ir entre ambas pantallas y localizar de manera más sencilla el pedido.

La última pantalla corresponde a la configuración del usuario:



Ilustración 39 - Diseño: Configuración del usuario

Como se puede observar, los filtros de búsqueda de momento se almacenarán aquí. Por tanto, no serán dinámicos y los usuarios no podrán modificar sus criterios de búsqueda de forma directa.

Para finalizar, hay que explicar las pantallas del registro de nuevos tiques. A este proceso sólo se puede acceder desde la pantalla principal de la aplicación y comienza solicitando al usuario el restaurante que ha visitado:



Ilustración 40 - Diseño: Nuevo tique, restaurante

Una vez se ha escogido el restaurante, hay que introducir cada uno de los platos. Para facilitar esta gestión, se incluye un buscador:

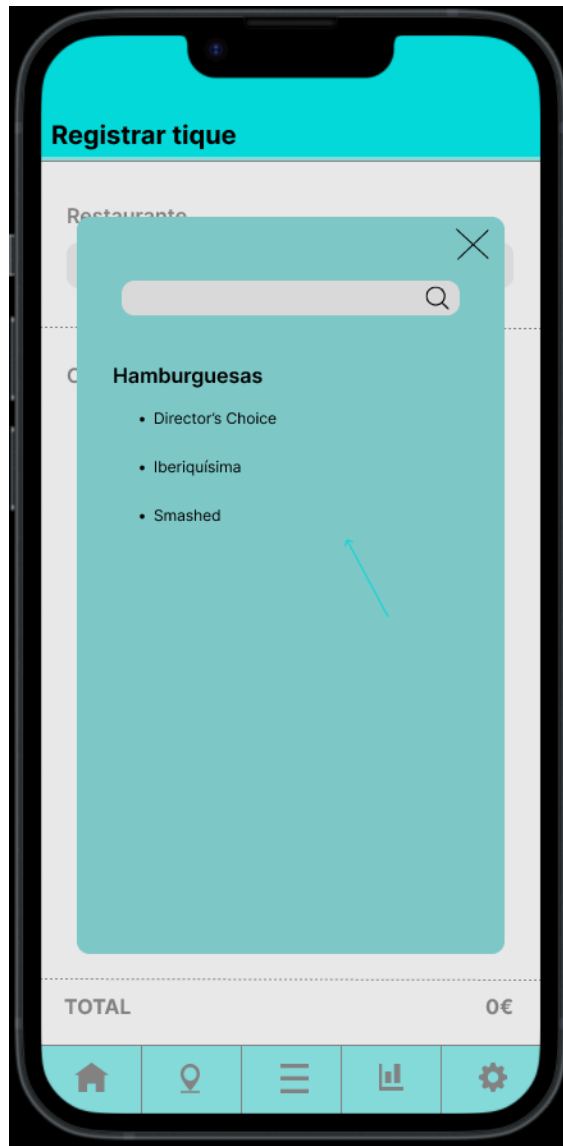


Ilustración 41 - Diseño: Nuevo tique, platos

Tras introducir los platos, el usuario podrá establecer su valoración y su coste. Estas acciones son totalmente opcionales, el usuario podrá:

- No introducir ni valoración ni precio.
- Introducir sólo valoración.
- Introducir sólo precio.
- Introducir valoración y precio.

Esto, se ha diseñado así para abarcar todas las posibilidades que pueden ocurrir cuando realizas este tipo de ocio. Vamos a listar algunos ejemplos:

- En la gran mayoría de los casos, la gente no va a querer valorar un refresco, porque el sabor siempre es el mismo → Se introducirá su precio, pero no una valoración.
- Un amigo ha invitado a la cena → Se puede valorar porque se ha comido pero no se ha pagado → Se introducirá una valoración, pero no un precio.
- El usuario quiere recordar que ha visitado el restaurante, pero no quiere dar información de valoraciones ni precios porque prefiere volver en otro momento → Añade un nuevo tique, pero no introduce ni valoración ni precios.
- El usuario ha consumido y pagado un plato → Puede introducir valoración y precio.

De todos modos, lo ideal siempre será introducir todos los datos posibles para aprovechar al máximo las funcionalidades que ofrece la aplicación.

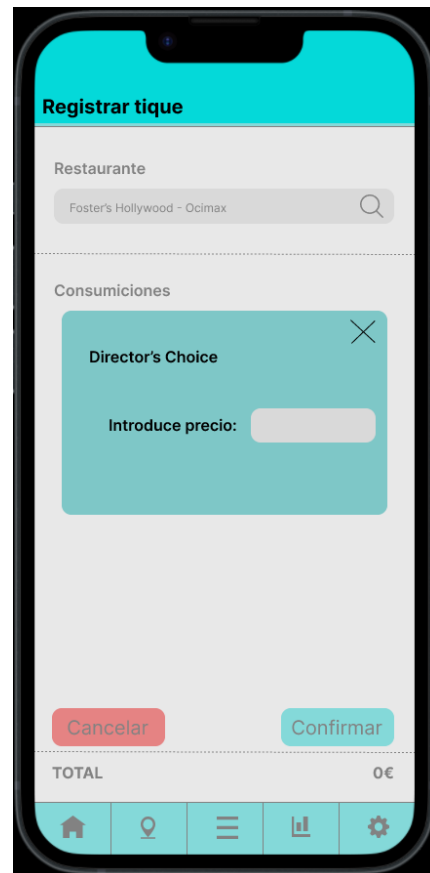
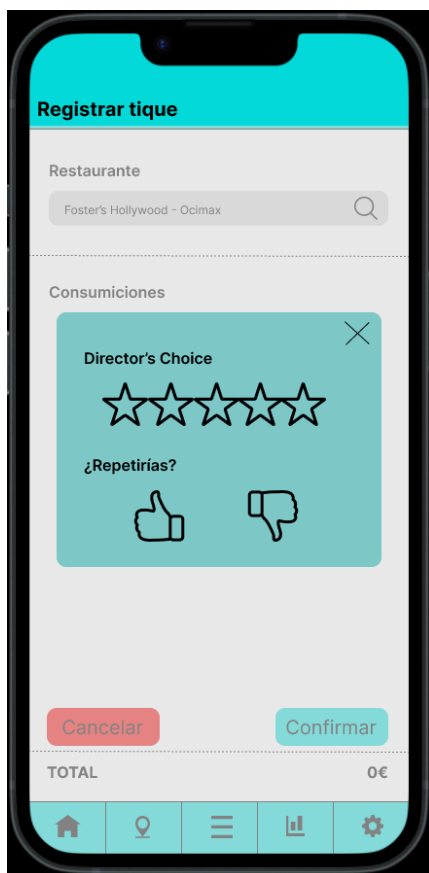


Ilustración 42 - Diseño: Nuevo tique, valoración
Ilustración 43 - Diseño: Nuevo tique, precio

El último paso de este flujo será en el cual el usuario dará los detalles generales y se le mostrará un resumen de toda la información introducida:



Ilustración 44 - Diseño: Nuevo tique, resumen

2.5 Resultados del test con usuarios

Se han llevado a cabo sesiones presenciales con 4 personas utilizando el prototipo horizontal y planteando las siguientes tareas sin un orden específico:

- **F-1:** Tus amigos han sugerido un nuevo sitio donde ir a comer, pero no sabes si ir porque no sabes si te gustará lo que ofrece para comer.
- **F-2:** Estas pensando hacer un pedido a domicilio, pero no sabes si ya has pedido demasiado este mes y deberías controlar los gastos antes de hacerlo.
- **F-3:** Un familiar va a visitar un restaurante que te suena que habías ido hace casi un año y no recuerdas si te gustó lo que pediste.
- **F-4:** Quieres sorprender a tu pareja saliendo a cenar a un sitio diferente del que soléis ir pero que aun así tengas la certeza de que os puede gustar.
- **F-5:** Acabas de salir del restaurante y deseas registrar la visita para no volver más, porque no te ha gustado nada la experiencia.
- **F-6:** Has salido de las pruebas de alergia y te han detectado un nuevo componente alérgeno, ¿cómo lo reflejarías en tu perfil para evitar consumirlo?

Una vez el usuario complete las tareas y pasada su ronda de preguntas y sugerencias, se realizarán las siguientes preguntas si no ha sido el propio usuario quien ya las había comentado previamente:

- ¿Consideras que es intuitivo?
- ¿Cambiarías algún paso de alguna funcionalidad?
- ¿Qué te ha parecido?

	Alicia Pino - Familiar	
	<i>Inputs positivos</i>	<i>Inputs negativos</i>
Concepto general	Le gusta el color y la idea.	N/A
F-1	Le gusta lo sencillo que es acceder a esta búsqueda.	No entiende el icono que indica que la gente repetiría.
F-2	Le gustan los gráficos.	Le ha confundido que la suma del histórico no coincida con el mensual (por cómo se ha preparado el prototipo).
F-3	Le gusta lo sencillo que es acceder a esta búsqueda y que se indique su valoración separada del resto de usuarios.	N/A
F-4	Le parecen acertadas las puntuaciones y que se ordenen de mayor a menor.	N/A
F-5	N/A	No encontraba el botón para registrar el tique y le ha confundido después no poder modificar el restaurante, valoración, tipo de pedido, etc (por cómo se ha preparado el prototipo).
F-6	Le ha resultado intuitivo encontrar la opción.	N/A
Otros	N/A	Considera que se podrían trabajar las siguientes cosas:

		<ul style="list-style-type: none">• Añadir alguna foto.• Los formularios son sosos.• Los tamaños son muy pequeños.• El icono de comer en restaurante no se ve claro.• Las recomendaciones de platos las cerraría en cajas que sólo se mostrasen al pulsar en el restaurante.• Podrían salir en un mapa las recomendaciones.• Las valoraciones podrían tener decimales y mostrar el número total.
--	--	--

	Claudia López - Amiga	
	<i>Inputs positivos</i>	<i>Inputs negativos</i>
Concepto general	Le gusta la idea general.	No le ve mucha utilidad a controlar los gastos desde aquí.
F-1	Le gusta que para cadenas de restaurante aparezcan todos los establecimientos, puesto la calidad de la comida varía.	Le gustaría ver fotografías de los platos.
F-2	Le gusta poder ver el desglose de tipos de pedidos.	No hay totales por tipo de pedido.
F-3	N/A	Echa en falta una nota descriptiva.
F-4	Le gusta las puntuaciones por plato.	No ve útil que se diferencien por tipo de pedido.
F-5	Le ha resultado cómodo.	No entiende por qué no le pide directamente la valoración y el precio del plato.
F-6	Le resulta intuitivo.	Considera que se le tendría que dar más importancia.
Otros	Siente mucha curiosidad del sistema de recomendaciones.	Considera que algunas pantallas son demasiado simples.

	Juan Cobo - Familiar	
	<i>Inputs positivos</i>	<i>Inputs negativos</i>
Concepto general	Le parece una buena idea.	N/A
F-1	N/A	N/A
F-2	Le parece bien que se vea el total de cada mes.	N/A
F-3	Le gusta que se vean sus reseñas y al lado las globales.	Ha completado la tarea llegando primero al histórico de una de las veces que visitó al restaurante, después se dio cuenta que no era ahí y utilizó el buscador.
F-4	Le parece muy útil.	N/A
F-5	N/A	N/A
F-6	Le gusta poder indicar esta información.	Ha confundido el botón de reiniciar datos con el de confirmar los cambios.
Otros	Le gusta la paleta de colores y que sea todo tan minimalista.	Se ha confundido al principio con los botones de añadir tique y configurar la aplicación porque le faltaba un poco de contexto. Aun así, gracias a ello se ha detectado que sería útil que nada más registrarte, se le

		solicitar al usuario que rellene su configuración inicial.
--	--	--

	Juan Cobo - Amigo	
	<i>Inputs positivos</i>	<i>Inputs negativos</i>
Concepto general	La utilizaría constantemente.	N/A
F-1	N/A	N/A
F-2	Le gusta poder comparar con el resto de los meses.	Echa en falta algún color o indicador que de contexto a cuanto lleva gastado durante el mes.
F-3	N/A	Ha completado la tarea llegando primero al histórico de una de las veces que visitó al restaurante, después se dio cuenta que no era ahí y utilizó el buscador.
F-4	Le gusta mucho el sistema de puntuaciones y que vaya por tipo de pedido.	Le gustaría que haya filtros para indicar el tipo de pedido y el tipo de restaurante.
F-5	Lo ha visto muy intuitivo.	N/A
F-6	N/A	N/A
Otros	Le gusta mucho el home, sobre todo que se vea la última actividad de los usuarios.	Le gustaría poder indicar en su perfil un límite mensual y que según lo consumido al mes vaya cambiando de

		color.
--	--	--------

Tras las pruebas con los usuarios se detectan los siguientes puntos de mejora:

- **PM-1:** En el apartado de recomendaciones sólo se mostrará la puntuación del restaurante. Si se pulsa encima, se desplegará y se mostrará la puntuación de cada plato. De este modo, se ganará mucho espacio y cargará menos la vista.
- **PM-2:** Se modificarán los tamaños de letra para facilitar la lectura a los usuarios.
- **PM-3:** Se modificarán los iconos para que sean más intuitivos.
- **PM-4:** Las estrellas de las valoraciones mostrarán los decimales y el total. Por ejemplo: 4.4 (120).
- **PM-5:** En los controles de gastos se mostrará un desglose del total por tipo de pedido.
- **PM-6:** Añadir una configuración de usuario para que este tenga un control de los gastos mensuales. Por ejemplo: 150€. A medida que se acerque al número, se cambiará el color para avisar al usuario.

Se valorará implementar las siguientes mejoras, aunque en principio quedan fuera del alcance del proyecto:

- **PM-7:** Añadir logotipos a los restaurantes que serán visibles para todos los usuarios de la aplicación.
- **PM-8:** Permitir a los usuarios añadir fotografías de los platos. En ningún caso serán públicas, cada uno verá las suyas.
- **PM-9:** En el apartado de recomendaciones, añadir una pestaña que en lugar de mostrarlas en listado las muestre en un mapa parecido a cuando se busca por Google Maps.
- **PM-10:** Al registrar un tique se permitirá añadir una nota para cada plato. De este modo, el usuario podrá recordar si solicitó algún cambio de ingredientes, punto de la carne, extras...

3. Diseño técnico

3.1 Definición de los casos de uso

Estos son los casos de uso de la aplicación:

Identificador	CU-001
Nombre	Registrarse en la aplicación.
Prioridad	Alta.
Descripción	El usuario quiere registrarse para poder utilizar la aplicación.
Actores	Usuario.
Pre-Condiciones	El usuario debe tener una cuenta de Google.
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none">1. Acceder a la aplicación.2. Pulsar sobre el botón de iniciar sesión con Google.3. Completar el perfil de usuario.
Post-Condiciones	Se crea el nuevo usuario en la aplicación y se muestra la pantalla inicial.
Notas	-

Identificador	CU-002
Nombre	Iniciar sesión.
Prioridad	Alta.
Descripción	El usuario quiere iniciar sesión para entrar en la aplicación.
Actores	Usuario.
Pre-Condiciones	El usuario debe estar registrado (CU-001).
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none">1. Acceder a la aplicación.2. Pulsar sobre el botón de iniciar sesión con Google.
Post-Condiciones	Se muestra la pantalla inicial de la aplicación.
Notas	-

Identificador	CU-003
Nombre	Añadir nuevo tique.
Prioridad	Alta.
Descripción	Se quiere registrar la visita a un restaurante.
Actores	Usuario.
Pre-Condiciones	El usuario debe tener su sesión iniciada (CU-002). El usuario debe haber realizado una visita a un establecimiento.
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none">1. Acceder a la aplicación.2. Pulsar sobre el botón flotante con el icono '+

	<ol style="list-style-type: none"> 3. Introducir restaurante y día de la visita. 4. Introducir los platos consumidos. 5. Opcionalmente, añadir las valoraciones y los precios de cada plato. 6. Indicar el tipo de pedido, si le ha gustado y si volvería al establecimiento. 7. Revisar los datos introducidos y enviar el tique.
Post-Condiciones	El tique se registra correctamente y se muestra la pantalla inicial de la aplicación.
Notas	-

Identificador	CU-004
Nombre	Revisar histórico.
Prioridad	Normal.
Descripción	Revisar cuantas veces se ha salido este mes, cuanto hace que no se va a un restaurante en específico, etc.
Actores	Usuario.
Pre-Condiciones	El usuario debe haber registrado, como mínimo, un tique (CU-003).
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none"> 1. Acceder a la aplicación. 2. Darle al botón de 'Ver histórico completo' de la pantalla inicial o a la tercera opción del menú inferior.
Post-Condiciones	Se muestra el histórico de reseñas del usuario agrupadas por mes.
Notas	Si el usuario no ha introducido aun ningún tique, le aparecerá un mensaje sugiriéndole que primero debería registrar alguno.

Identificador	CU-005
Nombre	Control de gastos.
Prioridad	Baja.
Descripción	Revisar el gasto del mes actual y el histórico mensual.
Actores	Usuario.
Pre-Condiciones	El usuario debe haber registrado, como mínimo, un tique (CU-003).
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none"> 1. Acceder a la aplicación. 2. Darle a la cuarta opción del menú inferior.
Post-Condiciones	Se muestra el control de gastos del usuario agrupados por mes y las estadísticas de consumo del mes actual (tipos de restaurante y tipos de pedido).
Notas	Si el usuario no ha introducido aun ningún tique, le aparecerá un mensaje sugiriéndole que primero debería registrar alguno.

Identificador	CU-006
Nombre	Modificar perfil.
Prioridad	Normal.
Descripción	Modificar los datos del usuario como la moneda utilizada en la aplicación, habilitar el acceso a la ubicación, etc.
Actores	Usuario.
Pre-Condiciones	El usuario debe tener su sesión iniciada (CU-002).
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none"> 1. Acceder a la aplicación. 2. Darle a la quinta opción del menú inferior. 3. Modificar la información deseada del perfil.
Post-Condiciones	Se actualiza el perfil del usuario.
Notas	-

Identificador	CU-007
Nombre	Ver recomendaciones.
Prioridad	Alta.
Descripción	Descubrir nuevos restaurantes afines en base a lo que opinan el resto de los usuarios.
Actores	Usuario.
Pre-Condiciones	El usuario debe tener su sesión iniciada (CU-002).
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none"> 1. Acceder a la aplicación. 2. Darle al botón 'Ver recomendaciones' de la pantalla inicial o a la segunda opción del menú inferior.
Post-Condiciones	Se muestran los restaurantes y sus platos con gustos afines al usuario ordenados en base a una puntuación numérica.
Notas	-

Identificador	CU-008
Nombre	Ver valoraciones de un restaurante.
Prioridad	Alta.
Descripción	Revisar cuantas veces se ha valorado cada plato, opinión global de los usuarios, etc.
Actores	Usuario.
Pre-Condiciones	El usuario debe tener su sesión iniciada (CU-002).
Iniciado por	Usuario.
Flujo	<ol style="list-style-type: none"> 1. Acceder a la aplicación. 2. Pulsar sobre el buscador de la pantalla inicial. 3. Opcionalmente, filtrar los restaurantes que aparecen por las palabras que introduzca el usuario. 4. Elegir el restaurante deseado entre las opciones disponibles.

Post-Condiciones	Se muestran todos los platos del restaurante con las reseñas del usuario y la valoración global de todos los usuarios.
Notas	Si el usuario ha permitido el acceso a la aplicación para obtener su ubicación, los restaurantes aparecerán ordenados por distancia. Es decir, los más cercanos a él le aparecerán en primer lugar.

3.2 Diagrama UML de la base de datos

Al utilizar una base de datos no relacional, se tratan de duplicar todos los campos que se consideran necesarios en lugar de utilizar las relaciones:

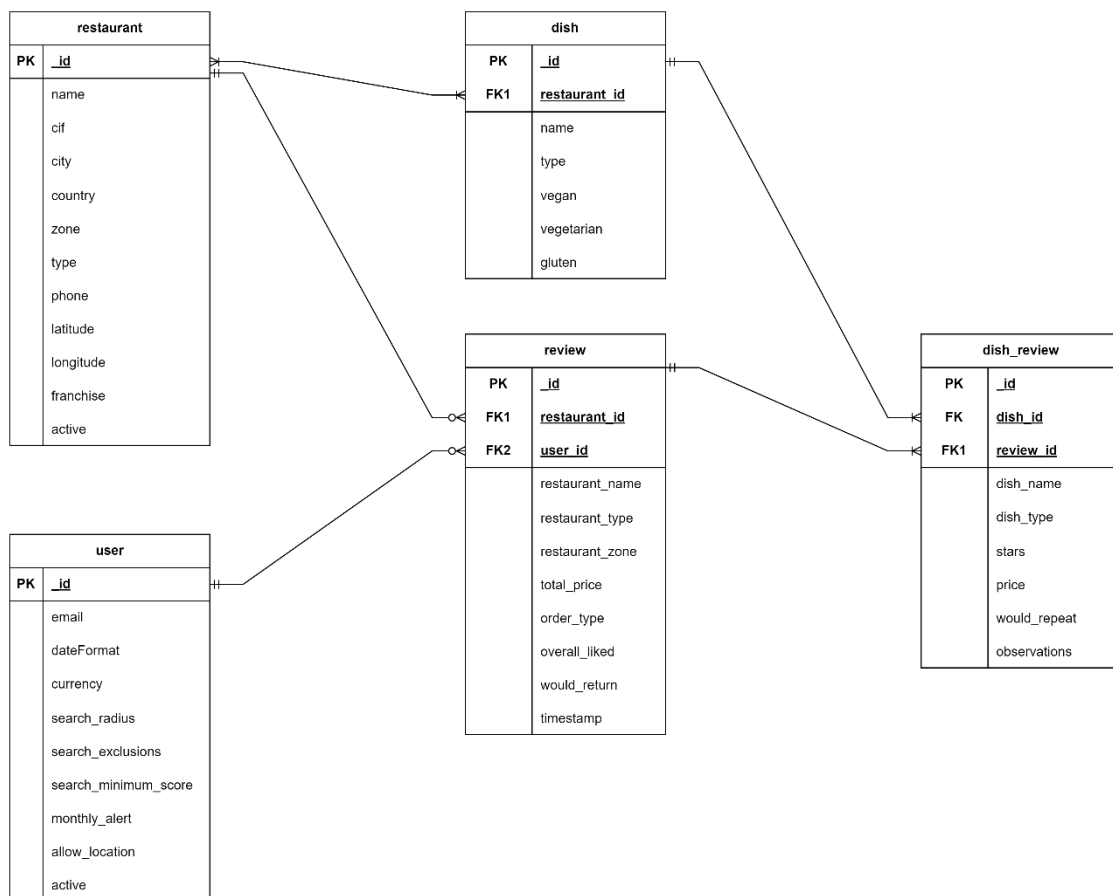


Ilustración 45 - Diseño técnico: Diagrama UML

De todos modos, en el esquema mostrado, se establecen las relaciones entre las tablas para que sea más comprensible la relación que hay entre ellas.

Sin embargo, tal y como se ha mencionado anteriormente, a nivel de aplicación, cada tabla es independiente y se duplica toda la información necesaria.

Como observación, sería más correcto denominarlos “documentos” en lugar de “tablas”. Ya que este concepto no existe en las bases de datos no relacionales.

3.3 Arquitectura del sistema

Actualmente, la aplicación utiliza una arquitectura REST con la siguiente estructura:

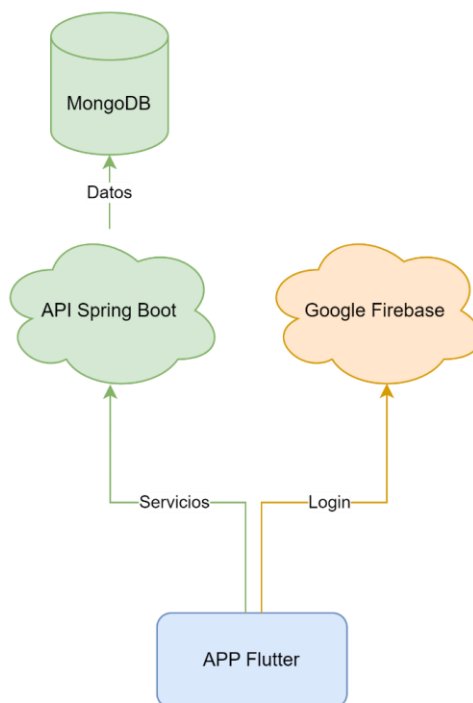


Ilustración 46 - Diseño técnico: Arquitectura de la aplicación v1

La APP se ejecutará en los dispositivos móviles y consumirá los servicios de Spring Boot y de *Google Firebase* mediante un cliente API REST.

Los servicios de *Google Firebase* servirán para controlar el acceso y registro de usuarios a la aplicación. Esto no pasará por los servicios de Spring Boot puesto que *Firebase* ya actúa como un back-end.

En futuras versiones de la aplicación se desglosará el servicio de Spring Boot en varios microservicios, resultando en una arquitectura similar a la del siguiente diagrama:

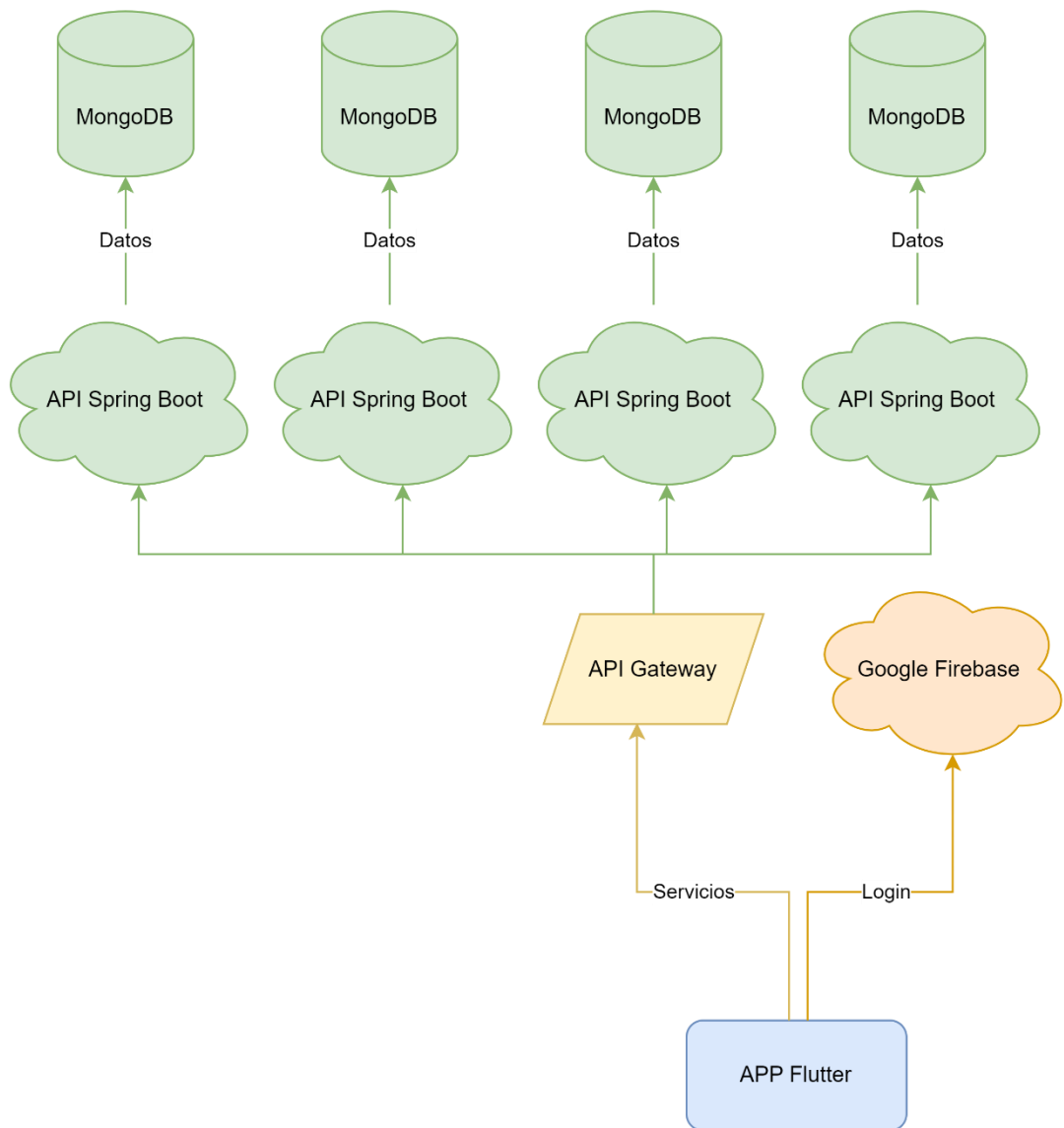


Ilustración 47 - Diseño técnico: Arquitectura de la aplicación v2

Lo que permitirá que los servicios sean más escalables, resilientes e independientes.

4. Herramientas utilizadas durante el desarrollo

El desarrollo de la aplicación se va a realizar desde un MacBook con el IDE IntelliJ IDEA en su versión *Ultimate*, que se puede obtener de manera gratuita mientras se es estudiante de la UOC.

Se ha elegido este IDE debido a la experiencia que se tiene con él del entorno laboral y dado que es uno de los más famosos, robustos y completos del mercado. Como alternativa, se ha planteado Android Studio, que es bastante similar a este y gratuito.

A nivel de APIs, como se ha mencionado anteriormente, se van a utilizar dos:

- *Google Firebase*
- Back-end Spring Boot propio

El primero, será el encargado de la integración con Google para el acceso a la aplicación. Durante la investigación de cómo acceder mediante la cuenta de Google en aplicaciones móviles, esta era la opción más común y sencilla de implementar.

Se hubiese preferido evitar más plataformas e implementar la integración con Google directamente mediante OAuth 2.0, pero prácticamente no había documentación de cómo implementarla y parecía más complicado que con Firebase.

El resto de la aplicación consumirá una API propia que informará de los restaurantes y los platos que estos ofrecen. Se intentaron buscar APIs que ofrezcan todos los restaurantes de manera gratuita, pero no se encontró nada que pudiese ser de utilidad.

La mayor ventaja de hacerlo de este modo es tener control total de lo que ofrece la API, cómo funciona, qué datos maneja, etc. Aun así, tiene una desventaja muy grave: que no está completa. Es decir, hay que dedicarle mucho tiempo a ir introduciendo todos los restaurantes y platos existentes para que los usuarios los puedan elegir.

Para paliar este comportamiento, en la aplicación se les permitirá a los usuarios introducir manualmente restaurantes y platos que no estén dados de alta. De este modo, aunque no estén registrados en la aplicación, los usuarios podrán seguir llevando su control.

De manera similar, en un futuro panel de gestión para los administradores de la aplicación, estos podrán ver cuales son los restaurantes y/o platos más demandados por los usuarios y podrán introducir estos valores. Otra opción, sería ofrecer estos servicios a los restaurantes y que ellos decidan si quieren o no estar afiliados a la aplicación.

En otras palabras, tener nuestra propia API puede utilizarse para, en un futuro, facilitar acciones comerciales y de marketing.

Por último, para facilitar el desarrollo de la aplicación se han utilizado las siguientes librerías de Flutter:

- **dropdown_search** ^[7] → Facilita la creación de desplegables con opciones de búsqueda y carga asíncrona de datos.
- **dio** ^[8] → Utilizado para comunicarse con las APIs mediante un cliente API REST.
- **flutter_form_builder** ^[9] → Facilita la creación de formularios donde el usuario puede introducir datos.
- **fl_chart** ^[10] → Incluye diversas gráficas para realizar funciones estadísticas.
- **carousel_slider** ^[11] → Añade la opción de crear carruseles en la aplicación.
- **url_launcher** ^[12] → Permite abrir aplicaciones del teléfono móvil como realizar llamadas, enviar SMS, abrir el navegador...
- **geolocator** ^[13] → Facilita la gestión de la ubicación actual del dispositivo.
- **fluttertoast** ^[14] → Permite mostrar mensajes que desaparecen al cabo de ciertos segundos.

5. Pruebas de la aplicación

Durante el desarrollo de la aplicación se han estado realizando pruebas con un dispositivo virtual iPhone 14 Pro max dado que es el que proporciona por defecto el MacBook con el que se lleva a cabo el desarrollo. Con este dispositivo se ha validado toda la aplicación, ya que ha sido el dispositivo principal durante todo el desarrollo.

Una vez se comenzaron a desarrollar versiones probables y estables de la aplicación, se distribuyeron varias APK y se instalaron en diferentes dispositivos físicos Android y se siguió el principio de *shadowing* para observar los comportamientos de los usuarios.

En las primeras pruebas, se detectó que el acceso a la aplicación no funcionaba correctamente en Android. Desde ese momento, se preparó un entorno en Android Studio con un segundo dispositivo virtual (Píxel 7) con el que se validó que el funcionamiento en Android fuese el esperado.

Con las siguientes versiones, en la mayoría de los casos las pruebas físicas fueron bien y los usuarios estaban contentos con la aplicación. Sin embargo, en todas las pruebas fueron ocurriendo algunos fallos de visualización debido a los diferentes tamaños de los dispositivos móviles. Estos, se han ido corrigiendo a medida que se desarrollaba la aplicación.

Algunos de los dispositivos físicos probados son:

- Nothing phone 2.
- Alcatel 1SE.
- Redmi 13C.

A nivel de pruebas del funcionamiento de la aplicación, se incluyen varias pruebas unitarias que validan que los fragmentos comunes del código mantengan el funcionamiento esperado. De esta manera, se validará que, con cualquier cambio en la aplicación, estos paquetes comunes sigan funcionando como corresponde.

Este nivel de pruebas debería ampliarse a pruebas de integración para asegurarse que todas las funcionalidades siempre serán utilizables y no contendrán errores. Sin embargo, debido a que las funcionalidades están ligadas al acceso de un usuario, las pruebas no serían del todo realistas y no se podría asegurar que ese usuario se comporte del mismo modo que un usuario real.

6. Estado actual del proyecto

6.1 Objetivos del proyecto

A fecha de este documento, el proyecto se encuentra en una fase estable y completamente usable para los usuarios. Sin embargo, no se ha cumplido con la planificación propuesta ni con todos los requisitos funcionales establecidos.

A continuación, se van a volver a listar todos los requisitos que no ha dado tiempo a implementar y, seguidamente, se dará una breve explicación del motivo:

- **RF7:** Se deben visualizar los restaurantes que más frecuenta el usuario.
- **RF8:** Diferenciar las valoraciones en base al tipo de pedido (mesa, a recoger, a domicilio, *riders* [6], etc.) para controlar si la calidad del producto disminuye.
- **RF9:** Apartado de sugerencias donde el usuario pueda acudir cuando no sepa qué comer. Irán apareciendo sugerencias de pedidos anteriores y el usuario podrá aceptar la sugerencia o solicitar otra nueva. Sólo incluirá sugerencias en base a sus gustos. Es decir, mediante lo que ha ido introduciendo en la aplicación.
- **RF10:** Apartado de descubrimiento donde el usuario pueda encontrar restaurantes con gustos afines. Es decir, platos valorados por otros usuarios cerca del rango establecido por el usuario y que coincidan en gustos. Por ejemplo, hamburgueserías con estilo *smash*.
- **RF12:** El usuario debe poder hacer una foto con su teléfono móvil y automáticamente se deberían precargar todos los platos dándole la opción al usuario de revisarlo, así podrá: revisar que los platos son correctos, eliminar los que no ha consumido y dar su valoración de ellos.

Con respecto a **RF7**, se había olvidado diseñar cómo debía quedar este requisito y no ha habido tiempo de pensar cómo implementarlo en la aplicación.

Por otra parte, el **RF8** sí que está pensado implementar en la pantalla de las reseñas de cada restaurante, pero simplemente no ha dado tiempo a implementarlo con la planificación actual. De todos modos, es un requisito que tendrá su dificultad de implementar porque habrá que crear nuevas vistas en la base de datos para poder diferenciar el número de reseñas por tipo de pedido.

Por último, el resto de los requisitos no implementados (**RF9**, **RF10** y **RF12**) se han dejado para futuras fases del proyecto debido a su complejidad de implementación.

En la planificación inicial, parecía bastante sencillo implementarlos. De hecho, esta opinión se mantiene tras un análisis más exhaustivo.

Sin embargo, la complejidad de estos requisitos viene dada por el tiempo que hay que dedicar a validar que su funcionamiento es correcto y a la falta de datos en la aplicación.

Los algoritmos de recomendación y sugerencias requieren que muchos usuarios hayan introducido datos en la aplicación para observar que estos se comportan correctamente. Con lo cual, es necesario que el resto de las funcionalidades de la aplicación funcionen de una manera muy cómoda y fluida para que los usuarios utilicen la aplicación.

Si a esto, le sumamos que no se había planificado un tiempo para introducir datos en la aplicación (restaurantes y sus platos) hacia muy complicada que la implementación funcionase correctamente para el día de la entrega, ya que los usuarios no podrían introducir datos reales porque estos no estaban dados de alta en la aplicación.

Con lo cual, se coordinó que era mejor dejar estos procesos para fases futuras del proyecto y centrarse en las funcionalidades “básicas” de la aplicación.

Aun así, hay pensada una estrategia a seguir para implementar estos algoritmos, y es la siguiente:

En primera instancia, estas recomendaciones se realizarán mediante un algoritmo programado en Java (en el back-end) mientras que, al mismo tiempo, se alimentará un modelo de datos que irá creciendo a medida que los usuarios utilicen la aplicación.

Una vez este modelo tenga suficiente información y esté bien entrenada, se realizará un cambio y se dejará de utilizar el algoritmo propio y se comenzará a consumir este modelo generado.

Esto, se debe a que tras varias conversaciones con conocidos que tienen experiencia con modelos de datos, coinciden que al principio este modelo no realizará buenas recomendaciones debido a que no tendrá datos suficientes. Con lo cual, mientras se recolecta toda esta información, se hará uso del algoritmo mencionado anteriormente.

Cabe destacar, que este algoritmo aun no se ha diseñado, hay algunas ideas en el aire, pero no está definido.

Por último, pasa algo parecido con el **RF12**, aunque se tiene experiencia con tecnologías de extracción información de textos, no se ha podido dedicar tiempo a insertar todos los restaurantes y platos en la aplicación.

Además de esto, se ha estado analizando el conjunto de tiques que se han ido recolectando a lo largo de este año y, por desgracia, en muchas ocasiones el texto que se imprime no corresponde al plato consumido. Por ejemplo, hay ocasiones que en el restaurante se pide una “Pizza cuatro quesos” y en el tique aparece “pizza”, haciendo imposible que se sepa a cuál de todas se refiere.

Si a esto, le sumamos que tampoco ha habido tiempo a implementar un sistema de edición de una reseña que aun no has introducido. Es decir, mientras el usuario está registrando un nuevo tique, si se equivoca con una valoración, deberá volver atrás y repetir la introducción de valoraciones y precios.

Se ve que, aunque sea sencillo escanear el tique, no existiría la información para que la aplicación la muestre y el usuario tampoco podría corregir los errores que pueda haber cometido este proceso.

Debido a esto, también se decidió que era mejor idea dejar este requisito a una fase futura del proyecto. Esta decisión, también coge peso al saber que este requisito es “únicamente” una mejora en la calidad del proceso de registrar información en el tique. El usuario va a llegar al mismo punto, aunque sea de una forma más manual.

Al tomar esta decisión, se ha planteado que esta funcionalidad pueda ser exclusiva a una suscripción de la aplicación o a su versión de pago. En el apartado de valoración económica del trabajo se darán más detalles de ello.

6.2 Mejoras implementadas y pendientes de implementar

Durante el desarrollo, se detectaron varios puntos que no estaban contemplados a nivel de requisitos pero que eran importantes tener en cuenta para que la aplicación siempre funcione correctamente.

Algunos de ellos, se corrigieron inmediatamente y otros quedaron pendientes para futuras versiones de la aplicación. Estos son algunos de los puntos que se han implementado pero que no se habían pensado y se encontraron una vez se empezó la implementación:

- El usuario debe poder registrar tiques de restaurantes y platos que no estén dados de alta en la aplicación. Estos **no** deben aparecer al resto de usuarios para evitar mensajes de odio.

- En el flujo de registro de tique, debe aparecer un buscador en la parte superior de la pantalla que permita al usuario filtrar los platos que está buscando.
- La aplicación debe funcionar correctamente cuando un usuario no tiene información que mostrar. Por ejemplo, cuando se acaba de registrar en la aplicación.
- El usuario debe poder escribir en los desplegados de restaurante para filtrar los resultados y que le sea más cómodo encontrar el que quiere.

Los puntos pendientes de implementar sobre todo corresponden a mejoras de rendimiento en la aplicación que, aunque ahora no son molestias, asumiendo que la aplicación vive durante años, acabarían apareciendo y podrían resultar en problemas:

- No se debe permitir introducir más de 5 tiques en un mismo día. Es decir, el máximo de comidas cada día será de 5. → Esto, paliará que las empresas generen muchas reseñas falsas para promocionar sus productos.
- Se debe limitar la introducción de un tique cada 8 horas. Una vez introduces uno, debes esperar 8 horas al siguiente. → Similar al punto anterior, será una medida de contingencia hacia falsas reseñas.
- Al registrar los precios del tique, se debe permitir al usuario introducir gastos adicionales y/o descuentos → Este punto es muy importante según las conclusiones de la encuesta con usuarios. Por ejemplo, cuando se sale con amigos y paga uno, no se introducirá el precio individual de cada producto, sino que se añadirá un gasto con el concepto "Bizum" y eso será el coste total del tique.
- Hay que modificar los textos de la aplicación para que funcione en más idiomas → La aplicación está preparada para ello, sólo faltaría crear las traducciones.
- En las páginas de históricos los meses deberían cargar según el usuario va desplazándose lateralmente en lugar de cargarlos todos directamente → Asumiendo que la aplicación vive años, si la dejamos tal y como está ahora, cada año tardaría ligeramente más en cargar estas pantallas.
- En la pantalla de control de gastos falta mostrar el desglose de la visita. Es decir, qué costó cada plato.

- Mejorar los formatos de las gráficas mostradas → Añadir iconos en lugar de nombres y mostrar una leyenda indicando a qué corresponde cada icono. De este modo, será mucho más clara visualmente.

6.3 Planificación del proyecto

Mediante lo previamente expuesto, si se revisa la planificación del proyecto, se pueden vincular las observaciones anteriores a los desvíos correspondientes:

Los desvíos de “Fase I Desarrollo aplicación móvil” vienen dados por el diseño de las pantallas de registrar un nuevo tique y los ajustes de la interfaz que hubo que realizar sobre las pantallas de consumos mensuales y historial de pedidos del usuario. En lugar de los 5.25 días propuestos han acabado siendo alrededor de 15 días. Para evitar que el desvío se incrementara más, se decidió optar por una versión que más o menos se viera bien en todos los dispositivos y, dedicar una versión futura únicamente a revisar los apartados visuales de las pantallas.

La “Fase III Desarrollo Back-end” sufrió desvíos debido a la integración con *Google Firebase*, que funcionaba correctamente en Apple, pero no en Android. En lugar de las 10 horas propuestas, fueron alrededor de 30 entre que se realizaban pruebas en dispositivos físicos y se configuraba un dispositivo virtual Android. Para evitar más desvíos por el dispositivo utilizado, a partir de ese momento siempre se validó la aplicación con ambos dispositivos virtuales.

A raíz de esos desvíos, no se ha podido completar la Fase III, dado que esta incluye los requisitos funcionales no completados: **RF7, RF8, RF9, RF10, RF12**. Como se ha mencionado anteriormente, se prescindió de los estos porque se considera que son los que menos afectaban al comportamiento básico de la aplicación.

7. Valoración económica del trabajo

Actualmente, mantener la aplicación tiene un coste fijo de 5.99€ mensuales debido a que el back-end Spring Boot y la base de datos MongoDB se encuentran desplegadas en un VPS para que sean accesibles desde cualquier lugar mediante Internet.

En un futuro, como se quiere publicar la aplicación tanto en *Google Play* como en la *Apple Store* supondrá costes adicionales:

- 25€ por registrarse como desarrollador en *Google Play*.
- 100€ anuales por estar como desarrollador en la *Apple Store*.

Además de esto, se quiere solicitar la creación de un logo (o varios) para la aplicación. De este modo, será más reconocible y tendrá su propia marca. Este coste se estima que será alrededor de 250€.

En primeras versiones no se espera obtener beneficios económicos con esta aplicación. Además, bajo ningún concepto se quiere monetizar la aplicación mediante anuncios. Aunque esto, sería la forma más sencilla para recibir algún ingreso.

Sin embargo, se ha planteado que se podría ofrecer una suscripción a la aplicación que permitiría a los usuarios utilizar funcionalidades adicionales. Algunas de las ideas serían las siguientes:

- Escaneo de tiques mediante fotografías.
- De base, hacer que la aplicación sólo muestre los N primeros meses (se había pensado en 3) en los históricos de pedidos y control de gastos y, con la suscripción, que muestre todo lo que ha introducido el usuario.
- Al registrar un tique, permitir subir fotografías de los platos. Estas fotografías serían privadas y tan sólo las podría ver el usuario que las ha subido.
- Habilitar más tipos de filtrado en la sección de recomendaciones. Por ejemplo, por tipo de comida.

Todo esto no dejan de ser ideas que podrían estar detrás de una suscripción, no quiere decir que sí o sí vayan a acabar así. Seguramente, la mayoría de ellas se acaben implementando de manera gratuita y sólo sean algunos detalles los que podrían permanecer monetizados.

En cualquier caso, con en el estado actual de la aplicación se puede decir claramente que esta no es viable, porque sólo genera costes y ningún tipo de ingreso.

8. Conclusiones

Gracias a este trabajo, se han adquirido una serie de conocimientos tanto técnicos como no técnicos que, de otro modo, no se hubiesen adquirido.

A nivel técnico, hemos aprendido:

- Trabajar con el *framework* Flutter y ser capaces de crear una aplicación móvil completamente usable.
- Trabajar con bases de datos no relacionales y explotar sus datos. En este caso, MongoDB.
- Emular dispositivos móviles y desplegar aplicaciones en dispositivos físicos.

En rasgos generales, se ha aprendido lo siguiente:

- Aprender a gestionar el alcance de un proyecto definiendo requisitos funcionales y no funcionales.
- Planificar el tiempo estimado para cumplir la implementación estos requisitos.
- Definir un diseño y moldear este en base a las opiniones de los usuarios entrevistados.
- Redactar todo el flujo de implementación, profundizando en los temas más importantes y marcando los objetivos no cumplidos.

Con todo lo mencionado, se puede decir con total seguridad que se ha pecado de optimismo y **no** se han cumplido todos los objetivos del trabajo. Aunque se ha llegado a una fase muy avanzada y la aplicación es totalmente usable, no se han cumplido todos los requisitos funcionales definidos.

Sobre todo, esto se debe a que las primeras semanas del desarrollo se vieron muy afectadas por la falta de experiencia técnica con estas tecnologías. Debido a esto, para cada paso que se quería avanzar en la implementación, se tenía que investigar qué formas había para hacerlo y cuál era la más conveniente. Con lo cual, se perdía más del doble de tiempo de lo definido para cada requisito.

A esto, también hay que sumarle los desvíos causados por: *bugs* detectados durante el desarrollo, modificaciones de las pantallas para que se vieran correctamente en diferentes dispositivos móviles, definición más técnica de cómo optimizar la carga de las pantallas, etc.

Para paliar todos estos desvíos, se decidió prescindir de las funcionalidades más complejas en la primera versión de la aplicación, ya que estas, aunque muy vistosas, no afectan a la funcionalidad principal de la aplicación: llevar un control de los platos que consumes y revisar la calidad de estos.

Así pues, la primera versión de la aplicación ha quedado mucho más sencilla de lo que se esperaba, pero se han cumplido los tiempos y es totalmente funcional.

Gracias a esto, se ha podido definir una línea de trabajo futuro que incluirá todas las funcionalidades pendientes y profundizará en algunas más. De momento, se podría resumir del siguiente modo:

- Fase II
 - Añadir botones para introducir gastos y descuentos en el registro de un tique → Por ejemplo, cuando se va en grupo podrías no introducir el precio de cada plato y añadir un gasto con el concepto “Bizum” y el importe que has pagado.
 - Modificar modal de la pantalla de histórico de pedidos para que este esté en una pantalla aparte → Así tendrá más espacio y nos permitirá introducir más funcionalidades a futuro.
 - Modificar pantalla de control de gastos para mostrar el desglose de cada plato en una pantalla aparte, similar a lo que se quiere hacer en el histórico de pedidos.
 - Permitir modificar valoraciones cuando estás registrando un tique.
 - Permitir eliminar reseñas que ya has introducido.
 - Habilitar acceso a la aplicación mediante Apple ID → Así se podrá publicar la aplicación en la *App Store*.
- Fase III
 - Implementar sistema de recomendaciones en base a los gustos del usuario.
 - Permitir añadir fotografías de los platos.
- Fase IV
 - Permitir a los usuarios escanear un tique mediante una fotografía y que facilite lo máximo posible la introducción de estos datos.

9. Glosario

A

API REST

Se trata de un método de comunicación utilizado para comunicarse entre diferentes aplicaciones mediante el protocolo HTTP.5

Apple Store

Servicio oficial para descargar aplicaciones en dispositivos iOS.....55

B

Back-end

Parte de la aplicación invisible para el usuario y que se encarga de ejecutar la lógica de negocio.5

bugs

Error o fallo no esperado en la ejecución de un programa o aplicación que hace que esta acabe en un estado no controlado.56

F

Figma

Herramienta que permite realizar diseños de alta precisión en línea.....22

Flutter

Kit de herramientas de diseño de Google centradas en el diseño de interfaces de usuario para aplicaciones móviles.5

framework

Conjunto estandarizado de reglas, conceptos y prácticas centrado en facilitar la labor de programación en un determinado lenguaje.5

Front-end

Interfaz de la aplicación, aquello que ven y con lo que interactúan los usuarios.5

G

Google Firebase

Herramienta de Google utilizada para facilitar la integración de sus aplicaciones desde aplicaciones móviles.45

Google Forms

Herramienta de Google diseñada para realizar encuestas públicas.....9

Google Play

Servicio oficial para descargar aplicaciones para dispositivos Android.55

I

IDE

Integrated development environment, aplicación que proporciona servicios, herramientas y utilidades para facilitar el desarrollo del software.47

M

microservicios

Enfoque en el desarrollo de software centrado en tener un conjunto de aplicaciones pequeñas en lugar de una única aplicación que abarque todas las funcionalidades.....5

monolito

Enfoque en el desarrollo de software que incluye todas las funcionalidades en una única aplicación. .5

O

OAuth 2.0

Estándar de seguridad web que permite a aplicaciones acceder a recursos alojados en otras webs. .47

S

shadowing

Observar el comportamiento y las rutinas de una persona durante un período de tiempo.9

Spring Boot

framework de Java.....5

V

VPS

Virtual Private Server, servidor privado virtual. Suelen ser máquinas virtuales con muy pocos recursos cuya finalidad es desplegar aplicaciones o servicios web.....55

10. Bibliografía

- [1] <https://bizum.es/> - 01/09/2023
- [2] <https://www.just-eat.es/> - 01/09/2023
- [3] <https://www.ubereats.com/es> - 01/09/2023
- [4] <https://www.google.es/maps/preview> - 01/09/2023
- [5] <https://www.tripadvisor.es/> - 01/09/2023
- [6] <https://www.aturnos.com/blog/2021/10/que-son-los-riders-y-cual-es-su-normativa/> - 30/09/2023
- [7] https://pub.dev/packages/dropdown_search - 10/11/2023
- [8] <https://pub.dev/packages/dio> - 10/11/2023
- [9] https://pub.dev/packages/flutter_form_builder - 13/11/2023
- [10] https://pub.dev/packages/fl_chart - 25/11/2023
- [11] https://pub.dev/packages/carousel_slider - 27/11/2023
- [12] https://pub.dev/packages/url_launcher - 02/12/2023
- [13] <https://pub.dev/packages/geolocator> - 03/12/2023
- [14] <https://pub.dev/packages/fluttertoast> - 06/12/2023

Recursos bibliográficos no referenciados:

- <https://www.kellton.com/kellton-tech-blog/top-10-mobile-app-development-frameworks-in-2023> - 01/09/2023
- https://www.reddit.com/r/FlutterDev/comments/14n19ss/is_flutter_really_that_good/ - 01/09/2023
- <https://medium.com/nerd-for-tech/flutter-series-connecting-ui-to-spring-boot-backend-f9874dc3dcd5> - 01/09/2023
- <https://www.youtube.com/watch?v=uisLoOmtlSk> - 08/10/2023
- <https://medium.com/flutter-community/simple-ways-to-pass-to-and-share-data-with-widgets-pages-f8988534bd5b> - 11/11/2023
- <https://stackoverflow.com/questions/73534183/how-can-i-fetch-data-from-async-function-in-flutter> - 11/11/2023
- <https://www.youtube.com/watch?v=HB5WMcxAmQQ> - 08/11/2023
- <https://www.youtube.com/watch?v=VRt66cN6PLw> - 11/11/2023
- https://github.com/flutter-form-builder-ecosystem/flutter_form_builder/issues/833 - 12/11/2023
- <https://stackoverflow.com/questions/55237311/how-to-create-nullable-variable-in-dart> - 12/11/2023
- <https://docs.flutter.dev/cookbook/navigation/navigation-basics> - 13/11/2023

- <https://blog.logrocket.com/flutter-floatingactionbutton-a-complete-tutorial-with-examples/> - 13/11/2023
- <https://docs.flutter.dev/cookbook/navigation/navigation-basics> - 13/11/2023
- https://www.youtube.com/watch?v=1U8_Mq1QdX4 - 15/11/2023
- <https://docs.flutter.dev/cookbook/forms/validation> - 18/11/2023
- <https://www.youtube.com/watch?v=FdsqU02xwsE> - 21/11/2023
- <https://www.dhiwise.com/post/infinite-list-in-flutter-scroll-more-load-less> - 21/11/2023
- <https://blog.logrocket.com/implement-infinite-scroll-pagination-flutter/> - 22/11/2023
- <https://www.youtube.com/watch?v=Gsfjcpo6wcA> - 22/11/2023
- <https://medium.flutterdevs.com/grouped-listview-in-flutter-a3a7cc13133c> - 22/11/2023
- <https://betterprogramming.pub/how-to-build-an-online-searchable-dropdown-flutter-widget-3ede426f4d17> - 23/11/2023
- <https://stackoverflow.com/questions/63410839/flutter-failed-to-build-on-android-due-to-icon-tree-shaker> - 25/11/2023
- <https://stackoverflow.com/questions/70997381/flutter-error-no-named-parameter-with-the-name-displaylarge> - 25/11/2023
- <https://stackoverflow.com/questions/60475481/flutter-doctor-error-android-sdkmanager-tool-not-found-windows> - 26/11/2023
- <https://stackoverflow.com/questions/61993738/flutter-doctor-android-licenses-gives-a-java-error> - 26/11/2023
- <https://stackoverflow.com/questions/49045393/message-flutter-run-no-connected-devices> - 26/11/2023
- <https://stackoverflow.com/questions/45523370/how-to-make-a-phone-call-from-a-flutter-app> - 02/12/2023
- <https://stackoverflow.com/questions/47046637/open-google-maps-app-if-available-with-flutter> - 02/12/2023
- <https://stackoverflow.com/questions/63224980/is-it-possible-to-hide-show-items-in-bottomnavigationbar-in-flutter> - 02/12/2023

- <https://stackoverflow.com/questions/45916658/how-to-deactivate-or-override-the-android-back-button-in-flutter> - 06/12/2023
- <https://www.youtube.com/watch?v=ZHdg2kfKmjI> - 08/12/2023
- <https://stackoverflow.com/questions/48927928/how-to-add-clear-button-to-textfield-widget> - 09/12/2023
- <https://www.dhiwise.com/post/practical-guide-to-integration-testing-in-flutter-app> - 11/12/2023
- <https://docs.flutter.dev/cookbook/testing/integration/introduction> - 11/12/2023
- <https://docs.flutter.dev/testing/integration-tests> - 11/12/2023
- <https://docs.flutter.dev/cookbook/testing/unit/introduction> - 11/12/2023
- <https://stackoverflow.com/questions/60355792/how-to-setup-a-widgettester-for-multiple-tests-in-flutter> - 11/12/2023
- <https://stackoverflow.com/questions/76303364/how-to-set-focus-on-search-bar-after-clicking-dropdownsearch-in-flutter> - 14/12/2023
- <https://stackoverflow.com/questions/43214271/how-do-i-supply-an-initial-value-to-a-text-field> - 14/12/2023
- <https://stackoverflow.com/questions/70434058/flutter-dropdownsearch-no-data-found-message-customization> - 14/12/2023
- <https://stackoverflow.com/questions/68726243/how-to-dismiss-an-open-dropdown-in-flutter> - 14/12/2023

11. Anexos

Se entrega junto a la memoria un manual de compilación para generar nuevas versiones de la aplicación y poder distribuirlas a dispositivos Android.