



# Celifinder: diseño y desarrollo de una aplicación de localización de restaurantes aptos para celíacos

**Blanca Lendoiro Valle**

Máster en Desarrollo de Aplicaciones Móviles

**Eduard Martín Lineros**

Enero de 2019



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

<b>Título del trabajo:</b>	Celifinder: diseño y desarrollo de una Aplicación de localización de restaurantes aptos para celíacos
<b>Nombre del autor:</b>	Blanca Lendoiro Valle
<b>Nombre del consultor:</b>	Eduard Martín Lineros
<b>Fecha de entrega (mm/aaaa):</b>	01/2019
<b>Titulación:</b>	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>La enfermedad celíaca es una enfermedad autoinmune provocada por la intolerancia permanente al gluten. El único tratamiento posible para esta enfermedad es la dieta estricta sin gluten, que excluye todos los alimentos que contengan este conjunto de proteínas, presentes fundamentalmente en cereales como el trigo, la cebada, el centeno y la avena, así como todos los alimentos que hayan entrado en contacto con algún alimento con gluten, lo que se considera contaminación cruzada.</p> <p>Las personas celíacas tienen numerosos problemas para encontrar establecimientos donde puedan comer, ya que el medio más utilizado para saber si pueden comer en un restaurante es preguntar directamente al encargado, lo que puede llegar a ser tedioso, tanto para la persona celíaca como para sus acompañantes. Actualmente, los restaurantes que adaptan su oferta a intolerancias alimenticias, no es muy grande y muchos usuarios se quejan de la poca fiabilidad de estas ofertas.</p> <p>Este trabajo consiste en el diseño y desarrollo de una aplicación que localice los restaurantes aptos para celíacos, más cercanos al usuario, evitando que tenga que preguntar en cada establecimiento si puede comer o no, añadiendo la información que el encargado daría al usuario cuando éste preguntara. Con ella se pretende añadir fiabilidad a los datos existentes sobre estos restaurantes en multitud de páginas web que suelen estar desactualizados y pueden llevar a errores a los usuarios, provocando así un perjuicio a su salud.</p>	

**Abstract (in English, 250 words or less):**

Celiac disease is an autoimmune disorder caused by permanent intolerance to gluten. The treatment for this disease is the strict diet without gluten, which excludes all foods that contain this set of proteins, mainly present in cereals such as wheat, barley, rye and oats, as well as all foods that have come into contact with any food with gluten, which is considered cross-contamination.

People with celiac disease have many problems when it comes to finding establishments where they can eat, because when they want to know if they can eat at a restaurant, they ask the manager, which can be tedious for both the person celiac as for those who accompany him.

This project consists of the design and development of an application that locates the restaurants suitable for coeliacs, closer to the user, avoiding having to ask in each establishment whether they can eat or not, adding the information that the manager would give the user when he asked . It aims to add reliability to existing data on these restaurants in a multitude of web pages that are often outdated and can lead to errors for users, causing harm to their health.

**Palabras clave (entre 4 y 8):**

Celíacos, aplicación, gluten, restaurantes, geolocalización, fiabilidad

# 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.....	8
2. Usuarios y contextos de uso.....	9
2.1. Fichas de usuario.....	9
2.2. Contextos de uso .....	11
3. Diseño conceptual .....	13
3.1. Escenarios de uso .....	13
3.2. Mapa de experiencia.....	14
3.3. Requisitos funcionales.....	16
3.4. Requisitos no funcionales .....	16
4. Prototipo.....	18
4.1. Sitemap .....	18
4.2. Prototipo de baja fidelidad.....	18
4.3. Moodboard.....	22
4.4 Navegación y módulos .....	24
4.5 Prototipo de alta fidelidad.....	25
4.6 Presentación del producto.....	28
5. Diseño Técnico.....	30
5.1 Diagrama UML de casos de uso .....	30
5.2 Listado de casos de uso.....	31
5.3 Diseño de la Base de Datos.....	40
5.4 Diseño de clases .....	41
5.5. Diagrama de la arquitectura de la aplicación.....	42
6. Implementación Backend.....	43
6.1 Spring Boot .....	43
6.2. API REST .....	44
6.3 MySQL .....	45
6.4 JPA .....	45
6.5 Lombok.....	46
7. Implementación Frontend .....	47
7.1 React-Native.....	47
7.2 Redux.....	48
7.3 NPM.....	49
7.4 Ramda.....	49
7.5 ESLint.....	50
7.6 React-Navigation.....	50
7.7 React-Native-Elements.....	50
7.8 React-Native-Maps.....	51
8. Otras herramientas.....	52
8.1 Visual Studio Code.....	52
8.2 XCode.....	52

8.3 Expo.....	53
8.4 Draw.io .....	53
8.5 Sketch .....	54
8.6 StarUML .....	54
8.7 InkScape.....	55
8.8 GitLab.....	55
8.9 SourceTree .....	55
8.10 Trello.....	56
8.11 Gantt Project.....	56
8.12 STS .....	57
8.13 Android Studio.....	57
9. Plan de pruebas.....	59
10. Líneas futuras .....	62
11. Conclusiones .....	64
12. Bibliografía.....	67
13. Anexo I. Manual de usuario .....	69
14. Anexo II. Manual de instalación .....	73
14.1 Base de datos.....	73
14.2 Backend .....	73
14.3 Instalación del Frontend.....	74
14.3.1 Arranque de la app con Expo .....	74
14.3.2 Arranque de la app con los proyectos nativos.....	75

## Lista de figuras

Ilustración 1. Respuestas a la pregunta ¿Padece usted la enfermedad celíaca y/o la padece alguien de tu entorno?.....	2
Ilustración 2. Respuestas a la pregunta ¿Tiene dificultades para encontrar restaurantes aptos para celíacos? .....	2
Ilustración 3. Respuestas a la pregunta ¿Cómo sabe si un restaurante es apto para celíacos?.....	2
Ilustración 4. Respuestas a la pregunta ¿Utiliza alguna herramienta para buscar restaurantes celíacos? .....	3
Ilustración 5. Logo de la aplicación ‘TripAdvisor’ .....	3
Ilustración 6. Logo de la página 'Viajar sin gluten' .....	3
Ilustración 7. Respuestas a la pregunta: Si ya ha utilizado alguna aplicación móvil para buscar restaurantes aptos para celíacos ¿Cuál fue la característica que más le gusto? ....	6
Ilustración 8. Respuestas a la pregunta ¿Qué cambiaría o mejoraría? .....	6
Ilustración 9. Diagrama de Gantt de la planificación del proyecto .....	7
Ilustración 10. Ficha de persona de María Pérez .....	9
Ilustración 11. Ficha de persona de Jorge Martín .....	10
Ilustración 12. Ficha de persona de David Sánchez.....	10
Ilustración 13. Diagrama de los parámetros que determinan el contexto de uso.....	11
Ilustración 14. Mapa de experiencia de la búsqueda por ubicación.....	15
Ilustración 15. Mapa de experiencia de la búsqueda mediante filtros.....	15
Ilustración 16. Sitemap de la aplicación.....	18
Ilustración 17. Moodboard de la aplicación.....	22
Ilustración 18. Logo de Celifinder para fondo oscuro.....	24
Ilustración 19. Logo de Celifinder para fondo claro .....	24
Ilustración 20. Presentación de la aplicación como producto .....	28
Ilustración 21. Diagrama UML de casos de uso .....	30
Ilustración 22. Diagrama de la base de datos.....	41
Ilustración 23. Diagrama de la relación entre clases .....	41
Ilustración 24. Diagrama de la arquitectura de la aplicación.....	42
Ilustración 25. Logo de Spring Boot .....	43
Ilustración 26. Esquema de funcionamiento de Spring Boot.....	43
Ilustración 27. Esquema de funcionamiento de API REST .....	44
Ilustración 28. Logo de MySQL .....	45
Ilustración 29. Arquitectura de las aplicaciones desarrolladas con React-Native .....	47
Ilustración 30. Arquitectura FLUX .....	48
Ilustración 31. Logo de Redux .....	48
Ilustración 32. Logo de NPM.....	49
Ilustración 33. Logo de Ramda .....	49
Ilustración 34. Logo de ESLint .....	50
Ilustración 35. Pantalla de Visual Studio Code.....	52
Ilustración 36. Logo de XCode .....	53
Ilustración 37. Pantalla de Draw.io .....	53
Ilustración 38. Logo de SketchApp.....	54
Ilustración 39. Pantalla de StarUML.....	54
Ilustración 40. Logo de InkScape.....	55
Ilustración 41. Ejemplo de pipelines en GitLab.....	55

Ilustración 42. Pantalla de SourceTree .....	56
Ilustración 43. Logo de Trello.....	56
Ilustración 44. Pantalla del proyecto de backend en STS.....	57
Ilustración 45. Logo de Android Studio .....	57
Ilustración 46. Pantalla de inicio (a) y Pantalla de lista(b) .....	69
Ilustración 47. Vista de mapa (a) y Filtro (b).....	70
Ilustración 48. Pantalla de detalle (Información - a y Opiniones - b) .....	71
Ilustración 49. Añadir opinión (a), pantalla de Login (b) y pantalla de nuevo usuario..	71
Ilustración 50. Menú lateral (a), pantalla de modificar perfil (b), pantalla de cambiar contraseña (c) .....	72
Ilustración 51. Pantalla de contacto.....	72
Ilustración 52. Consola de instalación de App con Expo .....	75



# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

La enfermedad celíaca es una enfermedad autoinmune provocada por la intolerancia permanente al gluten. Entre un 1 y un 2% de la población española es celíaca, lo que supone una cifra de entre 450.000 y 900.000 personas, aunque esta enfermedad está infradiagnosticada, ya que alrededor de un 80% de personas celíacas no están diagnosticadas.

Actualmente, la cantidad de restaurantes que adaptan su oferta a ciertas intolerancias alimenticias no es muy grande y muchos usuarios se quejan de la poca fiabilidad de estas ofertas. En el Reglamento N° 1169/2011 del Parlamento Europeo y del Consejo del 25 de Octubre de 2011, se especifica que los establecimientos que sirvan alimentos deben indicar las sustancias que puedan causar alergias o intolerancias, pero esto no siempre se cumple y en ocasiones estas indicaciones no son fiables o no tienen en cuenta la contaminación cruzada que pueda existir.

Las asociaciones de celíacos de las distintas comunidades autónomas, así como la Federación de Asociaciones de Celíacos de España (FACE), colaboran como entidades autorizadas que permiten a los establecimientos obtener el Sistema de Licencia Europeo Espiga Barrada, que garantiza el control de estos procesos para tranquilidad de las personas con alergias o intolerancias alimenticias. Pocos establecimientos cuentan con esta certificación, aunque hay bastantes que ya están adaptando su oferta.

La inflexibilidad de este tipo de dieta, dificulta que los establecimientos de hostelería y restauración, puedan disponer de una oferta apta para celíacos, ya que supondría tener utensilios de cocina y incluso algunos electrodomésticos para uso exclusivo de ingredientes libres de gluten. Supone un gasto extra y una preocupación más para los hosteleros, aunque también es considerado como un elemento diferencial respecto al resto de establecimientos, que puede aumentar sus ventas.

En este contexto se detecta una necesidad principal, que las personas celíacas y sus acompañantes puedan saber qué restaurantes y bares ofrecen platos adaptados a su intolerancia, de forma fácil, fiable y especificando qué tipo de oferta tienen, es decir, si ofrecen solo comidas, también tapas, pan sin gluten, etc. La cantidad de personas a las que se le ha diagnosticado esta enfermedad aumenta cada año en un 15% en España, debido al aumento de la sensibilización de la población, por ello esta necesidad que se ha detectado cada vez es más relevante al afectar a un mayor sector de la población.

Para conocer los hábitos que tienen los celíacos y sus allegados en referencia a este tema, he realizado una encuesta en la que han participado 42 personas que tienen la enfermedad celíaca o que conocen a alguien cercano que la tengan.

El número de personas con esta enfermedad no es muy alto, pero sí el número de personas al que afecta de forma indirecta, como se puede observar a continuación:

### ¿Padece usted la enfermedad celíaca y/o la padece alguien de su entorno?

42 respuestas

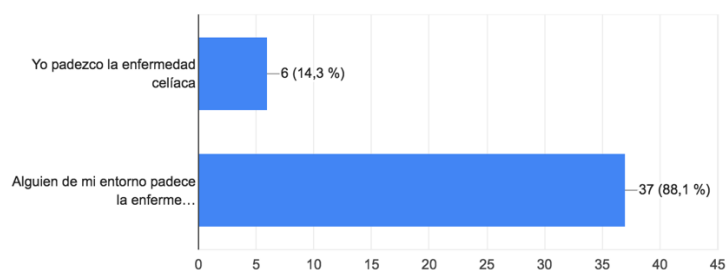


Ilustración 1. Respuestas a la pregunta ¿Padece usted la enfermedad celíaca y/o la padece alguien de tu entorno?

De todas estas personas, entorno a un 70% reconocen tener muchas o bastantes dificultades para encontrar restaurantes aptos para celíacos, y solo un 4% admite tener muy pocas dificultades, lo que refleja la problemática de los celíacos en este ámbito.

### ¿Tiene dificultades para encontrar restaurantes aptos para celíacos ?

42 respuestas

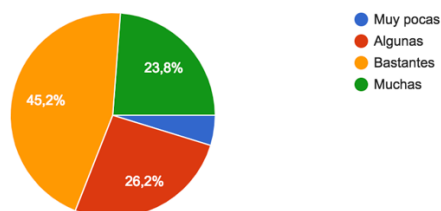


Ilustración 2. Respuestas a la pregunta ¿Tiene dificultades para encontrar restaurantes aptos para celíacos?

El medio más utilizado para conocer si un restaurante es apto para las personas con esta intolerancia es preguntar directamente a un camarero o encargado del propio establecimiento, seguido por investigar por Internet y preguntar a otras personas celíacas. De estos datos podemos deducir que la información que el usuario encuentra en internet no es suficientemente fiable o no sabe si está actualizada, ya que acaba preguntando a personas del establecimiento o a conocidos.

### ¿Cómo sabe si un restaurante es apto para celíacos?

42 respuestas

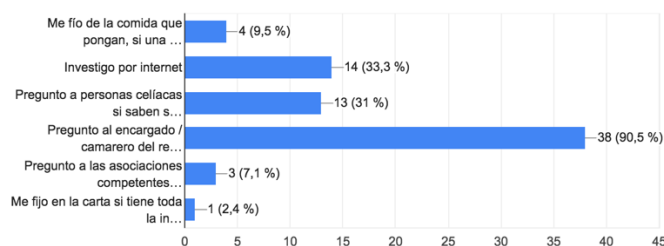


Ilustración 3. Respuestas a la pregunta ¿Cómo sabe si un restaurante es apto para celíacos?

Los medios tecnológicos utilizados para encontrar este tipo de establecimientos son las páginas web, pero en un porcentaje bastante bajo con respecto a lo que estamos acostumbrados con los

restaurantes en general, ya que solo un 45% aproximadamente, utiliza búsquedas en web o aplicaciones móviles. Sólo 2 personas de 42 han utilizado aplicaciones móviles con este fin.

¿Utiliza alguna herramienta para buscar restaurantes aptos para celíacos?

23 respuestas

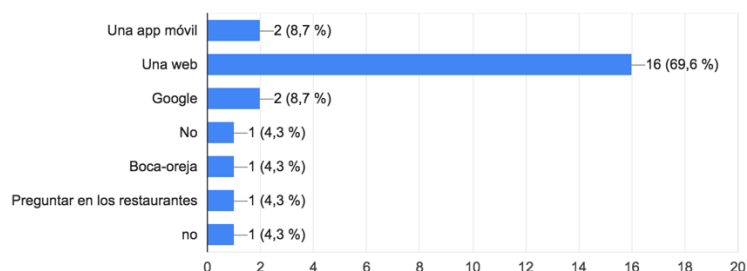


Ilustración 4. Respuestas a la pregunta ¿Utiliza alguna herramienta para buscar restaurantes celíacos?

De las webs utilizadas destacan TripAdvisor y Viajar SinGluten, además de la a web propia de los restaurantes. De las aplicaciones móviles utilizadas, solo se ha utilizado la App de TripAdvisor (2 de 42 personas).

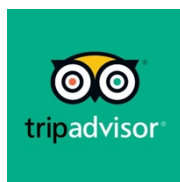


Ilustración 5. Logo de la aplicación 'TripAdvisor'

TripAdvisor es una web y aplicación de viajes muy completa que engloba restaurantes, hoteles, vuelos o actividades entre otras cosas. Uno de los filtros en la búsqueda de restaurantes es "Opciones sin gluten".

El problema que tiene este filtro es que sabemos que ese restaurante va a ofrecernos algo de comer, pero no sabemos si está certificado por la FACE y tienen una oferta variada, si sólo tienen un plato y ningún postre o si el menú sin gluten va a tener un precio mayor a otros menús.



Ilustración 6. Logo de la página 'Viajar sin gluten'

Viajar sin gluten es una web dedicada especialmente a la dieta sin gluten, ya que ofrece información sobre restaurantes, hoteles, albergues y tiendas sin gluten, entre otros. En general es una buena web que, pero no incluye restaurantes que tienen opciones sin gluten, pero no están certificados

Si realizamos una búsqueda de restaurantes en Valladolid, en TripAdvisor aparecen 55 restaurantes diferentes, mientras que en ViajarSinGluten 17 restaurantes. Estas diferencias son las que siembran la duda en los usuarios ya que no saben cuál de las dos opciones es válida.

A parte de estas dos opciones existen algunas aplicaciones más cuyo objetivo inicial era ser un buscador de restaurantes sin gluten, pero la mayoría de estas aplicaciones no están actualizadas, no contienen datos o han dejado de funcionar.

El resultado que se pretende obtener con este proyecto es crear una aplicación móvil que permita a los usuarios encontrar establecimientos con ofertas adaptadas a personas celíacas, ofreciendo la máxima información posible sobre estas ofertas, mediante el uso de etiquetas, que permita identificar fácilmente qué se van a encontrar los usuarios en los establecimientos.

## 1.2 Objetivos del Trabajo

El principal objetivo de este Trabajo de Fin de Máster es diseñar y desarrollar una aplicación, que tenga los siguientes objetivos:

- Permitir al usuario buscar establecimientos adaptados a una dieta sin gluten.
- Dar la información necesaria al usuario sobre los establecimientos sobre qué va a poder comer en ellos.
- Utilizar la geolocalización del dispositivo para poder mostrar los resultados de búsqueda Permitir al usuario logarse en la aplicación
- Añadir comentarios de los usuarios a los restaurantes para permitir recomendaciones entre ellos
- Etiquetar los establecimientos mediante un código de colores, que representen diferentes categorías, para facilitar así la identificación de los mismos.
- Reportar restaurantes en los que el usuario ha tenido experiencias negativas por aspectos como el trato o la incorrecta manipulación de los alimentos provocando contaminación cruzada.

Los objetivos secundarios que se persiguen en este Trabajo de Fin de Máster son:

- Conocer el estado del arte de las tecnologías de desarrollo de aplicaciones móviles
- Realizar un análisis completo de requisitos funcionales y no funcionales de la aplicación
- Planificar el proyecto con una metodología tradicional, marcando los hitos importantes en los que se realizarán las entregas pertinentes.
- Decidir las tecnologías que se van a utilizar en la aplicación y justificarlo explicando los motivos que han llevado a tomar esas decisiones.
- Hacer un diseño técnico de la aplicación especificando:
  - Diseño de las tablas de las bases de datos
  - Diseño del *backend*, describiendo los servicios REST que se van a exponer para la comunicación con la parte *frontend* de la aplicación
  - Diseño técnico de la parte *frontend*, explicando la estructura del proyecto, las tecnologías utilizadas y las librerías incluidas en el mismo.
- Realizar un diseño gráfico de la aplicación, detallando los siguientes aspectos:
  - Esbozando un prototipo de baja fidelidad para dar cuenta de la estructura visual de la aplicación
  - Proponer una paleta de colores y criterio cromático
  - Elegir las tipografías adecuadas y que representen el carácter del producto.
  - Crear un *Moodboard* con los criterios del diseño gráfico elegido.
  - Desarrollar una propuesta de diseño de interfaz visual con un Wireframe de alta fidelidad
  - Analizar si el diseño ofrece una buena experiencia de usuario comprobando los elementos relativos a la interacción del usuario con la aplicación

- Creación de una base de datos donde almacenar la información que necesita la aplicación
- Desarrollar el *backend* que conecte con la base de datos e implemente la lógica de los datos obtenidos para exponerlos a través de servicios REST.
- Desarrollar el frontend de la aplicación, acorde con los diseños creados anteriormente
- Optimizar el funcionamiento de la aplicación intentando reducir al máximo los tiempos de respuesta de los servicios.
- Crear un plan de pruebas que nos permita, al finalizar la aplicación, comprobar que se cumplen los requisitos planteados inicialmente en todos los casos de usuario.
- Redactar un manual de usuario que explique detalladamente qué funcionalidades tiene la aplicación y cómo se pueden utilizar.
- Desarrollar las conclusiones obtenidas durante el proyecto
- Investigar las líneas futuras de mejora del proyecto
- Familiarizarse con la redacción de artículos científicos para plasmar dichos conocimientos sobre la memoria de este proyecto.

### 1.3 Enfoque y método seguido

Las posibilidades que se nos presentan en este contexto son, por una parte, seguir la línea de las webs de búsqueda de restaurantes con un filtro concreto para este tipo de dieta y, por otra parte, crear una aplicación nueva, que se centre en las necesidades reales del usuario y que ofrezca una información específica y concreta sobre la oferta adaptada que tiene cada establecimiento.

La primera opción nos permitiría abarcar un amplio abanico de restaurantes añadiendo múltiples filtros, de forma que el espectro de usuarios será mucho mayor. El inconveniente que ofrece esta opción es que se competiría con webs que tienen una gran cantidad de datos, en muchos puntos de la geografía y sería realmente complicado poder conseguir usuarios, con una competencia de esas dimensiones.

Sin embargo, la segunda opción, la creación de una app específica y concreta, es la más viable, ya que permite que, desde un principio, se marquen características diferenciadoras y se capten usuarios más fácilmente. Por ello, se ha elegido esta estrategia, poniendo en valor las necesidades y opiniones de los usuarios.

La encuesta mencionada anteriormente, preguntaba a los usuarios sobre las características que más les gustaban de las webs o aplicaciones móviles que habían utilizado y las que les gustaría cambiar o mejorar. Los resultados son los siguientes:

Si ya ha utilizado alguna aplicación móvil para buscar restaurantes aptos para celíacos ¿Cuál fue la característica que más le gusto?

7 respuestas

Valoración, críticas...
Que lo indiquen por zonas (con un mapa en relación a tu zona más cercana)
Opiniones de otros usuarios
El rango de precio y la distancia
-
No procede
No he utilizado ninguna

Ilustración 7. Respuestas a la pregunta: Si ya ha utilizado alguna aplicación móvil para buscar restaurantes aptos para celíacos ¿Cuál fue la característica que más le gusto?

¿Qué cambiaría o mejoraría?

5 respuestas

La facilidad para encontrar este tipo de sitios.
Que fuera más fiable o que los encargados de los restaurantes confirmen que el restaurante es apto para celíacos.
-
La fiabilidad
No procede

Ilustración 8. Respuestas a la pregunta ¿Qué cambiaría o mejoraría?

Como características que más valoran los usuarios están las opiniones de otros usuarios, la localización y el rango de precios, mientras que la característica que más echan de menos es la fiabilidad de la información. Todo esto será el centro del enfoque que se ha elegido para poder atraer el mayor número de usuarios posibles y fomentar el uso de la aplicación.

## 1.4 Planificación del Trabajo

Los hitos marcados son los siguientes:

- 31 de octubre: Entrega del diseño
- 12 de diciembre: Entrega de la implementación
- 2 de enero: entrega final

La disponibilidad de horas por semana es la siguiente:

- Semana del 8 al 14 de octubre: 14h
- Semana del 15 al 21 de octubre: 22h
- Semana del 22 al 28 de octubre: 17h
- Semana del 29 de octubre al 4 de noviembre: 24h
- Semana del 5 al 11 de noviembre: 17h
- Semana del 12 al 18 de noviembre: 12h
- Semana del 19 al 25 de noviembre: 18h
- Semana del 26 de noviembre al 2 de diciembre: 20h

- Semana del 3 al 9 de diciembre: 25h
- Semana del 10 al 16 de diciembre: 18h
- Semana del 17 al 23 de diciembre: 20h
- Semana del 24 al 30 de diciembre: 29h

Suma un total de 236h a partir del entregable de la planificación del trabajo.

Las tareas a realizar se estiman de la siguiente forma:

1. Diseño (53h)
  - a. Usuario y contextos de uso: 5h
  - b. Diseño conceptual: 7h
  - c. Prototipo: 21h
  - d. Evaluación: 20h
2. Implementación (116h)
  - a. Creación de tablas BBDD: 6h
  - b. Diseño de API REST: 6h
  - c. Desarrollo de servicios: 30h
  - d. Desarrollo de pantallas: 69h
  - e. Definición de plan de pruebas: 3h
  - f. Realización de pruebas: 2h
3. Manual de usuario: 15h
4. Conclusiones: 4h
5. Líneas de mejora: 4h
6. Creación de paquetes para los diferentes SO móviles: 10h
7. Revisión de código: 6h
8. Revisión de memoria: 11h
9. Vídeo: 17h

Suma un total de 236h a partir del entregable de la planificación del trabajo.

A continuación, se muestra el diagrama de Gantt de la planificación del proyecto:

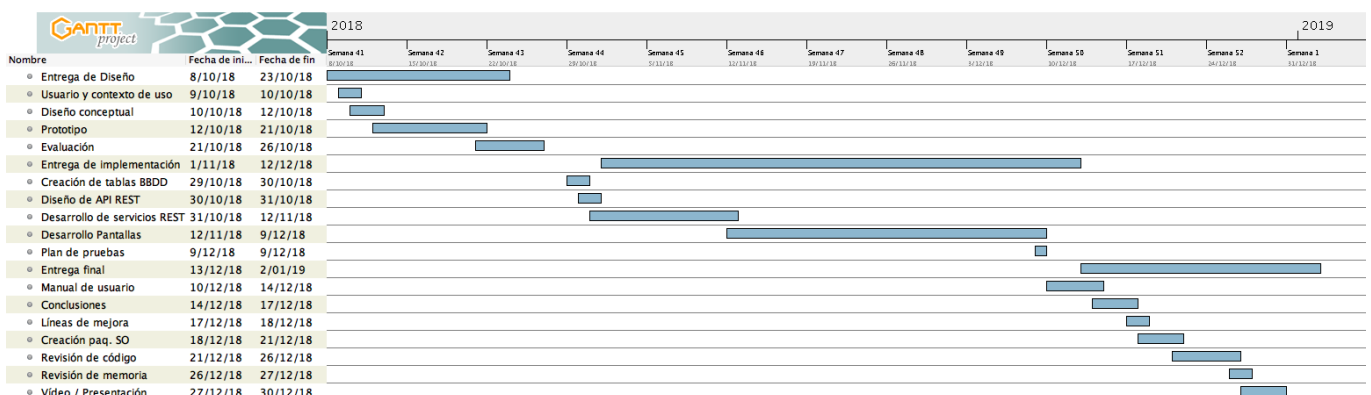


Ilustración 9. Diagrama de Gantt de la planificación del proyecto

## 1.5 Breve resumen de productos obtenidos

Los entregables producidos al finalizar el Trabajo Final Máster son:

- Memoria del Trabajo de Final de Máster, incluyendo
  - El plan de trabajo, incluyendo la justificación y contexto del trabajo, sus objetivos, la estrategia seguida y la planificación del proyecto
  - Análisis de los requisitos de la aplicación
  - Diseño técnico y gráfico de la aplicación
  - Las tecnologías utilizadas y su descripción
  - Plan de pruebas de la aplicación
  - Manual de usuario de la aplicación
  - Conclusiones obtenidas y líneas a seguir en un futuro
- Aplicación desarrollada atendiendo a los criterios técnicos descritos en la memoria
- Presentación que indique los aspectos más relevantes del proyecto realizado, mostrando el funcionamiento de la aplicación desarrollada.



## 2. Usuarios y contextos de uso

En este capítulo se abordará la primera fase de diseño de la aplicación. En primer lugar, se utilizará una herramienta que tienen en cuenta tanto la perspectiva de usuario como la perspectiva de negocio. (0) Se trata de la herramienta de fichas de personas que definió Alan Cooper. Esta herramienta consiste en segmentar lo máximo posible y definir un usuario específico para nuestra aplicación, inventando a una persona concreta que lo represente, con un nombre, unos objetivos, unos problemas y un entorno. Con ello perseguimos obtener perfiles a los que preguntar cuando tengamos que tomar una decisión de diseño. (1) En segundo lugar, se analizarán los contextos de uso de la aplicación, es decir, las condiciones bajo las cuales la aplicación se va a utilizar en una situación normal o cotidiana.

### 2.1. Fichas de usuario

A continuación, se van a describir las fichas de personas que representan a los usuarios objetivo de la aplicación:

*Primera ficha de persona: María Pérez*

**María Pérez**  
Inspectora de trabajo

“ Como fuera muchas veces, pero pierdo mucho tiempo buscando restaurantes aptos para celíacos ”

**Tipos de apps que utiliza**

Organización	_____●_____
Compartir ficheros	_____●_____
Comunicación	_____●_____
Viajes y restauración	_____●_____
Noticias	_____●_____
Juegos	_____●_____

**Biografía**  
María es inspectora de trabajo destinada en Valladolid. Estudió derecho y dedico los años posteriores a estudiar la oposición que daba acceso a la plaza que ostenta. A María se le diagnosticó la enfermedad celíaca hace 8 años, después de sufrir múltiples molestias gástricas. Esto le dificulta mucho algunas de las actividades sociales a las que estaba acostumbrada, tales como salir por las noches a tomar algo o comer fuera. Hoy en día, debido a su trabajo le resulta complicado comer en casa, y frecuentemente sale a comer con sus jefes y otros compañeros inspectores y resulta muy incómodo para ella, ir preguntando por los restaurantes si tienen una oferta apta para ella, mientras todos esperan a que encuentre un sitio adecuado

**Objetivos**

- Conseguir una buena relación con sus superiores, ya que quiere hacer propuestas en la Inspección
- Tener una vida social sin restricciones debido a su enfermedad celíaca
- Viajar con su familia

**Necesidades**


- Conocer qué lugares tienen una oferta apta para celíacos
- Evitar situaciones incómodas en las que tiene que hacer esperar a otras personas por no encontrar lugares aptos para ella
- Comer sin preocupaciones, allá donde vaya

**Conocimiento tecnológico**

Redes Sociales	_____●_____	Ofimática	_____●_____
Programación	_____●_____	Apps móviles	_____●_____

Ilustración 10. Ficha de persona de María Pérez

## Segunda ficha de persona: Jorge Martín



**Jorge Martín**  
Estudiante de Máster

“ Quiero buscar restaurantes aptos para celíacos de buena calidad para llevar a mi novia ”

**Tipos de apps que utiliza**

Organización	_____●_____
Compartir ficheros	_____●_____
Comunicación	_____●_____
Viajes y restauración	_____●_____
Noticias	_____●_____
Juegos	_____●_____

### Biografía

Jorge es licenciado en Administración y Dirección de Empresas y actualmente estudia un Máster en Dirección Comercial y Marketing en Valladolid. A Jorge siempre le ha gustado mucho salir con sus amigos y sobretodo, la gastronomía de su ciudad, que tiene mucha fama sobretodo por sus tapas. Tiene novia desde hace 5 años, ella también vive en Valladolid y ha sido diagnosticada con la enfermedad celíaca. Esto ha afectado en gran medida en los planes que tienen juntos ya que los establecimientos a los que acuden se han restringido mucho y en ocasiones no acaban de encontrar dónde ir para cenar o para picar algo con el grupo de amigos

### Objetivos

- Terminar el Máster en Dirección Comercial y Marketing
- Trabajar en el negocio de la restauración y montar su propio negocio
- Poder hacer los mismos planes que antes con su grupo de amigos y su novia

### Necesidades

- Saber a donde puede llevar a su novia a comer, sin tener que ir de bar en bar con sus amigos
- Conocer la oferta gastronómica de la ciudad y detectar las posibles carencias que tenga

### Conocimiento tecnológico



Ilustración 11. Ficha de persona de Jorge Martín

## Tercera ficha de persona: David Sánchez



**David Sánchez**  
Project Manager Tecnológico

“ Me gusta hacer sentir bien a todo el equipo, tenerlos a todos en cuenta ”

**Tipos de apps que utiliza**

Organización	_____●_____
Compartir ficheros	_____●_____
Comunicación	_____●_____
Viajes y restauración	_____●_____
Noticias	_____●_____
Juegos	_____●_____

### Biografía

David es Project Manager en WorldTecnoApps, una multinacional que se encarga de diseñar y desarrollar aplicaciones para grandes empresas con tecnologías punteras. Lidera un equipo de unas 15 personas, todas son ingenieros jóvenes a los que les gusta desarrollar soluciones diferentes que hace que la empresa destaca dentro de su ámbito en el mercado. Para David es especialmente importante, mantener la motivación dentro del equipo, hacer que todos se sientan partícipes dentro de él, asumiendo los diferentes roles que son necesarios. Para ello suele salir todos los jueves con ellos a lo que se conoce como “Jueves de Tapas”, para fomentar la unión, pero una chica de su equipo es celíaca y siempre tienen problemas para encontrar sitios donde puede picar algo con todos, ya que la mayoría de sitios de tapas no tienen una oferta apta para celíacos

### Objetivos

- Aportar soluciones punteras en los proyectos en los que trabaja
- Fomentar la unión y la motivación entre todos los integrantes del equipo

### Necesidades

- Saber qué establecimientos puede frecuentar cuando sale con todo el equipo para no dejar a nadie fuera
- Garantizar que cuando se hacen eventos de empresa, se elijan lugares con una oferta adaptada a las intolerancias de todos los empleados (no solo de su equipo)

### Conocimiento tecnológico

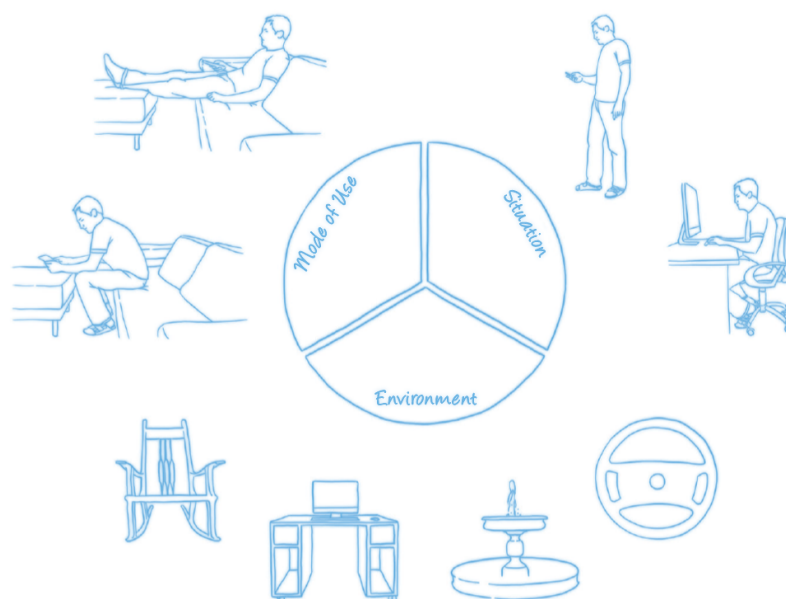


Ilustración 12. Ficha de persona de David Sánchez

Estos usuarios representan a los diferentes segmentos de clientes a los que está dirigida la aplicación. En primer lugar, el cliente objetivo principal son los propios celíacos que quieren encontrar establecimientos aptos para la dieta sin gluten, evitando problemas o molestias a los demás. En segundo lugar, la aplicación también tiene como objetivo el entorno más cercano de las personas con enfermedad celíaca tales como pareja, amigos cercanos o familiares, que tienen contacto permanente con ellos y que se adaptan a sus necesidades. Por último, la aplicación pretende ayudar también al círculo cercano de la persona celíaca, con los que tiene contacto ocasional, tales como compañeros de trabajo, de tal forma que ellos quieren tener en cuenta que no puede tomar gluten y buscan en qué lugares esta persona puede comer y sentirse bien, garantizando la seguridad alimentaria que necesita.

## 2.2. Contextos de uso

El contexto de uso se puede determinar a través de tres parámetros: el entorno, la situación y el modo de uso (2)



*Ilustración 13. Diagrama de los parámetros que determinan el contexto de uso*

La aplicación desarrollada va a tener dos contextos de uso concretos. Por un lado, el contexto correspondiente al usuario que se encuentra por la calle y tiene la necesidad de encontrar un restaurante celíaco y utiliza la aplicación con esa finalidad. Por otro lado, también se tiene en cuenta el contexto correspondiente al usuario que va a planificar una visita a un establecimiento de este tipo y lo busca con anterioridad, es decir, el día antes o antes de salir de casa.

Para el primer caso, al que denominaremos “búsqueda sobre el terreno”, tenemos el siguiente contexto:

- Entorno: el usuario se encuentra en la calle, en un lugar público, más concretamente en una ciudad o núcleo urbano donde existe una oferta

gastronómica variada. Puede estar acompañado de más personas, interesadas también en el resultado de la búsqueda.

- Situación: el usuario está en movimiento usando su *Smartphone*. Las interacciones con su móvil serán las menores posibles, ya que su posición no es estacionaria o pretende que deje de serlo lo antes posible al estar en el exterior. La localización del usuario puede cambiar en esta situación. La luz puede no ser siempre la más adecuada y puede dificultar la visión de la misma si los colores no se adaptan a esta característica. Lo más probable es que esté utilizando los datos móviles de su dispositivo al estar en movimiento, aunque es posible que en algún momento pueda tener conexión Wifi.
- Modo de uso: *lean forward*, ya que el usuario está concentrado en realizar la búsqueda, está interrumpiendo otras actividades y cambiando de dirección según los resultados que le muestre la aplicación. Su atención a lo que le muestra la aplicación es continuada.

Para el segundo caso, al que denominaremos “planificación de visita”, el contexto de uso es el siguiente:

- Entorno: el usuario se encuentra en un lugar cerrado, como su propia casa u hotel, en el caso de que se encuentre fuera de su ciudad. Normalmente estará solo, si el usuario es la persona con la enfermedad celíaca o acompañado, si el usuario es la persona acompañante del celíaco. Se encontrará en un ambiente relajado y sin prisas.
- Situación: El usuario se encuentra en una posición estacionaria, por lo que le es más cómodo realizar más interacciones con su dispositivo móvil. La luz a la que está expuesto el móvil es correcta, agradable y no supondrá un problema. La conexión que está utilizando será Wifi, en la mayoría de las ocasiones. La localización del usuario no cambia, pero no es muy relevante, ya que pretende planificar una visita a un establecimiento que se encuentre en un punto que no tiene que estar necesariamente cerca del punto donde se encuentra, pudiendo estar incluso en otra ciudad.
- Modo de uso: *lean back* ya que el usuario se encuentra relajado y pasivo, aunque interesado en la búsqueda, pero puede estar sometido a diferentes interrupciones ya que no necesita el resultado de forma inmediata. Su atención puede no ser continuada.

Tras analizar estos dos contextos de uso, se concluye que es necesario tener en cuenta ambos, priorizando las necesidades más acuciantes del usuario en cada uno de ellos. Para el usuario en “búsqueda sobre el terreno” debemos tener en cuenta que las interacciones con el dispositivo deben ser las menos posibles, que los colores deben tener el suficiente contraste como para verse correctamente en el exterior y el consumo de datos debe optimizarse lo máximo posible. Para el usuario en “planificación de visita” debemos tener en cuenta que puede querer más información para profundizar sobre los datos de un establecimiento (oferta apta para celíacos, dirección o datos de contacto).

## 3. Diseño conceptual

En este capítulo se profundizará en la información que se ha recopilado hasta el momento para poder especificar las necesidades de los usuarios y determinar así el diseño conceptual de la aplicación. Para ello se empezará describiendo diferentes escenarios de uso asociados a las personas y contextos de uso del capítulo anterior. A continuación, se va a presentar el mapa de la experiencia que viviría el cliente al utilizar la aplicación. Finalmente se describirán los requisitos tanto funcionales como no funcionales de la aplicación, que nos permitirán continuar con el diseño de la aplicación.

### 3.1. Escenarios de uso

A continuación, se van a describir diferentes situaciones en las que los diferentes perfiles de la aplicación utilizarán la aplicación

#### *Primer escenario de uso: María Pérez*

María trabaja como inspectora de trabajo en Valladolid. Periódicamente los altos cargos de su trabajo van a las distintas delegaciones en diferentes puntos de España. Cuando María está terminando su jornada laboral y recibe una llamada de un compañero inspector, también de Valladolid que le cuenta que los altos cargos de su trabajo están llegando en el AVE y que quieren comer con ellos. María se dirige a la estación de trenes rápidamente y se encuentra con su compañero y sus jefes. Ella siempre ha ido a restaurantes muy informales donde sabe que tienen una oferta apta para ella, pero esta vez deben ir a un restaurante formal, ya que se trata de una reunión muy importante. En ese momento María abre la aplicación de CeliFinder rápidamente, sin hacer perder el tiempo al resto y busca restaurantes cercanos a la estación de trenes.

Ve que a unos 300 metros de la estación se encuentra un restaurante conocido por su cocina Gourmet de los productos de la región y que además está Certificado por la FACE (Federación de Asociaciones de Celíacos de España). María les propone ese restaurante donde pueden comer y, además, probar la gastronomía típica de la zona. Según los comentarios que ha podido leer en la aplicación de camino a este restaurante, el resto de usuarios escriben muy buenos comentarios sobre él.

Finalmente comen en este establecimiento y efectivamente, tal y como venía en la información de la aplicación, tienen una amplia oferta de platos sin gluten y el trato es muy bueno.

#### *Segundo escenario de uso: Jorge Martín*

Jorge está estudiando su máster durante los días de la semana. El viernes es el cumpleaños de su novia, con la que lleva 5 años y quiere darle una sorpresa, llevándola a cenar a un restaurante. Su novia es celíaca y normalmente van a comer o a cenar a una hamburguesería que es apta para ella, pero quiere sorprenderla con un restaurante diferente al que van normalmente.

Para ello ha pensado en llamar a su grupo de amigos para que le ayuden a darle una sorpresa, quiere que queden todos de tapas, el viernes con la excusa de su cumpleaños, para después dejarlos solos e invitarla a cenar a un buen restaurante. Jorge, la semana

anterior había estado buscando en la app CeliFinder otros restaurantes aptos para celíacos, donde ella nunca había estado y reservo en un mexicano, una de las comidas que más le gustan. El grupo de Jorge suele ir a tomar tapas, pero su novia no suele, porque no sabe donde puede tomarlas así que cuando salen abren la aplicación de CeliFinder y buscan todos los bares donde puedan tomar tapas las personas con enfermedad celíaca.

La novia de Jorge pudo celebrar su cumpleaños con todos tomando tapas y cenando en un mexicano, pudiendo conocer sitios que no conocía anteriormente, donde tienen una oferta apta para celíacos.

*Tercer escenario de uso: David Sánchez*

David acaba de formar su equipo de ingenieros para la unidad de tecnologías avanzadas en la nueva oficina de su empresa. Para que se conozcan todos los integrantes y haya un buen ambiente desde el principio ha pensado que, periódicamente, va a intentar hacer una quedada cada semana con todos los compañeros que puedan. Además, en su ciudad, tienen una iniciativa para aumentar las ventas de los establecimientos de tapas, que se llama “Jueves de tapas”, donde se baja el precio de sus raciones.

Con todo ello David convoca a todos los que quieran el jueves a tomar tapas. Una vez allí, se da cuenta que una integrante del equipo, Sara, es celíaca, por lo que tiene bastante difícil encontrar algún sitio a donde ir. Sara le dice al resto que pueden ir a cualquier sitio, que no le importa no tomar nada de comer, ya que está acostumbrada a esto. David no está de acuerdo con esto, porque piensa que ella también tiene que poder tomar tapas como el resto. Por ello, David recurre a la app CeliFinder y comprueba que hay un filtro dentro de la búsqueda de los establecimientos para encontrar lugares donde tapear. Tras realizar la búsqueda se dan cuenta de que hay varios restaurantes donde puede tomar algunas tapas y se dirigen hacia allí.

Sara se siente muy agradecida y comprueba que se puede tomar tapas en sitios donde no conocía, además ven que hay más sitios por otras zonas que podrán ver en las próximas veces que vayan todos juntos de tapas.

## 3.2. Mapa de experiencia

El mapa de experiencia es un diagrama que representa el proceso que sigue el usuario desde que comienza su viaje en la aplicación hasta que deja de utilizarla. En ella se especifican diferentes pasos, describiendo los sentimientos y la actitud del usuario, así como la función de la aplicación y las interacciones entre el usuario y el dispositivo al usar la aplicación, en las diferentes fases.

Se ha creado dos mapas de experiencia diferentes, representando dos de los contextos de uso más representativos de la aplicación: la búsqueda por ubicación sobre el terreno y la planificación de visitas a un restaurante realizando una búsqueda mediante filtros:

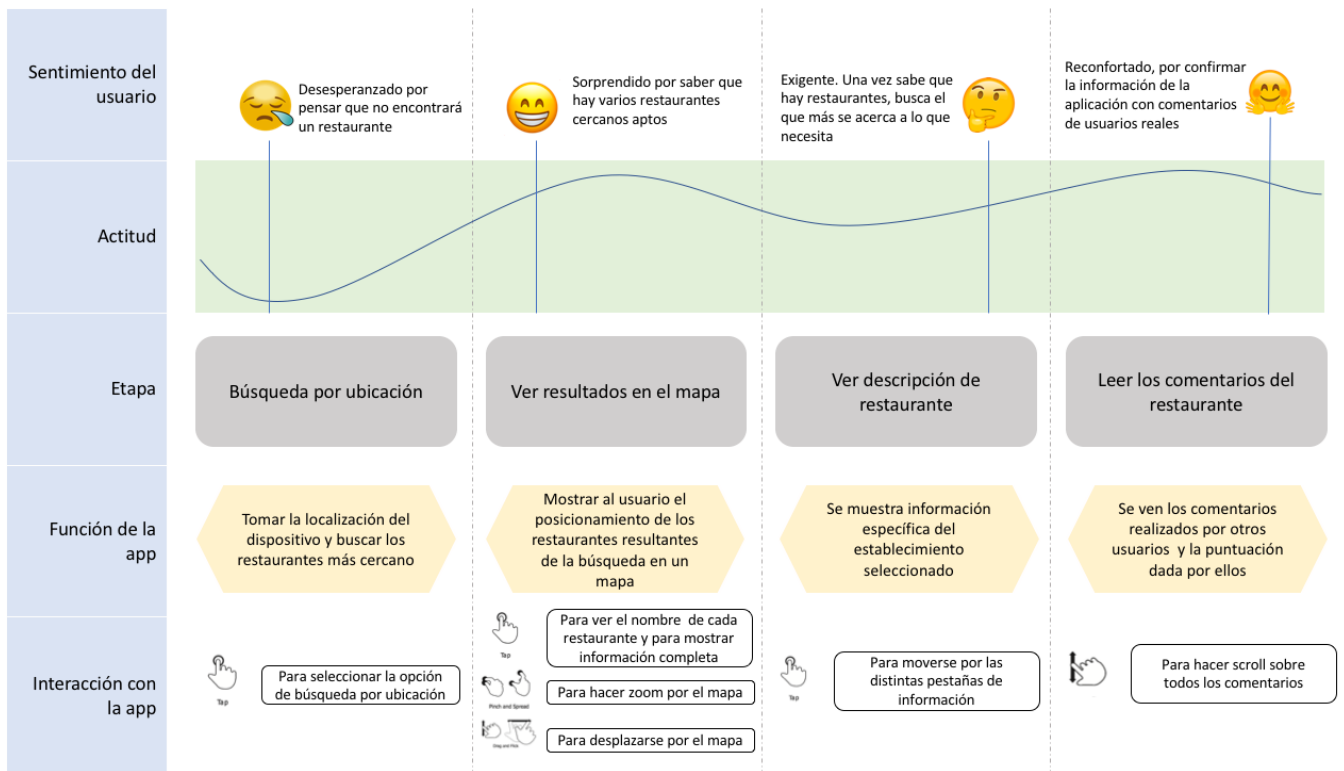


Ilustración 14. Mapa de experiencia de la búsqueda por ubicación

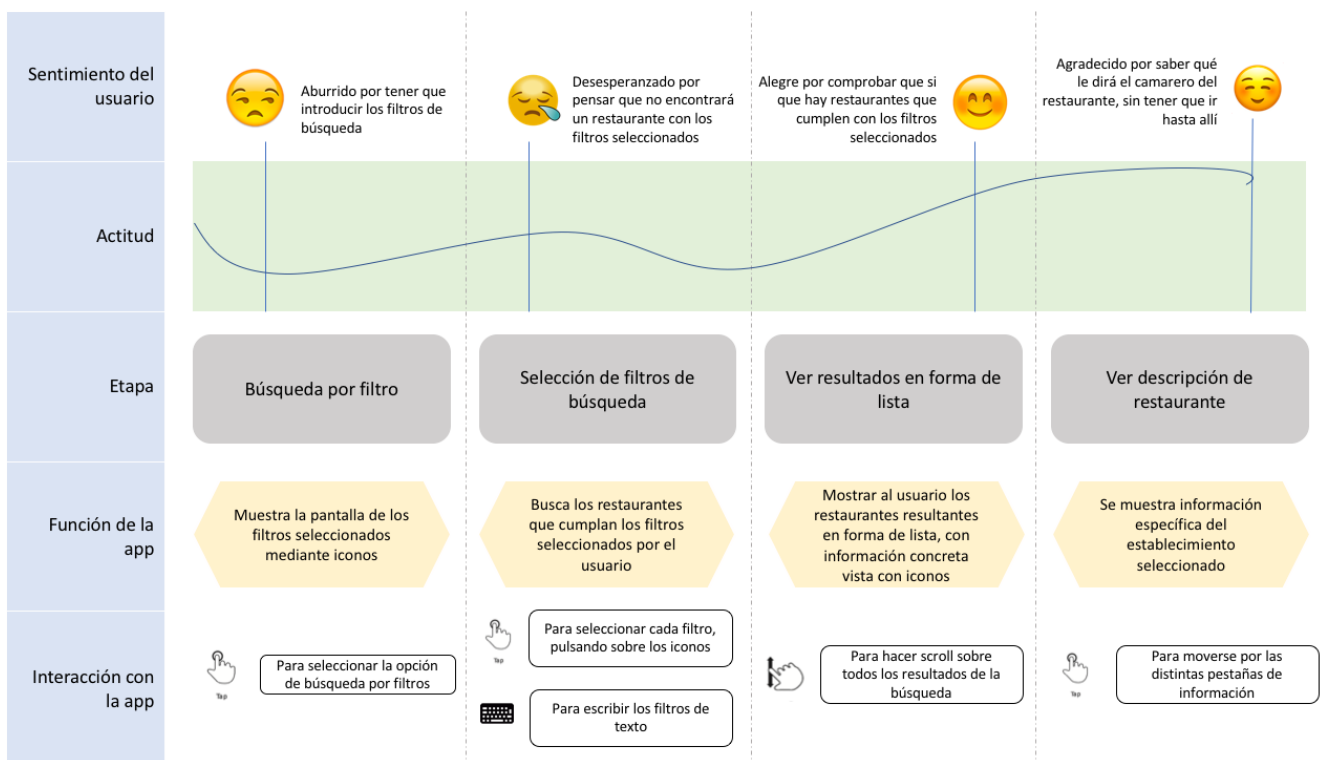


Ilustración 15. Mapa de experiencia de la búsqueda mediante filtros

### 3.3. Requisitos funcionales

Los requisitos funcionales identificados tras el análisis de las necesidades de los usuarios objetivo de la aplicación, son los siguientes:

- RF01: La aplicación permitirá logarse al usuario
- RF02: La aplicación permitirá buscar los restaurantes más cercanos al usuario mediante la geolocalización del dispositivo
- RF03: La aplicación permitirá buscar restaurantes a través de diferentes filtros
- RF04: La aplicación mostrará los resultados de la búsqueda a través de un mapa o a través de una lista, según la opción que elija el usuario
- RF05: La aplicación mostrará la información detallada del establecimiento seleccionado, organizándola a través de diferentes secciones
- RF06: La aplicación permitirá al usuario añadir comentarios a los establecimientos
- RF07: La aplicación permitirá al usuario reportar malas experiencias o información errónea mostrada en la misma
- RF08: La aplicación permitirá editar el perfil del usuario logado
- RF09: La aplicación permitirá cambiar la contraseña al usuario logado
- RF10: La aplicación permitirá crear un nuevo perfil a un usuario no logado

### 3.4. Requisitos no funcionales

Los requisitos no funcionales de la aplicación son los siguientes:

- RNF01: La aplicación va a ser desarrollada tanto para Android como para iOS
- RNF02: La aplicación tendrá control de errores, de forma que, si una consulta no se resuelve con éxito, se informará al usuario de la situación
- RNF03: La aplicación necesitará usar la geolocalización del dispositivo de forma opcional, para realizar la búsqueda por posicionamiento
- RNF04: Los colores usados en la aplicación tendrán el contraste suficiente para poder facilitar su visualización en el exterior con diferentes niveles de luminosidad



- RNF05: El tiempo de aprendizaje de la aplicación por un usuario deberá ser menor de 5 minutos
- RNF06: La aplicación contará con un manual de usuario como ayuda para su uso
- RNF07: La aplicación podrá mostrarse en español y en inglés.

## 4. Prototipo

En el cuarto capítulo de esta memoria se va a abordar la el diseño de la interfaz de la aplicación, diseño que ha sido obtenido como resultado del análisis del diseño conceptual confeccionado, así como de los requisitos a cumplir. En primer lugar, se mostrará el Sitemap que dará una idea de la organización de pantallas que tendrá la aplicación. A continuación, se plasmará un primer boceto de las pantallas en el prototipo de baja fidelidad. En tercer lugar, se desarrollará un Moodboard donde se definirá la gama cromática que predominar en las pantallas, así como la fuente de texto y los iconos. También se abordará los módulos de los que consta la aplicación y el tipo de navegación que permitirá al usuario moverse entre las pantallas. A continuación, se diseñará el prototipo de alta fidelidad que nos permitirá ver el aspecto real que tendrá la aplicación. Finalmente se realizará un análisis de UX (User Experience) que evaluará la usabilidad del diseño desarrollado, para terminar, presentando el diseño final del producto.

### 4.1. Sitemap

El sitemap que se va a mostrar a continuación es un diagrama que va nos servirá de ayuda para planificar la aplicación. Mostrará todas las pantallas que formarán la aplicación, así como la forma de acceder a ellas.

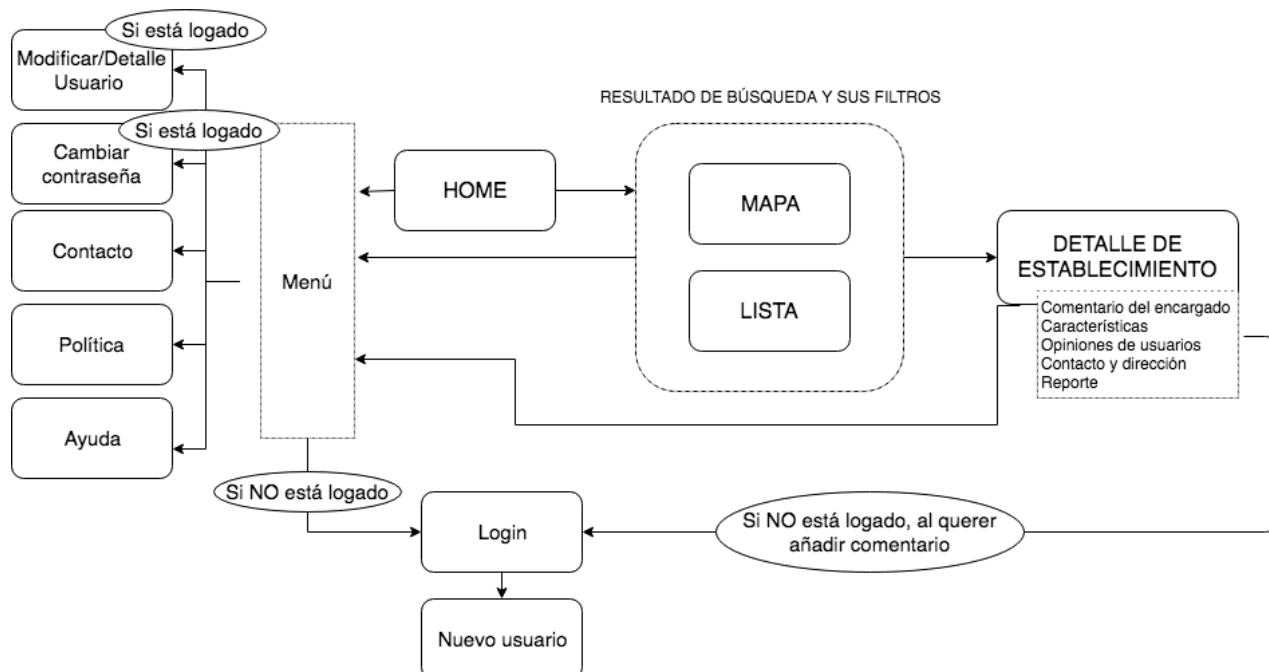


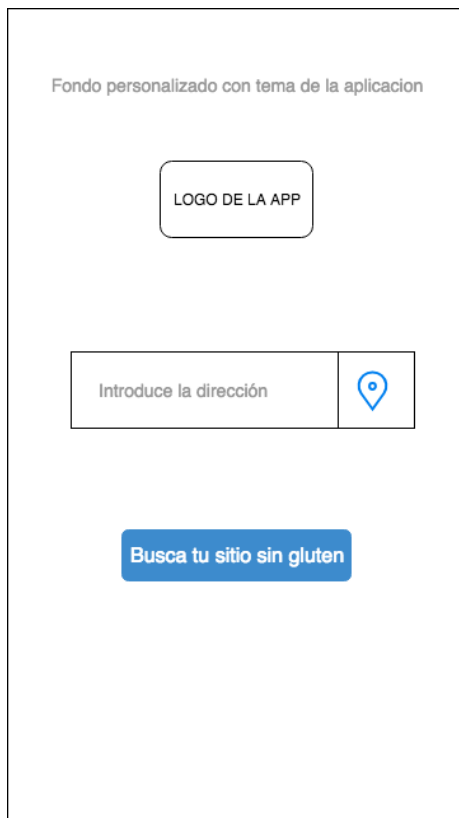
Ilustración 16. Sitemap de la aplicación

### 4.2. Prototipo de baja fidelidad

El prototipo de baja fidelidad muestra aspectos generales de la aplicación, sin entrar en detalles ni matices de colores, fuentes de textos o iconos que se diseñarán o escogerán a conciencia según sus características.

A continuación, se mostrarán los prototipos de baja fidelidad de las pantallas que se han señalado en el sitemap anterior.

## Pantalla de inicio



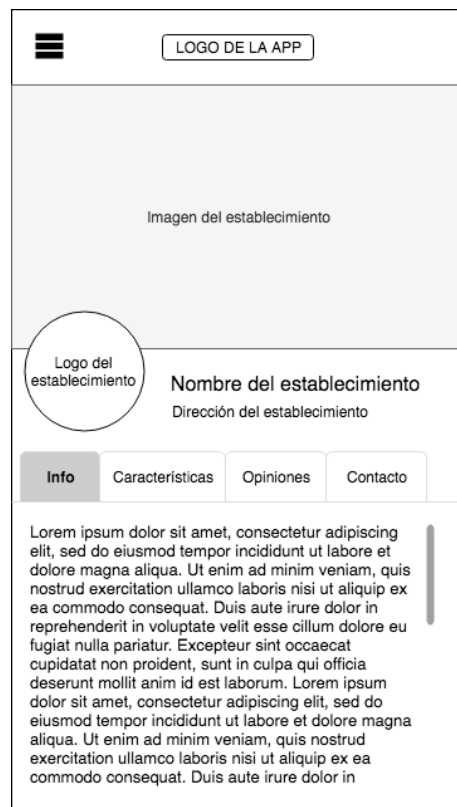
## Resultados mostrados en forma de mapa



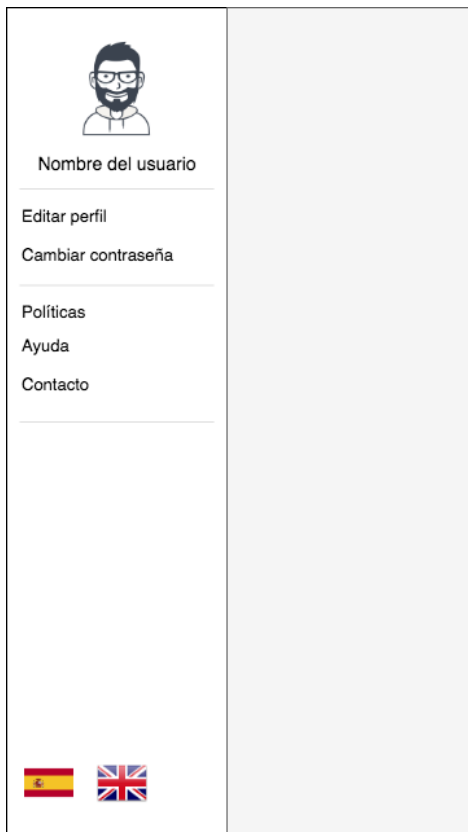
## Resultados mostrados en forma de lista




## Detalle del establecimiento



## Menú



Menú de usuario con un perfil de usuario y opciones de configuración.



Nombre del usuario



Editar perfil

Cambiar contraseña

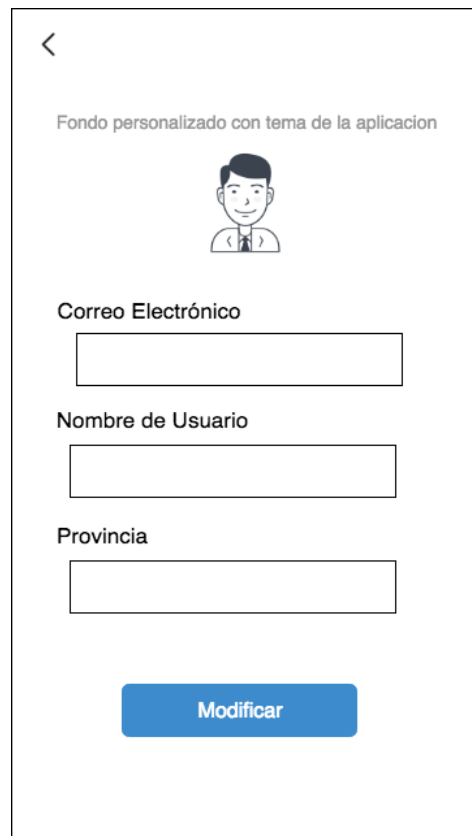
Políticas

Ayuda


Contacto


## Modificación / Detalle de usuario



Formulario para modificar los datos de usuario.



Fondo personalizado con tema de la aplicación



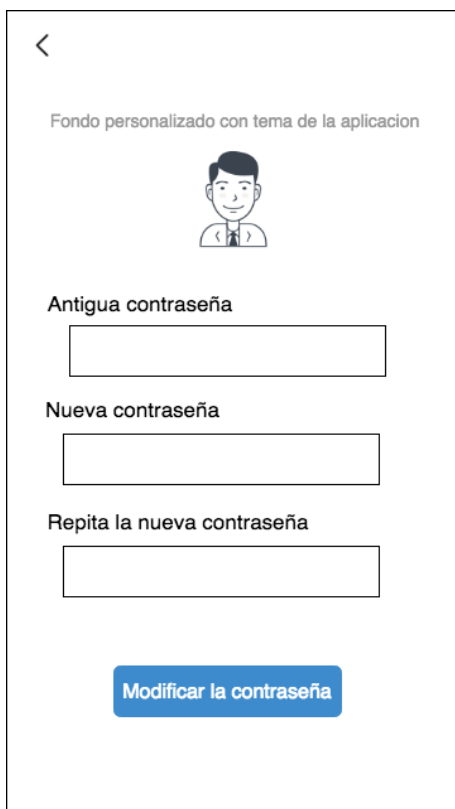
Correo Electrónico

Nombre de Usuario


Provincia

**Modificar**


## Cambiar de contraseña



Formulario para cambiar la contraseña de usuario.



Fondo personalizado con tema de la aplicación



Antigua contraseña

Nueva contraseña

Repita la nueva contraseña

**Modificar la contraseña**

## Login



Formulario de inicio de sesión.

Fondo personalizado con tema de la aplicación




Usuario

Contraseña

**Acceder**

## Contacto

<

  
**Contacto**

**Correo Electrónico**

**Nombre y apellidos**


**Contenido**

**Enviar**

## Ayuda


< **Ayuda**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

## Política

< **Política**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

## Nuevo usuario

< **Nuevo usuario**

**Correo Electrónico**

**Nombre de Usuario**

**Provincia**

**Nueva contraseña**

**Repita la nueva contraseña**

**Crear nuevo usuario**

### 4.3. Moodboard

A continuación, se van a determinar los colores, las fuentes, tanto de texto como de iconos, así como el logo de la aplicación. Todo ello se muestra en el siguiente Moodboard.

Un Moodboard es una herramienta visual que nos permite plasmar algunos aspectos de la interfaz visual de la aplicación, de tal forma que se muestra lo que queremos inspirar con la aplicación a través de sus colores y fuentes.



Ilustración 17. Moodboard de la aplicación

#### Colores

Los colores utilizados son los siguientes:



El color azul oscuro transmite, por una parte, credibilidad, confianza y profesionalidad, que representan tres aspectos que la aplicación pretende transmitir al usuario, poniendo especial atención en la credibilidad de los datos mostrados en la misma. Por otra parte, también transmite armonía y tranquilidad que permite atraer al usuario.



El color amarillo es un color fuerte que estimula ciertas facultades mentales e inspira alegría y creatividad. Al ser un color que también puede transmitir sentimientos negativos, se utilizará como color de contraste y de forma secundaria en la aplicación.



Finalmente, el color blanco, será el color de fondo que nos permitirá tener un gran contraste entre el fondo y el color principal, el azul oscuro.

(3) La W3C ha desarrollado una técnica de evaluación de los colores para comprobar que son suficientemente contrastables para poder verlos correctamente. Este algoritmo parte de los valores RGB que definen al color:

Brillo:

$$((R * 299) + (G * 587) + (B * 114)) / 1000$$

Contraste:

$$(Max(R1, R2) - Min(R1, R2)) + (Max(G1, G2) - Min(G1, G2)) \\ + (Max(B1, B2) - Min(B1, B2))$$

Para el color azul oscuro obtenemos como brillo: 47.72

Para el color blanco obtenemos como brillo: 255

Para el contraste obtenemos un valor de: 600

Según sus estándares la diferencia entre el brillo de ambos colores debe ser mayor de 125, en nuestro caso es de 205,28, por lo que si que cumpliría con este criterio.

El contraste debe tener un valor mayor a 500, en nuestro caso es de 600, por lo que también cumpliríamos con este criterio.

### Fuentes de texto

La fuente de texto principal será Helvética Neue, un tipo de fuente sans-serif, que se ha hecho popular cuando Apple la llevo a sus dispositivos. Al ser sans-serif, se trata de una tipografía que carece de terminaciones en sus letras.

Para títulos también se utilizará una letra sans-serif, en este caso Franklin Gothic Book, una tipografía muy característica de América del Norte.

### Fuentes de iconos

La fuente que se utilizará para los iconos comunes será Icomoon, que proporciona un paquete de iconos vectorizados. Algunos iconos se diseñarán especialmente para la aplicación al tener un contenido muy ligado a los celíacos y no ser iconos de uso común.

### Nombre y logo

La aplicación tendrá como nombre CeliFinder. La primera parte del nombre, “celi” se corresponde con los celíacos, ya que la aplicación está diseñada para las personas con enfermedad celíaca y su entorno. “Finder” representa a un buscador, en este caso de establecimiento con una oferta apta para celíacos.

El logo de la aplicación será el siguiente:



*Ilustración 18. Logo de Celifinder para fondo oscuro*



*Ilustración 19. Logo de Celifinder para fondo claro*

Tiene el nombre de la aplicación con la fuente de texto Savoye LET, poniendo la primera parte del nombre a una altura superior que la segunda parte del mismo para diferenciar bien ambas partes.

Además, se añade como icono una lupa, que representa la búsqueda que realiza la aplicación. Dentro de esta lupa esta una figura que representa el trigo el cereal más representativo que contiene gluten y que más problemas genera al colectivo celíaco. Sus espigas son pentágonos, que representan los cinco cereales más importantes que contienen gluten: trigo, centeno, cebada, avena y espelta.

## 4.4 Navegación y módulos

CeliFinder pretende ser una aplicación que destaca por su simplicidad por lo que los módulos principales de la aplicación se han reducido a los puntos clave de la misma. Son los siguientes:

- Configuración e información de la aplicación
- Búsqueda de restaurantes
- Resultado de la búsqueda
- Detalle de un establecimiento



Para movernos por estos módulos se utilizan los elementos de navegación. El más destacado es el menú lateral que emerge cuando se pulsa sobre un icono en la cabecera de la aplicación. En este menú aparecerán las opciones de configuración a las que el usuario puede acceder y le permite cambiar el idioma de la aplicación.

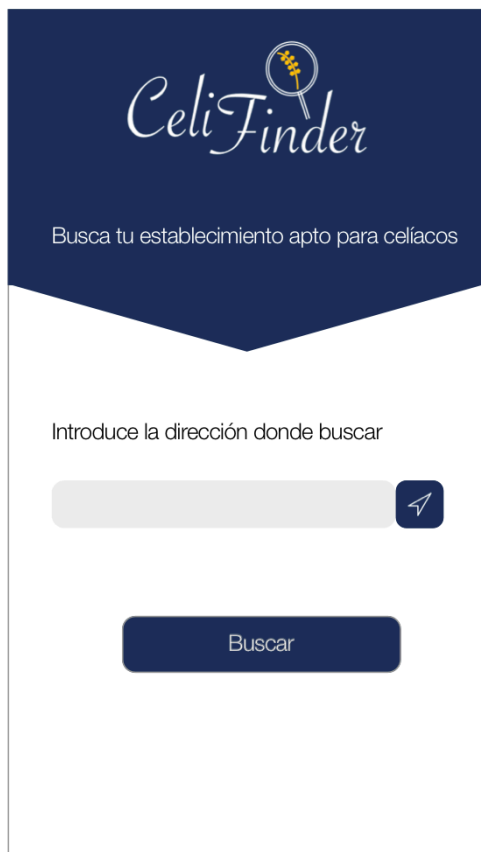
Los tres bloques principales se disponen de forma secuencial, de tal forma que la pantalla de bienvenida, permite al usuario hacer una primera búsqueda por ubicación, ya sea con su geolocalización o indicando una determinada dirección por texto. Una vez se hace esta búsqueda se pasa a la visualización de los resultados, en forma de mapa o de lista. Para volver a la pantalla anterior, se deberá pulsar sobre un icono de una flecha hacia la izquierda que se mostrará en la cabecera.

Al mostrar los resultados en forma de mapa, el usuario podrá clicar sobre los diferentes marcadores, de tal forma que cuando esto ocurre aparecerá un *tooltip* con el nombre del establecimiento y un icono de información. Al pulsar sobre este icono de información aparecerá la pantalla de detalle del establecimiento. Cuando los resultados se muestran en forma de lista, el usuario podrá pulsar sobre cada entrada, accediendo así al detalle del establecimiento.

## 4.5 Prototipo de alta fidelidad

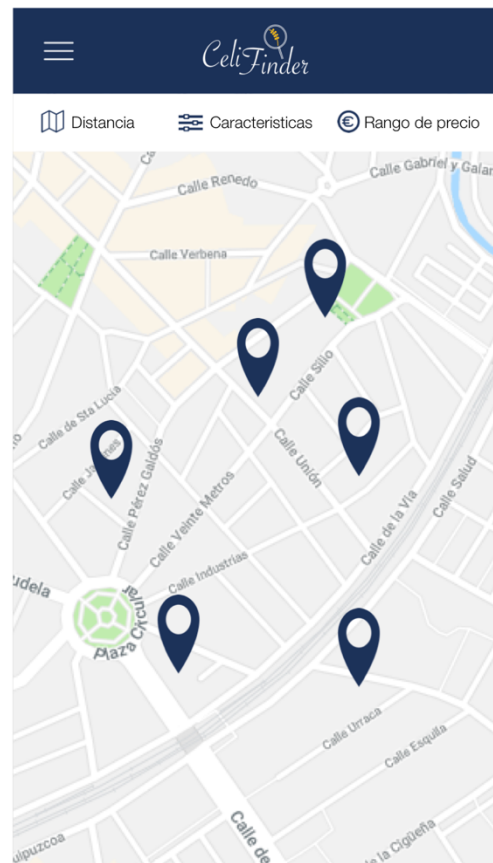
A continuación, se van a presentar los diseños de las pantallas de la aplicación en alta fidelidad. Para su composición se ha utilizado la herramienta de diseño Sketch.io

Pantalla de inicio

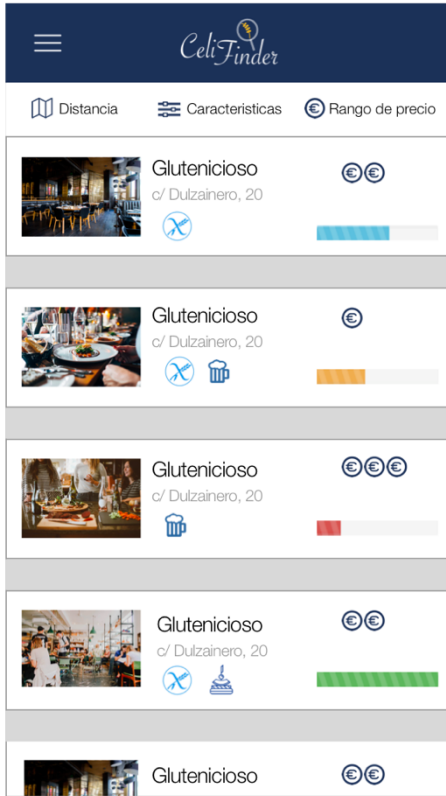


Resultados mostrados en forma de lista

Resultados mostrados en forma de mapa



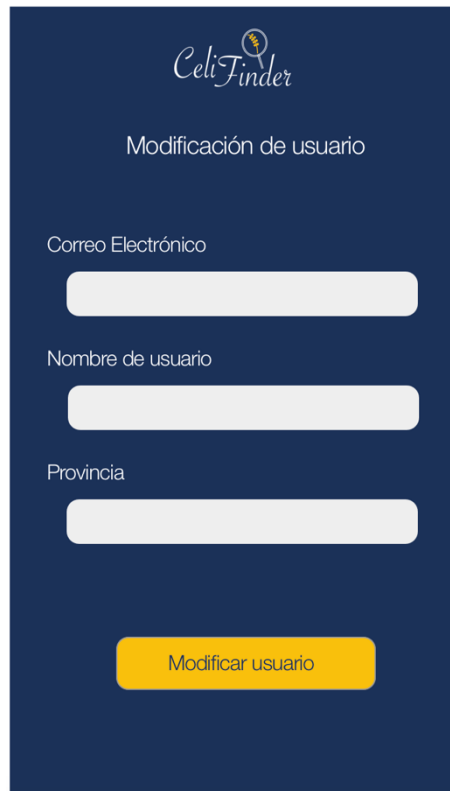
Detalle del establecimiento



Menú



Modificación / Detalle de usuario



## Cambiar de contraseña

CeliFinder

Cambio de contraseña

Contraseña antigua

Contraseña nueva

Repita la nueva contraseña

Modificar usuario

## Login

CeliFinder

Inicia sesión

Nombre de usuario

Contraseña

Iniciar sesión

¿No eres usuario? Regístrate

He olvidado mi contraseña

## Contacto

CeliFinder

Contacto

Correo Electrónico

Nombre y apellidos

Consulta

Enviar

## Ayuda

CeliFinder

Ayuda

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Distancia Características Rango de precio

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

## Política

Política

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

## Nuevo usuario

Nuevo usuario

Correo Electrónico

Nombre de usuario

Provincia

Nueva contraseña

Repita la nueva contraseña

Crear nuevo usuario

## 4.6 Presentación del producto

A continuación, se muestra la presentación de la aplicación como producto:



Ilustración 20. Presentación de la aplicación como producto

El objetivo de la presentación de la aplicación es dar al prototipo un aspecto más profesional y presentando la aplicación con un nivel de fidelidad alto respecto con el que debiera ser el resultado final. Con ello se pretende evaluar la validez de la aplicación para los usuarios objetivos de la misma.

La presentación debe ser lo suficientemente clara como para ser capaz de explicar el objetivo de la aplicación por si misma.

En este caso se presenta de forma principal el tema de la interfaz gráfica y el logo de la aplicación. Mediante la frase “Busca tu establecimiento apto para celíacos”, se especifica el objetivo principal de la aplicación.

También se añaden 3 pasos de uso de la aplicación que indican, por una parte, la simplicidad de la misma y, por otra parte, las claves de la aplicación que son la búsqueda por localización o dirección y la cantidad de información útil respecto a la oferta adaptada para celíacos que ofrece el establecimiento al usuario

## 5. Diseño Técnico

Una vez se ha confeccionado el diseño de la interfaz gráfica de la aplicación, se va a evaluar el diseño viendo los errores del mismo. Como el proceso de diseño es iterativo, en esta fase del proceso se evalúa todo lo realizado en la iteración presente. A continuación, se presenta la evaluación del prototipo, producto de la última iteración. En primer lugar, se presentará un diagrama UML que permitirá identificar los actores y el flujo de la aplicación. Basado en dicho diagrama, se presentará un listado de casos de uso explicando de forma clara las precondiciones, postcondiciones, flujos y actores que participan en cada caso de uso. En segundo lugar, se definirá la arquitectura del sistema, empezando por el diseño de la base de datos, especificando las entidades que la formarán, así como las relaciones entre ellas. Con esta información, se podrá definir las clases y objetos que se utilizarán para dar respuestas a las peticiones recibidas. Paralelamente se presentará el diseño técnico de la parte Frontend de la aplicación. Finalmente, se confeccionará un diagrama completo de la arquitectura del sistema.

### 5.1 Diagrama UML de casos de uso

El siguiente diagrama muestra las interacciones del usuario con el sistema, representando los diferentes casos de uso:

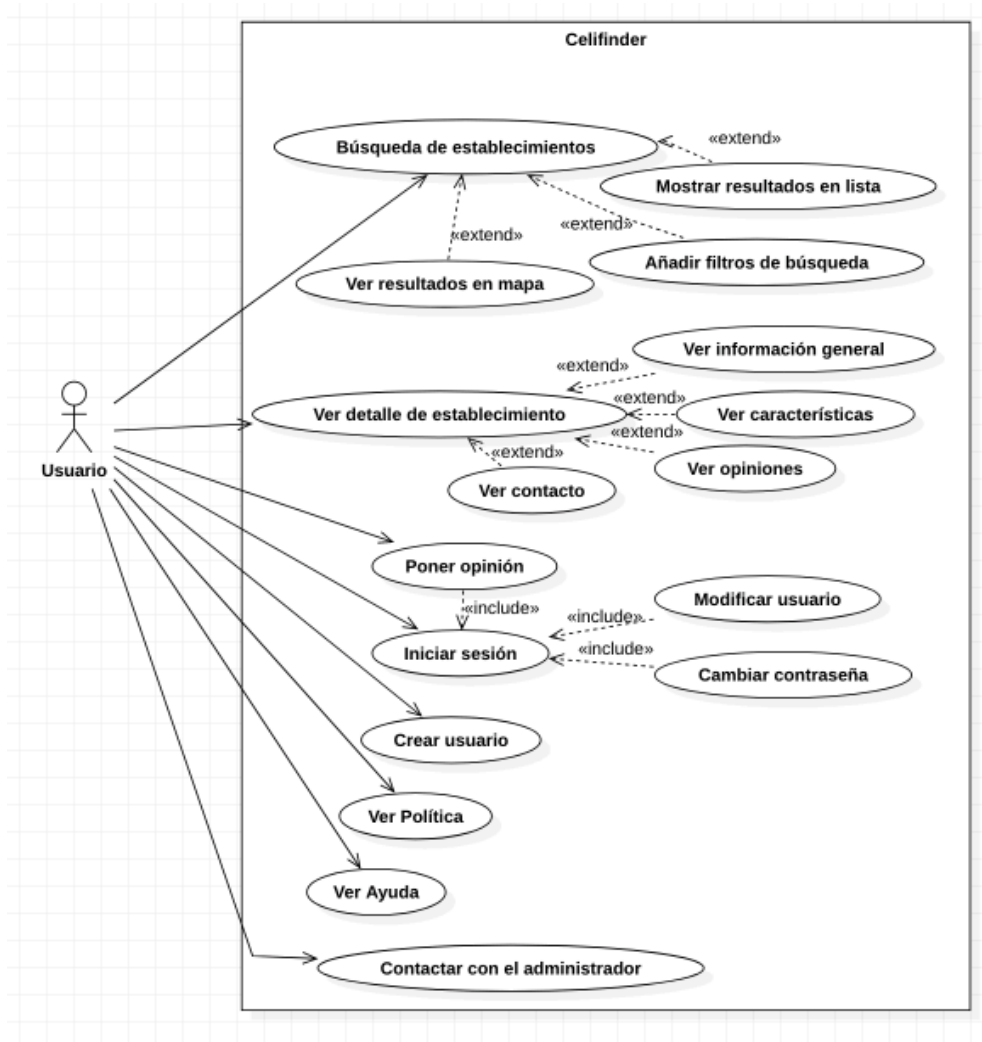


Ilustración 21. Diagrama UML de casos de uso

## 5.2 Listado de casos de uso

A partir del diagrama del apartado anterior se han identificado los casos de uso de la aplicación, por lo que a continuación se presentarán y analizarán cada uno de ellos individualmente.

Nombre	Búsqueda de establecimientos
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Encontrar restaurantes celiacos cercanos a la dirección indicada
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3b. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> </ol>
<b>Postcondiciones</b>	La búsqueda ha sido enviada al servidor.

Nombre	Ver resultados en mapa
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Visualizar los resultados de la búsqueda realizada en un mapa
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. Si la búsqueda se ha realizado por dirección escrita se mostrará una lista, el usuario pulsar sobre la opción para conmutar la vista de lista a la vista de mapa</li> </ol>

	5B. Si la búsqueda se ha realizado por localización, se muestran los resultados en la vista de mapa
<b>Postcondiciones</b>	-

<b>Nombre</b> Ver resultados en lista	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Visualizar los resultados de la búsqueda realizada en una lista
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. Si la búsqueda se ha realizado por dirección escrita se muestran los resultados en la vista de lista.</li> <li>5B. Si la búsqueda se ha realizado por localización, se muestran los resultados en la vista de mapa, el usuario pulsará sobre la opción para conmutar la vista de mapa a la vista de lista</li> </ol>
<b>Postcondiciones</b>	-

<b>Nombre</b> Añadir filtros de búsqueda	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Filtrar los resultados obtenidos según diferentes filtros
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> </ol>



	<ul style="list-style-type: none"> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5. El usuario abre el colapsable de los filtros de búsqueda para la categoría que prefiera.</li> <li>6. El usuario introduce los parámetros de los filtros que quiera aplicar</li> <li>7. El usuario pulsa sobre el botón de aplicar</li> </ul>
<b>Postcondiciones</b>	-

Nombre Ver detalle de establecimiento	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Obtener información concreta sobre un establecimiento
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente
<b>Escenario básico</b>	<ul style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. En la vista de lista el usuario selecciona una entrada de la lista</li> <li>5B. En la vista de mapa el usuario selecciona un marcador del mapa</li> <li>6B. El usuario pulsa sobre el icono de información del <i>Tooltip</i> que se ha desplegado al pulsar sobre el marcador del mapa</li> </ul>
<b>Postcondiciones</b>	-

Nombre Ver información general	
<b>Actores</b>	Usuario de la aplicación

<b>Objetivo</b>	Obtener información general del establecimiento, como tipo de establecimiento, comida que sirven, etc.
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente Haber seleccionado un establecimiento
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. En la vista de lista el usuario selecciona una entrada de la lista</li> <li>5B. En la vista de mapa el usuario selecciona un marcador del mapa</li> <li>6B. El usuario pulsa sobre el icono de información del <i>Tooltip</i> que se ha desplegado al pulsar sobre el marcador del mapa</li> <li>7a. El usuario pulsa sobre el <i>tab</i> de información (por defecto)</li> </ol>
<b>Postcondiciones</b>	-

Nombre	
Ver características	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Conocer las características concretas del establecimiento en relación a su oferta adaptada para celíacos
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente Haber seleccionado un establecimiento
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. En la vista de lista el usuario selecciona una entrada de la lista</li> </ol>

	<p>5B. En la vista de mapa el usuario selecciona un marcador del mapa</p> <p>6B. El usuario pulsa sobre el icono de información del <i>Tooltip</i> que se ha desplegado al pulsar sobre el marcador del mapa</p> <p>7a. El usuario pulsa sobre el <i>tab</i> de características</p>
<b>Postcondiciones</b>	-

<b>Nombre</b>		<b>Ver opiniones</b>
<b>Actores</b>		Usuario de la aplicación
<b>Objetivo</b>		Conocer las opiniones de otros usuarios sobre los diferentes establecimientos
<b>Precondiciones</b>		Haber realizado la búsqueda de establecimientos previamente Haber seleccionado un establecimiento
<b>Escenario básico</b>		<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. En la vista de lista el usuario selecciona una entrada de la lista</li> <li>5B. En la vista de mapa el usuario selecciona un marcador del mapa</li> <li>6B. El usuario pulsa sobre el icono de información del <i>Tooltip</i> que se ha desplegado al pulsar sobre el marcador del mapa</li> <li>7a. El usuario pulsa sobre el <i>tab</i> de opiniones</li> </ol>
<b>Postcondiciones</b>		-

<b>Nombre</b>		<b>Ver contacto</b>
<b>Actores</b>		Usuario de la aplicación

<b>Objetivo</b>	Dar información de contacto con el establecimiento, tales como teléfono, dirección o correo electrónico
<b>Precondiciones</b>	Haber realizado la búsqueda de establecimientos previamente Haber seleccionado un establecimiento
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. En la pantalla principal se pide al usuario indicar una dirección</li> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. En la vista de lista el usuario selecciona una entrada de la lista</li> <li>5B. En la vista de mapa el usuario selecciona un marcador del mapa</li> <li>6B. El usuario pulsa sobre el icono de información del <i>Tooltip</i> que se ha desplegado al pulsar sobre el marcador del mapa</li> <li>7. El usuario pulsa sobre el <i>tab</i> de contacto</li> </ol>
<b>Postcondiciones</b>	-

Nombre	
	<b>Iniciar sesión</b>
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Iniciar sesión en la cuenta personal del usuario
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Iniciar sesión”</li> <li>4. El usuario ingresa el usuario y la contraseña correspondiente</li> <li>5. El usuario pulsa sobre el botón de “Iniciar Sesión”</li> </ol>
<b>Postcondiciones</b>	El usuario se ha logado y podrá acceder a las funcionalidades de los usuarios registrados

<b>Nombre</b>		<b>Poner opinión</b>
<b>Actores</b>	Usuario de la aplicación	
<b>Objetivo</b>	Añadir un comentario de una experiencia u opinión de un establecimiento concreto	
<b>Precondiciones</b>	Haber iniciado sesión correctamente	
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>3. El usuario abre la aplicación</li> <li>4. En la pantalla principal se pide al usuario indicar una dirección <ol style="list-style-type: none"> <li>3A. El usuario escribe por teclado una dirección concreta</li> <li>3B. El usuario pulsa sobre el icono de localización para indicar su dirección actual como dirección en la que realizar la búsqueda</li> </ol> </li> <li>4. El usuario pulsa sobre el botón de búsqueda</li> <li>5A. En la vista de lista el usuario selecciona una entrada de la lista</li> <li>5B. En la vista de mapa el usuario selecciona un marcador del mapa</li> <li>6B. El usuario pulsa sobre el icono de información del <i>Tooltip</i> que se ha desplegado al pulsar sobre el marcador del mapa</li> <li>7. El usuario pulsa sobre el <i>tab</i> de opiniones</li> <li>8. El usuario pulsa sobre la opción de añadir comentario</li> <li>9. El usuario escribe el comentario y pulsa sobre el botón de “Añadir”</li> </ol>	
<b>Postcondiciones</b>	El nuevo comentario se ha añadido al establecimiento	

<b>Nombre</b>		<b>Modificar usuario</b>
<b>Actores</b>	Usuario de la aplicación	
<b>Objetivo</b>	Modificar los datos del usuario	
<b>Precondiciones</b>	Haber iniciado sesión correctamente	
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Modificar usuario”</li> <li>4. El usuario accede al formulario con sus datos y los modifica</li> <li>5. El usuario guarda los cambios pulsando sobre el botón “Guardar”</li> </ol>	

<b>Postcondiciones</b>	La información del usuario ha sido modificada

<b>Cambiar la contraseña</b>	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Modificar la contraseña del usuario registrado
<b>Precondiciones</b>	Haber iniciado sesión correctamente
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Cambiar contraseña”</li> <li>4. El usuario accede al formulario e introduce la contraseña anterior y la nueva contraseña, dos veces</li> <li>5. El usuario guarda los cambios pulsando sobre el botón “Guardar”</li> </ol>
<b>Postcondiciones</b>	La contraseña del usuario ha sido modificada

<b>Crear usuario</b>	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Registrarse como usuario en la aplicación
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Iniciar sesión”</li> <li>4. El usuario clica sobre la opción “¿No eres usuario? Regístrate”</li> <li>5. El usuario accede al formulario e introduce sus datos</li> <li>6. El usuario guarda la información del usuario pulsando sobre el botón “Guardar”</li> </ol>
<b>Postcondiciones</b>	Se crea un nuevo usuario con la información introducida en la aplicación.

<b>Nombre</b> Ver política	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Conocer las políticas y los avisos legales de la aplicación
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Política”</li> </ol>
<b>Postcondiciones</b>	-

<b>Nombre</b> Ver ayuda	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Conocer el funcionamiento de la aplicación
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Ayuda”</li> </ol>
<b>Postcondiciones</b>	-

<b>Nombre</b> Contactar con el administrador	
<b>Actores</b>	Usuario de la aplicación
<b>Objetivo</b>	Informar o sugerir al administrador sobre algún tema relacionado con la aplicación
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación</li> <li>2. El usuario abre el menú emergente lateral</li> <li>3. El usuario selecciona la opción “Contacto”</li> <li>4. El usuario accede al formulario de contacto e introduce los datos necesarios</li> </ol>

	5. El usuario envía el formulario al administrador, pulsando sobre el botón de “Guardar”
<b>Postcondiciones</b>	-

### 5.3 Diseño de la Base de Datos

Se ha elegido una base de datos relacional que se basa en la organización de la información en tablas relacionadas entre sí mediante identificadores. La decisión se ha basado en que los datos que vamos a utilizar van a ser validados por la aplicación y, por lo tanto, los datos serán consistentes y coherentes con el diseño de la base de datos.

Podemos distinguir tres tablas en la base de datos que se muestran en los siguientes diagramas:

Tabla restaurante:

restaurante		
PK	id_restaurante	INTEGER(11)
	nombre	VARCHAR(50)
	direccion	VARCHAR(255)
	informacion	VARCHAR(1023)
	telefono	VARCHAR(20)
	correo	VARCHAR(50)
	web	VARCHAR(255)
	fecha_alta	DATE
	fecha_actualizacion	DATE
	coordenada_x	DOUBLE
	coordenada_y	DOUBLE
	foto_portada	VARCHAR(255)
	foto_icono	VARCHAR(255)
	puntuacion	INTEGER(3)
	precio	INTEGER(4)
	certificado_face	BOOLEAN
	tapas	BOOLEAN
	cerveza	BOOLEAN
	porcentaje_gluten_free	INTEGER(3)
	restaurante	BOOLEAN

Tabla Usuario:

usuario		
PK	id_usuario	INTEGER(11)
	nombre_usuario	VARCHAR(30)
	correo	VARCHAR(50)
	provincia	VARCHAR(50)
	fecha_alta	DATE

Tabla Comentario:

comentario		
PK	id_comentario	INTEGER(11)
FK	id_usuario	INTEGER(11)
FK	id_restaurante	INTEGER(11)
	descripcion	VARCHAR(255)
	puntuacion	INTEGER(3)
	fecha_alta	DATE



A continuación, se muestran las relaciones entre las entidades:

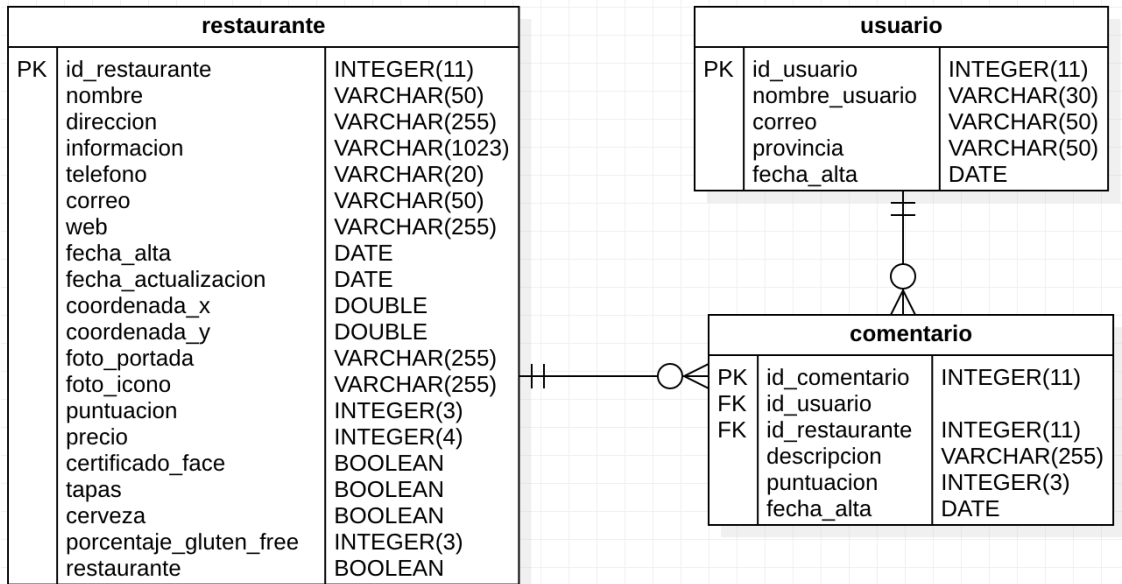


Ilustración 22. Diagrama de la base de datos

## 5.4 Diseño de clases

A continuación, se especifican las clases que se van a utilizar en la parte de backend de la aplicación, que utilizará Spring Boot para su implementación.

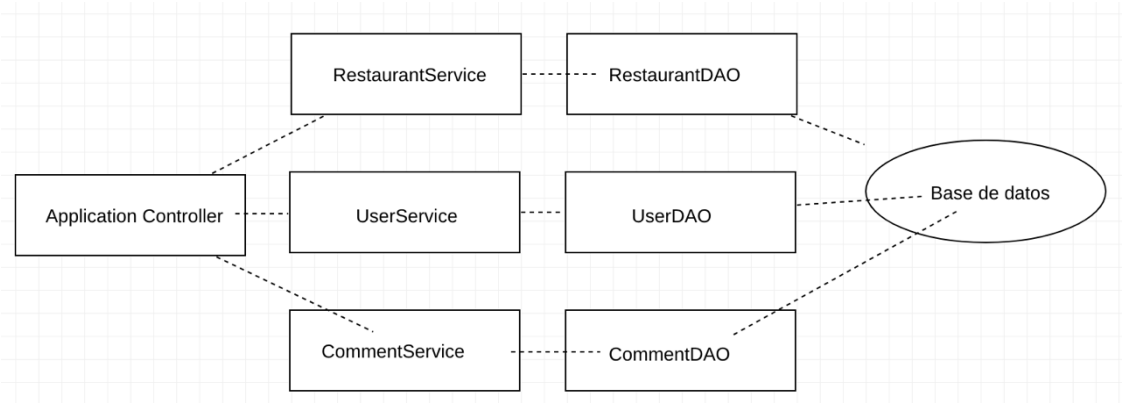


Ilustración 23. Diagrama de la relación entre clases

En primer lugar, el usuario a través de la aplicación va a acceder a pantallas que necesiten información. Esa información la pedirá la aplicación mediante REST API a la parte backend de la aplicación. La petición la recibirá el controlador de la aplicación y va a recoger los parámetros de la misma para, dependiendo de la información que pida, enviarla a un determinado servicio (servicio relacionado con la información del restaurante o con la información del usuario o con la información de los comentarios). Esos servicios llamarán al DAO correspondiente que se encargará de realizar una petición a la base de datos mediante una *query*, que nos devolverá una entidad ya definida.

Con todo ello el controlador va a devolver la información que el usuario ha pedido mediante la petición REST API.

Los servicios que se van a exponer son los siguientes

Método	Nombre
GET	restaurantesAll
GET	restaurantesByFilter
GET	comentariosByRestaurante
GET	restauranteById
POST	addComentario
PUT	updateComentario
DEL	deleteComentario
GET	usuarioById
PUT	updateUsuario
POST	addUsuario
GET	getAyuda
GET	getPolítica

## 5.5. Diagrama de la arquitectura de la aplicación

La arquitectura de la aplicación acoge distintas tecnologías que pretenden, en conjunto, optimizar el rendimiento y la experiencia de usuario de la aplicación. En la aplicación se va a distinguir notablemente la parte correspondiente al frontend y la parte correspondiente al backend.

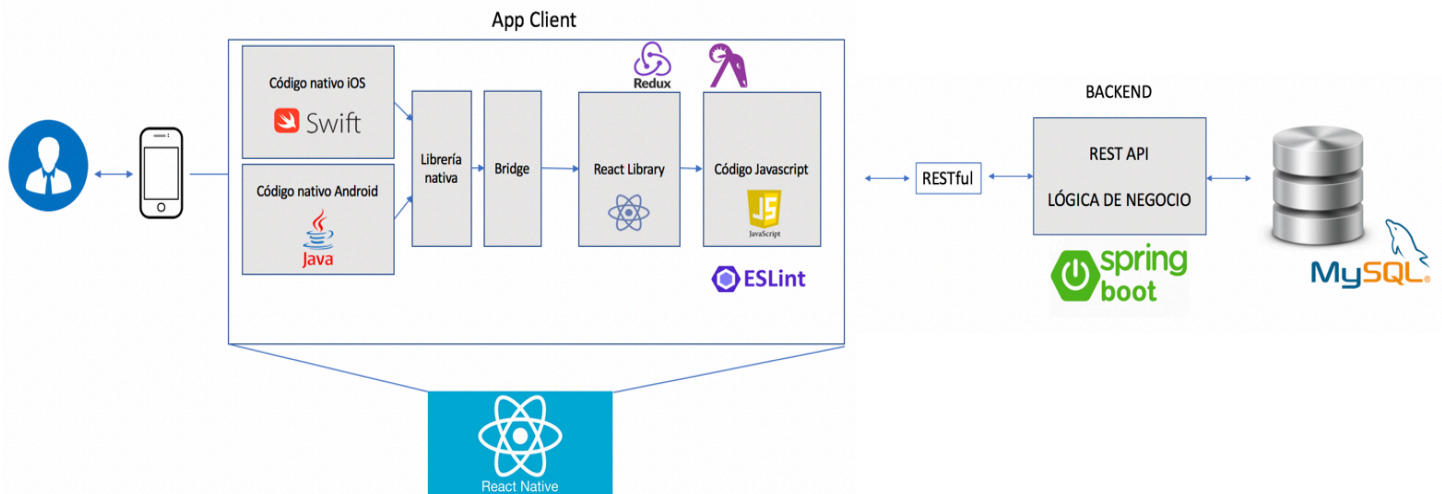


Ilustración 24. Diagrama de la arquitectura de la aplicación

Este diagrama muestra las principales tecnologías utilizadas, de tal forma que se puede afirmar que en la parte frontend la tecnología principal es React Native, mientras que en la parte backend las tecnologías principales son Spring Boot y MySQL como base de datos. Estas tecnologías se explicarán posteriormente.

## 6. Implementación Backend

En este capítulo se va a explicar las diferentes tecnologías, librerías y frameworks utilizados en la parte backend de la aplicación. Como se va a evidenciar a continuación, cada tecnología va a facilitar el desarrollo o a complementarlo añadiendo alguna *feature* concreta, formando así un proyecto a medidas según las necesidades especificadas en el diseño anterior.

### 6.1 Spring Boot



Ilustración 25. Logo de Spring Boot

(4) Spring Boot es un proyecto innovador de Pivotal, que consiste en un módulo del framework de Spring que proporciona una *feature* RAD (Rapid Application Development) a este framework.

La principal motivación de Spring Boot es simplificar el proceso de configuración y despliegue de aplicaciones Spring, permitiendo a los desarrolladores centrarse en la tarea de desarrollo y evitando configuraciones generales en las que hay que invertir mucho tiempo.

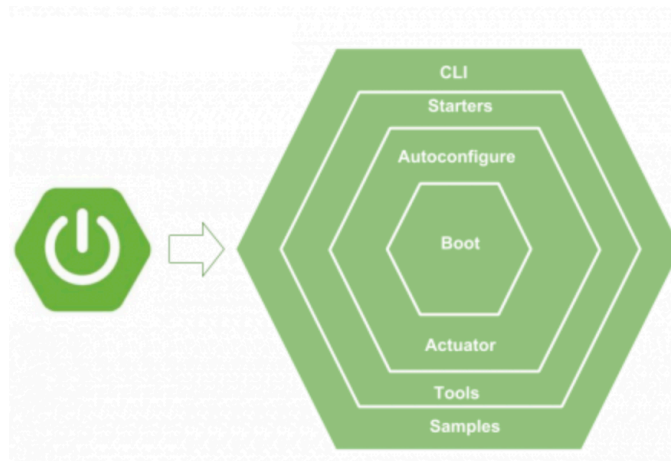


Ilustración 26. Esquema de funcionamiento de Spring Boot

De todo el desarrollo solo la parte del controlador es específica de la funcionalidad de Spring, el resto es un proceso genérico necesario en cualquier aplicación web desarrollada con Spring.

Sin embargo, Spring Boot no es ni un framework, ni un generador de código ni un servidor donde desplegar aplicaciones, sino una ayuda para construir empaquetar y desplegar aplicaciones que puede incluir un contenedor de servlets embebido como TomCat, donde podemos levantar nuestra aplicación.

Se basa en las librerías *starter*, con las que tienen una gran dependencia, ya que proporcionan una configuración inicial bastante completa pero difícil de personalizar.

En este proyecto se ha utilizado para desarrollar los servicios de Backend, de tal forma que el frontend llama a estos servicios y desde Spring se manejan los parámetros con los que se han llamado al servicio, así como el método utilizado para poder hacer una petición a la base de datos, procesarla y enviarla correctamente para que se pueda utilizar en la lógica de presentación de la aplicación. Se ha preferido utilizar Spring Boot por la gran cantidad de configuraciones por defecto que ayudan al desarrollo de aplicaciones backend poco complejas como esta, en vez de Spring framework en el que habría sido necesario un trabajo de arquitectura de la aplicación mayor.

## 6.2. API REST

REST significa Transferencia de estado representacional (*Representational State Transfer*) y actualmente es el estándar más lógico y habitual en la creación de APIs para el desarrollo de aplicaciones.

Se trata de una interfaz entre sistemas para obtener datos mediante HTTP o hacer operaciones sobre ellos en formatos tales como XML y JSON.

(5) Es un protocolo cliente/servidor sin estado, es decir, cuando se realizan peticiones HTTP, todos los datos necesarios están en esa petición de tal forma que ni el cliente ni el servidor necesita de datos sobre el estado anterior para poder satisfacer la petición. Por ello, REST nos ofrece una separación entre cliente y servidor, mejorando la reusabilidad de la interfaz en otras plataformas o aplicaciones.

Esta interfaz aplica acciones concretas para la obtención o manipulación de datos:

- GET: leer datos
- POST: crear datos
- PUT: modificar datos
- DELETE: borrar datos

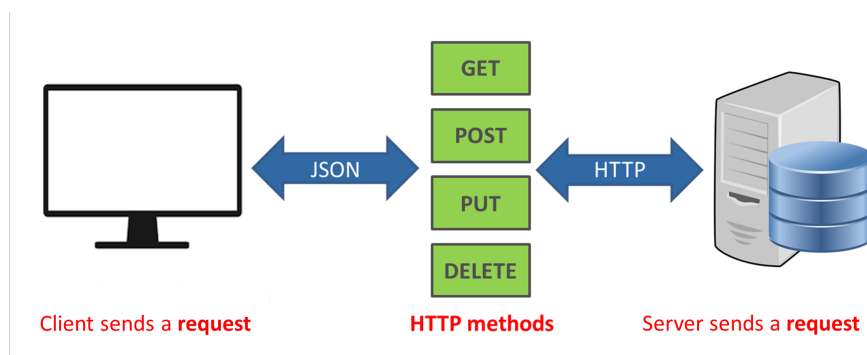


Ilustración 27. Esquema de funcionamiento de API REST

La separación entre cliente y servidor permite simplificar la tarea de escalar el producto o de migrarlo a otros servidores e incluso cambiar de base de datos si las peticiones se envían de forma correcta. Además, la API REST se adapta a la plataforma con la que se esté trabajando, pudiendo tener servidores de diferentes tipos como NodeJS, Python o Java. El formato de intercambio de datos siempre será JSON o XML.

Hoy en día, la gran mayoría de aplicaciones disponen de una API REST para crear servicios profesionales aprovechándose de todas las ventajas que se han descrito anteriormente.

En nuestra aplicación, API REST es la interfaz que se utiliza para intercambiar datos entre el servidor y el frontend de la aplicación.

## 6.3 MySQL

(6) MySQL es un tipo de base de datos relacional de código abierto respaldado por Oracle, que se ha convertido en uno de los más populares en la actualidad. Está basado en lenguaje estructurado de peticiones (SQL). Esta base de datos puede correr en distintos sistemas operativos tales como Windows, Linux o UNIX. Su aplicación principal se centra en el campo de las aplicaciones web, aunque puede ser utilizada en otros muchos.

MySQL está basado en el modelo cliente-servidor y maneja todas las instrucciones para bases de datos. Al ser una base de datos relacional, los elementos están organizados en una serie de tablas, previamente definidas y descritas por el usuario, especificando los campos, su obligatoriedad y el formato de los datos que pueden contener.

Estas tablas suelen definir una categoría como clave única, de tal forma que una fila en una de las tablas pueda hacer referencia a una fila de otra tabla a través de la *foreign key* o clave foránea procedente de la otra tabla. Con ello se pretende poder crear relaciones entre tablas de forma sencilla para el desarrollador.



Ilustración 28. Logo de MySQL

MySQL permite almacenar datos y acceder a ellos a través de diferentes mecanismos, que permiten otras operaciones tales como replicar datos, o dividir las tablas para mejorar su rendimiento.

En Celifinder, se ha utilizado como RDBMS (Relational Database Management System) de la aplicación, creando tres tablas de datos relacionadas entre sí. A pesar de tener otras opciones como PostgreSQL, MariaDB o DB2 entre otras, se eligió esta tecnología por su sencillez a la hora de crear las tablas y los registros, por su rapidez y por su alta disponibilidad en la mayoría de proveedores de hosting.

## 6.4 JPA

(7) JPA es la API de persistencia desarrollada para Java, concretamente la especificación JPA define la correlación de objetos internamente, en lugar de basarse en

implementaciones de correlación específicas del proveedor, representando una simplificación del modelo de persistencia.

Concretamente en la aplicación se ha utilizado el módulo JPA de Spring Data, que contiene un espacio de nombres que permite definir un repositorio de *beans*, así como otras *features* y características que facilitan el desarrollo de repositorios y su configuración.

Al crear repositorios JPA en la aplicación se nos proporciona una serie de métodos para realizar peticiones a la base de datos que nos permiten no tener que escribir los comandos de base de datos de forma manual y hacerlo mediante unos métodos de sintaxis sencilla. Además de estos métodos, es posible implementar *queries* personalizadas y adaptadas a las necesidades del proyecto. En este caso se han realizado ambas prácticas, es decir, se han utilizado tanto los *query methods* del repositorio JPA como las *queries* manuales.

## 6.5 Lombok

(8) Lombok es una librería que nos permite no repetir código y evitar fallos a la hora de escribir código que depende de otros datos y que se repite a lo largo de la aplicación.

Para utilizar esta librería solo es necesario añadirla a la lista de dependencias de compilación y las anotaciones serán procesadas, estando ya listas para su uso.

En el caso de esta aplicación se ha utilizado la anotación `@Getter` y `@Setter`. Estas anotaciones proporcionan una implementación de los métodos `get` y `set` de forma automática, de tal forma que, si varía algunos datos o características de la entidad en la que estamos trabajando, no es necesario preocuparse de modificar estos métodos para que se adapten a los cambios.

Esta librería también tiene otras anotaciones, con características similares a la explicada anteriormente, pero no ha sido necesario utilizarla en el desarrollo backend de la aplicación.

## 7. Implementación Frontend

A continuación, se van a explicar las librerías y tecnologías utilizadas en el desarrollo frontend de la aplicación. Sólo se describen las librerías principales, ya que se han utilizados algunas librerías de pequeño tamaño utilizadas para añadir ciertos componentes personalizados que no suponen una parte clave en la aplicación.

### 7.1 React-Native

(9) React-Native es una tecnología que nos permite construir aplicaciones nativas, tanto para iOS, como para Android desarrollando el proyecto con Javascript y con React.

React es una librería Javascript para el desarrollo de interfaces de usuario, que se centra en la componentización de las aplicaciones. Por ello los componentes son un concepto clave en esta librería, ya que permiten una mejor organización y reutilización del código. React fue creado por Facebook en el año 2013.

React-Native nos permite construir aplicaciones nativas, abstrayéndonos de la programación nativa, al desarrollar con Javascript, lenguaje utilizado para el desarrollo de aplicaciones web.

Su funcionamiento se basa en dos *threads* de programación. Por un lado, nos encontramos con el *thread* que corre en las aplicaciones nativas y que se encarga de procesar los gestos del usuario y de la presentación de elementos de la interfaz de usuario. Por otro lado, se añade un *thread* propio de React-Native, que se encarga de ejecutar el código en Javascript.

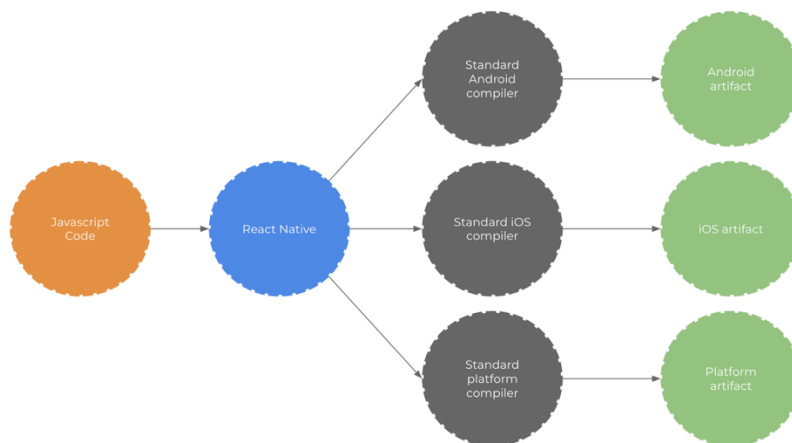


Ilustración 29. Arquitectura de las aplicaciones desarrolladas con React-Native

Ambos *threads* no interactúan nunca directamente, sino que lo hacen a través de un *bridge*, que es el *core* de React-Native. El *bridge* proporciona una comunicación bidireccional y asíncrona entre dos *threads* que están escritos en lenguajes totalmente diferentes. El *bridge* está desarrollado en C/C++ y tiene embebido el framework Javascript de Apple que permite acceder a través de una API a JavaScriptCore VM. De esta forma podemos ejecutar código JavaScript en un programa C/C++.

React-Native cuenta con una amplia variedad de componentes estilados que facilitan el desarrollo de la aplicación, aunque es necesario contar con librerías que complementen a React-Native para añadir características diferenciadoras a la aplicación

Es la base de la parte frontend de la aplicación y es necesario para poder, posteriormente ejecutarlo de forma nativa en dispositivos móviles. Se eligió esta tecnología ya que nos permite desarrollar dos aplicaciones nativas en dos plataformas diferentes (iOS y Android) a partir de un solo código en React, lo que supone un ahorro de tiempo y de recursos bastante importante.

## 7.2 Redux

(10) React es una librería, no un framework, y, por ello, es necesario complementarlo con ciertas librerías. Redux es una de estas librerías que nos permite gestionar el estado global de la aplicación de forma consistente. Redux está influenciado por la arquitectura Flux.

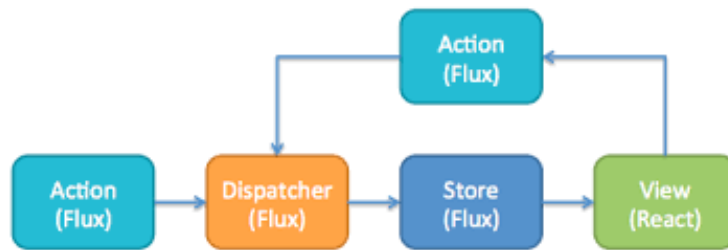


Ilustración 30. Arquitectura FLUX

El estado de la aplicación está contenido en un *store*, que es único. Esto constituye una gran ventaja ya que facilita la depuración de las aplicaciones. Este *store* es de sólo lectura, es decir, no se puede modificar, sino que se puede crear una copia de ese mismo *store*, pero realizando los cambios oportunos.

Para realizar estos cambios es necesario emitir una acción, que generará un *dispatch* de un tipo concreto. Los *reducers*, son funciones que reciben el estado actual del *store* y la acción que ha tenido lugar y devuelve una copia del estado actualizado del *store*.



Ilustración 31. Logo de Redux

En la aplicación usaremos Redux para almacenar datos globales de la aplicación que afectan a varios componentes y que, por lo tanto, no pueden almacenarse en el estado interno de los componentes. Un ejemplo de este tipo de datos es la información del usuario logado, una vez ha iniciado sesión.

Estos datos van a modificar la lógica de presentación, dependiendo de si el usuario ha iniciado sesión y tienen más permisos o si no lo ha hecho y se limita a realizar búsquedas y consultar información. Los datos obtenidos mediante llamadas API REST, también se almacenarán en los *stores* correspondientes.



## 7.3 NPM

(11) NPM es un gestor de paquetes para NodeJS, que comenzó como una forma de descargar y gestionar dependencias para proyectos de NodeJS, pero se ha convertido en una herramienta utilizada de forma muy mayoritaria en los proyectos Javascript.

NPM permite descargar y actualizar dependencias de librerías y gestionar las versiones de las mismas.



*Ilustración 32. Logo de NPM*

Además, NPM, también nos permite automatizar tareas, ya que también es una herramienta de *scripting*.

En Celifinder, se ha utilizado tanto como gestor de dependencias como se ha utilizado como herramienta de scripting para facilitar el arranque y automatización de tareas.

## 7.4 Ramda

(12) Ramda es una librería Javascript orientada a la programación funcional. Su forma de operar es más intuitiva y matemática que otras librerías, pero por el contrario su sintaxis es bastante compleja.

Se utiliza principalmente para realizar operaciones sobre objetos o arrays, añadiendo funcionalidades que los métodos nativos de Javascript no contemplan o no permiten hacerlo fácilmente.



*Ilustración 33. Logo de Ramda*

En la aplicación, se ha utilizado para manejar el estado global de la aplicación en los reducers. En definitiva, el store es un objeto que maneja información de diferentes aspectos como usuarios o restaurantes. Para realizar operaciones sobre él, es necesario modificar únicamente los datos que nos ocupan sin manipular el resto.

El método merge nos permite mezclar dos objetos sin modificar aquellas propiedades que no son necesarias.

## 7.5 ESLint

(13) ESLint es un proyecto de código abierto que consiste en un *linter* para proyectos de código Javascript.

ESLint nos permite especificar ciertas guías de estilos que se vayan a utilizar en el proyecto, de tal forma que garantizan la calidad del código, en ciertos aspectos. Existen ciertas guías de estilos creadas por compañías como Airbnb o Google, que proporcionan unas reglas básicas para el desarrollo de código.

Se pueden añadir reglas extra, personalizadas, tales como la indentación de las líneas en los ficheros, evitar los *console* o *debugger* o especificar un nivel máximo de complejidad o indexación.



*Ilustración 34. Logo de ESLint*

Se puede integrar ESLint con el IDE que se esté utilizando para desarrollar, de tal forma que al escribir código alerta al desarrollador que ha quebrantado alguna de las reglas especificadas en el fichero de configuración que tiene por nombre `.eslintrc`.

En este caso, se han utilizado las siguientes guías de estilo:

- ESLint: la guía más básica que proporciona ESLint
- React: la guía recomendada por React para el desarrollo de proyectos con esta librería

## 7.6 React-Navigation

(14) React Navigation es una librería que maneja la navegación entre pantallas en React Native. En las aplicaciones web, la navegación viene soportada por el navegador, pero en este caso, como las aplicaciones se van a ejecutar de forma nativa, no va a haber un navegador que soporte esta navegación.

Esta librería maneja la lógica de navegación entre pantallas, así como los parámetros que se pueden transferir de unas pantallas a otras.

Las vistas en React Native utilizan los componentes nativos y la librería *Animated* que añade animaciones a las transiciones entre pantallas que se ejecutan sobre el *thread* principal nativo.

En la aplicación, se utiliza para hacer transiciones hasta otras pantallas y para transferir parámetros de unas a otras.

## 7.7 React-Native-Elements

(15) React Native Elements es una librería de componentes estilados que complementa la que nos proporciona React Native por defecto.

La librería es transversal a varias plataformas, ya que obtenemos componentes diseñados de forma consistente tanto en Android como en iOS. Además, permite la personalización de estos componentes, aunque también proporciona un estilo por defecto en caso de que no queramos personalizar los componentes.

## 7.8 React-Native-Maps

(16) Librería que nos permite añadir mapas a las aplicaciones de React Native. Para ello, utiliza la API de Google Maps.

Nos proporciona el componente MapView, así como componentes complementarios como Marker (marcadores) o Callout (tooltip de los marcadores).

Al renderizar el MapView, podemos visualizar un mapa a pantalla completa en la aplicación y a través de la API de este componente podemos centrar el mapa, añadir marcadores o superponer capas en el mapa.

En nuestra aplicación, se utiliza al renderizar el mapa donde se localizan los restaurantes, así como en las acciones o eventos que se emiten al presionar sobre los marcadores.

## 8. Otras herramientas

A continuación, se van a describir las herramientas que se han utilizado para el desarrollo de este proyecto en todo aquello que no se refiere al desarrollo backend o frontend, incluyendo al diseño y planificación del mismo.

### 8.1 Visual Studio Code

(17) Visual Studio Code es un editor de código de Microsoft que se basa en Electron (framework para desarrollar aplicaciones de escritorio). Este editor incluye diferentes características o extensiones que hacen más fácil el desarrollo de código a los programadores y que se pueden descargar y habilitar o deshabilitar en función de las necesidades del programador.

Algunas de estas características son la gestión de versiones mediante integración con Git, sugerencias de autocompletado adaptadas a varios lenguajes de programación o frameworks o integración con ESLint para resaltar los errores del linter mientras se programa y corregirlos más fácilmente.

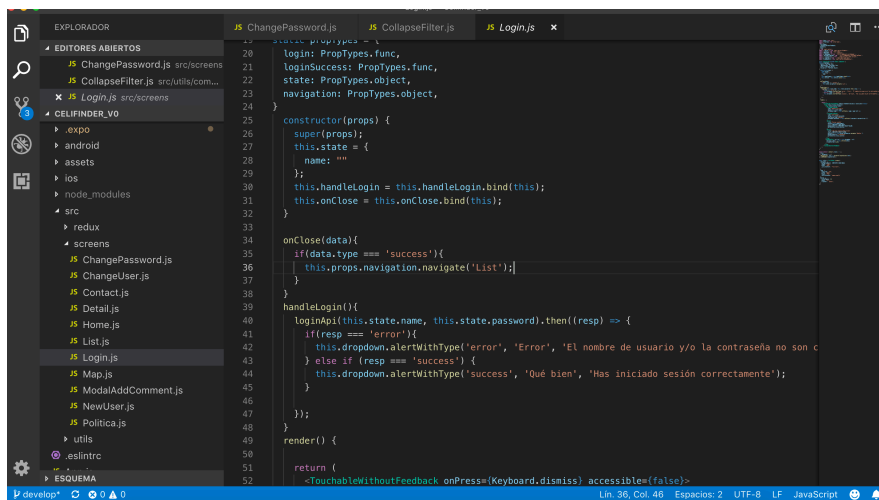


Ilustración 35. Pantalla de Visual Studio Code

En el proyecto se ha utilizado como editor de código para la parte frontend de la aplicación, utilizando las extensiones de ESLint, React Native Tools y React Native Snippet.

### 8.2 XCode

(18) XCode es el IDE (entorno de desarrollo integrado) para el desarrollo de las aplicaciones de Apple. XCode puede compilar código Objective C, Objective C++, C, C++ y Java.

Además, incluye un simulador de iOS que nos permite crear dispositivos virtuales de todo tipo, desde iPhone en sus diferentes versiones como iWatch o Apple TV y probar nuestras aplicaciones en ellos.

El IDE cuenta con diversas herramientas para desarrollar aplicaciones y configurar la interfaz gráfica de la aplicación, así como para depurarlas y hacer tests unitarios sobre el código desarrollado.



Ilustración 36. Logo de XCode

En la aplicación se ha utilizado para realizar configuraciones específicas de la plataforma iOS, tales como permisos para obtener la ubicación y para realizar peticiones HTTP a ciertos dominios. También ha generado el dispositivo virtual sobre el que se ha probado y creado la aplicación.

### 8.3 Expo

(19) Expo es un framework que nos facilita el desarrollo de aplicaciones nativas con React Native añadiendo una capa sobre las API de React Native. Expo utiliza Expo SDK una librería nativa y de Javascript que permite a la aplicación añadir funcionalidades nativas del dispositivo tales como el acceso a contactos o a almacenamiento local, sin necesidad de tener instalado XCode o AndroidStudio.

Expo SDK también nos permite gestionar otras funcionalidades complejas tales como las notificaciones Push o la construcción de ficheros binarios nativos para poder empaquetar la aplicación y posteriormente, subirlo a las diferentes stores.

### 8.4 Draw.io

Draw.io es una herramienta online para hacer diagramas como diagramas de flujo, control de procesos o esquemas de todo tipo. Está integrado con las herramientas de Google y se pueden almacenar los ficheros creados en Google Drive, así como exportarlos en otros formatos.

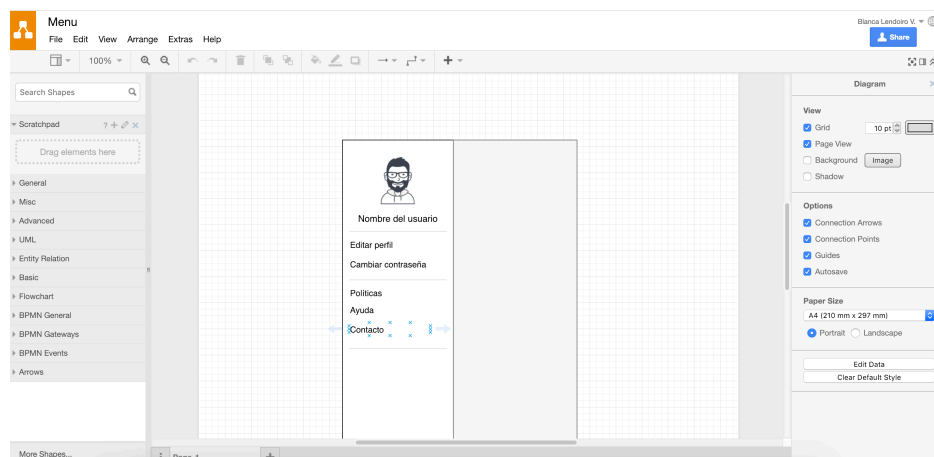


Ilustración 37. Pantalla de Draw.io

Draw.io proporciona una librería de formas que incluyen desde polígonos comunes como cuadrados círculos o rombos, hasta formas e iconos propios del sistema operativo iOS o formas propias de los diagramas UMLs.

En la aplicación se ha utilizado para hacer los bocetos o prototipos de baja fidelidad.

## 8.5 Sketch

(20) Sketch es un editor de gráficos vectoriales que permite agrupar diferentes textos, formas y assets para generar todo tipo de prototipos. Esta herramienta está formada por una barra de herramientas en la parte superior, que nos permite realizar acciones sobre las formas existentes (escalar, girar o transformar entre otras).



Ilustración 38. Logo de SketchApp

En este proyecto se ha utilizado para crear los prototipos de alta fidelidad, que pretenden ser imágenes que se parecen o se deben parecer lo máximo posible a la realidad en las diferentes pantallas por las que el usuario va navegando dentro de la aplicación.

## 8.6 StarUML

StarUML es una herramienta UML creada por MKLab. Soporta todos los diagramas especificados en UML 2.0, tales como diagramas de clases o diagramas de casos de uso entre otros.

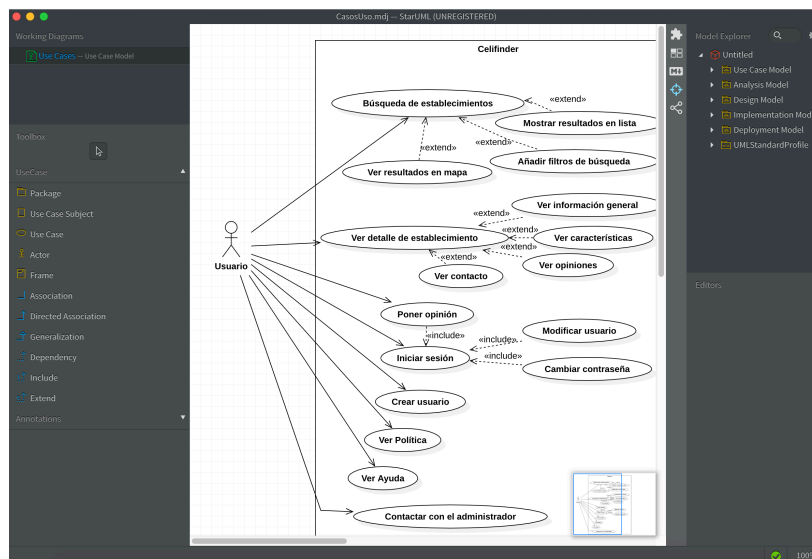


Ilustración 39. Pantalla de StarUML

En la aplicación se ha utilizado dentro del diseño técnico, para crear el diagrama UML de casos de uso, así como el diagrama de relación de tablas de la base de datos.

## 8.7 InkScape

(21) InkScape es un editor profesional de vectores gráficos libre y de código abierto. Con él es posible cambiar el tamaño, el color e incluso la forma de ciertos vectores gráficos. Es especialmente útil para personalizar iconos poniendo el color que la aplicación necesita dependiendo de su significado. Este es el uso que se le ha dado en el desarrollo de la aplicación.

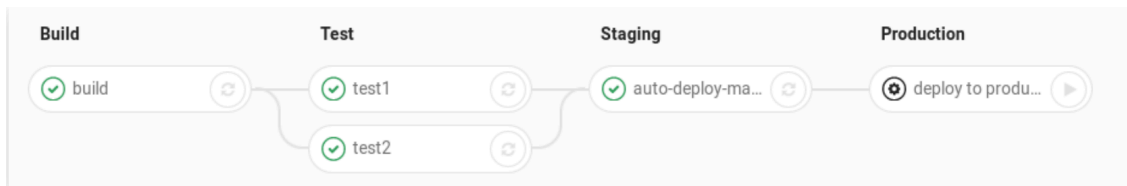


*Ilustración 40. Logo de InkScape*

## 8.8 GitLab

(22) GitLab es un servicio web de desarrollo de software y control de versiones basado en Git que nos permite gestionar repositorios, tanto públicos como privados.

Además, tiene varias herramientas de integración continua que nos permite realizar una serie de tareas que se ejecutan cuando se hacen commits al repositorio. Estas tareas se ejecutan en pipelines o conjuntos que tareas que se ejecutan en diferentes etapas.



*Ilustración 41. Ejemplo de pipelines en GitLab*

Con ello podemos asegurarnos que cada vez que subimos código, éste pasa una serie de controles, tales como que el proyecto compile correctamente o pase unos determinados tests.

GitLab se ha utilizado en este proyecto como repositorio web a través del cual se han manejado las versiones tanto del proyecto frontend de la aplicación como del proyecto backend de la misma, estando en diferentes repositorios para poder manejar las versiones de ambos de forma más clara e independiente.

## 8.9 SourceTree

(23) SourceTree es un cliente gratuito de Mercurial y de Git que ofrece una interfaz gráfica para poder gestionar las versiones y las ramas del repositorio de GitLab.

Su interfaz es bastante sencilla y amigable, permitiendo al desarrollador ver las modificaciones en los ficheros que han tenido lugar en cada commit, así como la rama en la que se ha realizado.

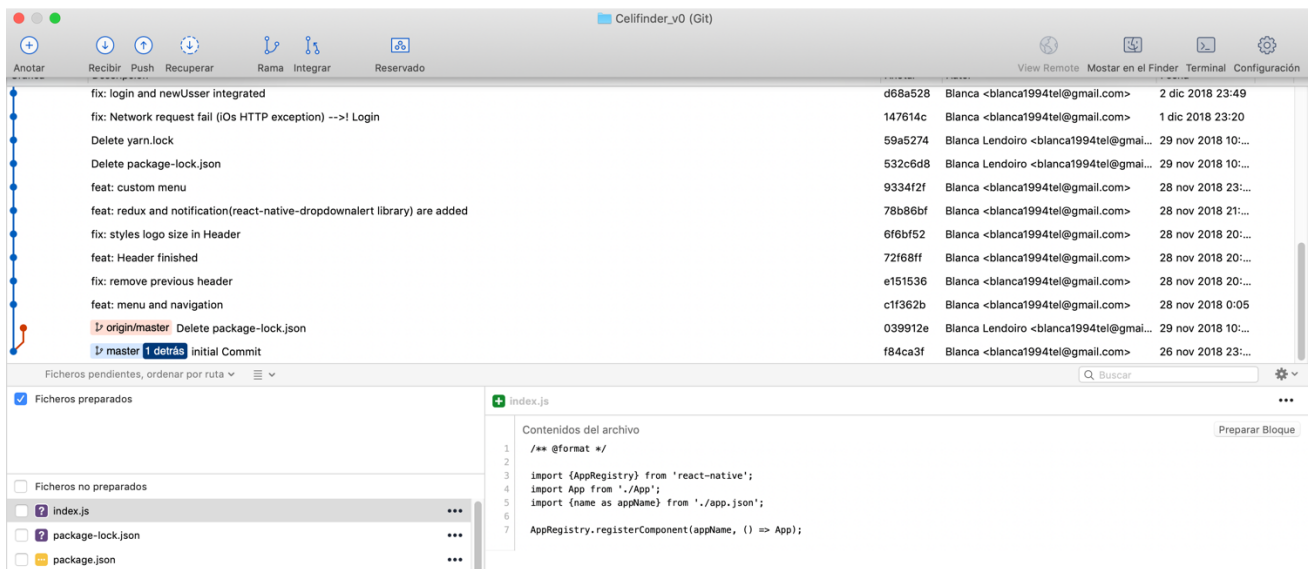


Ilustración 42. Pantalla de SourceTree

## 8.10 Trello

Trello es un software de gestión y administración de proyectos que consiste de tableros, que están formados por diferentes listas y en ellas se pueden añadir tarjetas y cambiarlas de lista dependiendo del estado en que se encuentren.



Ilustración 43. Logo de Trello

En este proyecto se ha utilizado como gestor de tareas al estilo ScrumBoard, es decir, hay 4 listas:

- Backlog
- Work In Progress
- Blocked
- Done

Cada tarjeta corresponde a una tarea concreta que se empieza colocando en la lista de BackLog, cuando se empieza a realizar, pasa a la lista de Work In Progress, si hay algún problema que impide que se realice completamente, pasará a la lista de Blocked. Cuando la tarea está completamente finalizada y subida al repositorio (tanto al local como al remoto) pasa a la lista de Done

## 8.11 Gantt Project

Software de código abierto escrito en Java que tiene como finalidad la gestión de proyectos a través de la creación y revisión del diagrama de Gantt.



## 8.12 STS

STS o Spring Tool Suite es un entorno de desarrollo basado en Eclipse que es específico para aplicaciones Spring.

Lleva preinstalado algunos de los plugins más comunes que nos permiten empezar a trabajar de forma más fácil evitando una serie de configuraciones por defecto de Maven, Git o Mylyn entre otros.

Permite generar proyectos Spring Boot mediante un wizard en el que el desarrollador especifica una serie de características propias de su proyecto y las configura automáticamente al generar el proyecto Spring Boot.

Cuenta con un Dashboard donde podemos gestionar las aplicaciones y su estado de ejecución actual.

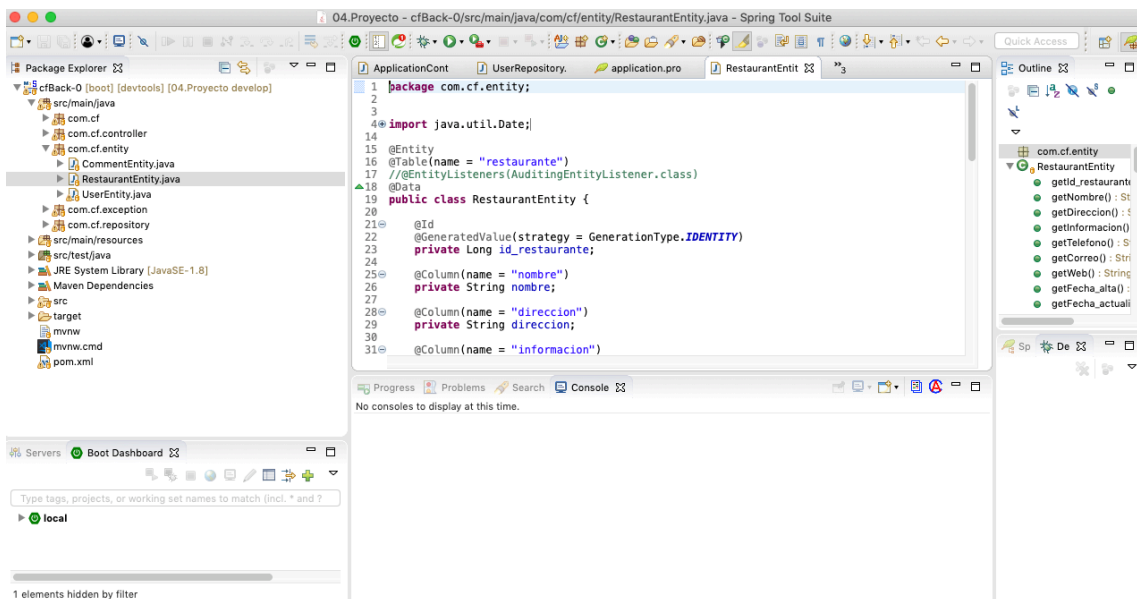


Ilustración 44. Pantalla del proyecto de backend en STS

## 8.13 Android Studio

(24) El software Android Studio es el entorno de desarrollo integrado (IDE) oficial que nos permite desarrollar aplicaciones para Android. Está basado en IntelliJ IDEA, un entorno de desarrollo integrado cuya finalidad es el desarrollo de programas informáticos, pertenece a la empresa JetBrains. Android Studio ofrece todas las herramientas de IntelliJ IDEA, un potente editor de códigos y otra multitud de utilidades que hace que este sea uno de los entornos más utilizados para el desarrollo de aplicaciones de Android.



Ilustración 45. Logo de Android Studio

En Celifinder se ha utilizado para generar los simuladores Android, además de para instalar ciertas librerías necesarias para la instalación de la aplicación o para modificar las versiones de ciertas librerías que no se habían actualizado en el core de React-Native.

## 9. Plan de pruebas

Acción	Prueba	Resultado
Home	Aparece la pantalla principal con un select y el botón de buscar por ciudad está deshabilitado puesto que no hay ninguna ciudad seleccionada	OK
Home	Cuando se selecciona una ciudad el botón se habilita	OK
Home	Al aparecer la pantalla principal el botón de buscar por localización si que está activo	OK
Home	Al clicar sobre la búsqueda de ciudad hace correctamente la búsqueda de establecimientos y los presenta en forma de lista	OK
Home	Al clicar sobre la búsqueda por localización hace correctamente la búsqueda de establecimientos y los presenta en forma de mapa	OK
Lista	Se presenta en la parte superior un colapsable cerrado y a continuación se presenta la lista de restaurante resultantes.	OK
Lista	En cada entrada se muestra el nombre del restaurante	OK
Lista	En cada entrada se muestra la dirección del restaurante	OK
Lista	En cada entrada se muestra el rango de precios del restaurante	OK
Lista	En cada entrada se muestra el porcentaje GF del restaurante	OK
Lista	En cada entrada se muestra los iconos con las características de ese restaurante	OK
Lista	En cada entrada se muestra la imagen de icono del restaurante	OK
Lista	Es posible filtrar por: <ul style="list-style-type: none"> <li>• Si tiene tapas</li> <li>• Si está certificado por la FACE</li> <li>• Si es un restaurante</li> <li>• Si tiene cerveza sin gluten</li> </ul>	OK
Lista	Es posible filtrar por rango de precio máximo	OK
Lista	Es posible filtrar por porcentaje GF mínimo	OK
Lista	En la parte superior izquierda aparece el icono del mapa y al presionarlo, el usuario navega al mapa	OK
Lista	Al presionar sobre una entrada de un restaurante, el usuario navegará a su detalle	OK
Mapa	Se presenta en la parte superior un colapsable cerrado y a continuación se presenta un mapa	OK
Mapa	Si hemos accedido al menú a través de la búsqueda por localización desde el botón de búsqueda por localización, el mapa se centra en la localización y busca a su alrededor	OK

Mapa	Si hemos accedido al menú por el icono de la cabecera de la lista, tras haber hecho la búsqueda por ciudad el mapa se centrará en el centro de esa ciudad	OK
Mapa	Si el mapa está en cualquier coordenada, pero se presiona sobre el botón de localización el mapa se centrará en el mapa y buscará a su alrededor restaurantes.	OK
Mapa	Los resultados de las búsquedas se verán como marcadores sobre el mapa.	OK
Mapa	Al presionar sobre cualquier marcador se abrirá un Tooltip con el nombre del restaurante	OK
Mapa	Si se presiona sobre el Tooltip que aparece con el nombre de cada restaurante se navegará al detalle del restaurante.	OK
Detalle	Al entrar sobre el detalle se muestra la foto de portada en la parte superior, el icono en un recuadro intermedio con el nombre y la dirección	OK
Detalle	Al entrar sobre el detalle aparecen 3 pestañas (información, opiniones y contacto) e inicialmente se enseñará la de información	OK
Detalle	Al pulsar sobre la pestaña de contacto se muestra los datos de contacto del restaurante	OK
Detalle	Al pulsar sobre la pestaña de opiniones se muestran las opiniones de otros usuarios.	OK
Detalle	Si el usuario presiona sobre el icono de añadir en la pestaña de opiniones aparecerá un modal y en el caso de no haber iniciado sesión le invita a navegar a la página de Login mediante un botón. Si el usuario presiona ese botón navegará a la pantalla de Login.	OK
Detalle	Si el usuario presiona sobre el icono de añadir en la pestaña de opiniones aparecerá un modal y en el caso de haber iniciado sesión, el usuario puede elegir una puntuación y un comentario	OK
Detalle	Si el usuario añade el comentario al volver a la pantalla de detalle verá su comentario añadido	OK
Detalle	Si un usuario ha iniciado sesión y ha añadido un comentario anteriormente en un restaurante y vuelve al detalle de ese restaurante, en la pestaña de opiniones, en la entrada de su comentario podrá ver un icono de una papelera. Al presionar sobre él, se borrará el comentario.	OK
Login	Se accede al Login a través del menú	OK
Login	Si el usuario y la contraseña son correctos aparece una notificación confirmando que se ha iniciado sesión y el usuario navega hasta la pantalla de lista. En el menú aparece en la parte superior el nombre de usuario y su inicial en el avatar	OK

Login	Si el usuario y la contraseña son incorrectos aparece una notificación avisando que no es correcto	OK
Login	Si el usuario no tiene cuenta y presionar sobre el enlace a nuevo usuario, navega hacia la pantalla de nuevo usuario	OK
Nuevo usuario	Si el usuario informa todos los campos correctamente y pulsa sobre crear usuario, éste se crea correctamente, se muestra una notificación y se navega a la pantalla de lista mostrando el nuevo usuario en el menú, estando ya logado	OK
Nuevo usuario	Si el usuario indica un correo electrónico erróneo, una notificación se lo indica.	OK
Nuevo usuario	Si el usuario indica la contraseña nueva diferente de cuando se le pide repetir la contraseña, una notificación se lo indica.	OK
Editar perfil	El usuario puede modificar su provincia, correo y nombre de usuario y tras pulsar el botón el usuario ve sus cambios reflejados en su perfil en el menú	OK
Cambiar contraseña	El usuario introduce la contraseña anterior correctamente y repite la nueva contraseña correctamente se realizará el cambio y una notificación informará de que el cambio se ha hecho correctamente	OK
Cambiar contraseña	Si el usuario introduce la contraseña anterior de forma errónea una notificación le informará de ello y no podrá modificar la contraseña	OK
Cambiar contraseña	Si el usuario no repite correctamente la nueva contraseña, una notificación le informará de ello y no podrá modificar la contraseña	OK
Política	Al pulsar la opción de Política en el menú, se mostrará la pantalla de política con la información en ella	OK
Ayuda	Al pulsar la opción de Política en el menú, se mostrará la pantalla de política con la ayuda en ella	OK
Contacto	Al pulsar la opción de Política en el menú, se mostrará un formulario con el correo electrónico, el nombre y el texto a enviar y al pulsar sobre enviar se redirige al cliente de correo electrónico por defecto del dispositivo	OK

## 10. Líneas futuras

La finalidad principal de este proyecto era el desarrollo de una aplicación que listase y localizase los establecimientos que tuvieran una oferta gastronómica o parte de ella, apta para personas con la enfermedad celiaca. A partir de esta versión de la aplicación se abren una serie de líneas de mejora e incluso de investigación que podrían dar más valor a la aplicación en un futuro.

La primera versión de Celifinder esta formada por una parte frontend desarrollada en React Native y una parte backend o de servidor desarrollada con Spring Boot, que tiene como finalidad recibir las peticiones que realiza la parte frontend, procesarlas, realizar *queries* a la base de datos MySQL en caso de que sea necesario y devolver los valores en formato JSON para que sean presentados correctamente.

El hecho de que se haya elegido React Native para realizar la aplicación no significa que, en un futuro, no pueda aparecer un framework de desarrollo de aplicaciones nativas a partir de código Javascript que sea mejor o es posible que se pueda optar por cambiar a otro framework ya existente como NativeScript. Por ello, una línea de mejora podría consistir en exponer los servicios de API Rest de una forma visual y sencilla de ver, con la finalidad de facilitar la modificación o migración de la aplicación en caso de que fuera necesario. Swagger es un software de código abierto que genera una interfaz visual en la que el desarrollador puede ver los servicios expuestos, los métodos que tiene cada uno de ellos, así como los parámetros o el *body* que necesita el servicio. Incluso permite hacer llamadas reales a los servicios añadiendo los datos o parámetros necesarios de la llamada a través de la interfaz y obteniendo la respuesta del servicio. Añadir Swagger y utilizarlo en el proyecto backend para facilitar la integración de frontend con backend, sería una mejora importante a nivel de integración de servicios.

Respecto a la interfaz visual, es decir, a la parte frontend, hay varios aspectos que podrían dar un valor añadido a la aplicación, mejorando la experiencia de usuario o ampliando el espectro de usuarios a los que se pretende llegar.

La aplicación actualmente solo se ofrece en castellano, pero está preparada para ofrecerse en más idiomas. Como puede observarse en la parte inferior del menú, se podrían añadir más iconos que indicasen que la aplicación se puede ver en otros idiomas. La internacionalización, podría hacerse tanto en la parte del servidor como en la parte del cliente, aunque, teniendo en cuenta que React Native cuenta con una librería propia para la internacionalización, ésta se podría utilizar sin necesidad de hacer llamadas al servidor para conseguir la mayoría de los literales específicos de cada idioma. Aunque la información de las bases de datos también tendría que estar internacionalizada, especialmente aquellas que contengan información o descripciones. Con esta mejora se podría intentar llegar al público no castellanoparlante y enfocarlo también hacia la industria del turismo, lo que sería un salto importante para el progreso de la aplicación.

En la sociedad actual, la importancia y la presencia de las redes sociales en nuestro día a día es indiscutible por lo que otra línea de mejora de la aplicación se centra en ellas. En primer lugar, las redes sociales podrían ser una fuente de publicidad para la aplicación, añadiendo opciones de compartir información de la aplicación a través de redes sociales como WhatsApp, Facebook o Twitter entre otras. De esta forma cuando un usuario

comparte información de un establecimiento que está contenido en Celifinder, también consigue hacer llegar el nombre de la aplicación a otras personas que pueden ser público objetivo de la aplicación, es decir, personas celíacas o su entorno. En segundo lugar, el uso de las redes sociales podría facilitar la funcionalidad del inicio de sesión en la aplicación. La integración del login con el inicio de sesión en redes como Facebook o Google podrían hacer el inicio de sesión más rápido para el usuario, fomentando así que los usuarios puedan iniciar sesión y por lo tanto poner opiniones y valoraciones sobre los establecimientos que contiene Celifinder. Como las opiniones es uno de los aspectos al que los usuarios dan más importancia, esta mejora podría aumentar el número de opiniones y, por lo tanto, el valor de la aplicación.

Respecto al aspecto de la experiencia de usuario, una de las mejoras a realizar sobre esta primera versión de la aplicación es añadir animaciones en la misma. Las animaciones, siempre que tengan sentido y sean útiles, dan al usuario un *feedback* de sus acciones, le guían en las interacciones con la aplicación y le dan una mejor impresión de la misma.

Otra línea de mejora a añadir a Celifinder, es una guía de *on-boarding*, en la que el usuario, la primera vez que accede a la aplicación es guiado por ella a través de una serie de interacciones dirigidas que le muestran las funcionalidades de la aplicación, así como el significado de algunos símbolos. En nuestro caso, por ejemplo, se podrían aclarar el significado de los iconos mostrados en cada entrada de la lista de establecimientos resultantes de la búsqueda. Esta guía, se podría saltar en el caso de que el usuario no quiera seguirla, evitando así molestar o crear un sentimiento negativo del usuario hacia la aplicación.

Finalmente, la aplicación para poder ser destacable en el mercado, necesita una mayor cantidad de datos y en más zonas geográficas. Actualmente sólo hay establecimientos en la ciudad de Valladolid, pero, como línea de investigación, es necesario obtener una gran cantidad de datos para poder ofrecer una mayor calidad de servicio al usuario y haciendo más útil a la aplicación para las personas con enfermedad celíacas y su entorno.

# 11. Conclusiones

El diseño, desarrollo y evaluación de una aplicación conlleva una serie de pasos concretos que nos permiten evaluar las ideas que han ido surgiendo a lo largo del proyecto, reflexionar sobre su sentido y su utilidad y ser capaz de desecharlas en caso de no encontrarlo.

En primer lugar, para desarrollar esta aplicación, se realizó un estudio del contexto y de la justificación del proyecto con la finalidad de investigar la situación actual de las aplicaciones y webs que tienen como objetivo aconsejar o recomendar establecimientos a personas con intolerancias alimenticias de diferentes tipos. Con esta investigación, se llegó a la conclusión de que, a pesar de existir algunos sitios de este tipo, tenían inconvenientes de diferentes tipos, tales como falta de fiabilidad o poca concreción a la hora de especificar la oferta apta para la intolerancia al gluten, entre otros.

A partir de este contexto, se pudieron definir concretamente los objetivos del proyecto, siendo los más importantes permitir al usuario buscar establecimientos adaptados a una dieta sin gluten, aportando la información necesaria para que el usuario sepa qué va a poder comer en ellos, usar la geolocalización y la visualización en mapas para poder situarlos fácilmente, añadir comentarios y puntuaciones por parte de los usuarios y, por último, identificar los establecimientos visualmente mediante iconos y colores especificando características de los establecimientos respecto a la oferta apta para celíacos.

Estos objetivos principales son los que han marcado la hoja de ruta de nuestra aplicación. Al tener estos objetivos siempre presentes, se han convertido en una herramienta de priorización de tareas, ya que, a la hora de ordenar las funcionalidades a implementar, solo había que preguntarse “¿En qué medida me acerca esta funcionalidad a mis objetivos principales?”

Al comienzo de este proyecto se realizó una encuesta a un grupo de personas que formaban parte de los usuarios objetivos de la aplicación, es decir, personas con enfermedad celíaca y su entorno. A través de estas respuestas, se pudo dar forma a los objetivos y elegir un método de para implementar la aplicación, escogiendo la aplicación móvil en diferentes plataformas, es decir, tanto en iOS como en Android.

El diseño de esta aplicación es un tipo de diseño centrado en el usuario y en sus necesidades, por lo que fue muy importante centrarnos en los tipos de usuarios a los que iba dirigida la aplicación. Para ello se siguió el método llamado ‘Personas’ que consiste en describir a una persona con imagen, nombre y apellidos como usuario de la aplicación. Esta técnica ha sido muy útil ya que permitía discernir si una funcionalidad era necesaria o no, para ello, de la misma forma que se había hecho anteriormente con los objetivos, es necesario preguntarnos si la persona (con nombres y apellidos) iba a dar valor a esa funcionalidad de la aplicación. Si es así, la funcionalidad sí que tenía sentido, sino no era necesaria y se podría volver a evaluar en versiones posteriores. Al principio del diseño de una aplicación, una gran cantidad de ideas, funcionalidades y características llamativas surgen, pero es necesario hacer uso de estas técnicas para ser conscientes de lo que realmente necesitamos en la aplicación y lo que nos sobra o es secundario en ella.



Una vez se definieron los usuarios, hay que pensar en cómo y cuándo va a utilizar la aplicación el usuario ya que no se puede diseñar una aplicación que va a ser utilizada mientras la persona que lo este usando esta sentada en una silla de la misma forma que una aplicación que va a ser utilizada por una persona que está en la calle en movimiento. Gran parte de la experiencia de usuario que ofrezca la aplicación va a venir dada por cómo el diseño de la aplicación ha tenido en cuenta de qué forma se va a utilizar esta. En este caso se tiene en cuenta que se puede utilizar tanto cuando el usuario está en el interior, de forma cómoda buscando establecimientos para ir en un futuro, ya sea cercano o lejano o se puede utilizar en el exterior mientras el usuario está en movimiento buscando un establecimiento para ir en el momento. A la hora de tomar decisiones, se ha utilizado el contexto más restrictivo, de tal forma que, por ejemplo, al decidir los colores de la aplicación se ha tenido en cuenta el contraste mínimo necesario para que un usuario pueda verlo cómodamente en la calle, aún sabiendo que si lo consulta en el interior lo podrá ver bien y de forma amigable.

Para la definición de los requisitos, tanto los funcionales como los no funcionales, se definieron unos escenarios de uso concretos en los que los usuarios, personificados en las fichas de usuario, usaban la aplicación. Con ellos, se concretaban situaciones en las que el usuario iba necesitar diferentes funcionalidades de la aplicación. Esta técnica me ha permitido saber qué funcionalidades son claves en la aplicación, ya que, con ella, se ve claramente cómo de frecuentes van a ser unos escenarios de uso de la aplicación en comparación con otros muy poco frecuentes.

Además de los escenarios de uso, también se han creado mapas de experiencia del usuario basados en los diferentes contextos de uso de la aplicación. Para cada uno de ellos se han analizado los sentimientos y la actitud del usuario ante ciertas interacciones y funcionalidades de la aplicación, lo que ha sido de gran utilidad para definir algunas interfaces potenciando aquellos puntos en los que el usuario se siente bien y tiene sentimientos positivos y reduciendo al máximo o evitando en la medida de lo posible, los momentos en los que el usuario está incómodo o tiene una actitud negativa ante las interacciones que lleva a cabo con la interacción. He podido comprobar la importancia que tienen estos pasos previos a la implementación, referentes al análisis del comportamiento del usuario, respecto a la aplicación, ya que se reflejan en gran medida en los resultados obtenidos.

A continuación, se realizó el diseño gráfico de la aplicación, que comenzó por un sitemap en el que se definía gráficamente a través de un diagrama, la navegación entre pantallas de la aplicación y los casos en los que el usuario podía acceder a ciertas pantallas. Por ejemplo, al intentar añadir un comentario, si el usuario está logado, puede añadirlo, pero sino se le invitará a iniciar sesión en la pantalla del login. Todo ello ha permitido organizar las pantallas en la implementación de la aplicación y tener claro cómo se podía llegar a cada una de ellas, evitando dejar al usuario bloqueado en una pantalla sin poder acceder al resto de pantallas de la aplicación.

Con toda la información recogida y descrita aquí anteriormente se realizaron dos prototipos, uno de baja fidelidad y esquemático en el que se situaron los diferentes componentes de la aplicación, y se hicieron un boceto de cual podría ser su apariencia sin entrar en detalles y, más tarde uno en alta fidelidad generando pantallas a las que se tenían que parecer las pantallas en la realidad. Entre ambos prototipos se realizó un Moodboard en el que se eligieron los colores, las fuentes y los iconos a utilizar en la aplicación,

teniendo en cuenta los contextos de uso en los que se iba a utilizar la aplicación. Para finalizar el diseño gráfico se creó una presentación del producto en la que se presenta la aplicación como un producto a vender al usuario. Esta es una técnica que me ha permitido evaluar cómo quedaban los colores y reflexionar sobre cómo de acertadas habían sido las decisiones tomadas en cuanto a diseño gráfico.

Una vez se definió el prototipo de la aplicación, se realizó el diseño técnico de la misma. Para ello me apoyé en el diagrama UML de casos de uso que, esquemáticamente, me permitió ver los casos de uso existentes y de qué forma debía diseñar el sistema para poder alcanzar los objetivos fijados para la aplicación de la forma más directa. En primer lugar, se describió la base de datos, con los modelos de datos y las tablas correspondientes, en segundo lugar, un diagrama de clases donde se especificaba la estructura del proyecto de backend de la aplicación, para finalizar definiendo, mediante un diagrama, la arquitectura de la aplicación, desde la parte frontend hasta la aplicación, especificando las tecnologías a utilizar.

Durante la implementación me he dado cuenta de la gran cantidad de herramientas tecnológicas que tenemos a nuestro alcance en todos los *stacks* de las aplicaciones tanto en frontend (React-Native, Redux, Ramda, Npm o Eslint entre otros), en backend (Spring Boot, MySQL, API Rest, JPA, Lombok, STS) y en otras herramientas transversales como SourceTree, GitLab, StarUML, Sketch, Trello o Expo entre otras. Las principales tecnologías, frameworks, librerías y herramientas que se han utilizado en este proyecto se describen en los capítulos 6, 7 y 8 de esta memoria.

Este proyecto me ha permitido lidiar con todas estas tecnologías, frameworks, librerías y herramientas y descubrir que, para conocerlas realmente, hay que utilizarlas en proyectos reales pudiendo verlas, investigar sobre ellas y aprovechar todo su potencial, sabiendo la gran cantidad de oportunidades que tenemos para crear aplicaciones de todo tipo con ellas. También he podido vivir la importancia que tiene en una aplicación un buen trabajo de análisis y diseño de la aplicación, que muchas veces no se le da el valor que tiene a esta parte de la creación de las aplicaciones.

## 12. Bibliografía.

0. **Dubberly Design Office.** Alan Cooper and the Goal Directed Design Process. [En línea] <http://www.dubberly.com/articles/alan-cooper-and-the-goal-directed-design-process.html>.
1. **Interaction Design Foundation.** The Glossary of Human Computer Interaction. [En línea] <https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/context-of-use>.
2. **Nagel, Wolfram.** *Multiscreen UX Design: Developing for a Multitude of Devices* . s.l. : Morgan Kaufmann, 2015.
3. **W3C (The World Wide Web Consortium).** Techniques For Accessibility Evaluation And Repair Tools. [En línea] <https://www.w3.org/TR/AERT/#color-contrast>.
4. **Spring by Pivotal.** Spring Boot. [En línea] <http://spring.io/projects/spring-boot>.
5. **BBVAOPEN4U.** API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. [. [En línea] Marzo de 2016. <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos..>
6. **MySQL.** MySQL Database. [En línea] <https://www.mysql.com/>.
7. **Spring by Pivotal.** *Spring Data JPA*. [En línea] <http://spring.io/projects/spring-data-jpa>.
8. **Lombok.** *Project Lombok*. [En línea] <https://projectlombok.org/>.
9. **Facebook.** *React Native · A framework for building native apps using React*. [En línea] <https://facebook.github.io/react-native/>.
10. **Redux.** *Getting Started with Redux*. [En línea] <https://es.redux.js.org/>.
11. **NPM.** *NPM - Build amazing things*. [En línea] <https://www.npmjs.com/>.
12. **Ramda.** *Ramda Documentation*. [En línea] <https://ramdajs.com/>.
13. **ESLint.** *ESLint - Pluggable JavaScript linter*. [En línea] <https://eslint.org/>.
14. **React Navigation.** *React Navigation · Routing and navigation for your React Native apps*. [En línea] <https://reactnavigation.org/>.
15. **React Native Elements.** *React Native Elements · Cross Platform React Native UI Toolkit*. [En línea] <https://react-native-training.github.io/react-native-elements/>.
16. **React Native Maps.** *React Native Mapview component for iOS + Android*. [En línea] <https://github.com/react-native-community/react-native-maps>.

17. **Visual Studio Code.** *Visual Studio Code - Code Editing. Redefined.* [En línea] <https://code.visualstudio.com/>.
18. **Apple.** *Xcode - Apple Developer.* [En línea] <https://developer.apple.com/xcode/>.
19. **Expo.** *Expo.io - The fastest way to build an app.* [En línea] <https://expo.io/>.
20. **Skecth.** *Sketch - The digital design toolkit.* [En línea] <https://www.sketchapp.com/> .
21. **InkScape.** *Inkscape: Draw Freely.* [En línea] <https://inkscape.org/es/>.
22. **GitLab.** *The first single application for the entire DevOps lifecycle.* [En línea] <https://gitlab.com/> .
23. **SourceTree.** *Sourcetree | Free Git GUI for Mac and Windows.* [En línea] <https://www.sourcetreeapp.com/> .
24. **Android.** *Conoce Android Studio | Android Developers.* [En línea] <https://developer.android.com/studio/intro/?hl=es-419>.

## 13. Anexo I. Manual de usuario

Al acceder a la aplicación, ésta nos mostrará una pantalla de bienvenida en la que el usuario tendrá dos opciones: por un lado, realizar una búsqueda por ciudad en la que se selecciona una ciudad en un combo y, por otro lado, realizar una búsqueda por localización tal y como se ve en [inicio (a) y Pantalla de lista]. Al realizar una búsqueda por ciudad se realizará una búsqueda de establecimientos alrededor del centro de la ciudad seleccionada y se mostrarán en forma de lista



Ilustración 46. Pantalla de inicio (a) y Pantalla de lista(b)

En la lista se podrá ver, para cada restaurante, el nombre, su dirección, así como las siguientes características:

- Tapas: si el establecimiento ofrece tapas en barra
- Certificado FACE: si el establecimiento está certificado por la FACE (Federación de Asociaciones de Celíacos de España)
- Cerveza: si tienen cerveza sin gluten
- Restaurante: si el establecimiento sirve comidas y cenas
- Rango de precio: clasificación por precio en 3 categorías
- Porcentaje GF (Gluten Free): cantidad relativa aproximada (en porcentaje) de platos aptos para celíacos con respecto al total de platos en el menú.

Si el usuario presiona sobre alguna de las entradas de la lista, navegará hacia el detalle de dicho establecimiento.

Cuando el usuario realiza una búsqueda por localización, los resultados se mostrarán en forma de mapa, centrado en la localización del usuario y realizando la búsqueda a su alrededor.

Los resultados se muestran en forma de chinchetas en el mapa. Cuando el usuario pulsa sobre alguna de ellas se mostrará el nombre del establecimiento en un Tooltip. Si el usuario presiona sobre el Tooltip, accederá al detalle del establecimiento.

Sobre el mapa habrá un botón en la esquina inferior izquierda con el icono de localización. Cuando el usuario presiona sobre él, el mapa se centra en el punto donde se localiza la posición del usuario.

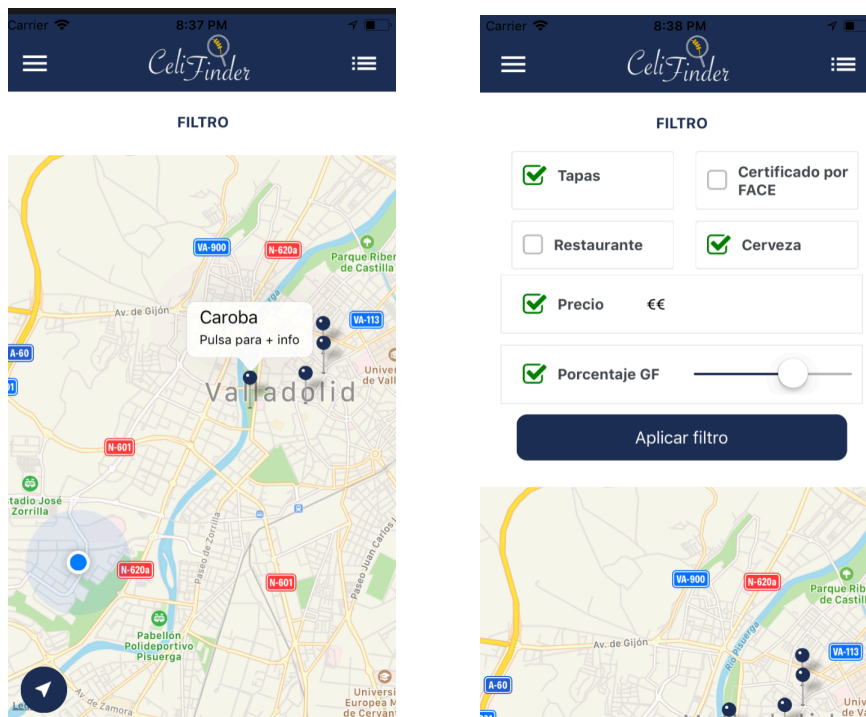


Ilustración 47. Vista de mapa (a) y Filtro (b)

Tanto en la vista de resultados del mapa como en la de la lista existe un filtro fijo en la parte superior, bajo la cabecera. Al desplegarlo aparecen una serie de características de los establecimientos por los que se puede filtrar la búsqueda.

Para que los filtros se apliquen, el usuario debe marcar el ‘check’ de cada criterio y, si es necesario seleccionar un valor, como en el caso del precio y del porcentaje GF. Posteriormente debe pulsar sobre el botón de ‘Aplicar filtro’.

En la pantalla de mapa, aparece un icono de lista en la parte derecha de la cabecera, para poder acceder a los resultados presentados en forma de lista. En la pantalla de lista, aparece un icono de mapa en la parte derecha de la cabecera, para poder navegar también a los resultados desde el mapa, centrado en el punto central de la ciudad.

Al acceder al detalle podremos ver la imagen de portada y el icono del establecimiento, así como su nombre y su dirección. En las pestañas inferiores se podrán ver tres pestañas: la de información, la de contacto y la de opiniones.

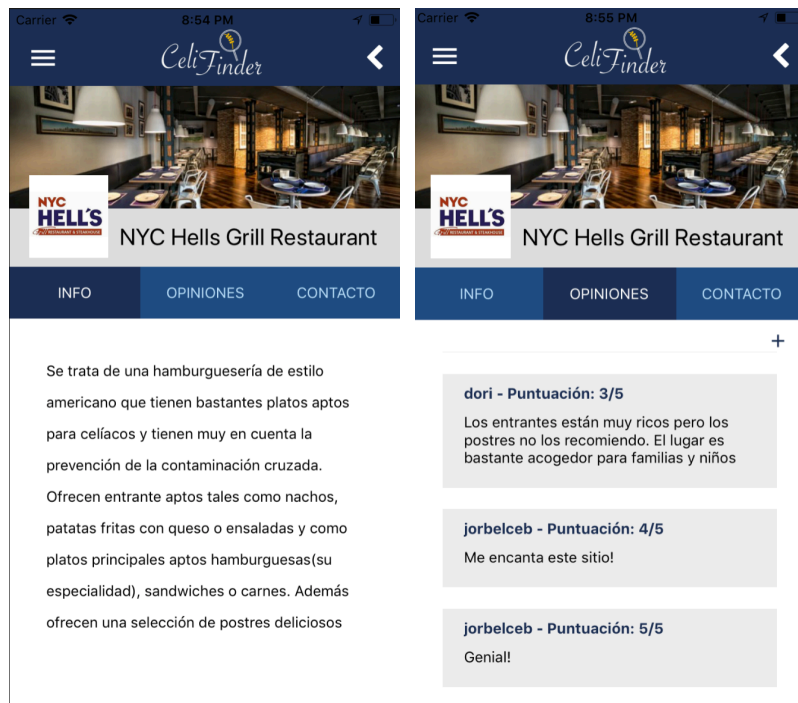


Ilustración 48. Pantalla de detalle (Información - a y Opiniones - b)

Si el usuario quiere añadir un comentario deberá haber iniciado sesión anteriormente, sino aparecerá un modal en el que le invitará a iniciar sesión. Si ha iniciado sesión podrá poner un comentario a través del siguiente modal [Ilustración 51. Añadir opinión]

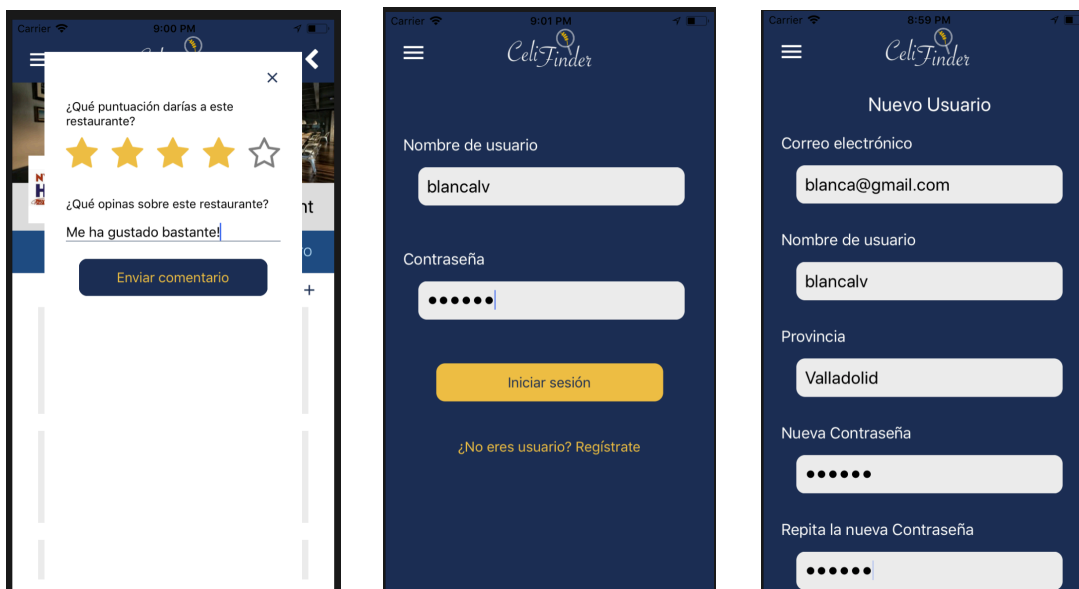


Ilustración 49. Añadir opinión (a), pantalla de Login (b) y pantalla de nuevo usuario

El usuario podrá llegar a la pantalla de Login a través del menú lateral emergente o cuando salga el modal de los comentarios y les invite a iniciar sesión.

Para iniciar sesión será necesario escribir el nombre de usuario y la contraseña del usuario.

Si el usuario no tiene cuenta hay un botón con el título ‘¿No eres usuario? Regístrate’ que los llevará a la pantalla de nuevo usuario, donde tendrá que especificar nombre de

usuario, correo, provincia y contraseña. Cuando el usuario se ha logado verá un menú como el siguiente donde podrá modificar su perfil de usuario o cambiar su contraseña.

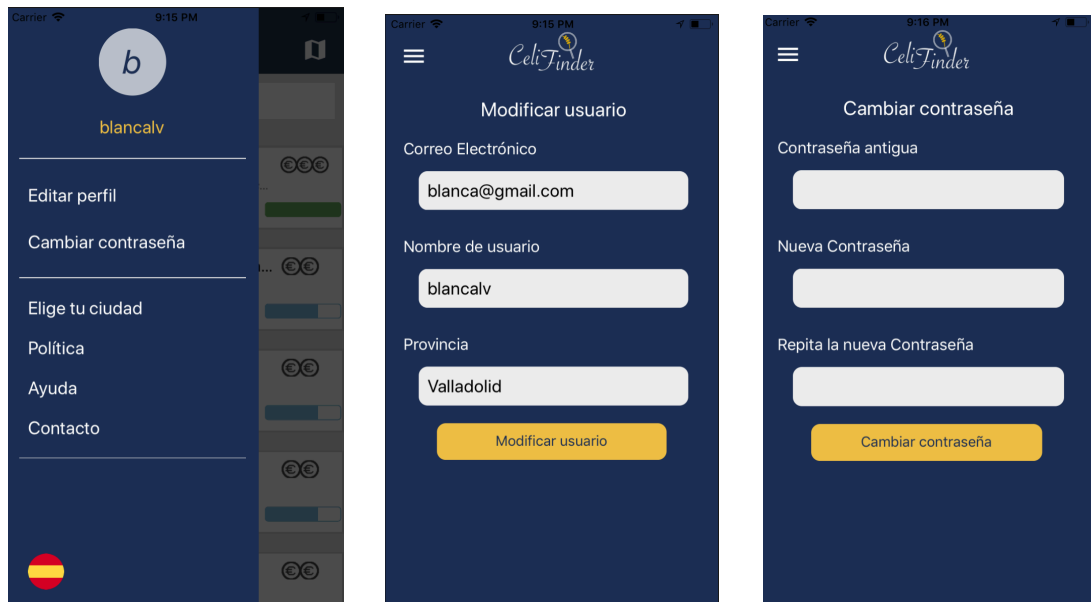


Ilustración 50. Menú lateral (a), pantalla de modificar perfil (b), pantalla de cambiar contraseña (c)

En el menú lateral aparecerá un avatar con la inicial del usuario logado y su nombre de usuario debajo de él. A través de las opciones de Política y Ayuda se puede acceder a la información de estas dos pantallas. En la pantalla de contacto aparecerá un formulario que se volcará sobre un correo en el cliente de mail del usuario, desde donde se podrá enviar el correo al soporte de Celifinder.

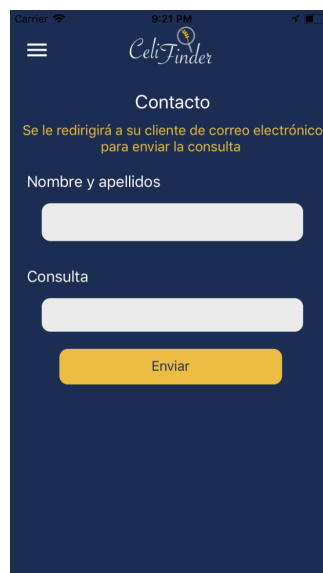


Ilustración 51. Pantalla de contacto



## 14. Anexo II. Manual de instalación

En el zip entregado, se encontrarán 3 ficheros de instalación:

- backup\_celifinder.sql  
Se trata del fichero backup de la base de datos en formato .sql
- back-project.zip  
Fichero comprimido con el proyecto de Spring Boot, que actuará como backend conectándose a la bbdd y dando respuesta a las peticiones front-end
- expo-pack.zip  
Fichero comprimido con el proyecto de React-Native desarrollado mediante Expo, que nos permitirá ejecutar el proyecto en los simuladores tanto de iOS como de Android **mediante Expo**
- native-pack.zip  
Fichero comprimido con el proyecto de React-Native desarrollado que nos permitirá generar los proyectos nativos de la app, con Android Studio y XCode, respectivamente

### 14.1 Base de datos

En primer lugar, es necesario obtener la base de datos a través del fichero backup\_celifinder.sql. La versión de MySQL que se ha utilizado es 8.0.12

Una vez la base de datos está instalada, se podrá comprobar que dentro de ella hay 3 tablas:

- restaurante
- usuario
- comentario

### 14.2 Backend

Al descomprimir el fichero se obtendrá un directorio con los ficheros necesarios para formar un proyecto de Spring Boot.

Es necesario tener STS (Sprint Tool Suite) instalada. La versión utilizada en este proyecto fue 3.9.6.RELEASE

Desde STS se importa el proyecto desde el directorio al que se haya descargado. Una vez podamos ver el proyecto importado en el Package Explorer, se selecciona sobre la carpeta raíz con el botón derecho y se selecciona: Run as > Maven install. Si han cambiado la contraseña o el nombre de usuario de la base de datos, habrá que cambiarlo en el fichero

```
src/main/resources/application.properties
```

Posteriormente en el BootDashboard se selecciona el nombre del proyecto y se arranca mediante el botón 'Run or restart process'

## 14.3 Instalación del Frontend

Para la instalación de la parte Frontend hay dos proyectos a partir de los cuales se puede arrancar la app.

- Por un lado, el proyecto contenido en expo-pack.zip que nos permite ver la app mediante Expo, evitando tener que generar los proyectos nativos de Android Studio y XCode.
- Por otro lado, el proyecto contenido en native-pack.zip está preparado y configurado para generar los proyectos nativos de XCode y de Android Studio respectivamente.

- 

Ambos comparten el mismo código fuente (contenido en /src y otros ficheros .js) pero la configuración de cada proyecto es diferente dependiendo de la finalidad de los mismos.

### 14.3.1 Arranque de la app con Expo

Al descomprimir el fichero expo-pack.zip, obtenemos un directorio. Para arrancar el proyecto front necesitamos las siguientes librerías:

- Node (v10.1.0)
- Yarn (v1.7.0)
- NPM (v5.6)

Mediante consola accedemos a este directorio e instalamos las dependencias mediante:

Yarn

o, como alternativa:

```
npm install
```

Es necesario comprobar en qué dirección IP y en qué puerto está arrancado el proyecto de Back. En el fichero /src/redux/apis/utils.api.js hay una variable string llamada basePath. En caso de ser una IP diferente deberá modificarse su valor con la iP y el puerto donde está expuesto el proyecto de backend.

Una vez se han instalado las dependencias correctamente se arranca el proyecto con el siguiente comando:

```
yarn start
```

Aparecerá lo siguiente en consola.



Ilustración 52. Consola de instalación de App con Expo

Si se tiene simulador de iOS o de Android (generados por XCode o Android Studio respectivamente) se podrá arrancar la aplicación pulsando sobre las letras 'i' (para iOS) y 'a' (para Android).

También se podrá escanear el código QR con un dispositivo móvil de tal forma que se abrirá la aplicación mediante la aplicación Expo.

La aplicación de Expo se puede descargar gratuitamente en:

- [iOS](#)
- [Android](#)

### 14.3.2 Arranque de la app con los proyectos nativos

Al descomprimir el fichero native-pack.zip, obtenemos un directorio. Para arrancarlo necesitamos las siguientes librerías (al igual que en el proyecto de arranque con Expo):

- Node (v10.1.0)
- Yarn (v1.7.0)
- NPM (v5.6)

Mediante consola accedemos a este directorio e instalamos las dependencias mediante:

```
yarn
```

o, como alternativa:

```
npm install
```

Los proyectos, tanto de XCode como de Android Studio, están generados, en el .zip, pero para que la app nativa funcione correctamente, antes hay que modificar la dirección IP donde está arrancado el servidor. Para ello, en el fichero /src/redux/apis/utils.api.js hay una variable string llamada basePath. En caso de ser una

IP diferente deberá modificarse su valor con la IP y el puerto donde está expuesto el proyecto de backend.

Una vez se han instalado las dependencias correctamente se vuelven a generar los proyectos de XCode y de Android Studio.

Para generar el proyecto de XCode se ejecuta el siguiente comando

```
react-native run-ios
```

En el directorio /ios estará el proyecto de XCode, donde se podrá ejecutar la app de forma nativa.

Para generar la app nativa de Android Studio se ejecuta el siguiente comando, teniendo un simulador de Android arrancado:

```
react-native run-android
```

Una vez se haya terminado de generar la app nativa, ésta se habrá instalado de forma nativa en el simulador de Android. En el directorio /android estará el proyecto de Android Studio.