



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una aplicación híbrida para la búsqueda y
comparación de comida apta para celíacos entre los
supermercados más visitados en España

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Martí Ripoll, Pau

Tutor/a: Sapena Vercher, Oscar

CURSO ACADÉMICO: 2021/2022

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor Óscar Sapena Vercher por la ayuda prestada durante el desarrollo del trabajo.

Por otro lado, también me gustaría agradecer todo el apoyo transmitido de mis compañeros, amigos y familiares durante la elaboración del proyecto.

Resumen

Este proyecto nace para ayudar a aquellas personas que compran alimentos sin gluten, ya que el mercado actual de este tipo de productos presenta unas condiciones de precio muy dispares. En consecuencia, se facilita una aplicación donde puedan comparar y descubrir nuevos productos. Además, tendrán la posibilidad de crearse una cuenta para formar parte de la comunidad de usuarios, aportando cada vez más productos al sistema si así lo desean. Esta herramienta estará disponible tanto en navegadores web como en dispositivos iOS y Android de forma nativa y en forma de PWA (Aplicación Web Progresiva). Para conseguir este propósito se utilizan tecnologías tales como Ionic y Capacitor en la parte frontal y Firebase como base de datos.

Como resultado se obtendrá una aplicación con un diseño muy intuitivo y sencillo para todos los públicos, con características modernas y una interfaz responsiva que agradará a los usuarios que la prueben.

Palabras clave: aplicación Web, Android, iOS, Ionic, Angular, Firebase, PWA, comparador de alimentos.

Abstract

This project was created to help those people who buy gluten-free food, since the current market for this type of products has very different price conditions. Consequently, an application is provided where they can compare and discover new products. In addition, they will have the possibility of creating an account to become part of the community of users, contributing more and more products to the system if they wish to do so. This tool will be available both in web browsers and on iOS and Android devices natively and as a PWA (Progressive Web Application). To achieve this purpose, technologies such as Ionic, Capacitor on the front end and Firebase as a database are used.

As a result you will get an application with a very intuitive and simple design for all audiences, with modern features and a responsive interface that will satisfy the users who try it.

Keywords : web application, Android, iOS, Angular, Firebase, PWA, Food Comparison.



Tabla de contenidos

1. Introducción.....	9
1.1 Motivación.....	9
1.2 Objetivos.....	10
1.3 Impacto esperado.....	11
1.4 Metodología.....	12
1.5 Estructura.....	13
2. Estado del arte	15
2.1 Clasificación de los productos.....	15
2.2 Alternativas.....	15
2.3 Conclusión.....	19
3. Análisis del problema	21
3.1 Especificación de requisitos	21
3.1.1 Actores	21
3.1.2 Requisitos funcionales	21
3.1.3 Requisitos no funcionales	32
3.2 Análisis del producto	33
3.2.1 Eficiencia algorítmica	33
3.2.2 Análisis del marco legal y ético	33
3.2.3 Análisis de riesgos.....	34
3.3 Plan de trabajo	36
3.4 Presupuesto.....	38
4. Diseño de la solución.....	41
4.1 Arquitectura del sistema	41
4.2 Diseño detallado	44
4.2.1 Modelo de datos.....	44
4.2.2 Prototipos	45
5. Implementación y despliegue	51
5.1 Tecnología utilizada.....	51
5.2 Desarrollo de la base de datos	56
5.3 Desarrollo de la parte frontal.....	57
5.4 Despliegue	61
6. Pruebas	65
6.1 Pruebas funcionales y pruebas de carga	65
6.2 Pruebas con usuarios	70
7. Conclusiones y trabajos futuros.....	75
7.1 Conclusión.....	75
7.2 Relación del trabajo desarrollado con los estudios cursados.....	76
7.3 Trabajos futuros	77
8. Referencias	79
Anexo	81



Índice de figuras

Figura 1. Gráfico comparador de la cesta de la compra con y sin gluten	12
Figura 2. Tablero de Trello de ejemplo	13
Figura 3. Soysuper página web.....	16
Figura 4. FACEMOVIL categorías.....	17
Figura 5. FACEMOVIL establecimientos.....	17
Figura 6. SinGLU10 menú principal	18
Figura 7. SinGLU10 comidas	19
Figura 8. Diagrama: casos de uso módulo buscar.....	22
Figura 9. Diagrama: casos de uso módulo detalles.....	25
Figura 10. Diagrama: casos de uso módulo lista	26
Figura 11. Diagrama: casos de uso módulo perfil	28
Figura 12. Diagrama de Gantt.....	37
Figura 13. Gráfica de costes totales	39
Figura 14. Arquitectura de la aplicación	42
Figura 15. Esquema multiplataforma.....	43
Figura 16. Diagrama de clases del modelo de datos.....	44
Figura 17. Prototipo: inicio	46
Figura 18. Prototipo: buscar.....	46
Figura 19. Prototipo: carrito.....	47
Figura 20. Prototipo: perfil	47
Figura 21. Prototipo: perfil - iniciar sesión.....	48
Figura 22. Prototipo: perfil - menú lateral	48
Figura 23. Prototipo: buscar - subcategoría	49
Figura 24. Prototipo: buscar - detalles	49
Figura 25. Prototipo: buscar - detalles - corregir error	50
Figura 26. Prototipo web: carrito	50
Figura 27. Logo de Ionic.....	51
Figura 28. Logo de Capacitor	52
Figura 29. Logo de Angular	52
Figura 30. Logo de Firebase	53
Figura 31. Logo de VS code	54
Figura 32. Logo de GitHub.....	54
Figura 33. Logo de Xcode	55
Figura 34. Logo de Android Studio	55
Figura 35. Almacenamiento de los productos en la base de datos.....	56
Figura 36. Servicios	57
Figura 37. Estructura de los directorios	58
Figura 38. Estructura del módulo search	60
Figura 39. PWA en iOS, Android y navegador web	62

Figura 40. Prueba de tiempos de carga en web.dev	68
Figura 41. Prueba de tiempos de carga en web.dev después de las correcciones	69
Figura 42. Resultado de encuesta: edad	70
Figura 43. Resultado de encuesta: compra frecuente sin gluten	71
Figura 44. Resultado de encuesta: ahorro en el ticket de compra	71
Figura 45. Resultado de encuesta: funcionalidad de comparar productos	71
Figura 46. Resultado de encuesta: descubrir nuevos productos	72
Figura 47. Resultado de encuesta: centralizar las compras en menos supermercados	73
Figura 48. Resultado de encuesta: facilidad de uso	73
Figura 49. Resultado de encuesta: rendimiento de la aplicación	74
Figura 50. Resultado de encuesta: recomendación a un conocido celíaco	74



Índice de tablas

Tabla 1. Comparativa de características de competidores	20
Tabla 2. Caso de uso: ver categoría	22
Tabla 3. Caso de uso: ver detalles	23
Tabla 4. Caso de uso: buscar un artículo.....	23
Tabla 5. Caso de uso: filtrar búsqueda	23
Tabla 6. Caso de uso: cancelar búsqueda.....	24
Tabla 7. Caso de uso: ir a una subcategoría	24
Tabla 8. Caso de uso: alerta de un error.....	25
Tabla 9. Caso de uso: añadir a la lista.....	25
Tabla 10. Caso de uso: agregar un comentario	26
Tabla 11. Caso de uso: ir a una subcategoría	27
Tabla 12. Caso de uso: ir a una subcategoría	27
Tabla 13. Caso de uso: ir a una subcategoría	27
Tabla 14. Caso de uso: ir a FAQs	29
Tabla 15. Caso de uso: iniciar sesión	29
Tabla 16. Caso de uso: crear cuenta.....	29
Tabla 17. Caso de uso: cerrar sesión.....	30
Tabla 18. Caso de uso: formulario nuevo artículo	30
Tabla 19. Caso de uso: formulario sugerencia	30
Tabla 20. Caso de uso: revisar solicitud.....	31
Tabla 21. Caso de uso: revisar corrección	31
Tabla 22. Caso de uso: activar notificaciones	31
Tabla 23. Caso de uso: modificar datos personales	32
Tabla 24. Niveles de probabilidad de riesgos	34
Tabla 25. Niveles de impacto.....	34
Tabla 26. Riesgos del desarrollo software	35
Tabla 27. Riesgos en la aceptación de usuarios.....	35
Tabla 28. Costes materiales.....	38
Tabla 29. Salario medio neto mensual	39
Tabla 30. Costes de recursos humanos	39
Tabla 31. Niveles posibles para medir el resultado.....	65
Tabla 32. Resultados funcionales: módulo buscar	66
Tabla 33. Resultados funcionales: módulo detalles	66
Tabla 34. Resultados funcionales: módulo lista.....	66
Tabla 35. Resultados funcionales: módulo perfil.....	67

1. Introducción

Una buena nutrición es vital para nuestra salud y bienestar, es increíble ver cómo los alimentos que ingerimos influyen tanto en nuestro día a día, por ejemplo, estar somnoliento, sentirse cansado, encontrarse mal, entre otros, son consecuencias de una mala alimentación. Es por esto que la nutrición es tan importante como cualquier otra actividad saludable en nuestro día a día. Hacer deporte, evitar estar sentado muchas horas, dormir bien y comer de forma equilibrada son hábitos saludables que están estrechamente relacionados con el bienestar tanto emocional como de salud, por esta razón, debemos elegir con cautela qué alimentos vamos a ingerir diariamente y no tomarnos tan a la ligera la nutrición.

Existe una proteína, el gluten, que es muy habitual para la mayoría de nosotros, la solemos comer en prácticamente todas las comidas, ya que la podemos encontrar entre otros, en el pan, trigo, cereales, dulces, pasteles, cerveza e incluso en el chocolate. El problema aparece cuando una persona no es capaz de digerir de forma correcta esta proteína, por lo tanto, se ve obligada a eliminar de su dieta todo alimento que contenga gluten. A estas personas se les llama celíacos y dependiendo de su tolerancia a esta proteína se ven obligadas en mayor medida a no comer bajo ningún concepto alimentos que contengan gluten.

Es por esto que, en su vertiente más aguda, muchas personas no pueden comer fuera de sus casas por miedo a que los restaurantes no tengan comida sin gluten o no cumplan unos estándares, obligándoles así, a comer en la mayoría de los casos en sus casas, cocinando cada plato de su dieta.

1.1 Motivación

Una de las primeras causas por la que me he sentido motivado para la realización de este proyecto ha sido ponerme en la piel de un celíaco durante varios meses. Mis dolores de estómago llevaron a un médico a proponer quitarme de mi dieta el gluten para observar si existía algún cambio en mi mal estar diario, y durante este periodo de tiempo, me sentí muy frustrado porque prácticamente todos los alimentos que conocía y utilizaba contenían gluten y no los podía comer. Sentí la necesidad de encontrar unos alimentos que sustituyeran a los habituales y no fue una tarea fácil. En los supermercados encontraba muy pocos alimentos sin gluten y los trabajadores desconocían qué comidas llevaban o no esta proteína [1].

Otra causa que me llevó al pensamiento de este proyecto fue, que durante mi periodo antigluten, me di cuenta de que todos los productos en los supermercados que no llevaban gluten eran más costosos. El ticket al final de cada compra semanal incrementó significativamente respecto a las compras anteriores, hecho que, me llevó a la idea de comparar

alimentos iguales o parecidos en los supermercados más cercanos a mi casa con el fin de evitar en mayor medida, ese incremento en cada compra.

Yéndonos a la parte más técnica, desde que empecé mis prácticas en empresa, he estado realizando proyectos relacionados estrechamente con la creación de páginas web, me he ido formando en este ámbito durante un tiempo y he descubierto el mundo de la programación web más de cerca. Este proyecto es una oportunidad de aprendizaje tanto de un nuevo framework, como de nuevas herramientas y paradigmas que fortalecerán mis conocimientos actuales.

En resumen, la realización de este proyecto surge gracias a la unión de las dos etapas comentadas hasta ahora, por un lado, crear desde cero una aplicación con un framework prácticamente desconocido me motivaba a crecer más en el mundo de la programación web, y por otro lado, la necesidad de las personas que no comen gluten en encontrar alimentos aptos y de la forma más económica posible, me ha motivado a llevar a cabo esta herramienta.

1.2 Objetivos

El objetivo principal de esta aplicación es compartir una herramienta a todas aquellas personas que no puedan o hayan decidido no comer gluten, la cual trata de ayudarles de diversas maneras. Para empezar, tratará de ayudar a estas personas a descubrir nuevos alimentos libres de gluten, gracias a la gran cantidad de productos que se mostrarán en la aplicación. Podrán encontrar diferentes productos organizados por categorías y de hasta cinco supermercados distintos en España. Para ayudar a los usuarios, dentro de estas categorías existirán subtipos diferentes y se podrán filtrar por supermercado. Además, dispondrán en todo momento de un buscador por categoría y un buscador global que filtrará por todos los productos de la aplicación, el objetivo de estas búsquedas, es para que cualquier usuario pueda encontrar algún producto que no sepa donde se encuentra realmente.

Otra característica fundamental de esta herramienta es que permite la comparación de productos. El usuario podrá añadir los artículos que le resulten interesantes a una lista y se mostrarán las diferencias de precio para cada uno de ellos dependiendo del supermercado. El objetivo es crear una lista de la compra por cada supermercado disponible en la aplicación, de modo que permita elegir qué productos son los más rentables o los que más le interesa, ahorrando así en el ticket final cuando vaya a hacer la compra. Además, gracias a esta estructura, el usuario sabrá en cada momento que supermercado le interesa más y no perderá tiempo yendo físicamente a todos los supermercados en busca de alimentos que sean aptos para él.

Hay que hacer hincapié en que vamos a construir una herramienta de comparación de artículos, no vamos a crear una tienda ni un supermercado donde podamos finalizar una compra. Es más bien, una aplicación donde dar a conocer nuevos productos aptos para celíacos, sus

precios, y donde encontrarlos. Además, se forjará una comunidad de personas gracias a los comentarios y las aportaciones que podrán realizar los usuarios de nuestra aplicación, ya que, otra meta en mente, sería la creación de formularios por parte de nuestros usuarios para incrementar la base de datos de la aplicación, de la misma forma, se dará la opción de corregir las ya existentes si estas están incorrectas o han sufrido alguna modificación en el mercado. Con todo esto conseguiríamos una herramienta cada vez más grande y actualizada con el paso del tiempo.

Por otro lado, aunque la aplicación es de libre acceso para todo el público, existirá la opción de iniciar sesión, la cual habilitará las opciones de comentar en los productos y generar los formularios mencionados anteriormente. Esta autenticación se podrá realizar con Google.

Por último, como la finalidad de este software es abarcar a todo el público celiaco en España, la aplicación deberá de ser intuitiva y agradable, con características de diseño cercanas a las ya utilizadas por otras aplicaciones en el mercado, para que facilite el uso en una primera toma de contacto con nuestros usuarios. Entre estas características se incluyen:

- Botones inferiores que estructuren la aplicación (tabs).
- En el apartado de búsqueda se incluirá un menú lateral deslizable el cual incorpore un índice de todos los tipos y subtipos de productos.
- Modo oscuro y modo claro (el fondo cambiará a oscuro y claro).
- Apartado de FAQs (preguntas frecuentes).
- Posibilidad de activar notificaciones cuándo se comente en un producto que el usuario ha comentado.
- Responsive en formato web.

Además, la aplicación deberá de ser de fácil acceso para todos los públicos, por consiguiente, se creará para las plataformas web, Android e iOS.

1.3 Impacto esperado

Siendo francos, cuando se decidió apostar por una aplicación con un margen de usuarios tan acotado, se pensó en que el impacto inicial no sería tan grande comparado con una herramienta enfocada a todo el público. Es por esto que nuestro objetivo es llegar al máximo de personas intolerantes al gluten para ayudarles en su alimentación diaria y conseguir generar un impacto en la forma en que compran estos alimentos y convertir esta aplicación en una herramienta habitual y normalizada para este sector.

Cabe señalar, que se espera que todas esas familias afectadas por la gran diferencia de los costes entre el tipo de comida con gluten y sin gluten (ver figura 1) puedan ahorrar en cada compra cuando van al supermercado.

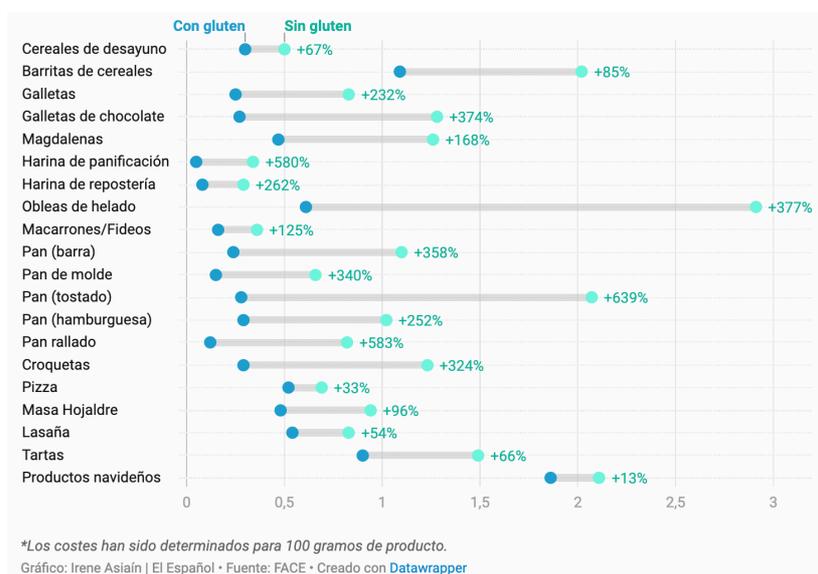


Figura 1. Gráfico comparador de la cesta de la compra con y sin gluten

Por otro lado, como veremos en páginas posteriores, entramos en un sector donde existen muchas herramientas ya consolidadas de uso frecuente en la población. Sin embargo, estas aplicaciones tienen un enfoque distinto y podemos afirmar que no competimos con ellas directamente, más bien se aproxima a una unión de funcionalidades donde nuestra aplicación crearía un punto de inflexión para el sector. La incorporación en el mercado llevará a cabo posibles contratos, permitiendo a nuestra aplicación la implementación de otras funcionalidades de aplicaciones externas.

1.4 Metodología

Para llevar a cabo este proyecto software se va a utilizar una metodología ágil. Nos enfocaremos en una de sus características más empleadas, dividir el producto final en pequeños proyectos capaces de dar un funcionamiento por si solos, generando así, partes del software con el que se pueda probar, desde la obtención de información de la base de datos, hasta el diseño de la página en cuestión. Para estructurar estas piezas de software y como el proyecto se llevará a cabo por una única persona, se usará la estructura Kanban [2], la cual consiste en crear tarjetas que incluyan funcionalidades, y estas tarjetas se desplazarán entre los siguientes estados (ver figura 2), el primero, *To Do*, indica la creación de la tarjeta y que está pendiente de realización; *In Progress*, le da a la tarjeta el significado de que se está implementando actualmente, y el último estado por el que pasan las tarjetas es, *Done*, el cual indica que la tarea se ha realizado.

No obstante, estos tres estados no serán absolutos, y dependiendo de la situación del proyecto podrán generarse otros. Del mismo modo, se estudiará el alcance del producto considerando el mínimo producto viable (MVP) [3].



Figura 2. Tablero de Trello de ejemplo

La herramienta que se utilizará para la realización de estas tarjetas es Trello [4], ya que nos permite la creación de un tablero donde se pintan los tres estados en forma de columnas. En cada columna se introducen las tarjetas anteriormente mencionadas, y cómo se puede ver en la figura 2, permite crear tantas columnas como necesitemos en cada momento del desarrollo.

1.5 Estructura

La estructura que se presenta en este documento es la siguiente:

- **Estado del arte.** En este capítulo se va a realizar un estudio del tema que ocupa este trabajo. Se elaborará una comparación de las aplicaciones o productos similares y se llevará a cabo una tabla comparativa de características entre esas aplicaciones y este proyecto.
- **Análisis del problema.** Para la elaboración del proyecto se efectuará un estudio previo, el cual incluirá una especificación de requisitos, un análisis de riesgos, un plan de trabajo y se concluirá con un presupuesto.
- **Diseño de la solución.** En este apartado se definirá la arquitectura del sistema, incluyendo un diagrama de clases y un diagrama con las API que se utilizarán. Además, se presentará el esquema de la implantación propuesta y se finalizará con los prototipos elaborados.

- **Implementación y despliegue.** Llegados a este punto se explicará como se ha desarrollado la aplicación y la estructura del código, con qué tecnologías se ha trabajado para lograrlo, y por último, se expondrá el despliegue efectuado con los errores encontrados durante el proceso.
- **Pruebas.** En este capítulo se abordarán las pruebas realizadas, en las que se incluyen: pruebas funcionales para revisar si los casos de uso se han cumplido en su totalidad, pruebas de carga para revisar la eficiencia de arranque de la aplicación y pruebas con usuarios en forma de encuestas para conocer diferentes opiniones.
- **Conclusiones y trabajos futuros.** En este último capítulo se hará un breve repaso de los objetivos y metas propuestos al principio del proyecto y se analizará si se han alcanzado completamente. Además, se enlazarán los conceptos utilizados en este trabajo con los vistos durante la carrera, y se finalizará con nuevas ideas que podrían llevarse a cabo en futuras versiones de la aplicación.

2. Estado del arte

En este apartado se realizará un estudio del estado actual de los productos similares a este proyecto y que características están cubiertas hoy en día, comparándolas con las soluciones propuestas por nuestra aplicación.

2.1 Clasificación de los productos

En el tema que aborda este proyecto es muy importante conocer la veracidad de los productos sin gluten. Saber diferenciar qué alimentos contienen esta proteína y con qué medidas son aspectos muy importantes a tener en cuenta para un celíaco. Es por eso que todas las herramientas relacionadas directamente con el tema, incluido la nuestra, deberían tener fuentes seguras para conocer la cantidad de gluten de los productos. Normalmente, esto se consigue gracias a una etiqueta impresa en el producto que nos indica cuando un alimento no contiene gluten.

Como veremos a continuación, existen aplicaciones capaces de escanear el código de barras de un producto para conocer todos sus ingredientes, facilitando así la compra para una persona celíaca.

2.2 Alternativas

Antes de citar el listado de productos, cabe resaltar que, no existe o no se ha encontrado una aplicación con los mismos propósitos que la nuestra hoy en día. Sin embargo, existen muchas aplicaciones enfocadas en los productos sin gluten.

Para comenzar, la herramienta más similar a la nuestra es Soysuper [5], la cual trata de una página web que compara todos los productos de distintos supermercados permitiendo hacer una cesta de la compra. Cuando el usuario completa la cesta, este puede terminar el pedido en la página web del supermercado y realizar una compra online.

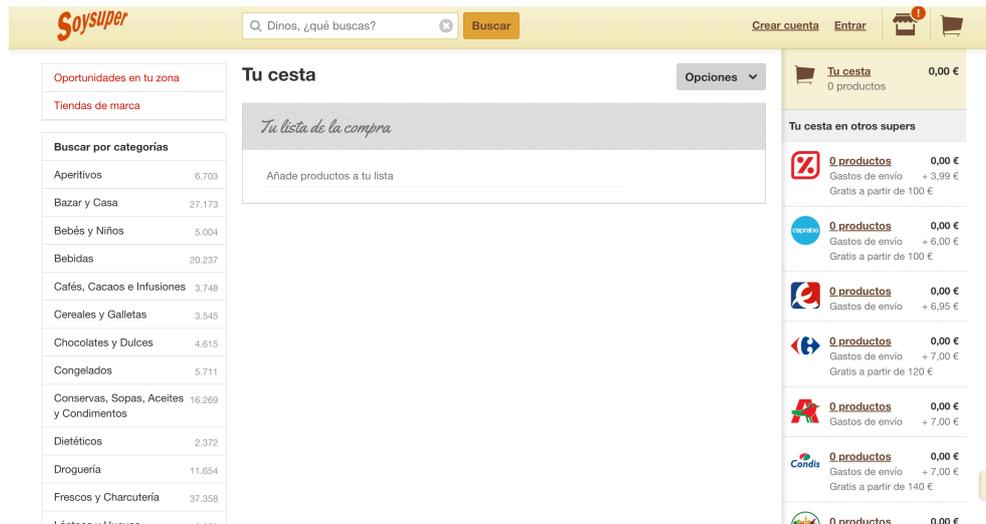


Figura 3. Soysuper página web.

Paradójicamente, nosotros buscamos un comparador similar al de esta página web, pero con un enfoque único para celíacos, aproximándonos más al público y cómo se ha explicado en el capítulo anterior, creando una comunidad de usuarios. Nuestra perspectiva es generar en el usuario una sensación de grupo y ayuda que no vemos en la página de Soysuper.

Por otro lado, se encuentran las aplicaciones para celíacos más famosas en el mercado actual. Como es el caso de FACEMOVIL [6], la cual está administrada por FACE (Federación de Asociaciones de Celíacos de España). Esta aplicación móvil trata de ayudar a las personas celíacas asegurando cuando un producto realmente no lleva gluten, ya que poseen una lista de más de 15.000 productos. Además, tienen la característica de escanear el código de barras de un producto directamente con el móvil en el supermercado, para alertar al usuario si el producto es conveniente o no.

Esta aplicación posee la característica de buscar los restaurantes de comida más cercanos al usuario con el fin de asegurarse de que el restaurante tenga comida sin gluten con cierta calidad. Además, al tratarse de una federación, los socios pueden conseguir descuentos en los servicios mostrados por la aplicación.



Figura 4. FACEMOVIL categorías.



Figura 5. FACEMOVIL establecimientos.

A pesar de que la aplicación tiene grandes características, esta no es pública para todos los usuarios. Para poder utilizarla debes pertenecer a FACE como socio de alguna asociación incluida en su programa, es decir, que esta aplicación no es gratuita.

La última aplicación que estudiaremos en este capítulo es sinGLU10 [7], la cual comparte características con la herramienta anterior. Por ejemplo, este programa contiene más de 18.000 productos revisados por otras asociaciones para que el usuario, en el momento de realizar la compra en el supermercado, pueda revisar si el artículo que está comprando realmente está aprobado como comida sin gluten, para hacer esto la aplicación emplea la cámara del dispositivo para escanear el código de barras y mostrarte información del producto. Además, también aporta la posibilidad de buscar restaurantes con comida sin gluten.

Otra característica importante de esta herramienta, es la sección de recetas sin gluten para preparar menús variados. En ella se pueden encontrar recetas sin gluten, con poco gluten o con un alimento que tiene gluten.

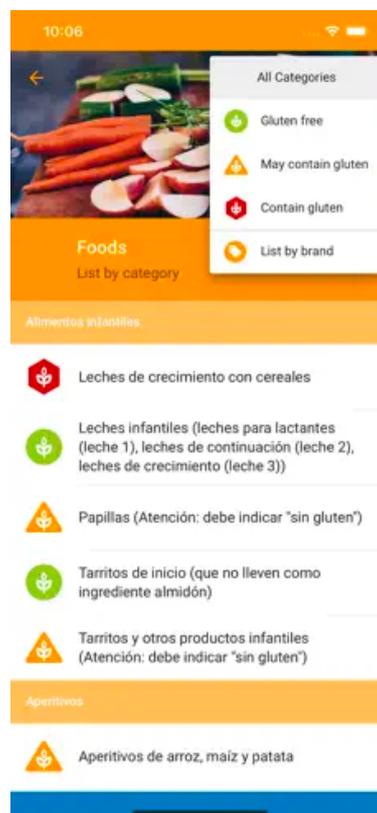


Figura 6. SinGLU10 menú principal



Figura 7. SinGLU10 comidas

2.3 Conclusión

Como se ha estudiado en el anterior apartado, la gran mayoría de aplicaciones que engloban nuestro tema se concentran en ofrecer la posibilidad de escanear los códigos de barras de cada producto en los supermercados para obtener información acerca del contenido de gluten de los mismos. Aunque en este estudio solo se muestran tres muy importantes, la realidad es que existen más aplicaciones con estas características que no se enumeran aquí por falta de funciones novedosas.

La comparación directa con nuestra aplicación nos da como resultado que únicamente una de estas alternativas se asemeja a los objetivos de comparación de productos por sus precios, pero justo esta no está estrechamente relacionada con la celiacía. Son por estas razones que nuestra herramienta puede hacer frente al mercado actual y puede relacionarse con estas aplicaciones sin llegar a sustituir ninguna de ellas, es más, como se menciona en el apartado de impacto esperado, se podría fusionar nuestra aplicación con algunas de las mencionadas anteriormente para incluir nuevas funcionalidades.

Para finalizar este capítulo se expone la tabla 1 con la comparativa de características de las alternativas propuestas.

Características	Nuestra aplicación	Soysuper	FACEMOVIL	sinGLU10
Enfocadas en la celiacía	Sí	No	Sí	Sí
Comparador de productos por precio	Sí	Sí	No	No
Comentarios en los productos (usuarios)	Sí	No	No	No
Filtrar productos por supermercado	Sí	Sí	No	No
Crear una lista de la compra por cada supermercado	Sí	Sí	No	No
Lector de código de barras	No	No	Sí	Sí
Recetas	No	No	No	Sí
Localizador de restaurantes	No	No	Sí	Sí

Tabla 1. Comparativa de características de competidores

3. Análisis del problema

En este capítulo haremos un análisis exhaustivo de nuestra aplicación, y para ello estudiaremos primeramente los actores, luego introduciremos la especificación de requisitos funcionales y no funcionales, incluyendo los diagramas de caso de uso. Por último, se harán unos últimos análisis del producto y se concluirá con un plan de trabajo y un presupuesto.

3.1 Especificación de requisitos

La especificación de requisitos es fundamental para el desarrollo de una aplicación software. En este contexto se estudiarán los requisitos funcionales y los no funcionales, diferenciándose entre ellos de la siguiente manera, los requisitos funcionales especifican “que” debe hacer el sistema, mientras que los requisitos no funcionales especifican el “cómo” se deben de hacer.

3.1.1 Actores

En nuestra aplicación se encuentran dos tipos de actores.

- Usuario no registrado. Se incluye todo el público que no ha iniciado sesión en la aplicación.
- Usuario registrado. Son los usuarios que han iniciado sesión en la aplicación, y pueden acceder a todas las funcionalidades de los usuarios no registrados y además tienen características añadidas.

3.1.2 Requisitos funcionales

Para este proyecto se hará uso de la especificación semi-formal es decir, los elementos en consideración y sus relaciones son declarados formalmente, pero usaremos sentencias que se describen en lenguaje natural.

A continuación se expondrán los casos de uso junto con sus diagramas relacionándolos con los actores. Estos se englobarán por módulos.



- **Módulo buscar**

Este módulo agrupa las funciones disponibles dentro de la página buscar. En la figura 8 se observa su diagrama de casos de uso. Cabe señalar que se ha utilizado la misma figura para representar los dos actores, usuario registrado y usuario no registrado.

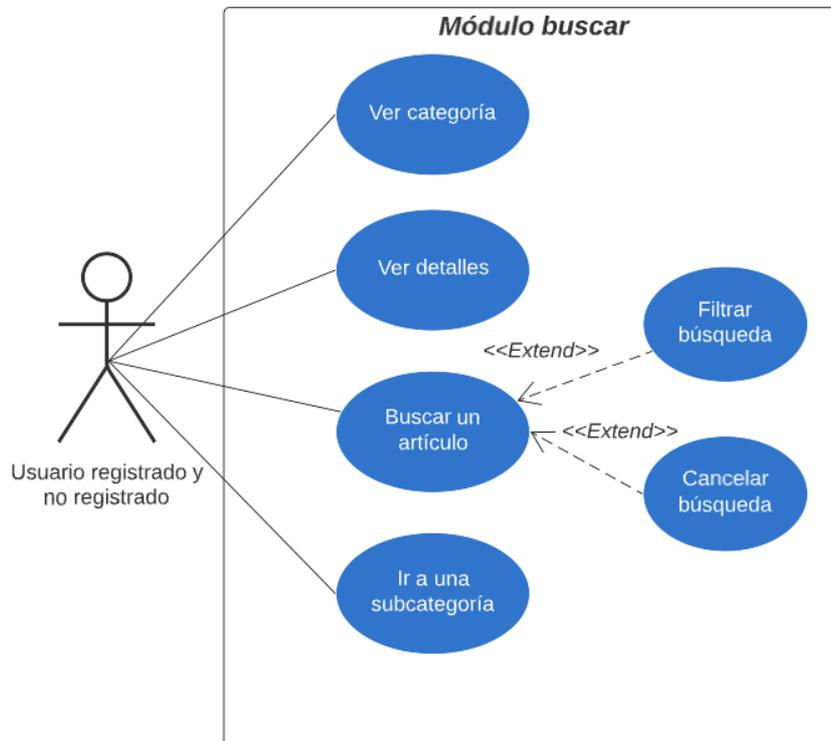


Figura 8. Diagrama: casos de uso módulo buscar

Ver categoría	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es la vista donde se listan los productos de una categoría en concreto.
Precondición	Que exista la categoría mostrada.
Secuencia principal	1. El usuario pulsa sobre la tarjeta de una de las categorías mostradas o el usuario pulsa sobre el botón dentro del menú de una de las categorías mostradas.
Salida	El sistema mostrará una nueva ventana con la lista de productos de esa categoría.

Tabla 2. Caso de uso: ver categoría

Ver detalles	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es la vista donde se muestran todos los detalles de un producto y sus comentarios.
Precondición	Que exista el producto en la base de datos junto sus imágenes.
Secuencia principal	1. El usuario pulsa sobre una de las tarjetas de producto que se muestran en pantalla.
Salida	El sistema mostrará una nueva ventana con la vista de detalles de ese producto.

Tabla 3. Caso de uso: ver detalles

Buscar un artículo	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es un input donde el usuario podrá buscar un producto por el nombre.
Precondición	Ninguna
Secuencia principal	1. El usuario pulsa sobre el input de búsqueda y introduce un nombre.
Salida	Automáticamente el sistema quitará de la pantalla todas las tarjetas y mostrará únicamente las que tengan el mismo nombre que el introducido por el usuario.

Tabla 4. Caso de uso: buscar un artículo

Filtrar búsqueda	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es un componente que añade un filtro al nombre del input de búsqueda. Este puede ser por ejemplo, filtrar por supermercado.
Precondición	Que se esté utilizando el input de búsqueda de una categoría o subcategoría.
Secuencia principal	1. El usuario pulsa sobre el botón de filtros. 2. Se despliega unos botones para elegir que filtro se quiere activar/desactivar. 3. El usuario pulsa sobre uno de los botones.
Salida	Automáticamente el sistema mostrará los productos que concuerden con el nombre del input de búsqueda y con el filtro seleccionado.

Tabla 5. Caso de uso: filtrar búsqueda

Cancelar búsqueda	
Actores	Usuario registrado y usuario no registrado.
Descripción	Cancelar una búsqueda activa
Precondición	Que haya una búsqueda activa
Secuencia principal	1. El usuario pulsa sobre el botón de cancelar búsqueda o simplemente elimina el nombre introducido en el input de búsqueda.
Salida	Automáticamente el sistema volverá a mostrar los datos que habían anteriores a la activación de la búsqueda en la pantalla.

Tabla 6. Caso de uso: cancelar búsqueda

Ir a una subcategoría	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es la vista donde se encuentran los productos dentro de una subcategoría.
Precondición	Que exista la subcategoría mostrada.
Secuencia principal	1. El usuario pulsa sobre el botón del menú lateral. 2. Se despliega un menú lateral con las categorías y subcategorías. 3. El usuario selecciona la subcategoría deseada.
Salida	El sistema mostrará una nueva ventana con la lista de productos de esa subcategoría.

Tabla 7. Caso de uso: ir a una subcategoría

- **Módulo detalles**

En este módulo se encuentran las funciones disponibles dentro de la página de detalles de un producto. En la figura 9 se muestra su diagrama de casos de uso.

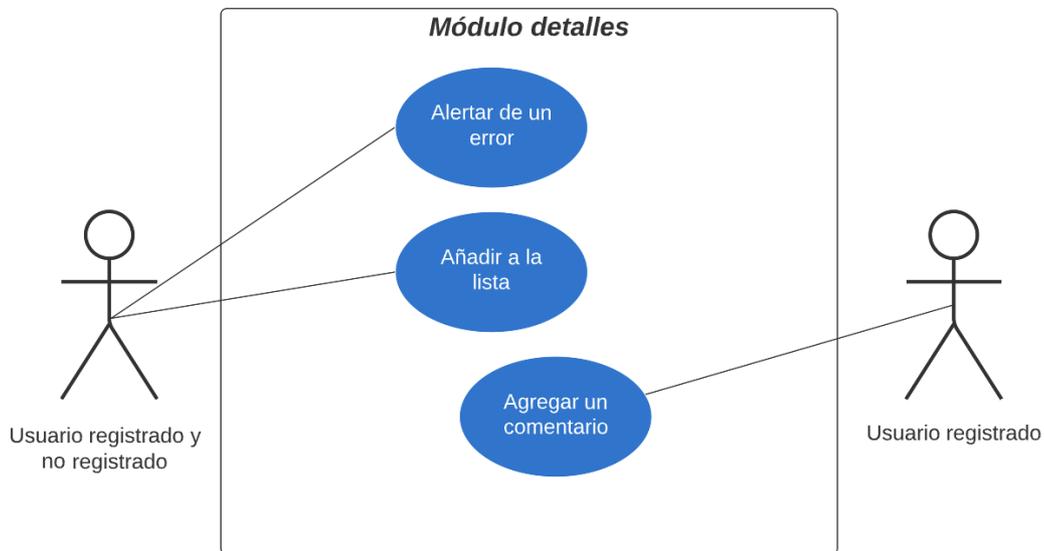


Figura 9. Diagrama: casos de uso módulo detalles

Alerta de un error	
Actores	Usuario registrado y usuario no registrado.
Descripción	Esta funcionalidad sirve para enviar un formulario con algún error en específico de entre estos: precio, existencia, cantidad, precio/cantidad, tipo, imagen y nombre. Posteriormente será evaluado por los demás usuarios y actualizado.
Precondición	Que el usuario se encuentre en la vista de detalles de un pedido y el pedido exista.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de opciones de la cabecera. 2. Se despliega una ventana modal donde aparecerá un selector con los distintos errores posibles. 3. El usuario selecciona el tipo de error y lo modifica. 4. El usuario pulsa sobre el botón enviar.
Salida	El sistema cerrará la ventana modal y enseñará un mensaje informativo.

Tabla 8. Caso de uso: alerta de un error

Añadir a la lista	
Actores	Usuario registrado y usuario no registrado.
Descripción	Se añade a la lista de la compra el producto en cuestión.
Precondición	Estar en la ventana de detalles y que exista el producto en la base de datos.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de añadir a la lista.
Salida	El sistema mostrará una notificación de confirmación. En el caso de que el producto ya esté en la lista, el sistema volverá a mostrar la notificación pero el producto no se duplicará.

Tabla 9. Caso de uso: añadir a la lista

Agregar un comentario	
Actores	Usuario registrado.
Descripción	Se añade un comentario en la sección de comentarios de un producto.
Precondición	Que el usuario esté registrado, es decir, haya iniciado sesión en la aplicación. Además el usuario debe de estar en la ventana de detalles del producto y que este exista.
Secuencia principal	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón de comentarios.2. Se despliega la vista de comentarios debajo del producto.3. El usuario introduce en el input un comentario.4. El usuario pulsa sobre el botón enviar.
Salida	El sistema mostrará el comentario en la lista de comentarios.

Tabla 10. Caso de uso: agregar un comentario

• Módulo lista

En este módulo se encuentran las funcionalidades relacionadas con la lista de la compra, la cual tiene intención de comparar los distintos productos por supermercados almacenados en ella. En la figura 10 se muestra su diagrama de casos de uso.

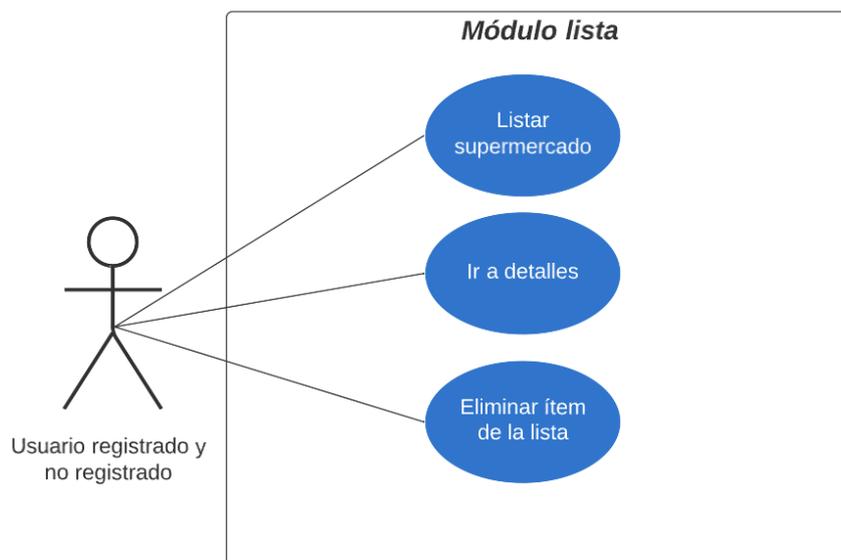


Figura 10. Diagrama: casos de uso módulo lista

Listar supermercado	
Actores	Usuario registrado y usuario no registrado.
Descripción	Se muestran los productos de la lista correspondientes al supermercado seleccionado.
Precondición	Que exista un producto en la lista de ese supermercado, de no ser así, no se mostrará nada.
Secuencia principal	1. El usuario pulsa sobre una de las cinco imágenes de supermercados.
Salida	El sistema mostrará los productos de la lista que estén en ese supermercado.

Tabla 11. Caso de uso: ir a una subcategoría

Ir a detalles	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es la vista donde se muestran todos los detalles de un producto y sus comentarios.
Precondición	Que exista un producto incluido en el supermercado.
Secuencia principal	1. El usuario pulsa sobre la imagen del producto o el usuario pulsa sobre el nombre del producto.
Salida	El sistema mostrará una nueva ventana con la vista de detalles de ese producto.

Tabla 12. Caso de uso: ir a una subcategoría

Eliminar ítem de la lista	
Actores	Usuario registrado y usuario no registrado.
Descripción	Se eliminará el producto de la lista.
Precondición	Que exista ese producto en la lista y que se esté listando ese supermercado
Secuencia principal	1. El usuario pulsa sobre el botón de eliminar producto.
Salida	El sistema dejará de mostrar el producto en la lista y actualizará el coste total de la lista.

Tabla 13. Caso de uso: ir a una subcategoría

- **Módulo perfil**

Este módulo engloba las funcionalidades correspondientes al perfil del usuario y a la creación / revisión de formularios. En la figura 11 se encuentra el diagrama de casos de uso.

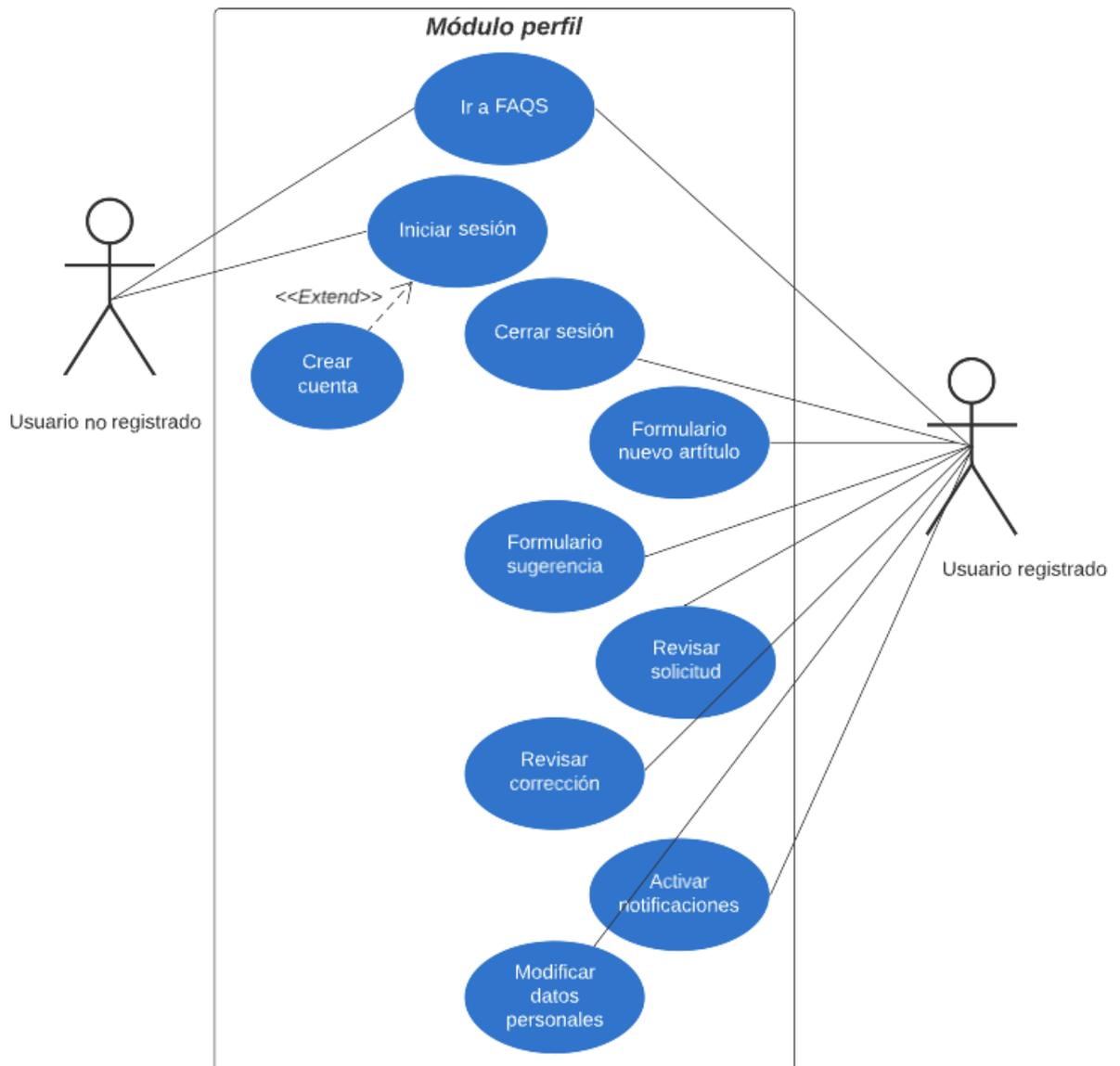


Figura 11. Diagrama: casos de uso módulo perfil

Ir a FAQs	
Actores	Usuario registrado y usuario no registrado.
Descripción	Es la vista donde se encuentran las preguntas frecuentes junto con sus respuestas.
Precondición	Ninguna.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de FAQs. 2. El sistema abrirá una nueva ventana con preguntas frecuentes. 3. El usuario selecciona una de las preguntas frecuentes mostradas.
Salida	El sistema desplegará la respuesta a la pregunta frecuente.

Tabla 14. Caso de uso: ir a FAQs

Iniciar sesión	
Actores	Usuario no registrado.
Descripción	Se inicia sesión con los datos del usuario.
Precondición	Que el usuario no esté registrado, es decir, no haya iniciado sesión en la aplicación.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de iniciar sesión. 2. El sistema abrirá una nueva ventana con la posibilidad de iniciar sesión con usuario y contraseña o iniciar sesión con Google. Además se mostrará un botón de crear cuenta. 3. El usuario elige el método de inicio de sesión. 4. El usuario introduce su usuario (email) y contraseña.
Salida	El sistema volverá a la vista de perfil con los datos del usuario actualizados.

Tabla 15. Caso de uso: iniciar sesión

Crear cuenta	
Actores	Usuario no registrado.
Descripción	Se crea una cuenta para un usuario que no ha iniciado sesión por primera vez en la aplicación.
Precondición	Que el usuario no haya iniciado sesión al menos una vez en la aplicación. Que el usuario se encuentre en la ventana de inicio de sesión.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de crear una nueva cuenta o inicia sesión con Google por primera vez. 2. El usuario introduce sus datos.
Salida	El sistema volverá a la vista de perfil con los datos del usuario actualizados.

Tabla 16. Caso de uso: crear cuenta

Cerrar sesión	
Actores	Usuario registrado
Descripción	Se cierra la sesión actual.
Precondición	Que el usuario haya iniciado sesión.
Secuencia principal	1. El usuario pulsa sobre el botón de cerrar sesión.
Salida	El sistema mostrará la misma ventana del perfil pero sin los datos del usuario actual.

Tabla 17. Caso de uso: cerrar sesión

Formulario nuevo artículo	
Actores	Usuario registrado.
Descripción	Se completa un formulario para solicitar la incorporación de un nuevo producto a la base de datos de la aplicación.
Precondición	Que el usuario esté registrado.
Secuencia principal	1. El usuario pulsa sobre el botón de nuevo formulario. 2. El sistema mostrará una nueva ventana con unos inputs a rellenar. 3. El usuario rellenará el formulario según los datos del nuevo producto. 4. El usuario pulsa sobre el botón seleccionar imagen y añade las imágenes al formulario. 5. El usuario pulsa en enviar formulario.
Salida	El sistema volverá a la ventana anterior enseñando un mensaje de confirmación.

Tabla 18. Caso de uso: formulario nuevo artículo

Formulario sugerencia	
Actores	Usuario registrado.
Descripción	Se completa un formulario con alguna sugerencia de mejora o incorporación de alguna nueva característica. Es la forma de tener un feedback directo del usuario.
Precondición	Que el usuario esté registrado.
Secuencia principal	1. El usuario pulsa sobre el botón de nueva sugerencia. 2. El sistema mostrará una nueva ventana con unos inputs a rellenar.. 3. El usuario rellenará el formulario. 4. El usuario pulsa en enviar formulario.
Salida	El sistema volverá a la ventana anterior enseñando un mensaje de confirmación.

Tabla 19. Caso de uso: formulario sugerencia

Revisar solicitud	
Actores	Usuario registrado.
Descripción	Es la vista donde se encuentran las solicitudes de nuevo producto. El usuario puede revisar las solicitudes y aprobarlas si le parece correcta. Cuando una solicitud se ha aprobado por más de 25 usuarios, y en ese momento únicamente un usuario o ninguno ha rechazado la solicitud, esta se introduce en la base de datos.
Precondición	Que el usuario esté registrado.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de revisión de solicitudes. 2. Se despliega una nueva ventana con una lista de solicitudes. 3. El usuario selecciona una solicitud y la revisa. 4. El usuario pulsa sobre validar solicitud o rechazarla.
Salida	El sistema volverá a la página anterior enseñando un mensaje de confirmación.

Tabla 20. Caso de uso: revisar solicitud

Revisar corrección	
Actores	Usuario registrado.
Descripción	Es la vista donde se encuentran las solicitudes de corrección. El usuario puede revisar las solicitudes y aprobarlas si le parece correcta. Cuando una solicitud se ha aprobado por más de 25 usuarios, y en ese momento únicamente un usuario o ninguno ha rechazado la solicitud, esta se introduce en la base de datos.
Precondición	Que el usuario esté registrado.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de revisión de solicitudes. 2. Se despliega una nueva ventana con una lista de correcciones. 3. El usuario selecciona una corrección y la revisa. 4. El usuario pulsa sobre validar solicitud o denegarla.
Salida	El sistema volverá a la página anterior enseñando un mensaje de confirmación.

Tabla 21. Caso de uso: revisar corrección

Activar notificaciones	
Actores	Usuario registrado.
Descripción	Se pueden activar notificaciones de la aplicación. Estas se muestran en forma de pop up cuando se envían.
Precondición	Que el usuario esté registrado.
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de notificaciones. 2. Se despliega unos botones indicando que notificación activar. 3. El usuario pulsa sobre uno de los botones.
Salida	El botón activado permanecerá en verde hasta que se vuelva a pulsar.

Tabla 22. Caso de uso: activar notificaciones

Modificar datos personales	
Actores	Usuario registrado.
Descripción	En esta sección se pueden modificar los datos personales: el nombre de usuario, el correo electrónico y la imagen del perfil.
Precondición	Que el usuario esté registrado.
Secuencia principal	<ol style="list-style-type: none">1. El usuario pulsa sobre el botón de ajustes.2. Se despliega un formulario con los datos actuales del usuario.3. El usuario edita el campo que le parece oportuno.
Salida	El sistema guardará los cambios automáticamente y mostrará un mensaje de confirmación.

Tabla 23. Caso de uso: modificar datos personales

3.1.3 Requisitos no funcionales

Los requisitos no funcionales para la aplicación se enumeran así:

- Todos los ítems son públicos, es decir, no es necesario estar registrado en la aplicación para visualizarlos.
- Se garantiza el acceso desde cualquier navegador web y desde las aplicaciones para Android e iOS, además estará activo las 24 horas al día.
- La interfaz debe ser sencilla e intuitiva para todos los públicos, incluyendo un diseño responsive en el caso de la página web, además se permite el cambio entre un fondo oscuro y uno claro.
- En el caso de que el usuario decida registrarse, se recoge la cantidad mínima de datos del usuario para su correcto funcionamiento.
- La carga inicial de la aplicación no debe de ser mayor a cuatro segundos. En la medida de lo posible, incluir un estado de carga para alertar al usuario de que se está cargando la información.

3.2 Análisis del producto

A continuación, se realizará un estudio de varios análisis imprescindibles para que la aplicación obtenga la calidad deseada.

3.2.1 Eficiencia algorítmica

En este apartado se expone los aspectos de eficiencia que debe de tener una aplicación como la nuestra. Recordemos que estamos ante una aplicación donde la gran cantidad de ítems almacenados en la base de datos es muy frecuente. Estos deben de mostrarse al usuario de forma prácticamente instantánea y puede llegar a suponer muchos problemas si no se trata con cautela.

Se enumeran posibles errores que se deben de evitar en la carga de productos cuando se utiliza la aplicación y su posterior solución.

- Cargar todos los productos cuando se inicia por primera vez la aplicación.

Este error puede ser un gran problema, ya que la aplicación tardará demasiados segundos en cargarse y puede saturar el navegador o el dispositivo móvil, así como drenar drásticamente la batería del mismo. Para solucionar esto, se deben de cargar únicamente los productos que el usuario ve en pantalla, es decir, se podría crear un algoritmo que cargase los productos a medida que el usuario navega por la página.

- Cargar de forma completa una tabla de información.

Cuando se genera una tabla muy grande donde se almacena mucha información, se debe crear distintas páginas para avanzar con los productos que se quieren mostrar. Para lograr esto, se puede hacer uso de un algoritmo de paginación [8] que haga más eficiente la carga de datos

3.2.2 Análisis del marco legal y ético

En una aplicación donde se involucra la creación de cuentas de usuario es muy importante garantizar el correcto cumplimiento de la ley de protección de datos personales. En concreto, prestaremos atención a la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales [9].

Se implantarán las medidas oportunas de las siguientes formas:

- Los datos que se recojan serán los mínimos para el correcto funcionamiento de la aplicación.

- Se utilizarán estos datos solo con el fin para los que se han solicitado y nunca se compartirán con terceros.
- Al registrarse, se informará al usuario de que datos van a ser almacenados y con qué fin.
- Los usuarios dispondrán en todo momento de una forma de contacto para solicitar la eliminación de su cuenta, incluyendo todos sus datos.

3.2.3 Análisis de riesgos

Para este proyecto se analizan dos tipos de riesgos, los riesgos en la realización del software y los riesgos de aceptación de los usuarios. Se exponen dos tablas para medir el impacto causado (tabla 24) y los niveles de probabilidad de que estos ocurran (tabla 25). Cabe mencionar que este análisis está explicado de forma detallada en el libro PMBOK [10].

Niveles de probabilidad	
Descriptor	Descripción
Improbable	El riesgo ocurrirá únicamente de forma excepcional.
Posible	El riesgo puede ocurrir en algún momento.
Bastante probable	Se espera que el riesgo ocurra de forma considerable.

Tabla 24. Niveles de probabilidad de riesgos

Niveles de impacto	
Descriptor	Descripción
Menor	El impacto sería bajo y traería pocas consecuencias.
Moderado	Es un impacto mediano, traería algunas consecuencias.
Mayor	El impacto más importante, tiene un gran efecto negativo.

Tabla 25. Niveles de impacto

- Riesgos del desarrollo software

Riesgo	Probabilidad	Impacto	Mitigación
Ambigüedad en la especificación de requisitos.	Posible	Moderado	Planificar más tiempo a la fase de especificación y realizar pruebas ficticias con el cliente.
Diseño de interfaces incorrecto.	Bastante probable	Moderado	Hacer encuestas con una muestra acotada de posibles usuarios. Tener claro los requisitos funcionales.
Desconocimientos de la tecnología seleccionada.	Posible	Mayor	Hacer un curso software con antelación. Posibilidad de obtener un mentor.
Componente software muy complejo.	Improbable	Mayor	Utilizar un componente de alguna librería de libre acceso, desconociendo cómo funciona internamente. O en la medida de lo posible, proponer otro componente con la misma funcionalidad.
Se añaden requisitos extra.	Posible	Menor	Dependiendo de la carga final de trabajo, añadir o no los requisitos extra a la planificación. En caso de una carga final de trabajo alta, no aceptar los nuevos requisitos.

Tabla 26. Riesgos del desarrollo software

- Riesgos en la aceptación de usuarios

Riesgo	Probabilidad	Impacto	Mitigación
Carga lenta de la aplicación.	Posible	Moderado	Cargar menos productos en cada página. Añadir indicadores de carga.
Poco intuitivo.	Posible	Mayor	Realizar pruebas con usuarios y obtener un feedback en forma de cuestionarios.
Interfaz no adaptada a la mayoría de dispositivos.	Posible	Mayor	Realizar tests en un gran abanico de emuladores para reducir el riesgo.
Errores en las funcionalidades	Improbable	Mayor	Crear pruebas unitarias.

Tabla 27. Riesgos en la aceptación de usuarios.

3.3 Plan de trabajo

A continuación se expone un diagrama de Gantt (figura 12) con el plan de trabajo estudiado.

El diagrama representa la cantidad de tiempo que se va a dedicar a cada apartado del proyecto. Dado que el proyecto se va a realizar por una sola persona, cada sección del diagrama está ligada a la anterior de forma secuencial, es decir, no existe una tarea que esté en paralelo con otra. Sin embargo, el apartado de desarrollo tiene un periodo de tiempo que se solapa con la maquetación, esto es porque según la metodología escogida, cada funcionalidad tiene que ser completa en todos sus aspectos, así se conseguiría un mínimo producto viable.

Seguidamente, se detalla cada una de las frases del diagrama:

- **Análisis.** En esta fase se estudia el comportamiento y los requisitos de la aplicación. Se hace una comparativa con las aplicaciones similares y se aprende de ellas los errores y las buenas prácticas.
- **Diseño.** Esta etapa engloba el diseño de prototipos en forma de mockups y el diseño que va a tener la base de datos.
- **Desarrollo.** Esta fase es la más larga, puesto que se desarrolla toda la aplicación. Primeramente, se construye y se crean todos los ítems de la base de datos, en nuestro caso se almacenan los productos y sus fotografías. Posteriormente, se programa la parte lógica y la parte de maquetación, es decir, la parte visual que verán los usuarios.
- **Pruebas.** En esta última etapa se prueba la aplicación final con una muestra de usuarios que posteriormente realizarán una encuesta para valorar la misma. Y luego se efectuarán las mejoras y correcciones pertinentes.

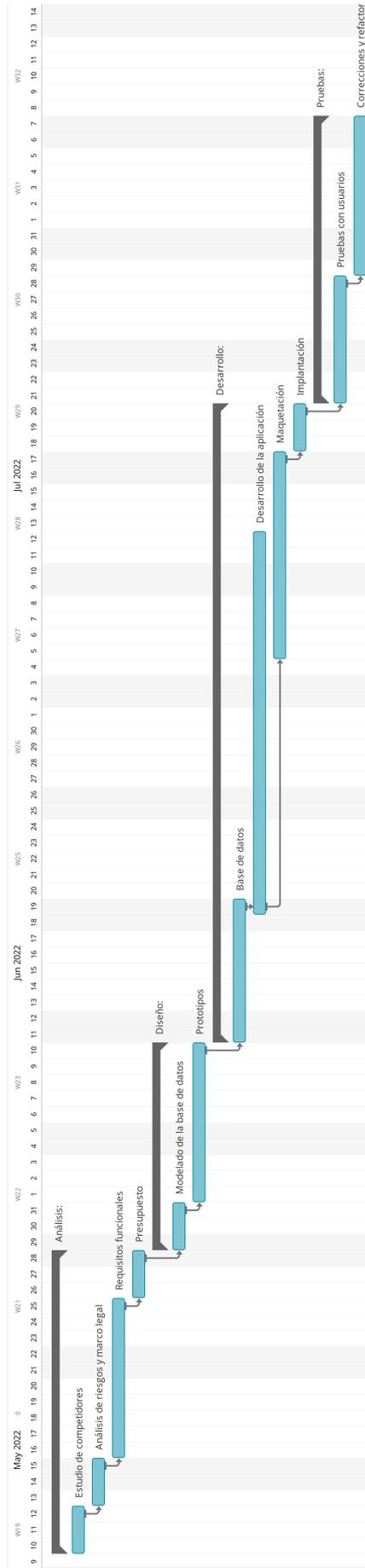


Figura 12. Diagrama de Gantt



3.4 Presupuesto

Para definir el presupuesto del proyecto se van a evaluar dos aspectos. El primero es un estudio del material necesario tanto software como hardware. Y el segundo se centra en el coste humano de desarrollar toda la aplicación.

- **Costes del material**

Este proyecto se basa en una aplicación híbrida, es decir, existe una aplicación para Android, otra para iOS y otra para los navegadores web. Hay que tener en cuenta que, para poder compilar una aplicación en iOS se necesita un ordenador de Apple, es decir, un Mac. Con él, se podría crear emuladores de los distintos iPhone del mercado. Además, este ordenador también valdría para emular los distintos Android que se necesiten.

Por otro lado, la base de datos estará alojada en Firebase, el cual tiene un coste de 0 euros con la versión de prueba y esta ascendería proporcionalmente respecto a las consultas necesarias.

Herramientas sw y hw	Coste
Ordenador Macbook	1.600 €
Cloud de google (firebase, autenticación y hosting)	Desde 0 € hasta 20 €
Publicar app en Play Store (Android)	25 €
Publicar app en App Store (iOS)	100 €
TOTAL:	1.745 €

Tabla 28. Costes materiales

- **Costes humanos**

En lo que respecta a los costes humanos, se estima un salario medio neto mensual para cada trabajador dependiendo de su especialidad. Como el proyecto se realiza únicamente por una persona, esta adoptará los distintos roles dependiendo de las fases del proyecto.

Los sueldos obtenidos (tabla 29) son una estimación extraída por la página de Jobted ¹

¹ <https://www.jobted.es/salario/programador>

Salario medio neto mensual	
Empleo	Sueldo [€/mes]
Analista	1.600 €
Diseñador informático	1.560 €
Programador	1.610 €

Tabla 29. Salario medio neto mensual

Según la planificación del apartado anterior, existen cuatro fases. Se asigna un empleo para cada fase y se calcula el coste por el tiempo empleado en cada una de ellas:

Fase	Empleo	Tiempo	Coste
Análisis	Analista	18 días	1310 €
Diseño	Diseñador	12 días	851 €
Desarrollo	Programador	39 días	2.855 €
Pruebas	Programador	17 días	1.244 €
			6.260 €

Tabla 30. Costes de recursos humanos

- **Resumen**

La suma de los costes asciende a **8.005 €**.

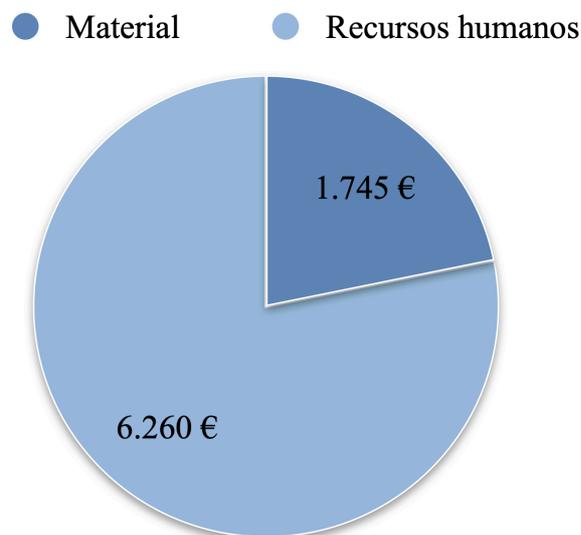


Figura 13. Gráfica de costes totales

4. Diseño de la solución

En este capítulo se define la arquitectura del sistema y el diseño de la aplicación, que esta última se compone de dos partes, el modelo de datos y los prototipos.

4.1 Arquitectura del sistema

La arquitectura de esta aplicación hace uso de la tecnología Serverless [11], es decir, una arquitectura sin servidor. Esta tecnología es una solución rápida y eficaz para ejecutar la aplicación sin necesidad de crear una infraestructura, ya que esta es proporcionada por un servicio externo ubicado en la nube, el cual es responsable de ejecutar el código de la aplicación, asignando dinámicamente los recursos necesarios para ello. Esta modalidad nos permite tener una serie de ventajas:

- No administraremos una infraestructura de servidor.
- Escalabilidad. Se obtiene una flexibilidad para poder escalar estos servicios según la demanda.
- Ahorro en los costos. Ya que está tecnología permite pagar únicamente por los recursos utilizados.

En nuestro caso, utilizaremos Firebase que pertenece a la infraestructura de Google Cloud para abastecernos de las siguientes funciones:

- Base de datos
- Almacenamiento de datos (imágenes)
- Servicios de autenticación
- Hosting

Junto a la tecnología de Serverless, se hace uso de la tecnología SPA [12] (Single-page application), la cual permite cargar únicamente una página web simultáneamente, y es el navegador web quien se encarga de cargar los futuros renderizados.



Por otro lado, en la parte front, se utiliza una arquitectura de componentes, esta se basa en englobar funcionalidades y diseños en componentes individuales. Se diseñan con el objetivo de cumplir un propósito único pero con características moldeables y personalizables. Por ejemplo, un componente muy básico sería un botón, el cual puede estar diseñado para usarse prácticamente en toda la aplicación con características personalizables como el color y el tamaño.

Los principales beneficios de utilizar los componentes son los siguientes:

- Independiente. Los componentes engloban la funcionalidad y el diseño de forma que no necesiten de otras funcionalidades o componentes para su correcto funcionamiento.
- Reutilizable. Se diseñan para usarse en la mayoría de escenarios y de la forma más genérica posible.
- Apilable. Existe la posibilidad de utilizar componentes dentro de otros componentes con el fin de aumentar sus funcionalidades.

En cuanto a las API que se emplean en esta aplicación son las mencionadas anteriormente, siendo todas ellas de la plataforma Firebase. Para que la aplicación las utilice, esta tiene una capa de servicios, que se crea dentro del proyecto y tiene como función englobar todos los métodos necesarios para hacer uso de Firebase. A continuación se muestra el esquema.

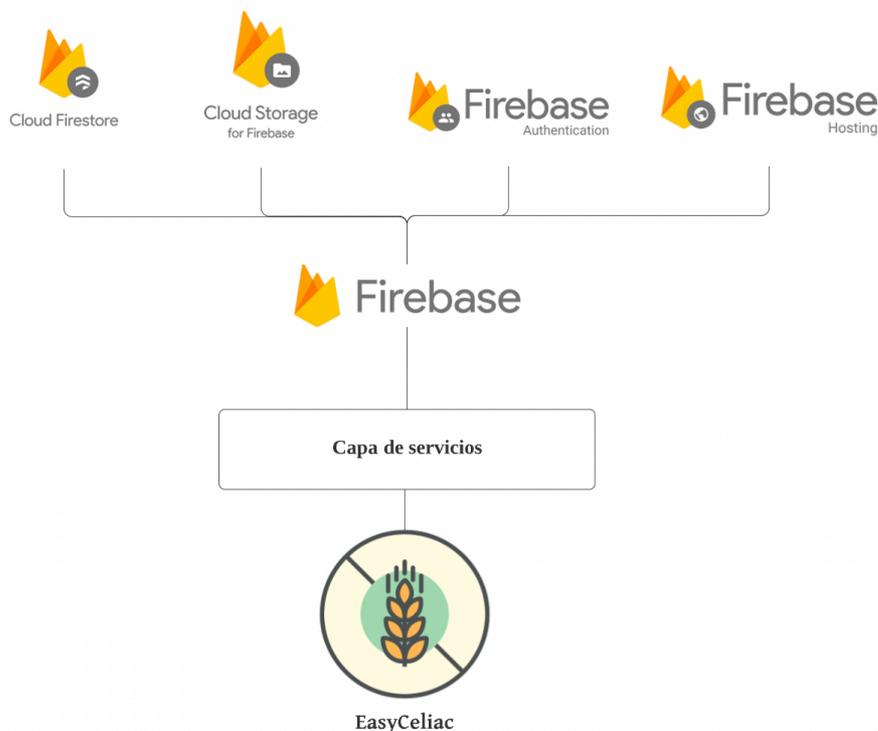


Figura 14. Arquitectura de la aplicación

Para finalizar este apartado se expone un esquema multiplataforma, donde se muestra en que dispositivos va a estar disponible la aplicación y de que forma el usuario podrá hacer uso de ella.

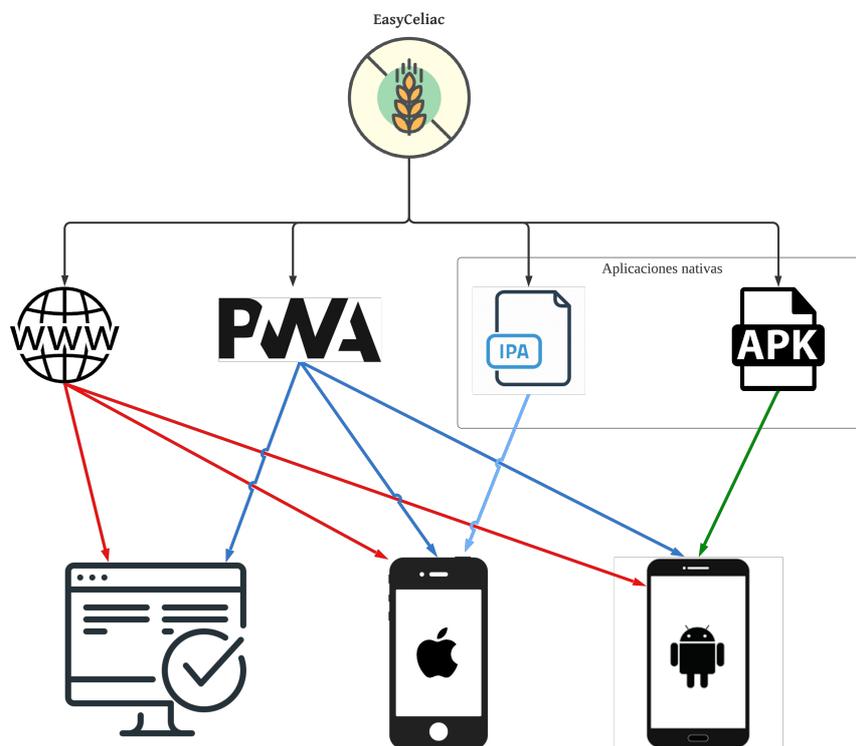


Figura 15. Esquema multiplataforma

La arquitectura de la aplicación está pensada para una aplicación híbrida, esto significa que la aplicación es capaz de ejecutarse en todos los dispositivos. Es más, los usuarios pueden hacer uso de ella de diferentes formas como veremos a continuación:

En la figura 15 vemos como nuestra aplicación será lanzada de cuatro formas posibles, estos son:

- **WWW.** Hace referencia a una página web, el cual es capaz de ejecutarse en todos los dispositivos que tengan un navegador, donde se incluye: ordenador, iPhone y Android.
- **PWA.** Una aplicación web progresiva, PWA [13], como veremos en el próximo capítulo, es un tipo de aplicación software que se instala a través de la web. Está destinado a instalarse tanto en el ordenador, en iPhone y en Android. Y una de las principales ventajas, es que parece una aplicación nativa, se instala en el escritorio o en el cajón de aplicaciones como si una aplicación nativa se tratase, pero con la peculiaridad que no deja de ser un software construido con Html y JavaScript, pero con un “service worker”, el cual se encarga de hacer las peticiones web que la aplicación necesite.

- IPA. IOS AppStore Package, es un formato de aplicaciones utilizado por Apple. El cual se usará para instalar la aplicación de forma nativa en iPhone.
- APK. Android Application Package, es un formato de aplicaciones empleado por Google, y se utilizará para instalar la aplicación de forma nativa en Android.

4.2 Diseño detallado

En este apartado se van a exponer dos tipos de diseños, el modelo de datos, que hace referencia a cómo son los objetos y cómo se relacionan entre ellos. Y el diseño de la parte front, es decir, se expondrán los prototipos de las diferentes vistas de la aplicación.

4.2.1 Modelo de datos

Los datos se dividen en dos APIs distintas. Firestore almacena el texto plano de los objetos y Cloud Storage almacena las imágenes.

A continuación se detalla el modelo de los objetos.

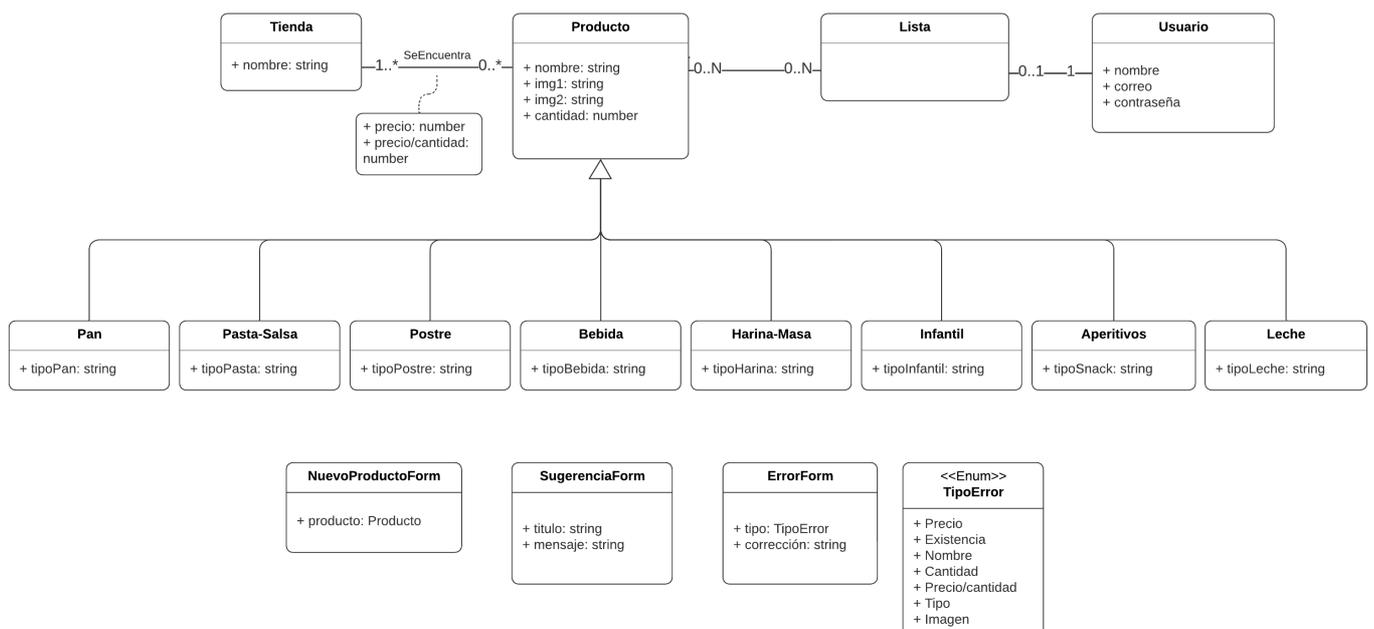


Figura 16. Diagrama de clases del modelo de datos

Tal como se observa en el diagrama anterior, se puede apreciar la base del modelo de datos, que es el producto. El producto hace referencia a un alimento dentro de nuestra aplicación y este debe de ser una de entre ocho diferentes categorías, cada una está diferenciada por el tipo de producto que contiene, por ejemplo, "tipoPan" puede ser: pan rallado, pan de molde, pan de barra, etc. Se ha decidido crear una especialización porque durante el estudio del proyecto se ha llegado a la conclusión que estas ocho categorías abarcan todos los productos del mercado, y es muy poco probable que esta lista se modifique con el tiempo. Cada producto se encuentra en al menos una tienda, creando así dos atributos de enlace, el precio y el precio por cantidad, los cuales son fruto de la relación producto y tienda, ya que el mismo producto puede tener precios diferentes dependiendo de la tienda donde se encuentre. Además, el producto contiene dos atributos llamados img1 e img2, estos hacen referencia a la url donde son alojados por el Storage de Firebase.

Por otro lado, se encuentra el usuario, el cual puede tener una lista de productos junto a la suma total del precio respecto de la tienda donde pertenece el producto. Si no existiera al menos un producto de una determinada tienda, su total sería cero.

Finalmente, se encuentran los formularios, donde destaca el formulario de nuevo producto. Este formulario contiene un producto con las mismas características del producto anteriormente mencionado.

Gracias a estas características, se estaría construyendo una estructura capaz de soportar cambios en versiones futuras, puesto que se podrán añadir más tiendas y más subcategorías a la aplicación si fuera necesario.

4.2.2 Prototipos

Antes de visualizar los prototipos cabe mencionar que, se ha decidido que el nombre final de la aplicación sea EasyCeliac. Este nombre, cuya traducción al inglés es "celíaco fácil", intenta dar una referencia a la ayuda de los celíacos haciéndoles la vida más sencilla.

Ahora sí, se exponen los prototipos por orden de profundidad, es decir, primeramente se enseñan las cuatro vistas principales que se pueden acceder con un simple toque. Estas son, inicio, buscar, carrito y perfil. Y posteriormente se muestran las vistas que se pueden acceder dentro de cada una de ellas.





Figura 17. Prototipo: inicio

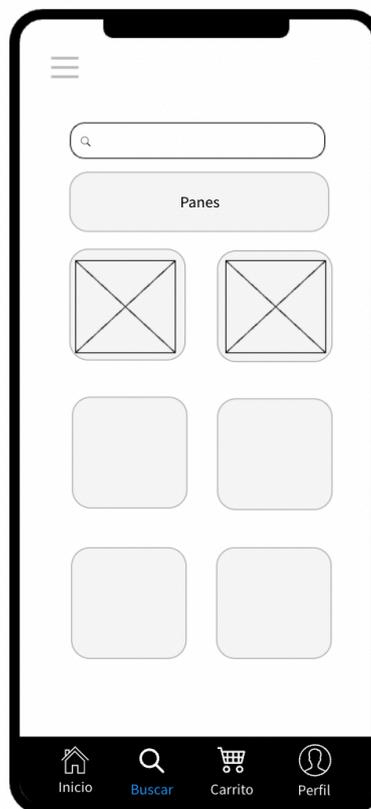


Figura 18. Prototipo: buscar

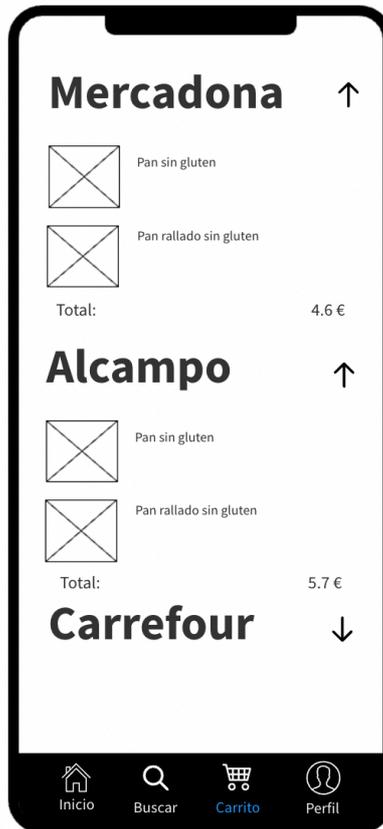


Figura 19. Prototipo: carrito

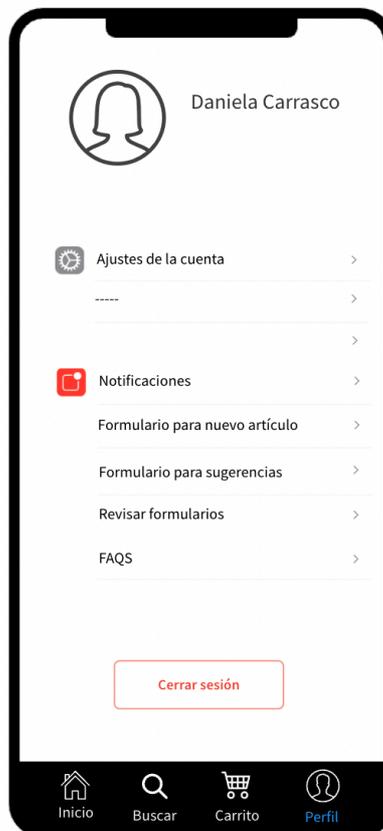


Figura 20. Prototipo: perfil

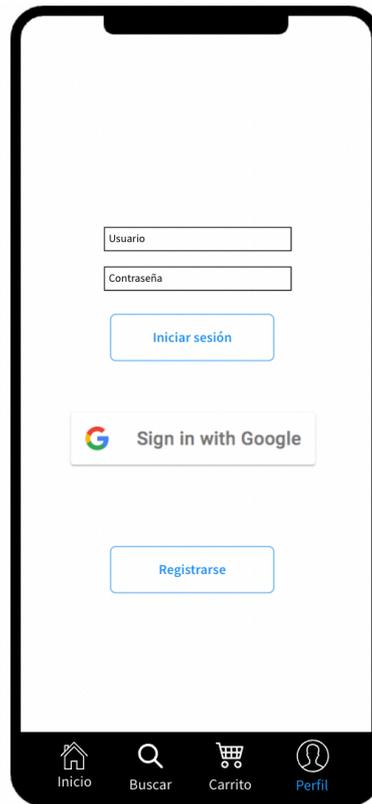


Figura 21. Prototipo: perfil - iniciar sesión

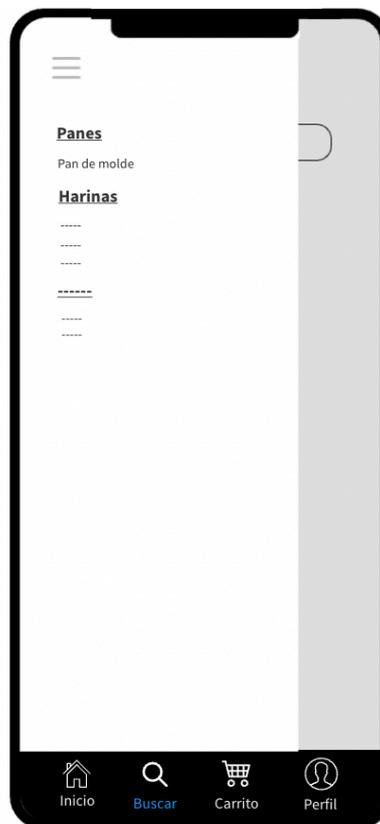


Figura 22. Prototipo: perfil - menú lateral

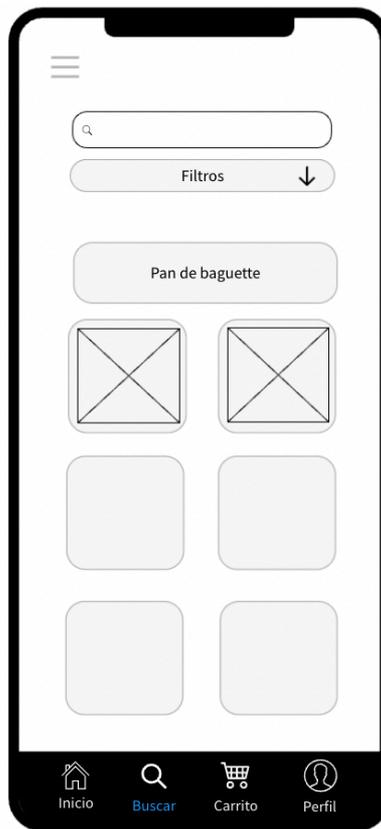


Figura 23. Prototipo: buscar - subcategoría



Figura 24. Prototipo: buscar - detalles



Figura 25. Prototipo: buscar - detalles - corregir error

Por último, se muestra un solo prototipo de la vista web a modo de ejemplo para no agrandar esta sección, ya que, tienen las mismas funcionalidades pero con diferente formato visual.

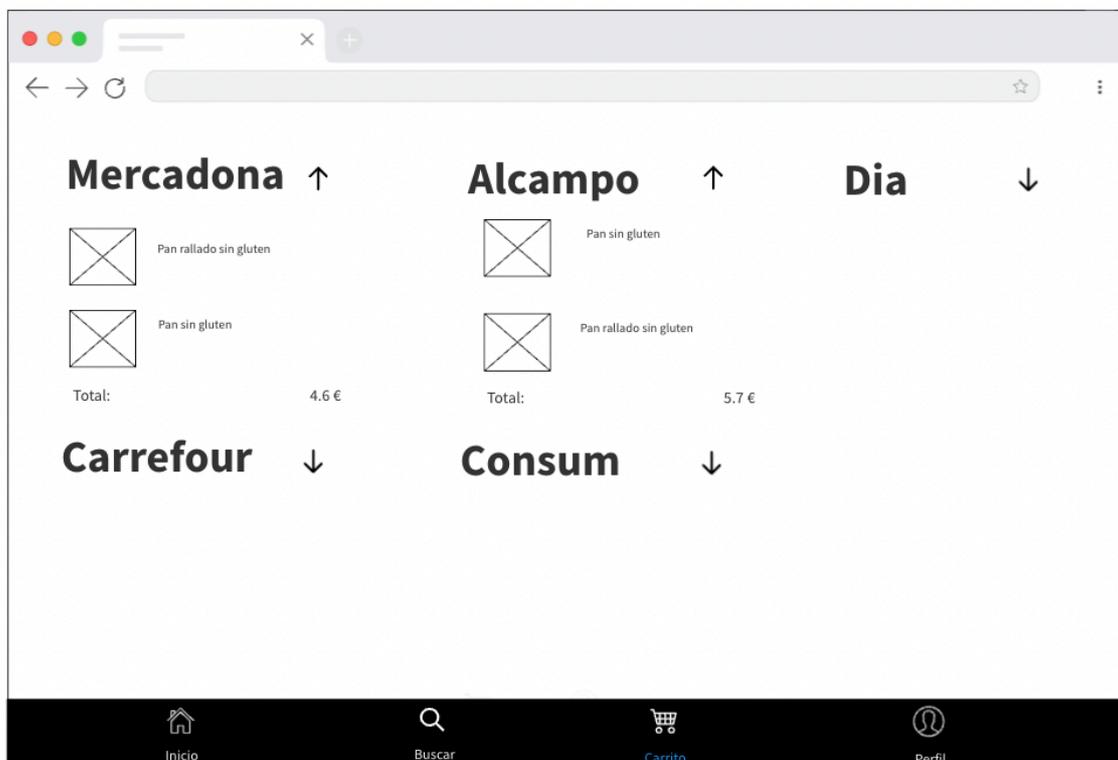


Figura 26. Prototipo web: carrito

5. Implementación y despliegue

Teniendo un análisis completo, una arquitectura estudiada y un diseño realizado, se procede a implementar y desarrollar EasyCeliac. En este capítulo nos adentraremos en detalle en la solución ejecutada, por un lado, se expondrán las tecnologías utilizadas junto a su justificación, y hablaremos del desarrollo tanto de base de datos como la parte frontal y lógica de la aplicación. Por otro lado, se efectuará el despliegue de acuerdo al esquema estudiado en el capítulo anterior.

5.1 Tecnología utilizada

A continuación se describen las principales tecnologías y herramientas empleadas para la realización de este proyecto.

- **Ionic**



Figura 27. Logo de Ionic

Ionic [14] es un framework de código abierto, formado por un conjunto de herramientas que permite la creación de aplicaciones web y aplicaciones nativas multiplataforma. Esta tecnología nos ofrece unos componentes preestablecidos para crear vistas móviles de alta calidad. Estas funcionan en dispositivos iOS y Android de forma nativa y cambian su forma dependiendo del sistema que lo ejecute. Gracias a este framework es posible generar las tres aplicaciones prácticamente con el mismo código, sin embargo, en ocasiones hay que hacer pequeñas modificaciones para obtener los resultados esperados.

Se ha optado por esta tecnología en particular porque, como se explicó en los objetivos, la idea de la aplicación es llegar a toda la población posible, y gracias a Ionic, todos los usuarios sin importar el sistema de su dispositivo podrán disfrutar de ella.

Como veremos a continuación, ionic depende de un framework para su funcionamiento, en este proyecto se ha elegido Angular para su integración, aunque las otras posibilidades de Ionic son, React y Vue.

- **Capacitor**



Figura 28. Logo de Capacitor

Capacitor [15] nace de la comunidad de Ionic y es el motor que convierte el proyecto de Ionic en las aplicaciones nativas, es decir, es el encargado de compilar el código del proyecto, en código nativo para cada sistema. Además, es el encargado de acceder a funcionalidades de los dispositivos móviles, por ejemplo, Capacitor nos brinda la oportunidad de acceder a la cámara del móvil. Cabe destacar que actualmente, aunque es propiedad de Ionic, se puede considerar de código abierto, lo que permite que la comunidad mejore sus prestaciones y que capacitor sea gratuito.

- **Angular**



Figura 29. Logo de Angular

Angular [16] es un framework de código abierto creado por Google, el cual se combina en este proyecto con Ionic. Angular aporta al conjunto el lenguaje de programación TypeScript, y este es el lenguaje principal del proyecto. TypeScript es un superconjunto sintáctico de JavaScript, aportando nueva sintaxis y un tipado en sus variables.

Ionic también ofrece la posibilidad de no elegir uno de los framework mencionados anteriormente para su funcionamiento, ya que se podría utilizar JavaScript directamente en él. Sin embargo, para este proyecto se ha decidido utilizar Angular porque aporta muchas facilidades a la hora de elaborar el código fuente, como la velocidad y la productividad al ofrecer diferentes plantillas de programación. Además, nos aporta la arquitectura de componentes necesaria, y la característica SPA. Las cuales se explicaron en el apartado de arquitectura del sistema.

Por otro lado, Angular trabaja con los lenguajes, HTML y SCSS para la elaboración de componentes visuales. Cabe mencionar que las versiones utilizadas en este proyecto son las más recientes hasta ahora: Ionic 6 y Angular 14.

Se ha decidido utilizar Angular junto con TypeScript como lenguaje de programación principal del proyecto por varias razones:

1. Angular aporta el lenguaje de TypeScript, que en nuestro caso, es una gran ventaja frente a JavaScript.
2. Nos ofrece la posibilidad de crear módulos de componentes y la creación de servicios.
3. La posibilidad de implementar “lazy loading” en el routing fácilmente. Esto nos aporta una gran ventaja en la eficiencia del programa, puesto que se encarga de que la web únicamente cargue la ventana donde se encuentra el usuario, y no inicialice toda la aplicación al completo.
4. Por último, Angular tiene un soporte de Google y una comunidad detrás muy grande. Que se traduce en mucha documentación y problemas resueltos por otros desarrolladores.

- **Firebase**



Figura 30. Logo de Firebase

Firebase [17] es una plataforma ubicada en la nube propiedad de Google. Cuenta con un gran abanico de ventajas como la rápida creación de una base de datos sin escribir una lógica muy compleja. Además, cuenta con la ventaja de escalabilidad mencionada en el apartado de arquitectura del sistema.

De esta plataforma se utilizan las siguientes API mencionadas en el capítulo anterior: Firestore, donde se almacena los datos de texto; Cloud Storage, donde se almacenan las imágenes; Firebase Authentication, la cual nos brinda la autenticación tanto de Google como la original con email y contraseña; y por último Firebase Hosting, que proporcionará las herramientas necesarias para subir la aplicación web a la red en el apartado de despliegue que veremos más adelante.

Se ha implementado esta plataforma como principal base de datos por varios motivos. La ausencia de crear una base de datos compleja y programar la lógica ha sido uno de los motivos principales de utilizar esta tecnología. Por otro lado, ofrece un diseño y una facilidad de uso bastante atractivo, y para las necesidades de la aplicación ha ayudado en centralizar todas las API en una sola plataforma. Como se mencionaba en la tecnología anterior, Firebase también tiene una gran comunidad detrás y una gran compatibilidad con Angular, ya que son propiedad de la misma empresa.

- **Visual Studio Code**



Figura 31. Logo de VS code

Visual Studio Code [18] es un editor de código fuente desarrollado por Microsoft. Se ha decidido utilizar este editor en el proyecto por las siguientes características:

- Se trata de un editor muy simple y liviano, además permite mucha personalización.
- Tiene una extensa lista de atajos de teclado muy útiles, incorpora un cursor múltiple y tiene un resaltado de palabras coincidentes.
- Se le puede añadir plugins para aumentar sus funcionalidades.

- **GitHub**



Figura 32. Logo de GitHub

GitHub es una plataforma de alojamiento de código en la nube, donde se sube el código fuente de esta aplicación. GitHub incorpora el protocolo de Git, el cual aporta un control de versiones y la posibilidad de crear distintas ramas según las necesidades del programa.

Además, en este proyecto se utiliza GitHub Desktop. Se trata de una aplicación de escritorio desarrollada por GitHub, la cual aporta una interfaz a los comandos de Git, haciendo mucho más sencilla su utilización. Por otro lado, esta herramienta aporta la posibilidad de ver el historial de cambios desde la propia aplicación, esta funcionalidad se emplea para revisar un fragmento de código anterior en caso de que el actual esté dando problemas, con el fin de encontrar una solución más rápida.

- **Xcode**



Figura 33. Logo de Xcode

Xcode es un entorno de desarrollo para macOS que contiene un conjunto de herramientas creadas por Apple destinadas al desarrollo de software. Esta herramienta es capaz de generar emuladores de dispositivos de Apple.

En nuestro proyecto se utiliza este entorno para compilar el proyecto nativo para iOS, en su lenguaje de programación Swift [19]. Además, se emula un iPhone con el fin de probar la aplicación antes de instalarla en un dispositivo real.

- **Android Studio**



Figura 34. Logo de Android Studio

Android Studio es el entorno de desarrollo oficial para la plataforma Android desarrollada por Google. Está disponible para Windows, macOS y Linux. Esta herramienta es capaz de generar emuladores de dispositivos Android.

Como en la herramienta anterior, Android Studio se emplea para compilar el proyecto nativo en Android con el lenguaje Kotlin [20] y emular un dispositivo Android con el objetivo de ejecutar la aplicación en un dispositivo nativo.

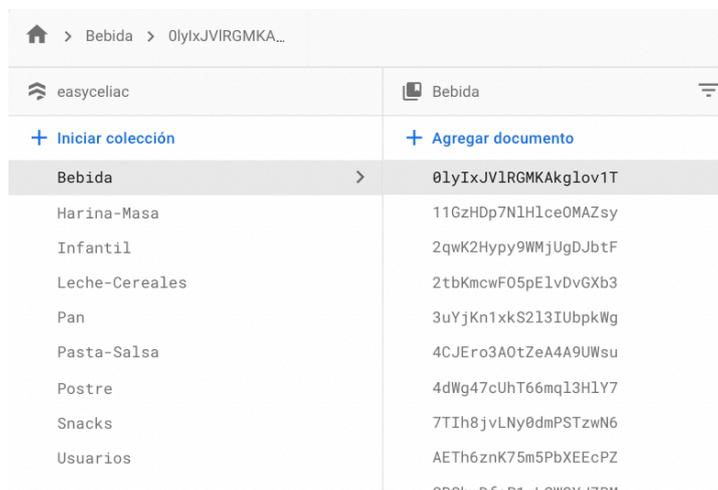
5.2 Desarrollo de la base de datos

El desarrollo de la base de datos se ha construido en dos fases, primeramente, se ha recopilado una cantidad generosa de productos sin gluten de los cinco supermercados y se ha ordenando según su categoría principal. Para que la primera versión de la aplicación pueda ser lanzada y tenga las funcionalidades deseadas, se ha llegado a recopilar aproximadamente 500 productos en total. Esta base de ítems será la necesaria para la primera puesta en marcha, y con el tiempo pueda ir creciendo.

Después de la recopilación de datos, se ha inicializado la base de datos en Firebase y se han subido las imágenes al Storage. Con las imágenes subidas a la nube, se han generado los productos con sus respectivos atributos, incluyendo los enlaces a sus respectivas imágenes. Para hacer más eficiente la carga de datos en la aplicación, se ha decidido almacenar los productos dependiendo de su categoría (figura 35).

Por último, se ha inicializado esta base de datos en la aplicación de ionic. Para hacer lograr esto se han seguido los siguientes pasos:

1. Instalar el SDK de Firebase con Node.js dentro del proyecto de Ionic.
2. Insertar la inicialización de Firebase en el archivo modules del proyecto de Ionic, incluyendo la inicialización de la autenticación por contraseña y por Google.
3. Crear los servicios donde se incluirán las llamadas a la base de datos y la autenticación de usuarios. Estos servicios serán utilizados por los componentes que requieran leer o escribir de la base de datos o iniciar y cerrar sesión en la aplicación (figura 36).
4. Programar los métodos necesarios para la llamada de los productos y los métodos necesarios para toda la autenticación.



Bebida > 0lyIxJVlRGMKA...	
easyceliac	Bebida
+ Iniciar colección	+ Agregar documento
Bebida >	0lyIxJVlRGMKAkg1ov1T
Harina-Masa	11GzHDp7N1Hlce0MAZsy
Infantil	2qwK2Hypy9WMjUgDJbtF
Leche-Cereales	2tbKmcwF05pE1vDvGXb3
Pan	3uYjKn1xkS213IUbpkWg
Pasta-Salsa	4CJEro3A0tZeA4A9UWsu
Postre	4dWg47cUhT66mq13H1Y7
Snacks	7TIh8jvLNy0dmPSTzwN6
Usuarios	AETH6znK75m5PbXEEcPZ
	0P5...0F...01...0WCV...170M

Figura 35. Almacenamiento de los productos en la base de datos

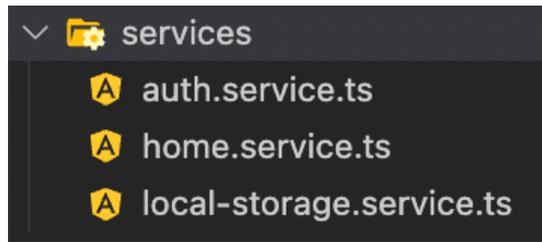


Figura 36. Servicios

Podemos observar en la figura 36 la creación de tres servicios diferentes, las cuales forman la capa de servicios de la aplicación. El primero, “auth.service” se encuentran los métodos encargados de toda la autenticación de la aplicación, los más importantes son: iniciar sesión, registrarse, iniciar sesión con Google y cerrar sesión.

Por otro lado, “home.service” gestiona todos los métodos relacionados con la base de datos de Firebase y Storage. A modo de ejemplo, se encuentran los métodos capaces de obtener los productos por tipo, por tienda, por nombre, entre otros.

Por último, se encuentra “local-storage.service” donde se gestiona la persistencia local, es decir, se guarda en el almacenamiento interno los datos necesarios para el correcto funcionamiento de la aplicación, por ejemplo, se almacena los datos de sesión del usuario para que este no tenga que iniciar sesión cada vez que refresca la aplicación. Este servicio es un tanto complejo, ya que no es lo mismo utilizar el almacenamiento del navegador web, que utilizar el almacenamiento interno de un dispositivo móvil, puesto que se tratan de forma distinta. Para solucionar esto, se ha tenido que instalar y configurar un plugin dentro de Capacitor con el fin de poder obtener acceso al almacenamiento de los diferentes dispositivos y poder crear los respectivos métodos.

5.3 Desarrollo de la parte frontal

En este apartado, se va a explicar la implementación de la parte visual y su lógica. Para empezar, se va a detallar la estructura del proyecto.

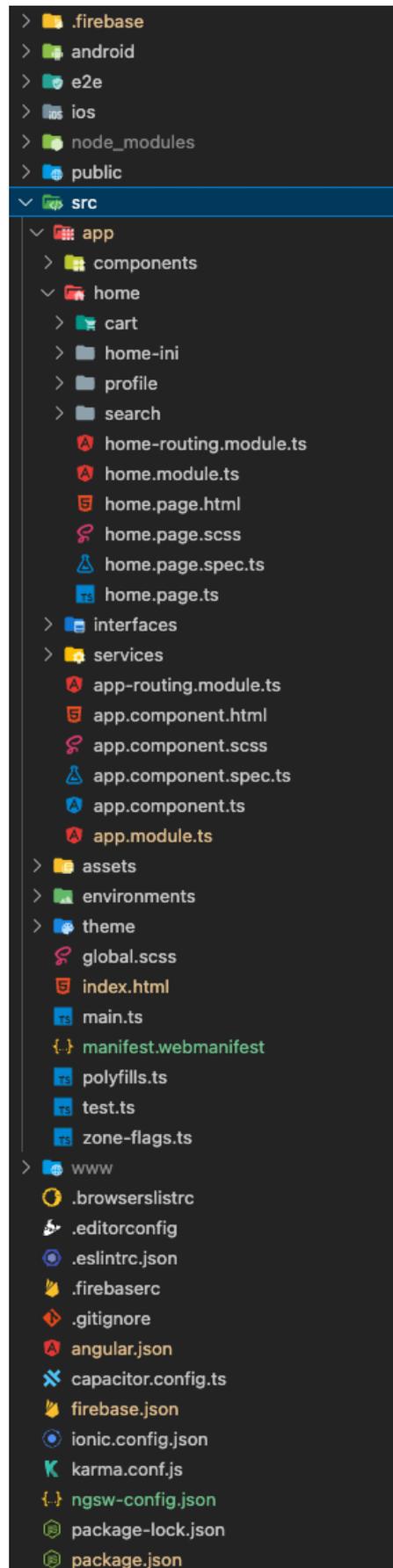


Figura 37. Estructura de los directorios

A continuación, se detalla brevemente el contenido de las carpetas y algunos archivos más relevantes de la figura 37 por orden.

- “android”: en esta carpeta se almacena el proyecto compilado de forma nativa para los dispositivos Android.
- “ios”: al igual que en la carpeta anterior, en esta carpeta se almacena el proyecto compilado de forma nativa para iOS. Ambas carpetas son las que se utilizan en sus respectivos entornos para ejecutar la aplicación y crear sus ejecutables.
- “src”: este es el directorio donde se escribe el código fuente del programa.
- “components”: aquí se engloban los componentes creados específicamente para este proyecto y son reutilizados en varias vistas.
- “home”: como vimos en los prototipos y podemos ver en el anexo, esta aplicación está formada por un menú inferior de cuatro botones distintos: inicio, buscar, carrito y perfil. Estos botones están estructurados en la aplicación por cuatro carpetas en el interior de home y son respectivamente: “home-ini”, “search”, “cart”, “profile”. Entonces, “home” es el componente global donde se construye los cuatro botones y en su archivo “home-routing.module.ts” se crean las rutas lazy-loading a cada componente mencionado. Gracias a esta estructura, la aplicación es capaz de cargar únicamente un módulo simultáneamente, consiguiendo así más velocidad de carga inicial, puesto que no se cargaría la aplicación al completo. Cabe mencionar, que la carpeta home se podría evitar y podríamos crear exactamente lo mismo con los archivos de “app”, con “app-routing.module.ts”, pero se ha decidido englobar este diseño en el componente “home” para tener una estructura más visual y organizada.
- “interfaces”: en esta carpeta se almacenan todas las interfaces de la aplicación. Una interfaz es el mecanismo que utiliza TypeScript para definir los tipos de los objetos como se ha visto en el diagrama de clases.
- “app.module.ts”: se utiliza este archivo para importar todos los elementos que se utilizan en toda la aplicación: componentes, módulos y servicios. Como veremos a continuación, cada módulo contiene este archivo, y funciona exactamente igual que este, pero con la peculiaridad que se importan únicamente los elementos que se utilizan en ese módulo en concreto, evitando así, importar todos los elementos en el “app.module.ts”.
- “assets”: en este directorio se almacenan los iconos y el logo.
- “manifest.webmanifest”: este peculiar archivo contiene metadatos, y se configura para que los navegadores puedan hacer uso de la PWA y logren instalar la aplicación en su sistema.



- “www”: en esta carpeta se compila la aplicación web, contiene los archivos necesarios para que los navegadores puedan ejecutar la página web. Como los navegadores únicamente entienden JavaScript, Html y Css, el proyecto se compila en estos lenguajes.
- “package.json”: por último, en este archivo se almacenan los metadatos de la aplicación y las dependencias de los distintos paquetes, incluyendo la versión de estos.

Por otro lado, la aplicación contiene cuatro módulos principales mencionados anteriormente: “home-ini”, “search”, “cart” y “profile”. Los módulos se caracterizan por tener el archivo “module.ts”. Además, estos contienen: “page.ts” dónde se escribe la lógica en el lenguaje TypeScript, “page.html” donde se escriben las etiquetas Html y “page.scss” que son hojas de estilo. En el caso de que el módulo tenga más vistas que la principal, se crea una carpeta llamada “views” y se añade el archivo “routing.module.ts” para redireccionar las vistas dentro del módulo.

A modo de ejemplo se expone la estructura del módulo “search” (figura 38).

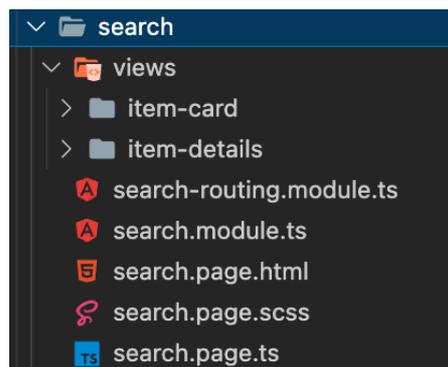


Figura 38. Estructura del módulo search

En el módulo “search” existen varias vistas, una de ellas es la vista de detalles del producto (item-details). En esta vista se utiliza el componente de detalles, el cual muestra un producto en profundidad.

5.4 Despliegue

Para implantar la aplicación se ha seguido el esquema expuesto en el capítulo anterior. A continuación, se explican los pasos que se han seguido para lograr construir todo el esquema.

- **Aplicación web**

Para esta modalidad se ha hecho uso de Firebase Hosting como se ha explicado en anteriores apartados.

1. Primeramente, se instala la herramienta de hosting de Firebase en el proyecto.
2. Luego, se ha compilado en la carpeta “www” el código fuente en el formato de producción, puesto que ya no nos encontramos en el entorno de desarrollo. Y se han corregido posibles errores que este formato puede ocasionar, por ejemplo, en nuestro caso, existía un error de dependencias de uno de los paquetes, y para solucionarlo, se ha tenido que buscar que versiones eran compatibles entre sí sin afectar a todo el proyecto restante, luego se ha actualizado el proyecto con esas versiones y se ha vuelto a compilar.
3. Se inicializa el proyecto en Firebase indicando la carpeta “www” y que el proyecto funciona con la tecnología SPA. Además, se podría cambiar el dominio en el caso de que tuviéramos uno.
4. Se suben los archivos y ya se puede disfrutar de la aplicación web en todos los dispositivos.

- **PWA**

Una vez compilado el proyecto, se puede generar la PWA para que todos los dispositivos puedan instalar la aplicación. Para lograr esto, se siguen los siguientes pasos:

1. Se instala el paquete necesario, el cual añade al proyecto el “service worker” y el archivo manifest.
2. Se configura el archivo manifest con las características de nuestro proyecto, como la caché que utilizará la aplicación progresiva y el lazy loading. Por otro lado, se tienen que elegir varias imágenes que actuarán como el icono de la app, pero estos tienen que tener formatos concretos para que la PWA funcione correctamente, ya que al primer intento se ha observado que si no se especifican las imágenes en formatos concretos, la aplicación progresiva no funciona.



3. Se repite el último paso realizado para la aplicación web, el cual vuelve a subir los archivos a la red, pero esta vez con la aplicación progresiva.

4. Finalmente, se consigue implementar la PWA y aparece un nuevo botón en la url. Para instalarla, únicamente se tendría que apretar ese botón. Se puede observar un ejemplo de cada sistema en la figura 39.

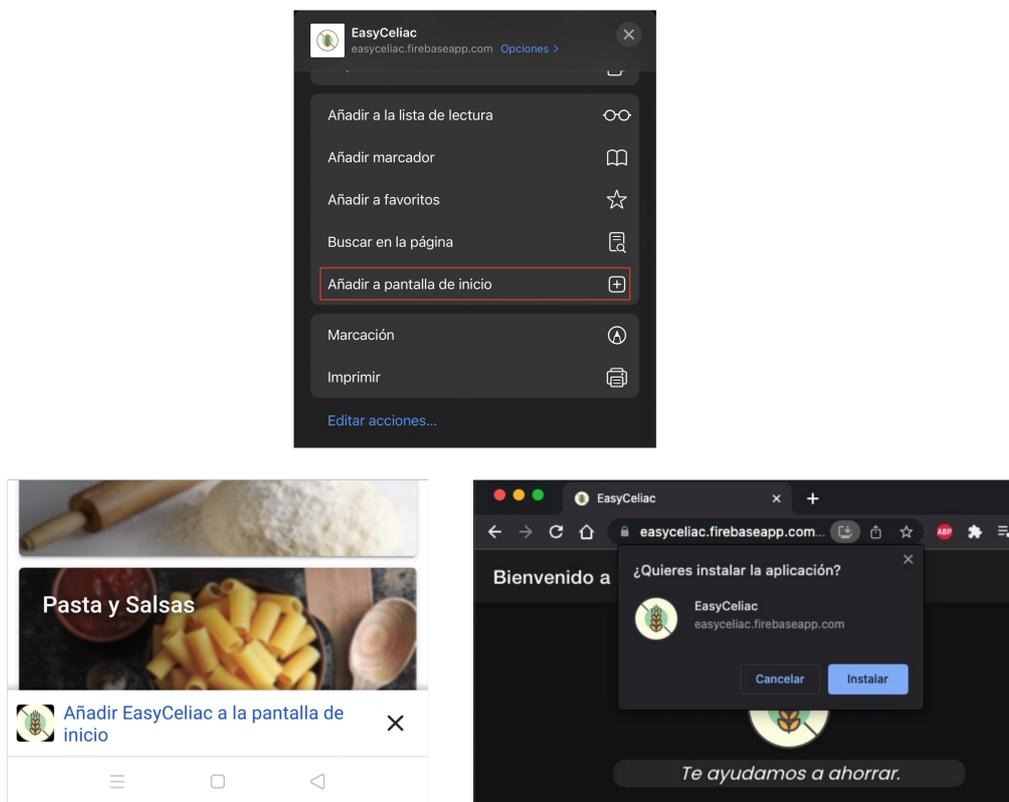


Figura 39. PWA en iOS, Android y navegador web

- **IPA**

Para la creación del archivo ejecutable IPA, se hace uso del entorno Xcode en un ordenador Mac. Se realizan estas acciones para su obtención:

1. Se hace uso de Capacitor para compilar el proyecto en el lenguaje Swift de Apple.
2. Con el proyecto abierto en Xcode, se elige un emulador de un iPhone, donde se instala la aplicación. Posteriormente, se confirma que la aplicación se abre perfectamente y no contiene errores.

3. Para exportar el ejecutable en iOS se tiene que verificar y firmar el código. Esto lo realizamos en el archivo de configuración de la aplicación nativa. Cabe destacar que en Android este proceso no es necesario.

4. Por último, hay dos posibilidades en un iPhone de instalar el ejecutable. El más sencillo es subir la aplicación en la App Store para poder descargarlo e instalarlo. Sin embargo, si se quiere prescindir de subirlo en la App Store, aunque no sea lo habitual, se tiene que instalar un certificado de confianza en el iPhone.

- **APK**

Para la creación del archivo ejecutable APK, se utiliza el entorno Android Studio. Para conseguir este ejecutable se siguen los siguientes pasos:

1. Se hace uso de Capacitor para compilar el proyecto en el lenguaje Kotlin de Android.
2. Con el proyecto abierto en Android Studio, se elige un emulador de un dispositivo Android, donde se instala la aplicación. Posteriormente, se confirma que la aplicación se abre perfectamente y no contiene errores.
3. Por último, se exporta el proyecto en formato APK, el cual se podrá instalar en dispositivos Android sin problemas.



6. Pruebas

La realización de pruebas son muy importantes en un proyecto de esta envergadura, ya que nos da a conocer el estado de la aplicación, si todas las funcionalidades actúan de forma esperada y si los usuarios se sienten satisfechos con ella. Para lograr este propósito se han realizado tres tipos de pruebas, pruebas funcionales, pruebas de carga y pruebas con usuario mediante una encuesta.

6.1 Pruebas funcionales y pruebas de carga

En este apartado se desarrollan pruebas funcionales, las cuales comprueban que el software creado cumple con la función deseada.

Primeramente, analizamos si los casos de uso de la aplicación cumplen con lo preestablecido. A continuación, se exponen los resultados de las pruebas englobadas en los cuatro módulos vistos en el capítulo de análisis.

Niveles posibles para medir el resultado	
Descriptor	Descripción
Problemas graves	El caso de uso presenta problemas con la funcionalidad y se deben corregir.
Problemas leves	El caso de uso presenta leves problemas con la funcionalidad, es posible que no afecte a todos los usuarios.
Problemas de diseño	Pueden ser problemas con el reescalado imágenes, o con el responsive de cierta pantalla.
Correcto	El caso de uso funciona correctamente en todos los aspectos.

Tabla 31. Niveles posibles para medir el resultado

Con los distintos niveles expuestos en la tabla 31, se califican los casos de uso para medir el grado actual de la aplicación en los diferentes dispositivos. Cada caso de uso representa la totalidad de su funcionalidad, incluyendo la parte frontal necesaria para su ejecución final.

Módulo buscar			
Caso de uso	Aplicación web	iOS	Android
Ver categoría	Correcto	Correcto	Correcto
Ver detalles	Correcto	Correcto	Correcto
Buscar un artículo	Correcto	Correcto	Correcto
Ir a una subcategoría	Problemas de diseño	Correcto	Correcto
Filtrar búsqueda	Correcto	Correcto	Correcto
Cancelar búsqueda	Correcto	Correcto	Correcto

Tabla 32. Resultados funcionales: módulo buscar

Módulo detalles			
Caso de uso	Aplicación web	iOS	Android
Alerta de un error	Correcto	Correcto	Correcto
Añadir a la lista	Correcto	Correcto	Correcto
Agregar un comentario	Problemas de diseño	Correcto	Correcto

Tabla 33. Resultados funcionales: módulo detalles

Módulo lista			
Caso de uso	Aplicación web	iOS	Android
Listar supermercado	Correcto	Correcto	Correcto
Ir a detalles	Correcto	Correcto	Correcto
Eliminar ítem de la lista	Correcto	Correcto	Correcto

Tabla 34. Resultados funcionales: módulo lista

Módulo perfil			
Caso de uso	Aplicación web	iOS	Android
Ir a FAQs	Correcto	Correcto	Correcto
Iniciar sesión	Problemas de diseño	Correcto	Problemas de diseño
Crear cuenta	Correcto	Correcto	Correcto
Cerrar sesión	Correcto	Correcto	Correcto
Formulario nuevo artículo	Problemas leves	Correcto	Problemas leves
Formulario sugerencia	Correcto	Correcto	Correcto
Revisar solicitud	Correcto	Correcto	Correcto
Revisar corrección	Correcto	Correcto	Correcto
Activar notificaciones	Correcto	Correcto	Correcto
Modificar datos personales	Correcto	Correcto	Correcto

Tabla 35. Resultados funcionales: módulo perfil

Se pueden observar unos resultados bastante favorables, siendo en su gran mayoría correctos y no se ha detectado ningún problema grave. Sin embargo, se destacan algunos problemas con el diseño, principalmente en la aplicación web. Esto es coherente, pues la aplicación se ha desarrollado con la filosofía “mobile first design”, esta consiste en realizar una aplicación pensada desde un principio para dispositivos móviles, y posteriormente adaptar el diseño a pantallas más grandes.

Por otro lado, se encuentra algún problema leve en la funcionalidad: formulario nuevo artículo, únicamente aparece el problema en aplicaciones web y Android. Este suceso es causado por un problema con Capacitor en el elemento responsable de abrir la galería de algunos dispositivos Android. Este elemento es importante para el caso de uso, puesto que el formulario de nuevo artículo requiere subir las imágenes necesarias por el usuario. Por el lado de la aplicación web, este problema leve es causado por la funcionalidad de arrastrar la imagen al formulario, ya que en algunos navegadores no funciona como debería.

En resumen, se confirma que la aplicación está lista para ser probada por usuarios reales para conocer sus opiniones.

Para finalizar este apartado, se han realizado pruebas de carga de la aplicación web y PWA. Cabe resaltar que estas pruebas son orientativas, ya que dependemos en gran medida del hosting que nos proporciona gratuitamente. Sin embargo, podemos hacernos una idea de cuánto demora la aplicación en cargar y si hay algún factor que impide el arranque eficiente.

Para la realización de esta prueba se utiliza la página Web.dev ². Esta plataforma está desarrollada por Google especialmente para los desarrolladores, con el objetivo de evaluar las aplicaciones web en desarrollo y aporta cursos para mejorar los resultados obtenidos

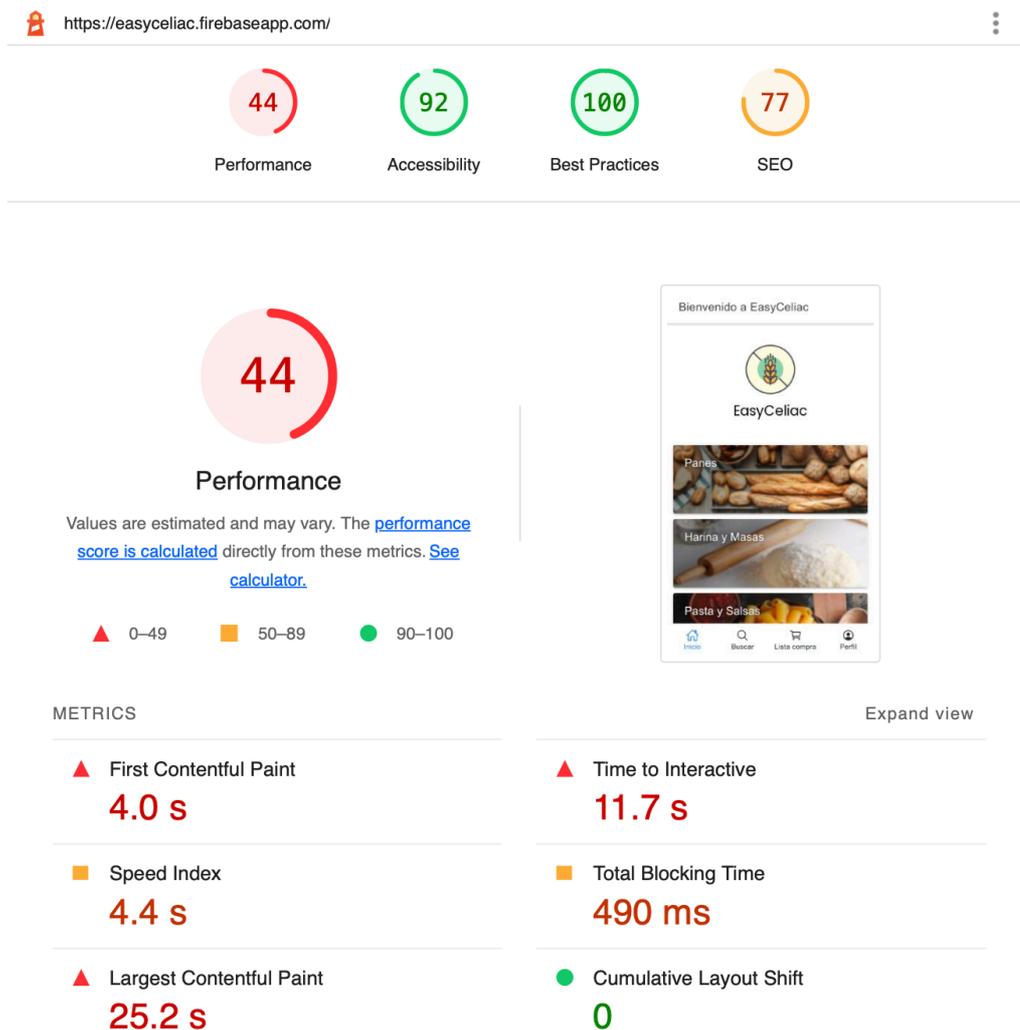


Figura 40. Prueba de tiempos de carga en web.dev

² <https://web.dev>

Estudiamos los resultados obtenidos por la herramienta (figura 40). Y se observan valores de carga bastante elevados, como la velocidad de indexado a 4.4 segundos y el tiempo de interacción en 11.7 segundos. Este último valor funciona de la siguiente manera, se calcula el tiempo que tardó en finalizar la última tarea larga y cuando está fue seguida por cinco segundos de inactividad en la red y el subproceso principal.

Como estos dos valores importantes son elevados, se ha seguido los siguientes pasos para intentar reducir los tiempos de carga:

1. Identificar imágenes del proyecto que no se utilicen y eliminarlos, como iconos o logos.
2. Reducir el tamaño de las imágenes pesadas dentro del proyecto.
3. Identificar las hojas de estilo (css) que no se usan y gastan recursos.
4. Revisar si hay algún bucle o método ineficiente que pueda estar causando el problema.

Posteriormente, se vuelve a ejecutar la herramienta.

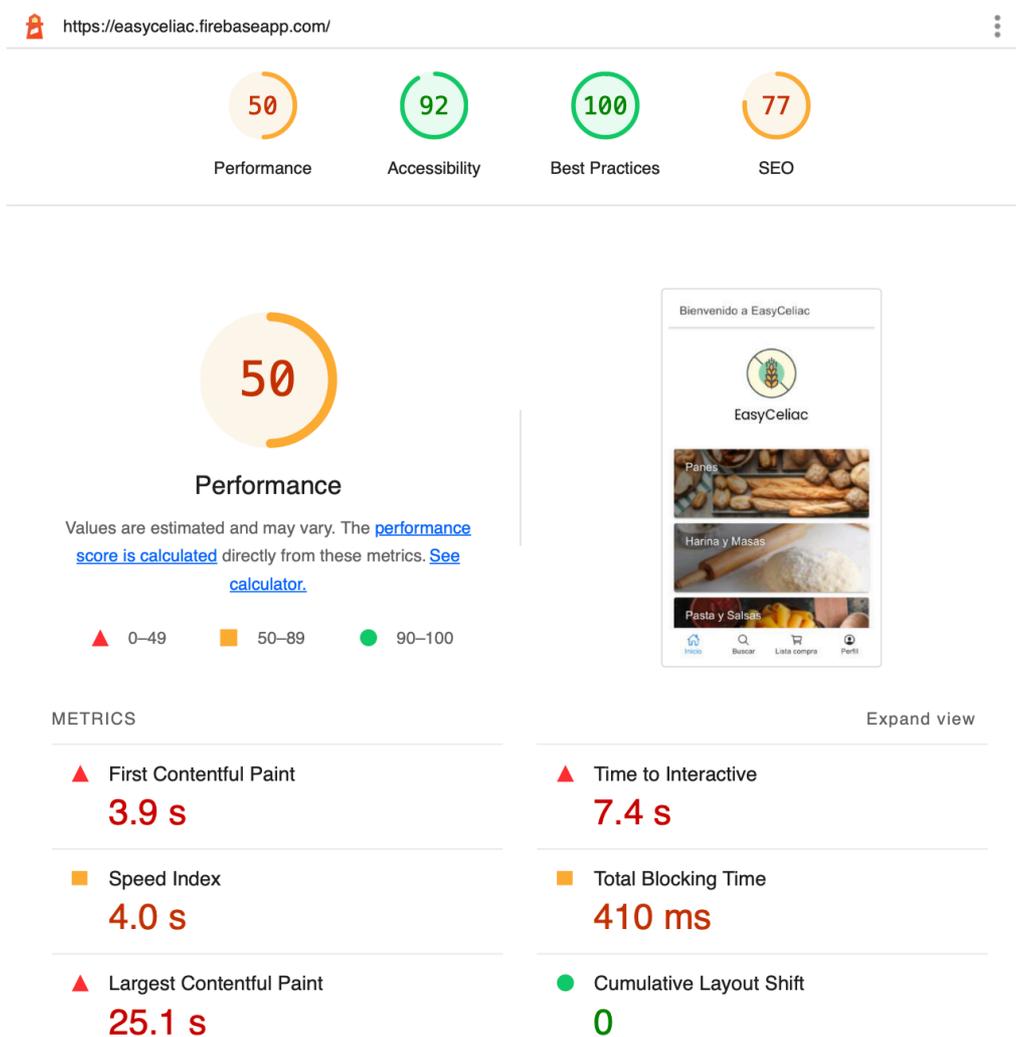


Figura 41. Prueba de tiempos de carga en web.dev después de las correcciones

Vemos de nuevo los resultados de la segunda prueba después de las correcciones (figura 41). Se observa un cambio considerable, pues la velocidad de indexado ha pasado de 4,4 segundos a 4 segundos y el tiempo de interacción ha bajado de 11,7 segundo a 7,4 segundos.

6.2 Pruebas con usuarios

Estas pruebas consisten en compartir la aplicación con algunas personas para que estén un tiempo probándola y conocer la opinión que tienen acerca de la aplicación. Para lograr esto, se ha creado una encuesta con Google Forms para que los candidatos la rellenen una vez hayan probado la aplicación. La encuesta se ha estructurado de la manera más sencilla posible con preguntas muy simples y cortas para que el usuario se centre en su opinión y no abarcarlo con muchas preguntas.

Se ha procurado tener una muestra de usuarios considerables de distinto rango de edad. En total, se han conseguido 21 candidatos, y a continuación, se analizan algunas de las respuestas obtenidas.

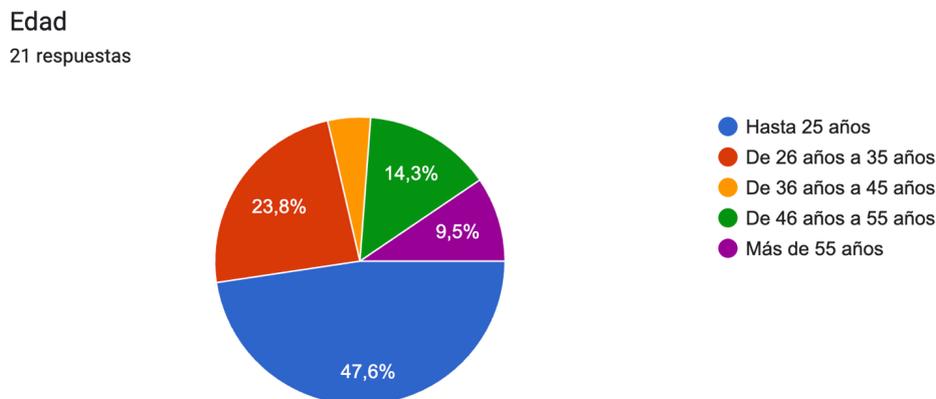


Figura 42. Resultado de encuesta: edad

Primeramente, en la figura 42, se confirma que las edades de los encuestados son muy variadas. Esta pregunta es muy importante en nuestro proyecto porque, como se comentó en el primer capítulo, la idea de la aplicación es abarcar al máximo público posible de personas. Y teniendo una muestra de usuarios de todas las edades, podemos obtener opiniones muy diferentes entre sí, puesto que no es lo mismo una aplicación intuitiva para una persona de menos de 25 años que para una persona de más de 55.

¿Sueles comprar productos sin gluten?

21 respuestas

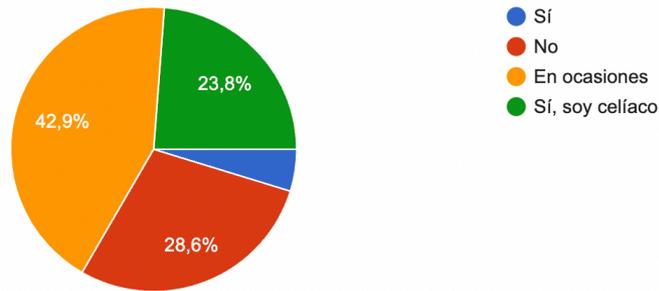


Figura 43. Resultado de encuesta: compra frecuente sin gluten

En esta pregunta (figura 43), podemos hacernos una idea de si la mayoría de usuarios que han probado la aplicación compran productos sin gluten o si directamente son celíacos. Cabe destacar que únicamente el 28,6 % de los encuestados ha respondido que no, esto quiere decir que el 71,4 % de usuarios restantes compran o suelen comprar productos celíacos. Esto es una gran ventaja para el proyecto, ya que se confirma que no solo las personas celíacas compran este tipo de productos.

¿Crees que EasyCeliac puede ayudarte a ahorrar en tus tickets de compra?

21 respuestas

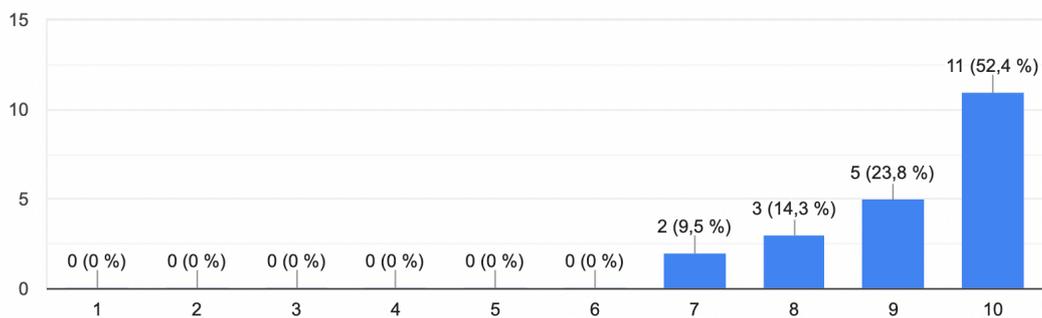


Figura 44. Resultado de encuesta: ahorro en el ticket de compra

Respecto a la función de comparación de productos. ¿Crees que EasyCeliac puede serte útil?

21 respuestas

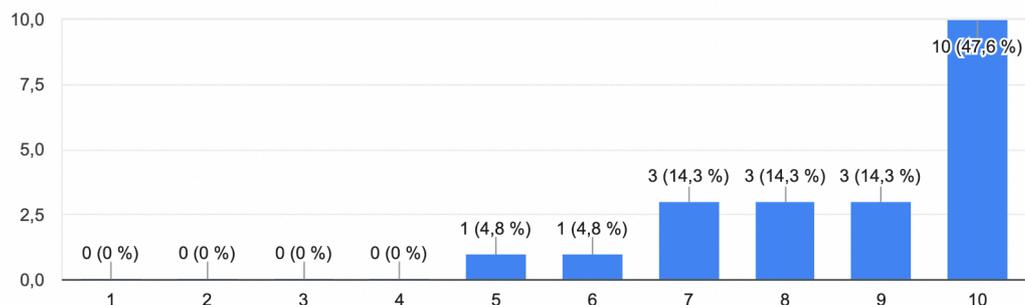


Figura 45. Resultado de encuesta: funcionalidad de comparar productos

En las figuras 44 y 45 se preguntaban uno de los objetivos principales del proyecto, si realmente creen los usuarios que EasyCeliac les puede ayudar ahorrando en el ticket de la compra. Estas dos preguntas están estrechamente relacionadas porque la función de comparación es precisamente para conseguir un ahorro en la compra, y se ha decidido preguntarlo de ambas formas.

Como podemos observar, las respuestas de los encuestados es muy positiva, siendo el máximo valor el más reiterado en ambas preguntas. Se confirma así, que la función de ahorro ha sido cumplido con una gran aceptación por parte de los usuarios.

¿Crees que EasyCeliac puede ayudarte a encontrar productos que no conocías?

21 respuestas

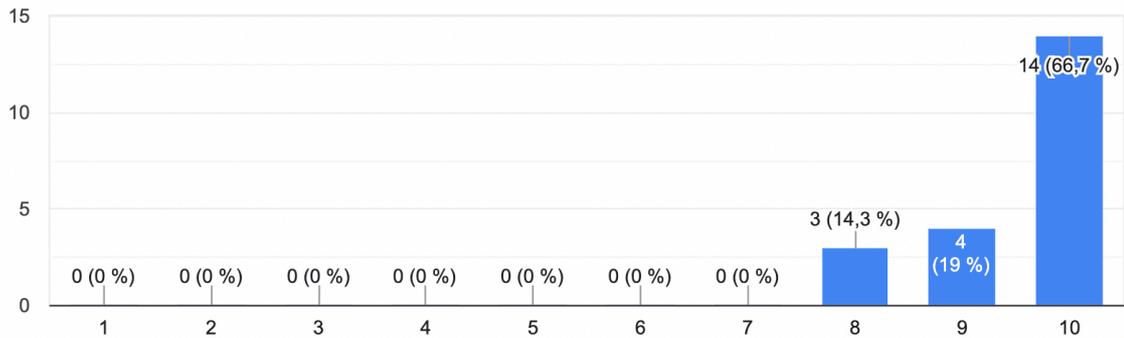


Figura 46. Resultado de encuesta: descubrir nuevos productos

Por otro lado, la pregunta de la figura 46 hace referencia a otro objetivo principal de la aplicación, si la herramienta que han estado probando les ayuda a descubrir nuevos productos sin gluten que no conocían. Analizando los resultados, podemos apreciar una gran influencia sobre este aspecto, y es que más del 60% de los encuestados ha votado la máxima nota, dándonos a entender que están muy satisfechos con esta funcionalidad.

¿Crees que EasyCeliac puede ayudarte a centralizar tus compras en menos supermercados?

21 respuestas

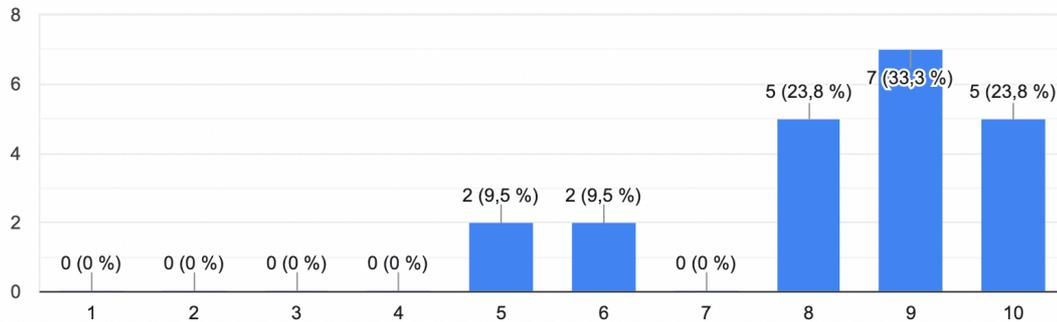


Figura 47. Resultado de encuesta: centralizar las compras en menos supermercados

La última pregunta que se basa en los objetivos del proyecto es la relacionada con la figura 47. En ella, se pregunta a los encuestados si la aplicación les puede ayudar a no tener que desplazarse entre varios supermercados para conseguir la totalidad de la compra deseada. Esta pregunta es quizá la más controvertida, puesto que hay diferentes opiniones. Se destaca la puntuación nueve siendo la más elegida, pero se pueden ver cuatro puntuaciones por debajo del siete. Esto podría ser, por ejemplo, que los usuarios que han votado por debajo del siete no suelen hacer la compra en más de un supermercado, dejando así a esta funcionalidad sin utilidad.

¿Te ha resultado una aplicación fácil de usar e intuitiva?

21 respuestas

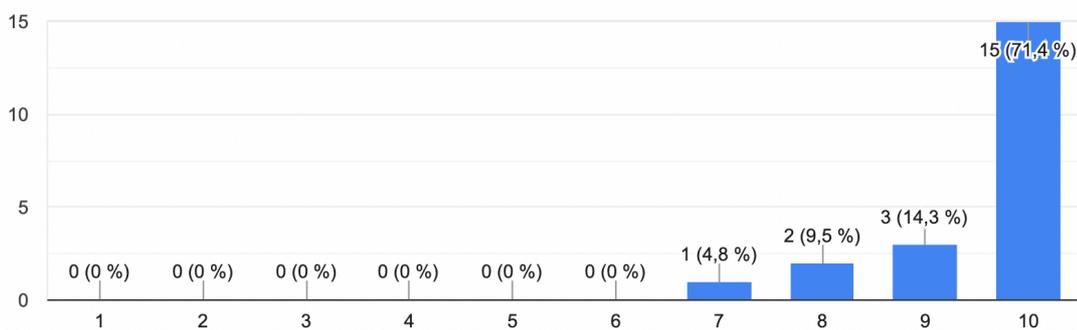


Figura 48. Resultado de encuesta: facilidad de uso

¿Crees que el rendimiento de la aplicación es bueno?

21 respuestas

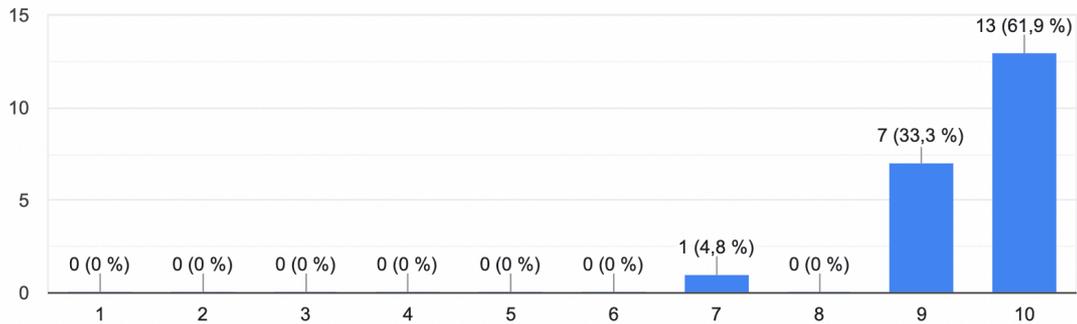


Figura 49. Resultado de encuesta: rendimiento de la aplicación

En estas dos cuestiones (figuras 48 y 49), se les pregunta a los usuarios si han encontrado la aplicación intuitiva y si han obtenido un buen rendimiento durante su uso. El objetivo de estas es conocer el grado de satisfacción del usuario durante el uso de la aplicación, para hacernos una idea de si finalmente se ha conseguido un diseño intuitivo y sencillo para todo el público.

Como se observan en los resultados, a los usuarios les ha parecido una aplicación fácil de usar con más de un 70 % en la máxima nota. Por otro lado, el rendimiento ha sido bueno y no parece haber quejas al respecto. Por lo que se ha cumplido con lo esperado.

¿Recomendarías esta aplicación a algún conocido celíaco?

21 respuestas

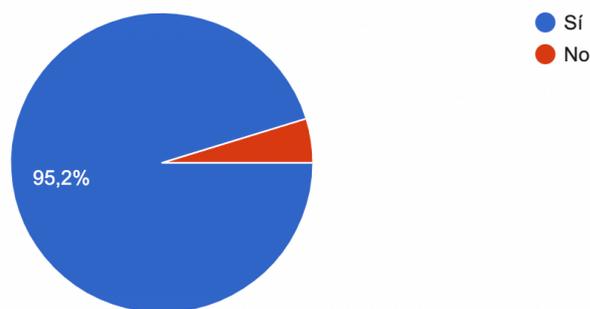


Figura 50. Resultado de encuesta: recomendación a un conocido celíaco

En la figura 50 podemos destacar que prácticamente todos los usuarios recomendarían la aplicación a las personas celíacas. Dándonos a entender un agrado general bastante elevado.

Gracias a esta encuesta podemos concluir una gran aprobación por parte de los usuarios sobre las funcionalidades que EasyCeliac les puede aportar. Además, la gran mayoría de los usuarios han encontrado la aplicación intuitiva sin importar el rango de edad dónde se encuentran.

7. Conclusiones y trabajos futuros

7.1 Conclusión

Es el turno de hacer una retrospectiva completa a este proyecto, desde que objetivos se querían cumplir hasta los resultados obtenidos al finalizarlo.

Primeramente, refrescaremos el planteamiento inicial del proyecto. Se partía de la idea de realizar una aplicación híbrida para las tres plataformas más comunes: navegador web, móviles Android y móviles iOS. Esta aplicación tenía como objetivo llegar al máximo público celiaco posible, construyendo una herramienta muy intuitiva capaz de abarcar tres características principales: ahorrar en la compra diaria haciendo una comparación de productos en distintos supermercados, la posibilidad de descubrir nuevos artículos y centralizar la compra en la menor cantidad de tiendas.

En lo que respecta a la aplicación multiplataforma, se ha construido la herramienta deseada con las características de diseño planteadas, y según los resultados de la encuesta, a los usuarios les ha parecido bastante intuitiva. Por otro lado, los tres objetivos principales se han cubierto al completo, siendo de mayor agrado para los usuarios las funciones de descubrir nuevos productos y la de comparativa.

Para abordar todos los objetivos, he hecho un gran esfuerzo con el estudio previo a la realización del desarrollo. Como se comentó en el primer capítulo, los conocimientos de este framework en concreto eran escasos, es más, al principio del proyecto desconocía al completo el término PWA y no conocía el funcionamiento de una aplicación híbrida. Gracias a la realización de este trabajo he aprendido en abundancia nuevas tecnologías y herramientas que aumentan las competencias profesionales. He conseguido un gran dominio con Ionic y con entornos de trabajo nuevos como Xcode y Android Studio. Además, he aprendido qué es una aplicación nativa y qué diferencia tiene respecto a la construida en este trabajo. Se podría decir que estoy bastante satisfecho con todos estos resultados.

En cuanto a los problemas durante el desarrollo de la aplicación, me he enfrentado a algunos complicados que han requerido mucho tiempo para solventar. Los cuales han aportado un conocimiento vital una vez se han solucionado, no solo por el aspecto tecnológico, sino también por el aspecto personal, como la perseverancia al no tirar la toalla en los momentos más críticos durante los inconvenientes. Algunos de estos han sucedido durante la realización de la PWA y durante la corrección de problemas con paquetes incompatibles entre sí.



7.2 Relación del trabajo desarrollado con los estudios cursados

Durante la realización de este proyecto se han utilizado tecnologías y herramientas de vanguardia que no se han visto en la carrera. Sin embargo, también se ha hecho uso de conocimientos adquiridos en las asignaturas cursadas, las cuales han ayudado a la elaboración de este trabajo. A continuación, se enumeran las asignaturas por orden cronológico con una breve explicación de como se relacionan con el proyecto.

- **IPC.** Interfaces persona computador ha sido la primera asignatura cursada en la carrera que ha aportado varios aspectos muy importantes a este proyecto. En esta materia se explicaron las leyes Gestalt [21] aplicadas al diseño de interfaces. Estas, ayudan a la construcción de diseños más profesionales y eficientes para el usuario. Gracias a estas, se ha construido un diseño más intuitivo y sencillo para todo el público.
- **ISW.** Ingeniería del software es la primera asignatura enfocada a la rama software. El diagrama de clases, visto en el capítulo de diseño de la solución, se ha desarrollado gracias a las competencias adquiridas durante el estudio de esta asignatura, ya que fue la primera asignatura que introdujo este tipo de diagramas UML. Además, se dieron otros aspectos tales como el ciclo de vida del software y sus procesos.
- **DDS.** Diseño del software ha aportado varios matices en cuanto a las buenas prácticas de programación. Los patrones de diseño estudiados en el temario de esta asignatura han sido muy útiles para crear un código limpio. Por otro lado, en esta asignatura se estudiaron las pruebas unitarias para proyectos software, y estas han sido de gran utilidad para desarrollar las pruebas y validaciones en este proyecto.
- **GPR.** Gestión de proyectos ha ayudado a desarrollar una correcta planificación del proyecto, ya que durante el transcurso de esta asignatura se han visto elementos como el diagrama de Gantt y el estudio de los riesgos. Además, se adquirieron conocimientos de planificación económica y de alcance.
- **AER.** Análisis y especificación de requisitos ha sido un pilar durante el inicio del proyecto, porque se han utilizado herramientas muy útiles como el llamado “brainstorming”. Por otro lado, en esta asignatura se han obtenido las habilidades necesarias para la especificación de requisitos y casos de uso desarrollados en este trabajo.
- **PSW y PIN.** Proceso de software y su continuación, proyecto de ingeniería de software, son las asignaturas donde se han visto en profundidad las metodologías ágiles como la implementada en este trabajo. Gracias a estas asignaturas, se ha obtenido el conocimiento necesario para entender y poner en práctica aspectos de estas metodologías, tales como el tablero Kanban y el mínimo producto viable.

- **MES.** Mantenimiento y evolución de software ha aportado a este proyecto el entendimiento de la tecnología Git empleada. Además, gracias a esta se ha podido hacer un seguimiento de las versiones del proyecto.

En cuanto a las competencias transversales que se han puesto en práctica en este proyecto, se destacan las siguientes. Análisis y resolución de problemas se ha visto en varias asignaturas de lógica y programación durante la carrera, y ha sido muy útil durante todo el desarrollo del proyecto para solventar los inconvenientes que han ido apareciendo. Por otro lado, diseño y proyecto ha sido una competencia transversal que ha aparecido en varias asignaturas de la rama de software. Además, planificación y gestión del tiempo también ha sido beneficioso para el trabajo, puesto que, como indica su nombre, ha sido de ayuda con la planificación inicial y su posterior gestión temporal. Por último, innovación y creatividad ha dado un punto de vista distinto en la creación de la interfaz y el diseño del proyecto.

7.3 Trabajos futuros

Gracias al gran número de productos que se han guardado en la base de datos y a las funcionalidades que se han desarrollado en el proyecto, podemos decir que tenemos una primera versión del software perfectamente capaz de ser publicada y utilizada por la población. Sin embargo, durante el desarrollo del mismo, han surgido ideas para nuevas funcionalidades que podrían enriquecer la satisfacción del usuario.

Para versiones futuras, se ha pensado en mejorar la función de comparación. Esta añadiría a la página de lista de la compra una fila deslizable debajo de cada supermercado, en ella se mostrarían productos iguales o muy parecidos pero de otros supermercados, con el fin de facilitar la búsqueda de productos.

Por otro lado, esta aplicación no está pensada para finalizar una compra, es decir, no existe la posibilidad de comprar productos ni tampoco de llevar la lista de la compra a un supermercado online donde si se pueda comprar los productos. Por tanto, se podría abordar en un futuro, la posibilidad de llevar la lista de la compra al supermercado elegido con el fin de comprar los productos de forma online. Para poder realizar esto, se tendría que hablar con los supermercados para conseguir una API que diese paso a su plataforma.

Además, se preguntó a los usuarios en la encuesta sobre que funcionalidades les gustaría que tuviese EasyCeliac en un futuro, y de las respuestas obtenidas se destacan las siguientes:

- Exportar/imprimir los artículos introducidos en la lista de la compra.
- Identificar productos congelados



- Cuando se rechace una propuesta de nuevo artículo, se podría desplegar una caja de texto donde se informe el motivo de rechazo. Este escrito se le enviaría al creador del formulario para que pudiera corregirlo.

Por último, se podría plantear para versiones posteriores un apartado con recetas sin gluten donde los usuarios puedan interactuar también, añadiendo comentarios en las recetas o añadiendo las propias recetas.

8. Referencias

- [1] DAVIS, William. *Sin trigo, gracias*. Aguilar, 2014.
- [2] MARTINS, Julia. ¿Qué es la metodología Kanban y cómo funciona? *Asana* [en línea]. 18 de julio de 2022 [consultado el 29 de julio de 2022]. Disponible en: <https://asana.com/es/resources/what-is-kanban>
- [3] DOUGLAS DA SILVA. ¿Qué es MVP? ¿Cuál es su concepto y cómo crear uno? *Zendesk* [en línea]. 11 de mayo de 2021 [consultado el 29 de julio de 2022]. Disponible en: <https://www.zendesk.com.mx/blog/que-es-mvp/>
- [4] Trello. *Trello* [en línea]. Disponible en: <https://trello.com/es>
- [5] Soysuper, tu supermercado online. *Soysuper* [en línea]. [consultado el 31 de julio de 2022]. Disponible en: <https://soysuper.com/>
- [6] App FACEMOVIL | FACE. *FACE* [en línea]. [consultado el 31 de julio de 2022]. Disponible en: <https://celiacos.org/que-hacemos/app-facemovil/>
- [7] SinGLU10, la aplicación que te ayuda a cortar con el gluten. | ACSG. *sinGLU10 | ACSG* [en línea]. [consultado el 31 de julio de 2022]. Disponible en: <https://singlu10.org/>
- [8] FIALKOVSKAYA, Nataliya. Pagination: what it is and how to implement it properly using 4 ways. *Sitechecker* [en línea]. 19 de junio de 2020 [consultado el 6 de agosto de 2022]. Disponible en: <https://sitechecker.pro/es/pagination/>
- [9] ESPAÑA. Jefatura del Estado. Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales. Ley Orgánica n.º 3/2018 de 5 de diciembre de 2018. *Boletín Oficial del Estado* [en línea]. 6 de diciembre de 2018, (294) [consultado el 9 de agosto de 2022]. Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>
- [10] Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK). Project Management Institute, 2013.
- [11] FLORES, Frankier. Qué es Serverless, ventajas y servicios. *OpenWebinars* [en línea]. 11 de octubre de 2021 [consultado el 12 de agosto de 2022]. Disponible en: <https://openwebinars.net/blog/que-es-serverless-ventajas-y-servicios/>
- [12] LAWSON, Katie. What is a single page application? *Bloomreach* [en línea]. 19 de julio de 2018 [consultado el 12 de agosto de 2022]. Disponible en: <https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application>



- [13] Aplicaciones web progresivas. *web.dev* [en línea]. [consultado el 13 de agosto de 2022]. Disponible en: <https://web.dev/i18n/es/progressive-web-apps/>
- [14] The cross-platform app development leader. *Ionic Framework* [en línea]. Disponible en: <https://ionicframework.com/>
- [15] Cross-platform native runtime for web apps. *Capacitor* [en línea]. Disponible en: <https://capacitorjs.com/>
- [16] Angular. *Angular* [en línea]. Disponible en: <https://angular.io/>
- [17] Firebase. *Firebase* [en línea]. Disponible en: <https://firebase.google.com/>
- [18] Visual studio code. *VSC* [en línea]. Disponible en: <https://code.visualstudio.com/>
- [19] Swift. *Apple Developer* [en línea]. [consultado el 16 de agosto de 2022]. Disponible en: <https://developer.apple.com/swift/>
- [20] Kotlin programming language. *Kotlin* [en línea]. [consultado el 16 de agosto de 2022]. Disponible en: <https://kotlinlang.org/>
- [21] BUSQUETS, Cris. Principios Gestalt aplicados al diseño UI/UX. *uiFromMars* [en línea]. [consultado el 27 de agosto de 2022]. Disponible en: <https://www.uifrommars.com/principios-gestalt-diseno-web/>

Anexo

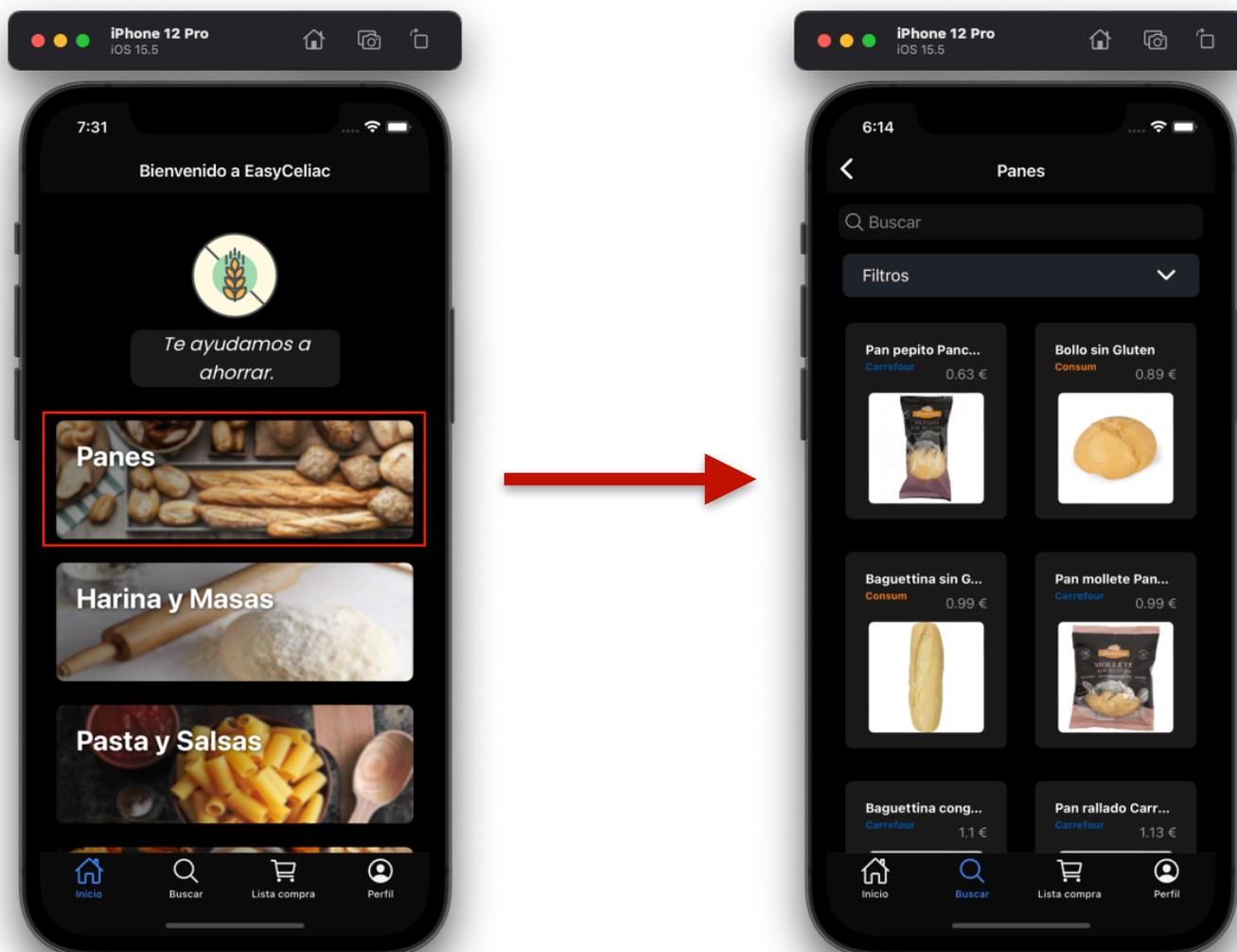
El anexo está formado por dos capítulos. El primero es un manual de usuario de EasyCeliac con algunas de las funciones más relevantes y con capturas de la aplicación final sobre un dispositivo iOS, y el segundo capítulo es una reflexión sobre la relación de este TFG con los objetivos de desarrollo sostenibles.

Manual de usuario

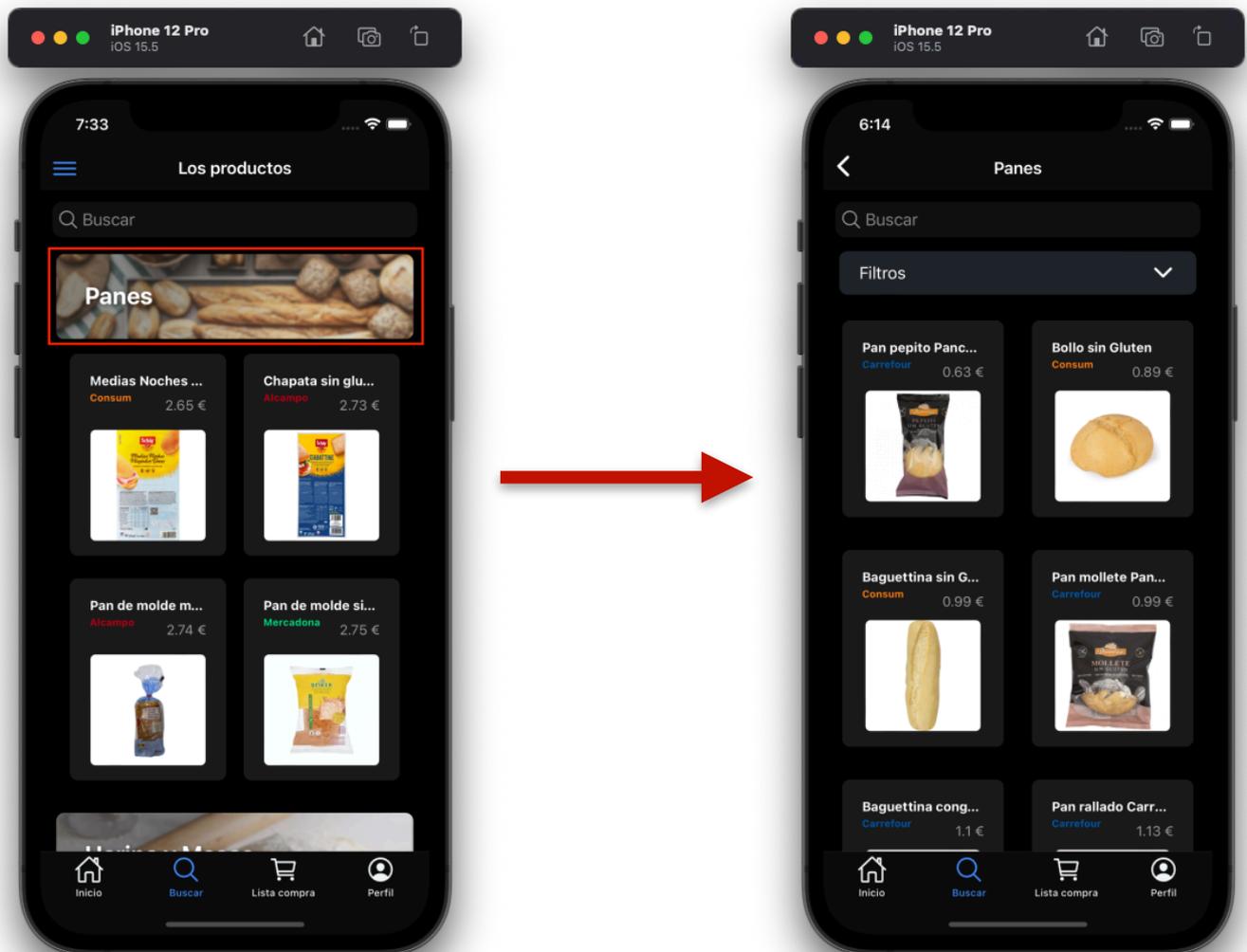
🕒 Buscar productos

Para la búsqueda de productos hay cuatro opciones.

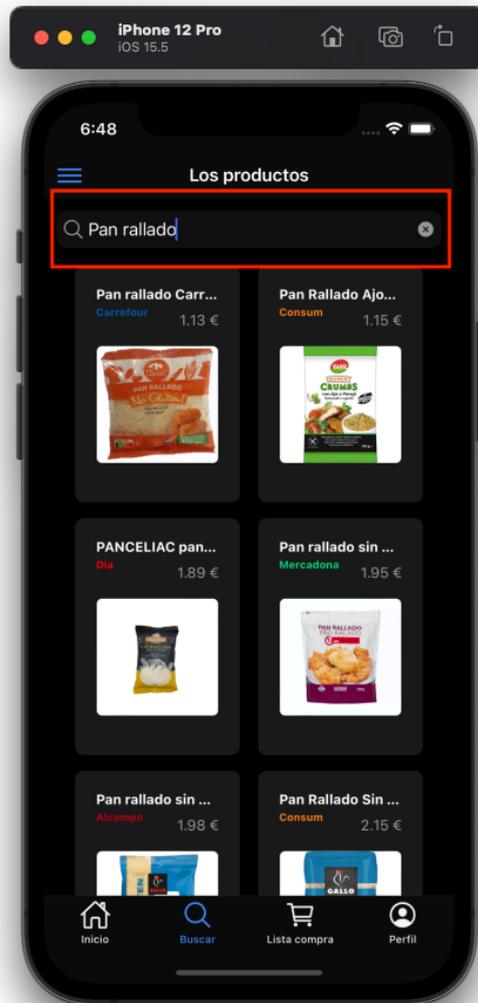
La primera es desde la pestaña Inicio. Se debe seleccionar una de las tarjetas que se muestran, y automáticamente la aplicación le llevará a los productos relacionados con la pestaña seleccionada. A continuación se presenta un ejemplo de uso:



La segunda opción se realiza desde la vista Buscar, y del mismo modo que la anterior, se debe pulsar una de las tarjetas que se muestran a lo largo de esta página y la aplicación le llevará a la pestaña de los productos relacionados con esa tarjeta. A continuación se muestra un ejemplo de uso:

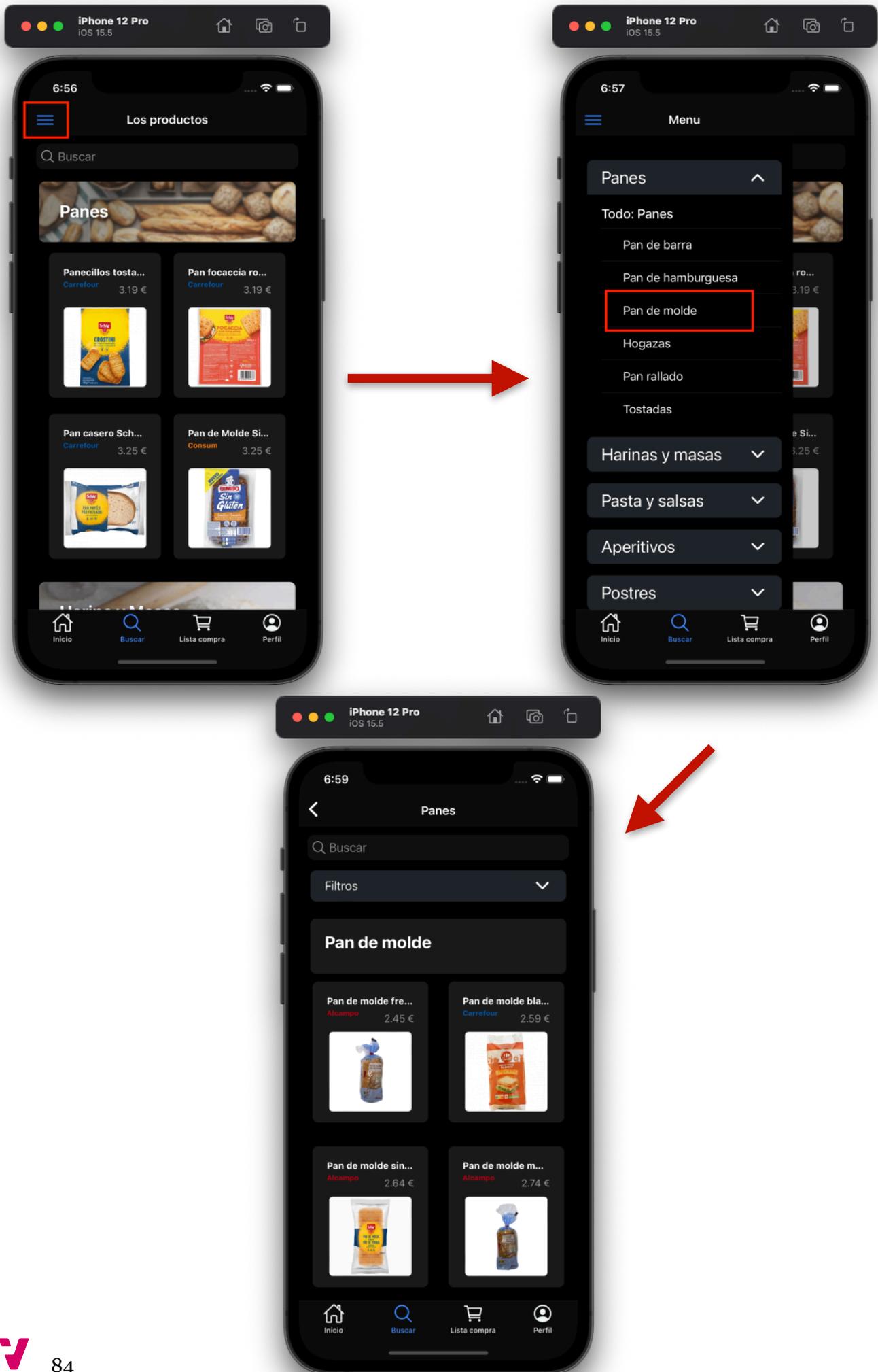


La tercera opción se realiza desde la página Buscar, y para encontrar productos debe hacer uso del buscador situado en la parte de arriba. Únicamente escriba el nombre de un producto y la aplicación le mostrará productos que contengan ese nombre, además, puede escribir el nombre de un supermercado. A continuación se muestra un ejemplo de uso:



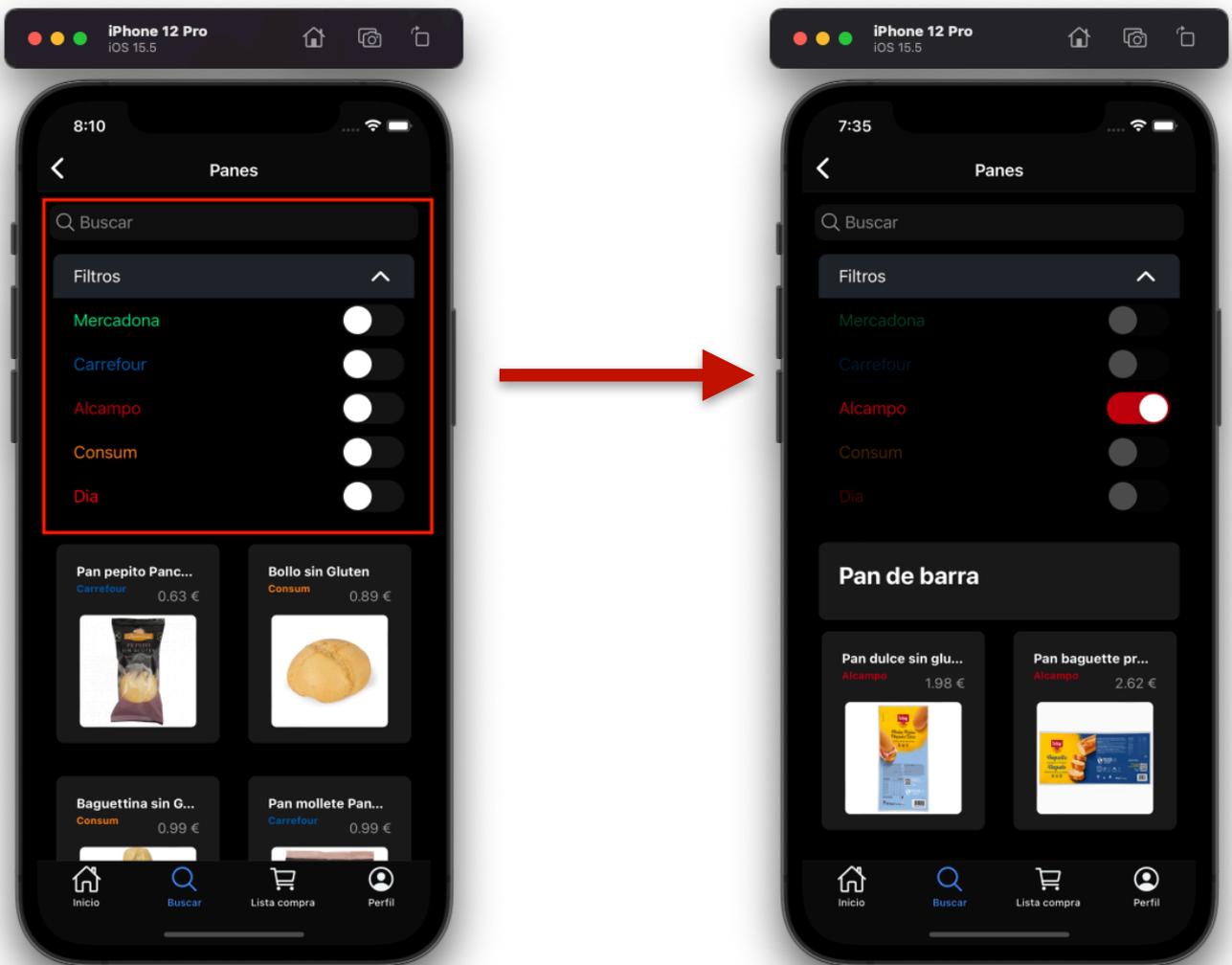
La última opción se realiza también desde la página de Buscar, a continuación se debe pulsar en el botón con el símbolo de tres rayas azules ubicado en la parte superior. Se abrirá una ventana lateral con una lista de subcategorías, al pulsar en una de ellas la aplicación le llevará a los productos de esa subcategoría en concreto. A continuación se muestra un ejemplo de uso:

Desarrollo de una aplicación híbrida para la búsqueda y comparación de comida apta para celíacos entre los supermercados más visitados en España



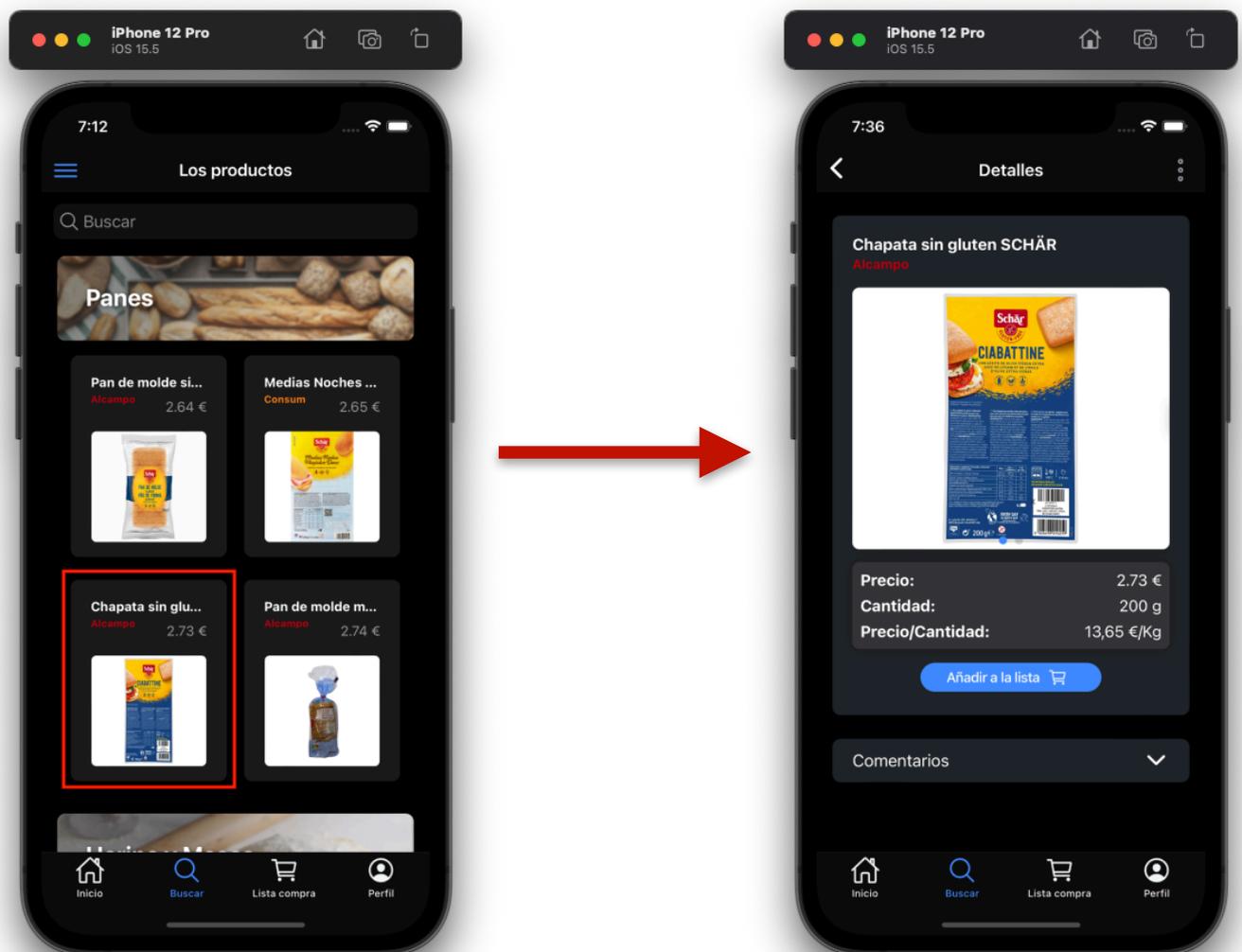
● Filtrar productos

Para poder filtrar productos debe estar en una página de alguna categoría o alguna subcategoría. Una vez en una de estas páginas se puede filtrar los productos utilizando el filtro desplegable, en él podrá seleccionar un supermercado y la aplicación mostrará únicamente productos de ese supermercado. Además, mientras se usa los filtros se puede buscar por nombre empleando el buscador en la parte superior. A continuación se muestra un ejemplo de uso:



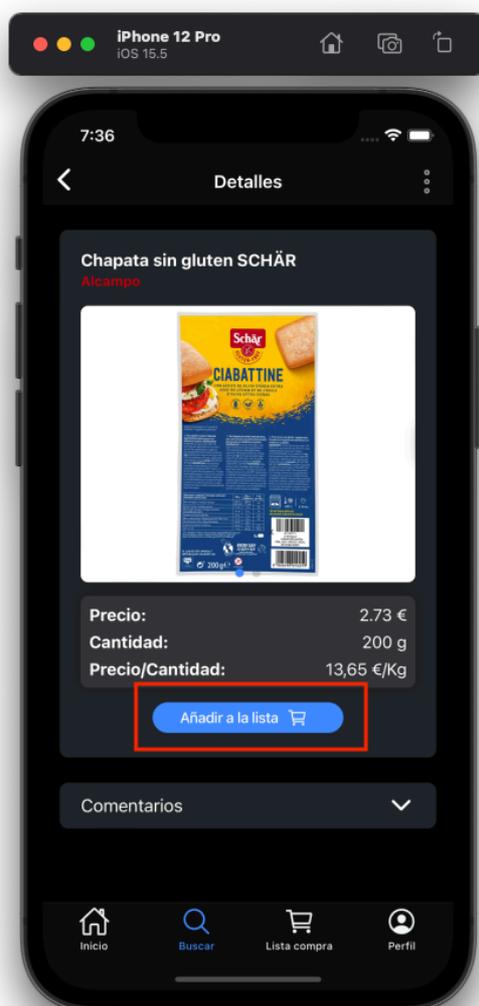
● Ver detalles de un producto

Para ver los detalles de un producto en específico es tan sencillo como pulsarlo desde la ventana de Buscar, puede hacerlo en todo su cuadrado, ya que el contenedor del producto es un botón seleccionable.



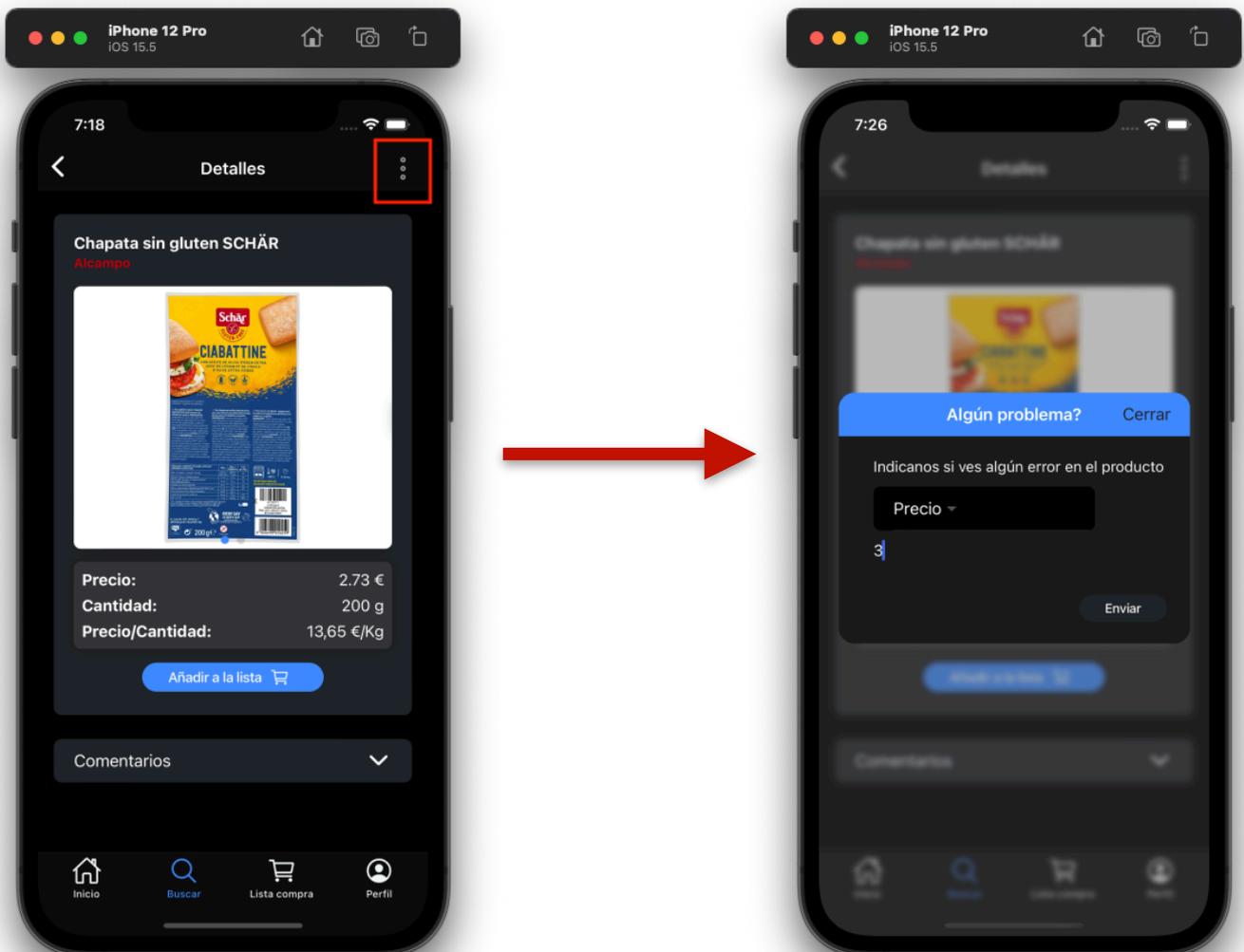
● Añadir el producto a la lista de la compra

Una vez se ha accedido a la pestaña de detalles de un producto, este se puede almacenar en la lista de la compra pulsando el botón “Añadir a la lista”. Luego de esto, la aplicación mostrará un mensaje en la parte inferior de confirmación y se podrá visualizar en la página Lista compra. A continuación se muestra un ejemplo de uso:



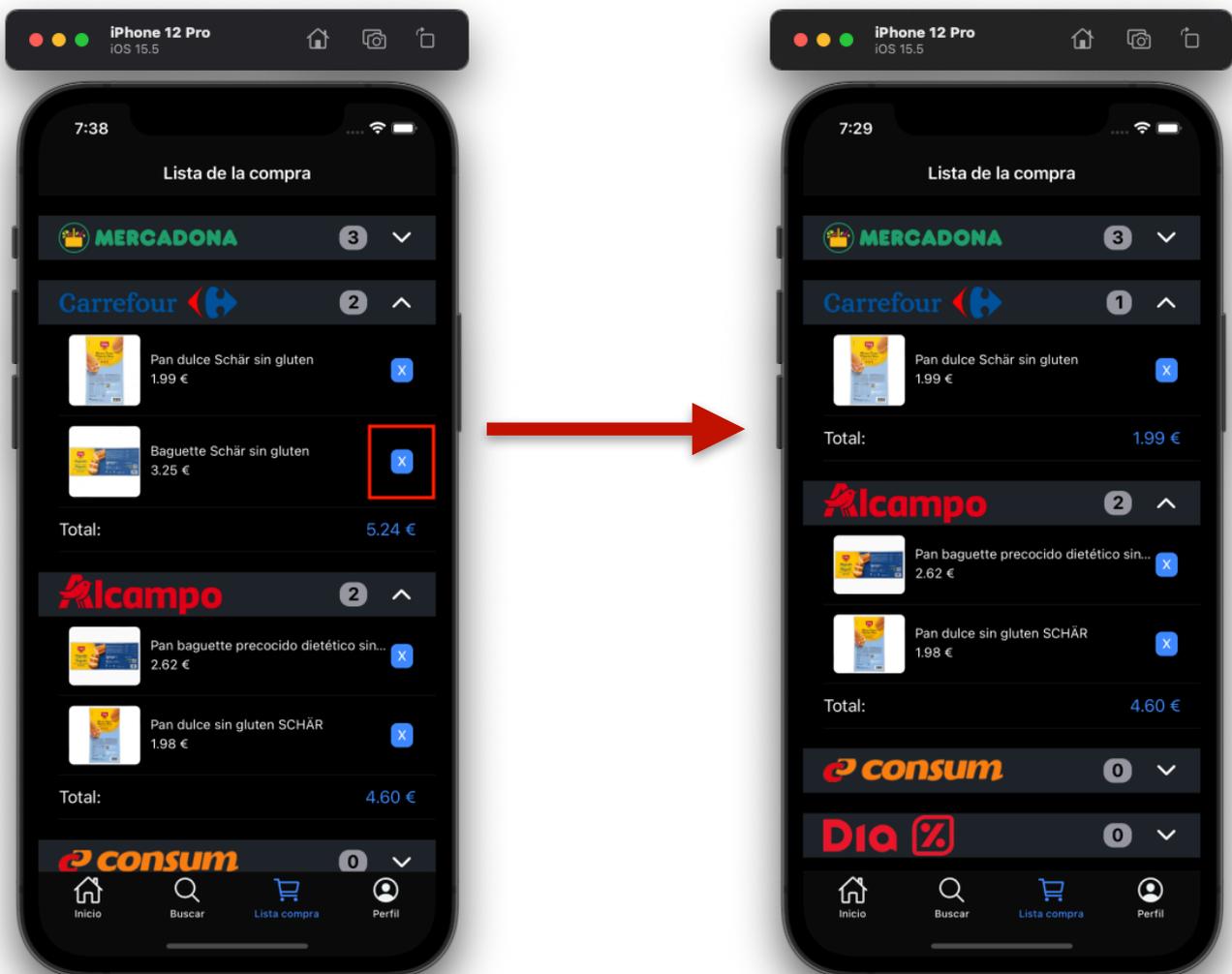
● Notificar un error en un producto

Desde la vista de detalles de un producto se puede informar de un error en el mismo. Para poder acceder se tiene que pulsar sobre los tres puntos verticales de la parte superior. Se abrirá automáticamente una ventana modal en el centro de la aplicación. Primeramente, debe seleccionar el tipo de error y este debe ser uno de entre: Nombre, Precio, Precio/cantidad, Tipo y si el producto sigue existiendo para ese supermercado. Luego de seleccionar el tipo de error debe escribir el valor corregido (no se debe escribir nada más, únicamente el valor) y posteriormente pulsar sobre el botón “Enviar”. A continuación se muestra un ejemplo de uso:



● Eliminar un producto de la lista

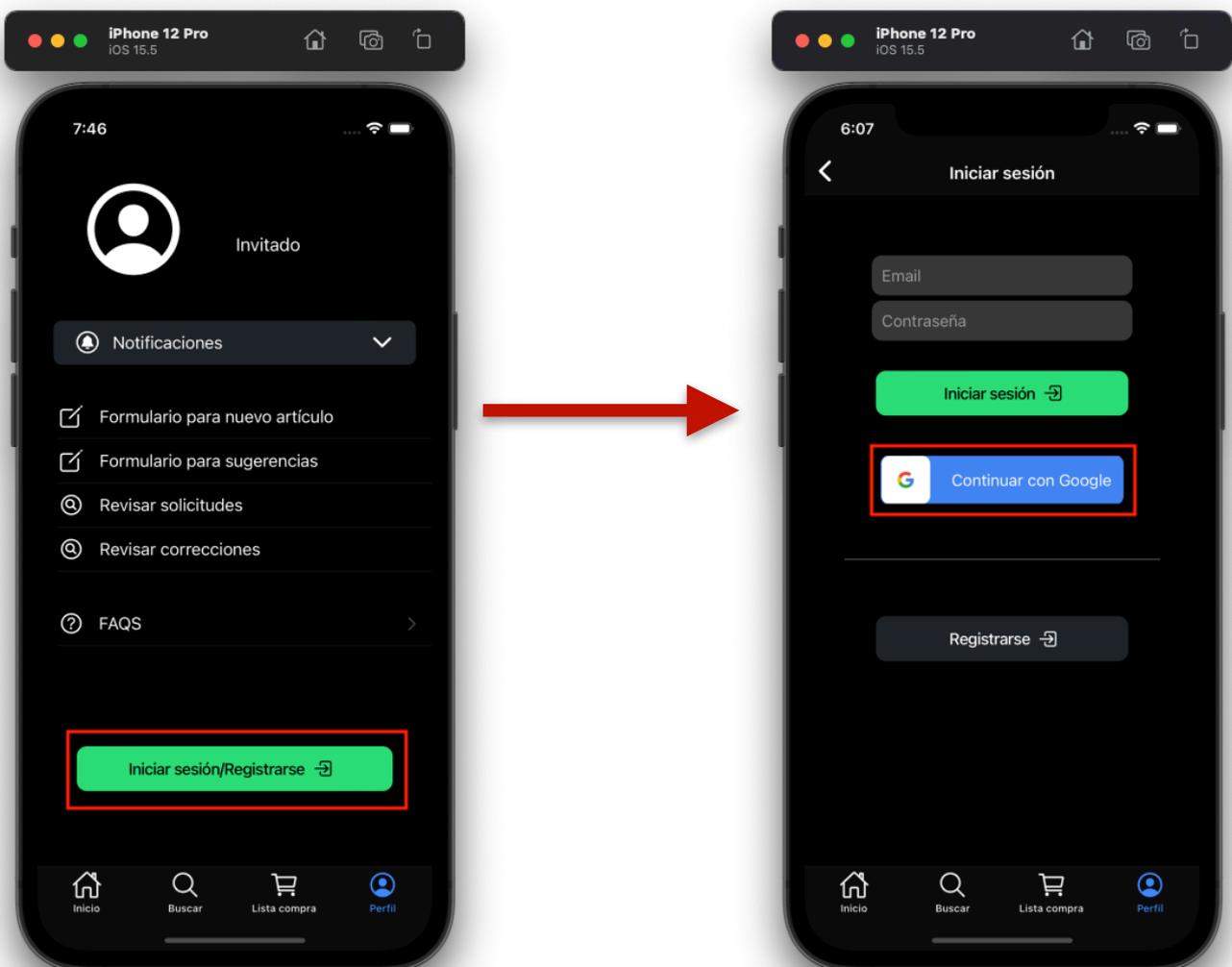
Desde la página Lista compra puede ver qué productos ha guardado anteriormente ordenados según el supermercado al que pertenece. Desde esta página puede eliminar uno de los productos pulsando el botón que contiene una “X”. Además, cabe destacar que puede visualizar los detalles de los productos si pulsa en su imagen o en su nombre. A continuación se muestra un ejemplo de uso:



● Iniciar sesión

Esta aplicación le permite iniciar sesión con el objetivo de desbloquear ciertas funcionalidades que en el modo invitado no se pueden utilizar. Estas son: añadir comentarios en los productos, crear formularios de nuevo artículo y formularios de sugerencias, revisar solicitudes y correcciones. Nótese que aunque en la vista de invitado aparezcan estas opciones, estas están desactivadas y si intenta acceder a ellas aparece un mensaje informativo en la parte inferior (se puede distinguir la ausencia de una pequeña flecha en cada una ellas).

Entonces, para poder iniciar sesión debe situarse en la página Perfil, luego pulsar en el botón verde donde aparece el texto “Iniciar sesión” y automáticamente la aplicación le llevará a una pestaña donde tiene tres opciones: introducir sus datos de acceso y pulsar sobre el botón “iniciar sesión”; iniciar sesión con Google apretando el botón “Continuar con Google” (si es la primera vez que accede con google se le creará la cuenta automáticamente); y por último puede registrarse apretando el botón correspondiente “Registrarse”. A continuación se muestra un ejemplo de uso:







OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
1. Fin de la pobreza.				X
2. Hambre cero.			X	
3. Salud y bienestar.		X		X
4. Educación de calidad.				X
5. Igualdad de género.				X
6. Agua limpia y saneamiento.				X
7. Energía asequible y no contaminante.				X
8. Trabajo decente y crecimiento económico.			X	
9. Industria, innovación e infraestructuras.				X
10. Reducción de las desigualdades.	X			
11. Ciudades y comunidades sostenibles.				X
12. Producción y consumo responsables.				X
13. Acción por el clima.				X
14. Vida submarina.				X
15. Vida de ecosistemas terrestres.				X
16. Paz, justicia e instituciones sólidas.				X
17. Alianzas para lograr objetivos.	X			



De los anteriores objetivos de desarrollo sostenibles mencionados, el presente trabajo está relacionado con:

- **Hambre cero**, puede estar relacionado con el propósito de este proyecto porque buscamos ayudar a las personas celíacas con el ahorro de sus compras en el supermercado. Esto abarca a toda clase de situaciones económicas, ya que comer sin gluten no es tarea fácil y no toda la población se lo pueden permitir.
- **Salud y bienestar**, la celiaquía es una enfermedad autoinmune más común y perjudicial de lo que pensamos. El bienestar de las personas celíacas está relacionado con este proyecto, puesto que la aplicación intenta conseguir una integración social y un apoyo comunitario para hacer su dieta más fácil y sencilla. Las personas que decidan utilizar la aplicación pueden verse influenciadas en la compra de productos más saludables, los cuales aumentarán su salud y bienestar.
- **Trabajo decente y crecimiento económico**, tiene una relación con el proyecto, dado que una de las metas a perseguir por este ODS es mantener el crecimiento económico de conformidad con cada circunstancia nacional, concretamente en España, está enfocado en un crecimiento del PIB per cápita³. En consecuencia, esta herramienta puede ser de utilidad a la hora de alcanzar los objetivos definidos. Su mayor relación pasa por conseguir un ahorro considerable en las familias que, en consecuencia, tendrá un efecto directo en la economía real. De esta forma, se tendrán en cuenta algunas actividades que hasta la fecha han sido minoritarias para proporcionarles una mayor consideración.
- **Reducción de las desigualdades**, como se ha mencionado anteriormente, el proyecto intenta apoyar la integración social de las personas celíacas, evitando así las desigualdades ocasionadas en la población. Esta aplicación tiene como objetivo desarrollar una comunidad de usuarios, los cuales puedan interactuar con la aplicación y con otros usuarios escribiendo comentarios en los productos. Además, en un futuro estos usuarios podrán publicar recetas de diferentes comidas que retroalimentan dicha integración.

³ Es un indicador económico que mide la relación entre el nivel de renta de un país y su población, su fórmula es el cociente entre el Producto Interior Bruto (PIB) y el número de habitantes de una zona geográfica determinada.



- **Alianzas para lograr objetivos**, este ODS tiene un apartado destinado al comercio, en el cual intenta promover un sistema abierto, no discriminatorio y equitativo. Estos aspectos están directamente relacionados con este proyecto, puesto que esta aplicación tiene como meta aproximar el valor económico de una cesta de la compra sin gluten con una cesta de la compra sin restricciones. Este movimiento debe influenciar al sistema comercial del país a un cambio radical con los precios de la comida sin gluten, porque en la actualidad estos son discriminatorios y claramente no equitativos comparados a habituales.