



Universidad de Valladolid

Escuela de Ingeniería Informática
Trabajo de Fin de Grado

Grado en Ingeniería Informática
Mención en Ingeniería de Software

Aplicación de gestión colaborativa de precios de productos en
supermercados usando tecnologías AWS

Alumno: Ignacio Arenas Guerra

Tutora: Margarita Gonzalo Tasis

Agradecimientos

Me gustaría agradecer, en primer lugar, a mi profesora y tutora del presente trabajo, Margarita Gonzalo Tasis, por la ayuda y el trato recibidos tanto en este trabajo como durante la carrera.

También quiero mencionar especialmente a mi familia, sin la cuál no podría haber realizado estos estudios. Les agradezco haber estado durante estos cuatro años apoyándome en todo momento y sin dudar que podría conseguirlo.

Por último quiero dar las gracias a Irene. Ha estado acompañándome cada día, incluso desde la distancia, haciéndome ver que “eso” que no hay manera de hacer funcionar va a salir adelante y sin su apoyo, sus consejos, su paciencia y su colaboración este trabajo no habría sido posible.

Resumen

En este documento se detalla el desarrollo de un sistema destinado a la comparación de precios de productos en diferentes establecimientos cuyo objetivo es ayudar a los usuarios a distinguir los mejores precios de productos similares siguiendo la metodología del Proceso Unificado. Con esta aplicación desarrollada para dispositivos Android utilizando los servicios en la nube de Amazon Web Services, los clientes podrán realizar un seguimiento conjunto de los productos de los establecimientos junto con sus precios y seleccionar los más interesantes para sus futuras compras.

Abstract

This document details the development of a system for comparing product prices in different establishments with the aim of helping users to distinguish the best prices for similar products following the Unified Process methodology. With this application developed for Android devices using Amazon Web Services cloud services, customers will be able to track products in establishments together with their prices and select the most interesting ones for their future purchases.

Índice general

1. Introducción	13
1.1. Consideraciones Iniciales	13
1.1.1. Contexto	13
1.1.2. Motivación	13
1.1.3. Objetivos	14
1.2. Estudio de mercado	14
1.2.1. Análisis de aplicaciones similares	14
1.2.2. Oportunidades en el mercado extraídas del análisis	17
1.3. Estructura de la memoria	18
2. Planificación del proyecto	19
2.1. Introducción	19
2.1.1. Propósito	19
2.1.2. Ámbito	19
2.2. UP. Descripción de la metodología utilizada	19
2.2.1. Características principales de la metodología UP	20
2.2.2. Fases de la metodología UP	21
2.2.3. Aplicación de la metodología en el proyecto	21
2.3. Planificación inicial del proyecto	22
2.3.1. Visión general de la planificación	22
2.3.2. Visión detallada de la planificación	22
2.4. Plan de gestión de riesgos del proyecto	27
2.4.1. Gestión de riesgos	28
2.5. Seguimiento de la planificación	34
2.5.1. Cambios sobre la planificación inicial	34
2.5.2. Monitorización de riesgos	37
2.6. Presupuesto del proyecto	38
2.6.1. Recursos necesarios	38
2.6.2. Presupuesto inicial	40
2.6.3. Presupuesto final	40
3. Análisis	42

3.1. Introducción	42
3.2. Actores y Roles del sistema	42
3.2.1. Actores Principales	42
3.3. Requisitos	42
3.3.1. Requisitos funcionales	43
3.3.2. Requisitos funcionales de baja prioridad	44
3.3.3. Requisitos funcionales de información	45
3.3.4. Requisitos no funcionales	45
3.4. Casos de uso	46
3.4.1. Presentación de los casos de uso	46
3.4.2. Diagrama de casos de uso	47
3.4.3. Descripción detallada de los casos de uso	48
3.4.4. Matriz de correspondencia de requisitos y casos de uso	71
3.5. Realización en Análisis de los casos de uso	72
4. Diseño	93
4.1. Arquitectura del sistema	93
4.1.1. Patrón arquitectónico MVVM	93
4.1.2. Arquitectura del cliente	95
4.1.3. Arquitectura del servidor	98
4.2. Otros patrones presentes en la aplicación	100
4.2.1. Patrón <i>Facade</i>	100
4.2.2. Patrón <i>Inversion of Control</i>	101
4.2.3. Patrón <i>Singleton</i>	101
4.2.4. Patrón DAO	102
4.3. Modelo de datos	103
4.4. Clases de diseño detalladas	103
4.5. Privacidad desde el diseño y por defecto	107
4.6. Despliegue de la aplicación	108
5. Implementación	109
5.1. Tecnologías utilizadas	109
5.1.1. Generación de recursos y documentación visual	109
5.1.2. Creación de la memoria	110
5.1.3. Desarrollo de aplicación móvil	111
5.1.4. Ionic Framework	112
5.1.5. Servicios AWS	113
5.2. Implementación del patrón MVVM. Angular Framework	117

5.3. Tecnologías de pruebas	118
5.3.1. <i>Postman</i>	119
5.4. Entorno de desarrollo. Visual Studio Code	119
5.5. Control de versiones. Git Windows y GitHub	119
6. Pruebas	120
6.1. Ejecución de las pruebas	120
6.2. Batería de pruebas	121
6.3. Resultados de las pruebas	127
7. Conclusiones y trabajo futuro	130
7.1. Objetivos conseguidos	130
7.1.1. Objetivos relacionados con las tecnologías	130
7.1.2. Objetivos relacionados con el negocio	130
7.2. Valoración personal	131
7.3. Futuras líneas de desarrollo	131
Bibliografía	131
Anexo I: Estructura del repositorio	134
Anexo II: Manual de instalación	135
Anexo III: Manual de uso	137

Índice de figuras

1.1. Capturas de la aplicación móvil Chollómetro	14
1.2. Capturas de la aplicación móvil SoySuper	15
1.3. Capturas de la aplicación móvil Mercadona	16
2.1. Integración de los casos de uso en el Proceso Unificado	20
2.2. Evolución de la arquitectura en el Proceso Unificado	20
2.3. Diagrama de Gantt de la Fase de inicio	23
2.4. Planificación de la Fase de inicio	23
2.5. Diagrama de Gantt de la Fase de elaboración	24
2.6. Planificación de la Fase de elaboración	24
2.7. Diagrama de Gantt de la Fase de construcción	25
2.8. Planificación de la Fase de construcción	25
2.9. Diagrama de Gantt de la Fase de transición	26
2.10. Planificación de la Fase de transición	26
3.1. Diagramas de casos de uso	47
3.2. Matriz de correspondencia de requisitos y casos de uso	71
3.3. Diagrama de secuencia del caso de uso “Registrarse”	72
3.4. Diagrama de secuencia del caso de uso “Iniciar Sesión”	73
3.5. Diagrama de secuencia del caso de uso “Cerrar sesión”	74
3.6. Diagrama de secuencia del caso de uso “Eliminar perfil”	75
3.7. Diagrama de secuencia del caso de uso “Modificar información de perfil”	76
3.8. Diagrama de secuencia del caso de uso “Visualizar detalles de un producto”	76
3.9. Diagrama de secuencia del caso de uso “Publicar producto”	77
3.10. Diagrama de secuencia del caso de uso “Buscar producto en el catálogo por criterios de búsqueda”	78
3.11. Diagrama de secuencia del caso de uso “Buscar productos en el catálogo por escaneo de código de barras”	79
3.12. Diagrama de secuencia del caso de uso “Consultar lista de la compra”	80
3.13. Diagrama de secuencia del caso de uso “Añadir producto a la lista de la compra”	81
3.14. Diagrama de secuencia del caso de uso “Eliminar producto de la lista de la compra”	82
3.15. Diagrama de secuencia del caso de uso “Guardar la lista de la compra”	83
3.16. Diagrama de secuencia del caso de uso “Modificar cantidad de un producto de la lista de la compra”	84

3.17. Diagrama de secuencia del caso de uso “Modificar información de un producto”	85
3.18. Diagrama de secuencia del caso de uso “Valorar producto”	86
3.19. Diagrama de secuencia del caso de uso “Añadir oferta”	87
3.20. Diagrama de secuencia del caso de uso “Reportar producto”	88
3.21. Diagrama de secuencia del caso de uso “Consultar histórico de precios”	89
3.22. Diagrama de secuencia del caso de uso “Marcar un producto como guardado”	90
3.23. Diagrama de secuencia del caso de uso “Desmarcar un producto como guardado”	91
3.24. Diagrama de secuencia del caso de uso “Consultar lista de productos guardados”	92
4.1. Representación gráfica del patrón MVC Pasivo	93
4.2. Representación gráfica del patrón MVVM	94
4.3. <i>Modules Style</i> y <i>Uses Style</i> general de la aplicación	95
4.4. Diagramas de la arquitectura del paquete de interfaz	96
4.5. Diagramas de la arquitectura del paquete de negocio	96
4.6. Diagramas de la arquitectura del paquete de servicios	97
4.7. Diagrama que representa la arquictura básica de la aplicación	98
4.8. Flujo básico de peticiones a la persistencia de la aplicación	99
4.9. Diagrama de recursos de la API REST	100
4.10. Esquema del uso del patrón Fachada en la aplicación	101
4.11. Ejempificación del patrón DAO	102
4.12. Modelo lógico de datos	103
4.13. Clases de diseño detalladas para la capa de Interfaz	104
4.14. Clases de diseño detalladas para la capa de Modelo	105
4.15. Clases de diseño detalladas para la capa de Servicios	106
4.16. Diagrama de despliegue de la aplicación	108
5.1. Capturas de pantalla de la aplicación “Diagrams.net”	109
5.2. Interfaz de usuario de la aplicación Overleaf	111
5.3. Ejemplos de componentes Ionic	113
5.4. Capturas de la consola de administración AWS para el servicio Amazon Cognito	114
5.5. Capturas de la consola de administración AWS para el servicio Amazon API Gateway	115
5.6. Capturas de la consola de administración AWS para el servicio AWS Lambda	116
5.7. “Hooks” del ciclo de vida de un componente Angular	118
5.8. Patrón MVVM aplicado a un componente Angular	118
5.9. Interfaz de usuario de la aplicación <i>Postman</i>	119

Índice de tablas

- 2.1. Descripción de las probabilidades de los riesgos 27
- 2.2. Categorización del impacto de los riesgos 28
- 2.3. Descripción del riesgo n^o1 28
- 2.4. Descripción del riesgo n^o2 29
- 2.5. Descripción del riesgo n^o3 29
- 2.6. Descripción del riesgo n^o4 29
- 2.7. Descripción del riesgo n^o5 30
- 2.8. Descripción del riesgo n^o6 30
- 2.9. Descripción del riesgo n^o7 31
- 2.10. Descripción del riesgo n^o8 31
- 2.11. Descripción del riesgo n^o9 32
- 2.12. Descripción del riesgo n^o10 32
- 2.13. Descripción del riesgo n^o11 33
- 2.14. Descripción del riesgo n^o12 33
- 2.15. Descripción del riesgo n^o13 34
- 2.16. Descripción del riesgo n^o14 34
- 2.17. Modificaciones en el calendario para la Fase de Inicio 35
- 2.18. Modificaciones en el calendario para la Fase de Elaboración 35
- 2.19. Modificaciones en el calendario para la Fase de Construcción 36
- 2.20. Modificaciones en el calendario para la Fase de Transición 37
- 2.21. Costes asociados a los recursos materiales utilizados 39
- 2.22. Comparación entre presupuestos inicial y final del proyecto 41

- 3.1. Detalles del caso de uso “Registrarse” 48
- 3.2. Detalles del caso de uso “Iniciar sesión” 49
- 3.3. Detalles del caso de uso “Cerrar sesión” 50
- 3.4. Detalles del caso de uso “Eliminar perfil” 51
- 3.5. Detalles del caso de uso “Modificar perfil” 52
- 3.6. Detalles del caso de uso “Visualizar detalles de un producto” 53
- 3.7. Detalles del caso de uso “Publicar producto” 55
- 3.8. Detalles del caso de uso “Buscar producto en el catálogo por criterios de búsqueda” 56

3.9. Detalles del caso de uso “Buscar producto en el catálogo por escaneo de código de barras” 57

3.10. Detalles del caso de uso “Consultar lista de la compra” 58

3.11. Detalles del caso de uso “Añadir producto a la lista de la compra” 59

3.12. Detalles del caso de uso “Eliminar producto de la lista de la compra” 60

3.13. Detalles del caso de uso “Guardar lista de la compra” 61

3.14. Detalles del caso de uso “Modificar cantidad de un producto de la lista de la compra” 62

3.15. Detalles del caso de uso “Modificar información de un producto” 63

3.16. Detalles del caso de uso “Valorar producto” 64

3.17. Detalles del caso de uso “Añadir oferta” 65

3.18. Detalles del caso de uso “Reportar producto” 66

3.19. Detalles del caso de uso “Consultar histórico de precios” 67

3.20. Detalles del caso de uso “Marcar producto como guardado” 68

3.21. Detalles del caso de uso “Desmarcar producto como guardado” 69

3.22. Detalles del caso de uso “Consultar lista de productos guardados” 70

6.1. Caso de prueba 1 121

6.2. Caso de prueba 2 121

6.3. Caso de prueba 3 121

6.4. Caso de prueba 4 121

6.5. Caso de prueba 5 122

6.6. Caso de prueba 6 122

6.7. Caso de prueba 7 122

6.8. Caso de prueba 8 122

6.9. Caso de prueba 9 123

6.10. Caso de prueba 10 123

6.11. Caso de prueba 11 123

6.12. Caso de prueba 12 123

6.13. Caso de prueba 13 124

6.14. Caso de prueba 14 124

6.15. Caso de prueba 15 124

6.16. Caso de prueba 16 124

6.17. Caso de prueba 17 125

6.18. Caso de prueba 18 125

6.19. Caso de prueba 19 125

6.20. Caso de prueba 20 125

6.21. Caso de prueba 21 126

6.22. Caso de prueba 22	126
6.23. Caso de prueba 23	126
6.24. Caso de prueba 24	126
6.25. Caso de prueba 25	127
6.26. Caso de prueba 26	127
6.27. Resultados de la ejecución del Caso de prueba 1	127
6.28. Resultados de la ejecución del Caso de prueba 3	127
6.29. Resultados de la ejecución del Caso de prueba 6	128
6.30. Resultados de la ejecución del Caso de prueba 9	128
6.31. Resultados de la ejecución del Caso de prueba 16	128
6.32. Resultados de la ejecución del Caso de prueba 18	128
6.33. Resultados de la ejecución del Caso de prueba 24	129

Capítulo 1: Introducción

1.1. Consideraciones Iniciales

1.1.1. Contexto

Gran parte de las actividades que una persona desarrolla en la sociedad actual tiene que ver con la provisión o adquisición de bienes y servicios. Este modelo de sociedad pone al alcance de la mano de los consumidores unas opciones de compra cada vez más variadas en lo que se refiere a satisfacer una determinada necesidad. Este modelo económico ha derivado en una constante competición de las empresas por atraer a los clientes utilizando estrategias cada vez más sofisticadas para imponerse a sus competidores en el mercado.

El cliente, en el centro de esta situación, necesita poder tomar las decisiones sin verse influido por los mecanismos previamente mencionados. Esta es la idea básica que ha motivado la realización de este trabajo.

El proyecto que se presenta en este documento se realiza en el ámbito del Grado en Ingeniería Informática, en concreto como Trabajo de Fin de Grado. En este documento van a presentarse todas las actividades llevadas a cabo desde la concreción de la idea que ha dado lugar al proyecto hasta su resultado final, la aplicación GangApp.

1.1.2. Motivación

Cuando un cliente entra en una tienda es muy probable que en ella encuentre numerosos artículos en oferta. Los reconoce rápidamente por el lugar en el que se encuentran expuestos o el color de los carteles que los anuncian. Ese cliente encuentra inmediatamente nuevas necesidades de compra que no tenía antes de entrar en la tienda. Las irresistibles ofertas con las que los negocios buscan afianzar clientes y otorgarles una sensación de ahorro pueden jugar, sin embargo, en contra de sus intereses.

En muchas ocasiones el cliente encuentra, buscando un artículo o servicio en concreto, otros complementarios que acabará adquiriendo solo por evitar perder “la gran oportunidad” que supone aprovechar el precio en oferta que sugiere la etiqueta. El cliente no lo sabe, pero es muy probable que ese precio en oferta no sea tan diferente del precio con el que cuenta ese mismo artículo en otros establecimientos.

Quizá incluso existan productos suplementarios menos conocidos que satisfagan la misma necesidad, que reduzcan el desplazamiento necesario para su compra (con su reducción directa en el impacto medioambiental) o simplemente que puedan ser proporcionados por tiendas de menor escala cuya supervivencia es vital para nuestra economía.

Estas estrategias de venta basadas en el *neuromarketing* están especialmente popularizadas en las grandes cadenas de tiendas de alimentación como supermercados o hipermercados. No es el objetivo de este trabajo estudiar en profundidad estas tácticas de venta, pero su existencia ha motivado la creación del producto que se presenta en este documento, una aplicación a modo de foro virtual centrado en los productos alimenticios.

Con esta aplicación web y móvil se pretende proveer al consumidor de una serie de herramientas que faciliten la toma inteligente de decisiones a la hora de comprar, lo cual derivará indudablemente en un ahorro para sus bolsillos.

La aplicación permitirá a los usuarios tener a la vista todos los productos de los supermercados presentes en la aplicación y poder efectuar comparaciones entre ellos. Esta tarea de recopilación de información podría suponer demasiado esfuerzo para un solo individuo por lo que la aplicación está pensada para el uso en comunidad. Juntos,

todos los clientes podrán recopilar toda la información necesaria para optimizar cada compra lo máximo posible.

1.1.3. Objetivos

Se busca desarrollar una aplicación móvil capaz de permitir a los usuarios la comparación de productos de supermercados utilizando para ello los servicios en la nube de Amazon Web Services y el SDK para el desarrollo de aplicaciones híbridas de Ionic. Durante este proceso se generarán los artefactos de documentación que la metodología de trabajo, UP en este caso, requiera.

1.2. Estudio de mercado

1.2.1. Análisis de aplicaciones similares

Ecosistema Chollómetro

Se ha elegido analizar esta aplicación ya que ofrece una perspectiva interesante sobre la gestión de contenido basada en aportaciones de los usuarios. Además, comparte objetivo con el sistema a desarrollar ya que busca destacar los productos en oferta que pueden ser encontrados en multitud de tiendas para facilitar su acceso a los potenciales consumidores.

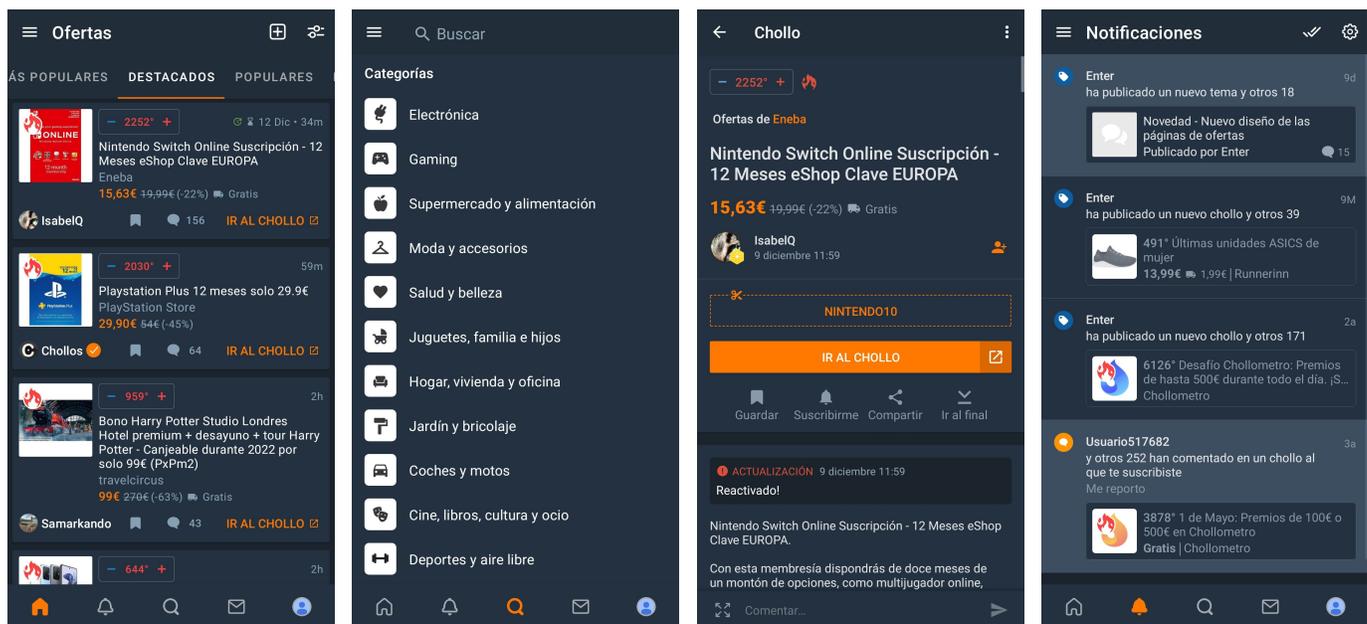


Figura 1.1: Capturas de la aplicación móvil Chollómetro

Sistema en estudio: Aplicación móvil de publicación de ofertas de Chollómetro [22]

Descripción: Este sistema compuesto por una aplicación web y una aplicación móvil permite a las personas registradas descubrir nuevas ofertas aportadas por los usuarios. Además permite la interacción de los usuarios por medio de las funcionalidades clásicas de las redes sociales como pueden ser las valoraciones, comentarios, suscripciones o reportes.

Funcionalidad destacada:

- La interfaz de usuario cuenta con un modo oscuro correctamente implementado.
- El uso de la aplicación es sencillo y de rápido aprendizaje.
- Las publicaciones relacionadas con ofertas ya finalizadas se destacan para evitar confusiones y posteriormente son eliminadas.
- Los usuarios pueden destacar fácilmente las ofertas que les interesan mediante un medidor de popularidad (implementado visualmente utilizando la analogía de un termómetro, que ayuda a entender inmediatamente su función).
- La implementación del menú de navegación inferior.

Debilidades detectadas:

- La página principal cuenta con mucha información y, siendo lo primero que ve el usuario al entrar, es posible que saturé a ciertos sectores de población menos relacionados con el manejo de aplicaciones de este tipo.
- La forma de gestionar los criterios por los que se ordena la información en el registro de eventos (conocido normalmente como “feed”) dificulta acceder a las publicaciones más recientes y, por lo tanto, beneficia a usuarios con mejores reputaciones y unas pocas aportaciones con mejor valoración.

Ecosistema SoySuper

Esta aplicación está planteada como un comparador de productos en diferentes supermercados. Cuenta con una funcionalidad similar en su página web y su aplicación móvil por lo que se utilizará esta última en su análisis.

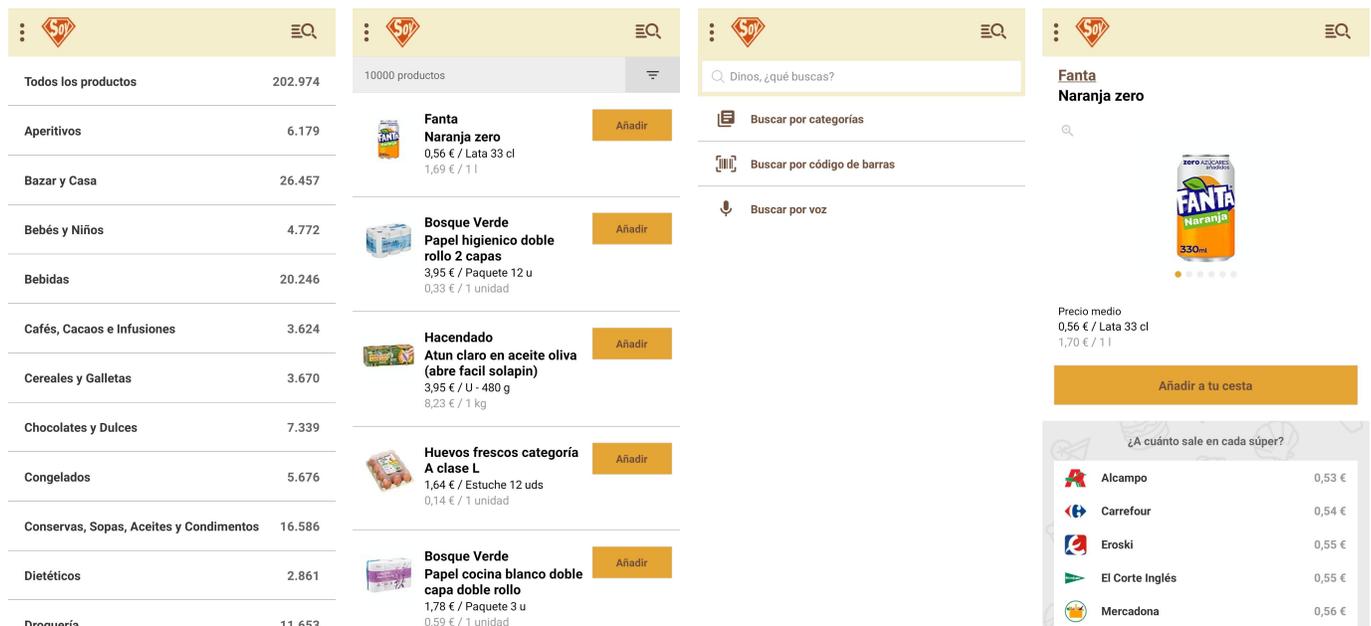


Figura 1.2: Capturas de la aplicación móvil SoySuper

Sistema en estudio: Aplicación de supermercado SoySuper [25]

Descripción: Este sistema, compuesto de una aplicación web y una aplicación móvil está pensado para que los usuarios puedan comparar los precios que ofrecen diversos supermercados sobre el mismo producto. También incluye la posibilidad de hacer listas de la compra para cada una de las empresas con las que colaboran y realizar un pedido directamente a todas ellas a la vez.

Funcionalidad destacada:

- La capacidad de realizar una lista de la compra de cada supermercado.
- Listas de los precios actualizados de un producto por supermercados.
- Búsqueda de artículos por código QR.
- Todos los productos incorporan imágenes.
- Información nutricional de cada producto.
- Ofertas con compañías asociadas.
- Catálogo muy amplio.

Debilidades detectadas:

- El diseño de la interfaz no es estéticamente novedoso.
- La navegación no está personalizada para ofrecer información al usuario que pueda ser de su interés.
- No existe ningún apartado de ofertas de los supermercados.

Mercadona

En el caso de Mercadona nos vamos a centrar en analizar su aplicación web, ya que en una primera aproximación se ha apreciado que su página web incorpora otros procesos del negocio como puedan ser la búsqueda de nuevas incorporaciones o la atención al cliente, siendo estos aspectos irrelevantes para nuestro interés.

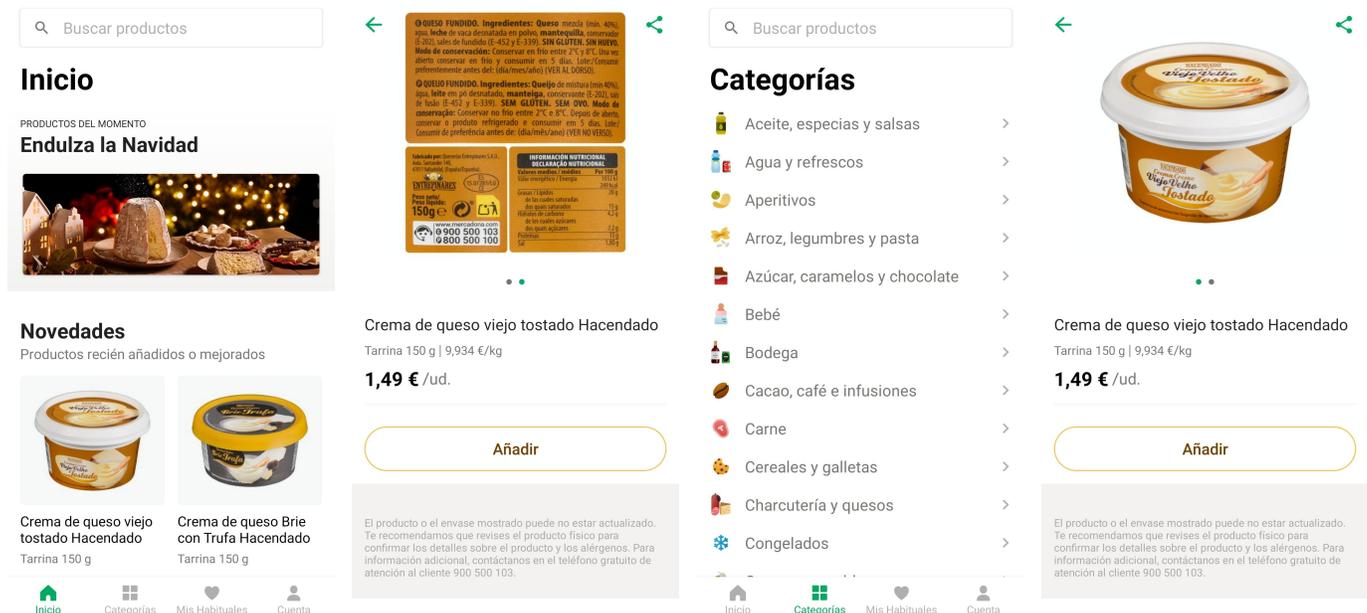


Figura 1.3: Capturas de la aplicación móvil Mercadona

Sistemas en estudio: Aplicación del supermercado Mercadona [21]

Descripción: Es la aplicación oficial de la cadena de supermercados. Permite la realización de una compra de forma *online* mediante la búsqueda de los productos deseados en su catálogo.

Funcionalidad destacada:

- La aplicación es capaz de personalizar los productos en función de la localización del cliente.
- Todos los productos incorporan imágenes.
- Se detallan los ingredientes y aporte nutricional de cada uno de los productos.
- Cuenta con diversas categorías de productos que facilitan la búsqueda.
- Permite gestionar tu “carro” de la compra y posteriormente realizar un pedido.
- Realiza sugerencias de productos en función de los pedidos realizados.

Debilidades detectadas:

- La aplicación tiene una gestión de la lista de la compra muy limitada.
- No se ofrecen productos similares a los que se buscan, siendo necesario realizar un pedido para que la aplicación decida los productos que pueden interesar al cliente.
- Las ofertas no están suficientemente destacadas en relación con otros productos.

1.2.2. Oportunidades en el mercado extraídas del análisis

Podemos extraer una serie de puntos a tener en cuenta del estudio de las aplicaciones similares que se ha desarrollado en las anteriores secciones. En concreto, se han descubierto guías de desarrollo que conviene tener en cuenta en el proyecto:

- **La importancia de enfatizar los eventos temporales en el sistema:** en un sistema cuya función principal es destacar productos en oferta es crucial que la interfaz de usuario ayude a posicionarlos en los puntos más accesibles de la aplicación.
- **La existencia de un modo oscuro en la interfaz de usuario:** en la actualidad, casi todas las aplicaciones buscan implementar de la mejor manera posible una interfaz constituida por colores con poca luminosidad. Esta tendencia se fundamenta en aumentos de productividad, menor cansancio ocular y ahorros de batería en los dispositivos.

No es objetivo de este estudio analizar si los temas oscuros conllevan realmente algún beneficio, pero es innegable que este aspecto se ha posicionado como un factor crucial en el desarrollo móvil, existiendo guías de desarrollo publicadas por empresas muy importantes en el sector como Google y su guía “Material Design” [20]

- **Vistas sencillas con información compacta:** al probar las aplicaciones se han experimentado mayores dificultades en el manejo de aquellas cuyas vistas presentaban mayor complejidad, aunque la funcionalidad en los sistemas era similar.

Por la naturaleza de la aplicación como un producto novedoso que debe resultar atractivo a los consumidores desde el primer uso, se va a implementar una interfaz sencilla, dividiendo la funcionalidad en diferentes secciones del sistema

- **Múltiples referencias entre páginas:** para favorecer que el usuario aprenda lo más pronto posible a manejar la aplicación se debe incorporar un sistema de navegación básico que permita seguir el flujo normal de uso. Para los usuarios con mayor experiencia en la aplicación deben añadirse referencias y atajos desde unas secciones del sistema a otras a fin de que los usuarios experimenten una evolución en el manejo

- **Manejo con una mano:** lo más probable es que los usuarios de la aplicación la utilicen en ciertos momentos mediante una sola mano (por ejemplo en el supermercado escaneando un producto). La herramienta debe tener en cuenta esta posibilidad y estar diseñada para el uso de esta forma (expresamente en determinados procesos críticos).
- **Alternativas para un mismo proceso:** se busca potenciar al máximo la capacidad de atracción de usuarios. Para ello se deben contar con mecanismos variados para realizar los procesos básicos y que sea el propio usuario el que decida cuál le resulta más fácil utilizar.
Un ejemplo al respecto pueden ser la búsqueda mediante códigos de barras u otros criterios basados en características de las aportaciones. También puede apreciarse al presentar la información de los productos, mediante fotos o formularios en la aplicación.
- **Imágenes como centro de las publicaciones:** el sistema debe asegurarse de que toda la información que se incluye en el sistema esta acompañada de imágenes.

1.3. Estructura de la memoria

El documento se estructura en capítulos, secciones y subsecciones siguiendo la siguiente estructura:

- **Capítulo 1. Introducción:** explica las consideraciones previas, motivos, situación y otras informaciones interesantes para conocer el alcance y objetivos del trabajo.
- **Capítulo 2. Planificación del proyecto:** trata todo lo referente a la metodología que se va a utilizar y como se va a estructurar el trabajo durante la duración del proyecto
- **Capítulo 3. Análisis:** explicación de la etapa y los artefactos generados.
- **Capítulo 4. Diseño:** explicación de la etapa y los artefactos generados.
- **Capítulo 5. Implementación:** explicación de la etapa y los artefactos generados.
- **Capítulo 6. Pruebas:** análisis de los procedimientos llevados adelante para detectar y corregir errores en el sistema.
- **Capítulo 7. Conclusiones y futuras líneas de investigación:** experiencias y aprendizajes adquiridos junto con posibles mejoras en el producto creado.
- **Bibliografía y Webgrafía.**
- **Anexos:** incluye los documentos del Manual de instalación, Manual de Uso y Contenido del repositorio.

Capítulo 2: Planificación del proyecto

2.1. Introducción

En este capítulo se expone el plan de desarrollo propuesto para el presente Trabajo de Fin de Grado. Se presentará primero la metodología que se ha utilizado, UP. Se detallarán las características básicas de este tipo de procesos incluyendo las fases, roles, artefactos a generar y las modificaciones necesarias para adecuarlos al contexto de aprendizaje en el que se van a aplicar.

Posteriormente se introduce una planificación de las tareas a llevar a cabo. A continuación, se presenta la planificación real de las tareas fruto del seguimiento elaborado durante el proyecto que recoge todas las replanificaciones derivadas de la materialización de los riesgos.

Asimismo, se dedicará también una sección a la lista de riesgos que pueden afrontarse durante el desarrollo del proyecto. Existirá también un apartado de revisión de los riesgos que finalmente se han materializado (incluyendo algunos que no se hayan tenido en cuenta en la lista inicial) explicando la forma en la que han afectado al proyecto, y su impacto en la consecución de los objetivos.

La sección final se dedicará al presupuesto necesario que un desarrollo como el planteado conlleva, desglosando la factura que la compra de materiales, el alquiler de equipos, las horas de trabajo y otros factores suponen. Siguiendo el esquema de las anteriores secciones, primero se desglosarán los costes que se han tenido en cuenta previamente al desarrollo del proyecto para, en la siguiente subsección, corregir esta previsión con los gastos finales que se han afrontado.

2.1.1. Propósito

El objetivo principal de esta planificación es mantener un control sobre el estado del desarrollo de la aplicación. Se pretenden establecer unos límites tanto temporales como de recursos y comprobar la calidad de las previsiones realizadas.

Otros objetivos son ensayar la planificación de proyectos tecnológicos y poner en práctica los métodos de control, prevención y mitigación (aprendidos durante el grado) para corregir los problemas que puedan surgir.

2.1.2. Ámbito

Este capítulo describe en detalle la planificación y metodologías utilizadas en la creación de la aplicación.

2.2. UP. Descripción de la metodología utilizada

Como ya se ha mencionado previamente, la metodología a utilizar en este proyecto va a ser el UP o “Proceso Unificado”. A continuación se van a abordar las características principales, las fases y los artefactos a generar propios de esta forma de organización de un proyecto *software*, basando las explicaciones en la documentación de IBM sobre

la metodología “RUP” [1, 23], que es un refinamiento de la metodología utilizada. Al final de la sección se detallarán los roles propios de un desarrollo de estas características utilizando una clasificación genérica [17].

2.2.1. Características principales de la metodología UP

- **Está dirigida por casos de uso:** entendiendo los casos de uso como la descripción detallada de las funcionalidades que los usuarios pretenden y deben poder llevar a cabo en el uso del sistema, simplificando mucho, responden a la pregunta “¿Qué va a hacer el sistema?”. Esta metodología los usa como eje principal de todas las etapas que se van a seguir ya que fijan un objetivo claro del desarrollo.

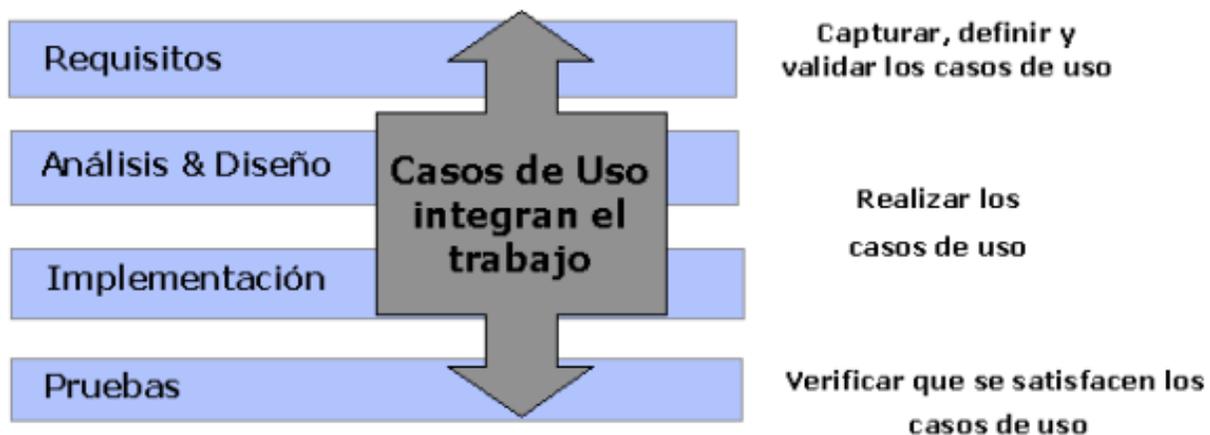


Figura 2.1: Integración de los casos de uso en el Proceso Unificado

- **Es un proceso centrado en la arquitectura:** la arquitectura nos ofrece una instantánea del sistema a definir y sirve como guía para el resto de artefactos. Por eso en un proceso inherentemente cíclico como es el UP es crucial tener siempre presente la arquitectura objetivo. Esto no implica plantear un diseño de arquitectura inflexible durante el la creación del producto, tan importante es la arquitectura en sí, como la revisión de la misma durante todas las fases.

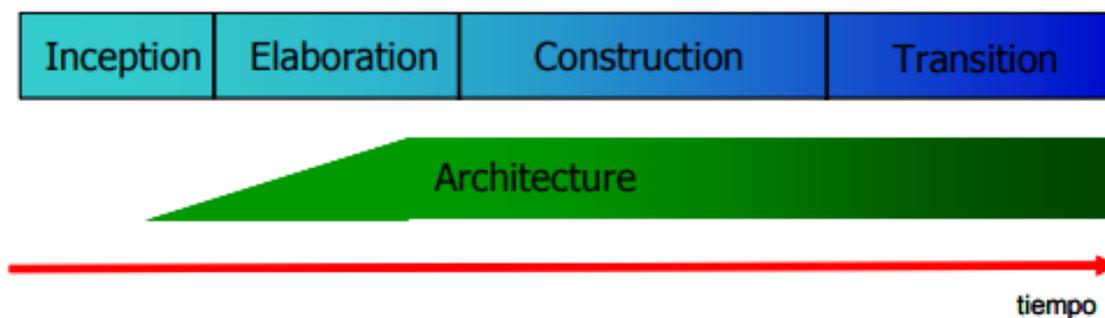


Figura 2.2: Evolución de la arquitectura en el Proceso Unificado

- **Es un proceso iterativo e incremental:** el equilibrio entre qué queremos hacer (definido por los casos de uso) y cómo lo queremos hacer (definido por la arquitectura) sólo puede obtenerse mediante el paso del tiempo y la corrección de los desajustes que puedan darse. Para evitar que la acumulación de estos desajustes derive en algo mucho más importante, consideraremos el proyecto como un conjunto iteraciones con objetivos diferentes y, por

lo tanto, ligeras variaciones también en las fases que las componen. Con esto lograremos que existan puntos de revisión donde poner el foco de nuevo en que los casos de uso y la arquitectura se desarrollen en armonía.

2.2.2. Fases de la metodología UP

Las fases que componen esta variante del Proceso Unificado son similares al proceso original, al menos, en una primera división: Inicio, Elaboración, Construcción y Transición. En esta sección se describirá brevemente en qué consiste cada una de ellas y que artefactos generan.

- **Inicio (*Inception*)**. Durante esta fase buscamos concretar el proyecto con una imagen inicial. Determinaremos información básica como su ámbito o visión y detallaremos los aspectos del negocio que nos son de importancia. Conlleva diagramas que reflejen los casos de uso de mayor importancia, planificaciones que tengan en cuenta los recursos necesarios, presupuestos y riesgos. Pese a ser una primera toma de contacto deben quedar bien definidos el alcance de lo que se quiere conseguir y la forma de hacerlo. También es el primer punto en el que estudiaremos la viabilidad del proyecto y si fuese necesario, su abandono.
- **Elaboración (*Elaboration*)**. En la elaboración buscamos conocer más información acerca del problema que el producto desea resolver. Describiremos la gran mayoría de casos de uso, desarrollaremos las planificaciones y los análisis de riesgo, revisaremos la arquitectura y, en general, buscamos concretar todo lo que hemos planteado en esa fase “origen” del proyecto. Se trata, en realidad, de la última oportunidad de, considerando todos los aspectos descritos, abandonar el proyecto si este no se considera viable.
- **Construcción (*Construction*)**. Debe abarcar el grueso del trabajo ya que en esta fase se realiza el desarrollo como tal mediante iteraciones. Tras ellas, debemos tener un producto finalizado, probado y perfectamente documentado con todos los diagramas que hayan sido necesarios en su construcción, como pueden ser los ya mencionados casos de uso, la arquitectura o el despliegue.
- **Transición (*Transition*)**. Como su nombre sugiere, es una fase destinada a traspasar el proyecto de los desarrolladores a los usuarios. El objetivo es conseguir que las personas que van a usar el sistema sepan cómo hacerlo y además, se sientan cómodos con ello, ya que esto determinará también si el resultado es exitoso. Algunos ejemplos del tipo de artefactos que se generan en esta fase son los manuales de usuario o los documentos legales necesarios.

Como se ha comentado en la sección anterior, todas estas fases tienen en común el desarrollo mediante iteraciones de un tiempo similar que permiten tomar el pulso al desarrollo frecuentemente y evitar que los riesgos que se presenten aumenten en gravedad.

2.2.3. Aplicación de la metodología en el proyecto

Por último queda describir cómo vamos a aplicar todo lo presentado durante esta sección a la creación del producto que ha originado este trabajo. Se seguirá la metodología presentada aunque la estructuración de los artefactos generados no se corresponde con su presentación en el documento puesto que se ha considerado que facilita la comprensión presentar los diagramas y tablas relacionados de forma conjunta.

No se van a presentar, pues, todas las versiones de los diagramas que se han ido refinando durante las diferentes iteraciones sino los puntos de partida y los resultados finales fruto de la revisión constante.

En los siguientes apartados de este capítulo se presentarán los aspectos más relacionados con la definición del proyecto que se realiza en la fase de inicio como son: la planificación dividida por fases y teniendo en cuenta los recursos (roles en este caso), la lista de riesgos y el presupuesto.

2.3. Planificación inicial del proyecto

A lo largo de este apartado se expondrá la planificación que ha determinado el ritmo del trabajo durante su realización. Como no existe una experiencia previa por parte del alumno en desarrollos de este alcance y ha sido necesario compaginarlo con otros requerimientos formativos y laborales se ha optado por una planificación semanal en lugar de diaria.

El alcance inicial del proyecto se ha fijado en 300 horas porque es esa la cantidad de horas propuesta para la finalización de un Trabajo de Fin de Grado.

2.3.1. Visión general de la planificación

Se ha establecido que se asignarán a cada una de las semanas del transcurso del proyecto 20 horas de trabajo que se utilizarán en generar los artefactos propios del proceso UP así como en la generación de la memoria.

La razón de las horas a las semanas se ha tomado atendiendo a diferentes criterios:

- Se ha buscado un número fijo de horas realizable durante todas las semanas. Esto favorece el establecimiento de una rutina de trabajo aun incluso con la flexibilidad inherente a la planificación semanal
- Se ha decidido realizar 4 horas de trabajo de lunes a viernes durante la semana, contando con días extras útiles en caso de una planificación de tiempos que no se ajuste a la realidad para alguna de las tareas, así como otros riesgos que puedan surgir, cuya lista se expondrá más adelante en este capítulo.
- La última semana se utilizará para algo no contemplado en las bases de la metodología UP: durante esa semana se realizarán pruebas con personas voluntarias que ofrecerán feedback sobre los errores que se hayan producido, a fin de poder corregir algún defecto ligero o de generar las conclusiones pertinentes

Siguiendo esta asignación de horas el trabajo quedaría finalizado en dieciséis semanas.

2.3.2. Visión detallada de la planificación

En este apartado se definirán los artefactos propios de la metodología elegida que se generarán en cada una de las semanas de desarrollo, así como otra información importante de cara a la consecución de los objetivos, como son los recursos asignados (aunque a efectos prácticos no afecta a la organización ya que el único desarrollador efectúa todos los roles), la duración en horas y las predecesoras para cada una de las tareas.

Cada fase es considerada como un hito (conocido por su denominación en inglés “milestone” en UP) en el desarrollo del proyecto y por lo tanto, todas las tareas de una fase son predecesoras de aquellas que se encuentren en fases posteriores.

En la realización de este trabajo no se han planificado iteraciones dentro de las fases de la metodología. Aun en contra de los principios establecidos anteriormente en este capítulo, se ha decidido realizarlo así por las restricciones de tiempo que convertirían las iteraciones en un impedimento para poder llevar a cabo el trabajo. A efectos prácticos si que se repetirán elementos de fases anteriores durante todas las fases, de modo que, ciertamente, se cumple la característica de incrementalidad.

FASE DE INICIO

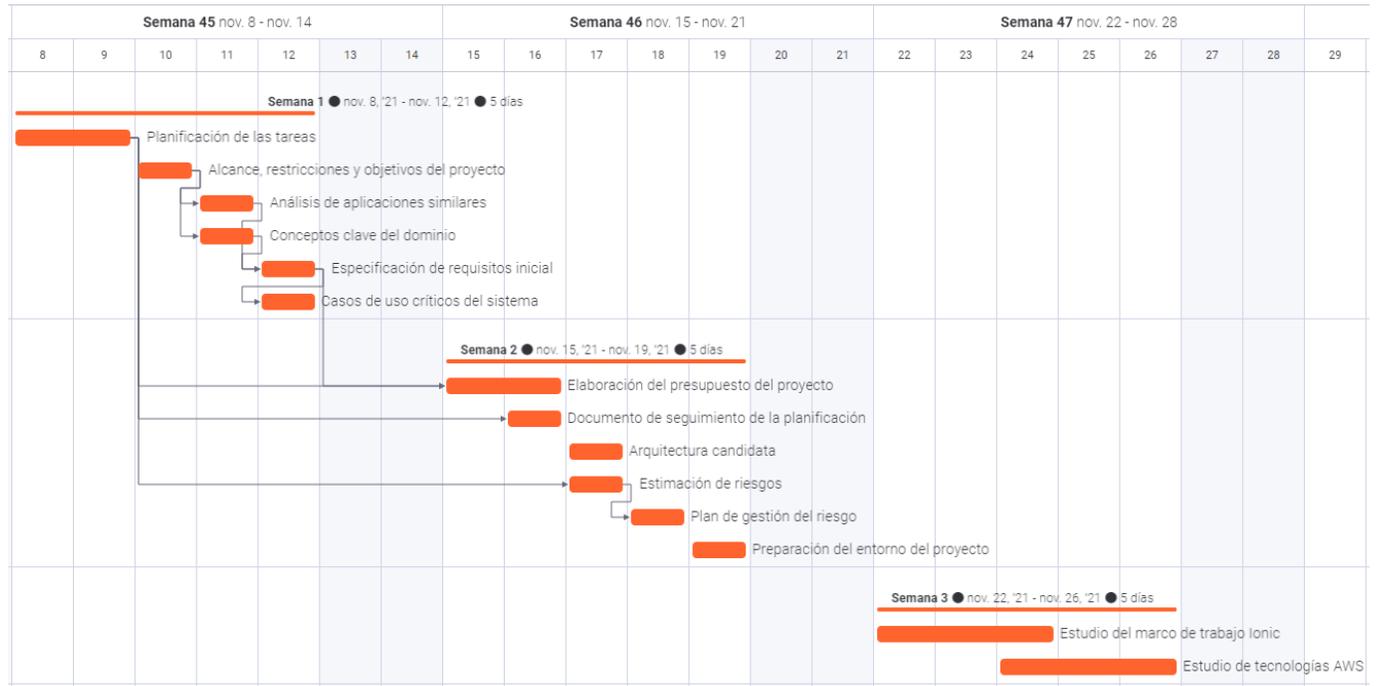


Figura 2.3: Diagrama de Gantt de la Fase de inicio

Semana 1	Cronograma	Recursos necesarios	Duración estimada	Depende de
Planificación de las tareas	nov. 8, '21 - nov. 9, '21	Jefe de Proyecto	8	-
Alcance, restricciones y objetivos del proyecto	nov. 10, '21	Jefe de Proyecto	2	-
Análisis de aplicaciones similares	nov. 11, '21	Analista	4	Alcance, restricciones y objetivos del proyecto
Conceptos clave del dominio	nov. 11, '21	Analista	2	Alcance, restricciones y objetivos del proyecto
Especificación de requisitos inicial	nov. 12, '21	Analista, Stakeholder	2	Análisis de aplicaciones similares Conceptos clave del dominio
Casos de uso críticos del sistema	nov. 12, '21	Analista, Diseñador	2	Especificación de requisitos inicial
+ Agregar Elemento			20 Total	
Semana 2	Cronograma	Recursos necesarios	Duración estimada	Depende de
Elaboración del presupuesto del proyecto	nov. 15, '21 - nov. 16, '21	Jefe de Proyecto	6	Planificación de las tareas Especificación de requisitos inicial
Documento de seguimiento de la planificación	nov. 16, '21	Jefe de Proyecto	2	Planificación de las tareas
Arquitectura candidata	nov. 17, '21	Arquitecto Software	2	-
Estimación de riesgos	nov. 17, '21	Jefe de Proyecto	2	Planificación de las tareas
Plan de gestión del riesgo	nov. 18, '21	Jefe de Proyecto	4	Estimación de riesgos
Preparación del entorno del proyecto	nov. 19, '21	Jefe de Proyecto	2	-
+ Agregar Elemento			18 Total	
Semana 3	Cronograma	Recursos necesarios	Duración estimada	Depende de
Estudio del marco de trabajo Ionic	nov. 22, '21 - nov. 24, '21	Especialista Software	10	-
Estudio de tecnologías AWS	nov. 24, '21 - nov. 26, '21	Especialista Software	10	-
+ Agregar Elemento			20 Total	

Figura 2.4: Planificación de la Fase de inicio

FASE DE ELABORACIÓN

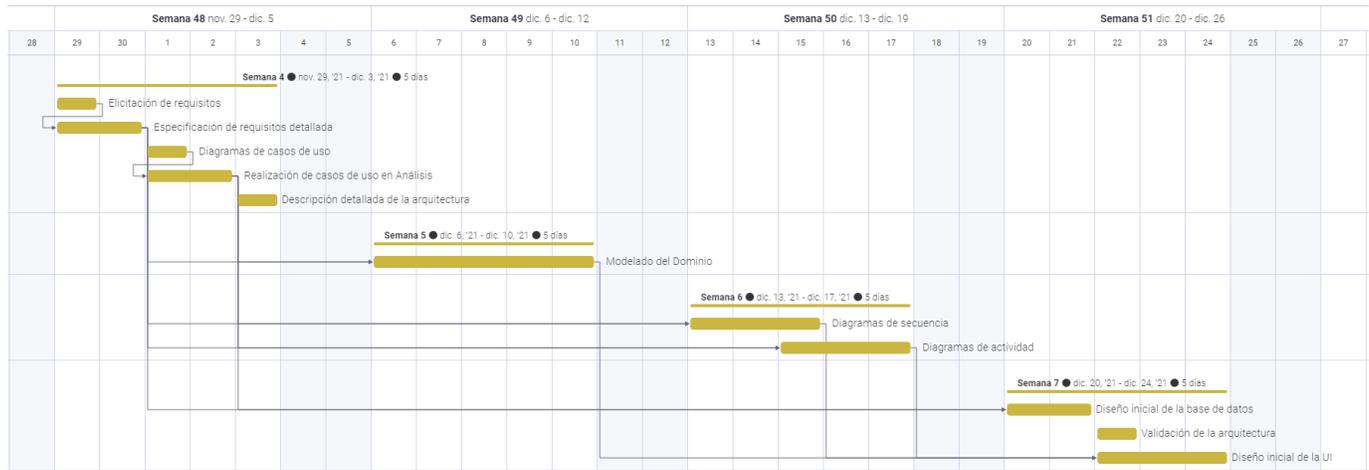


Figura 2.5: Diagrama de Gantt de la Fase de elaboración

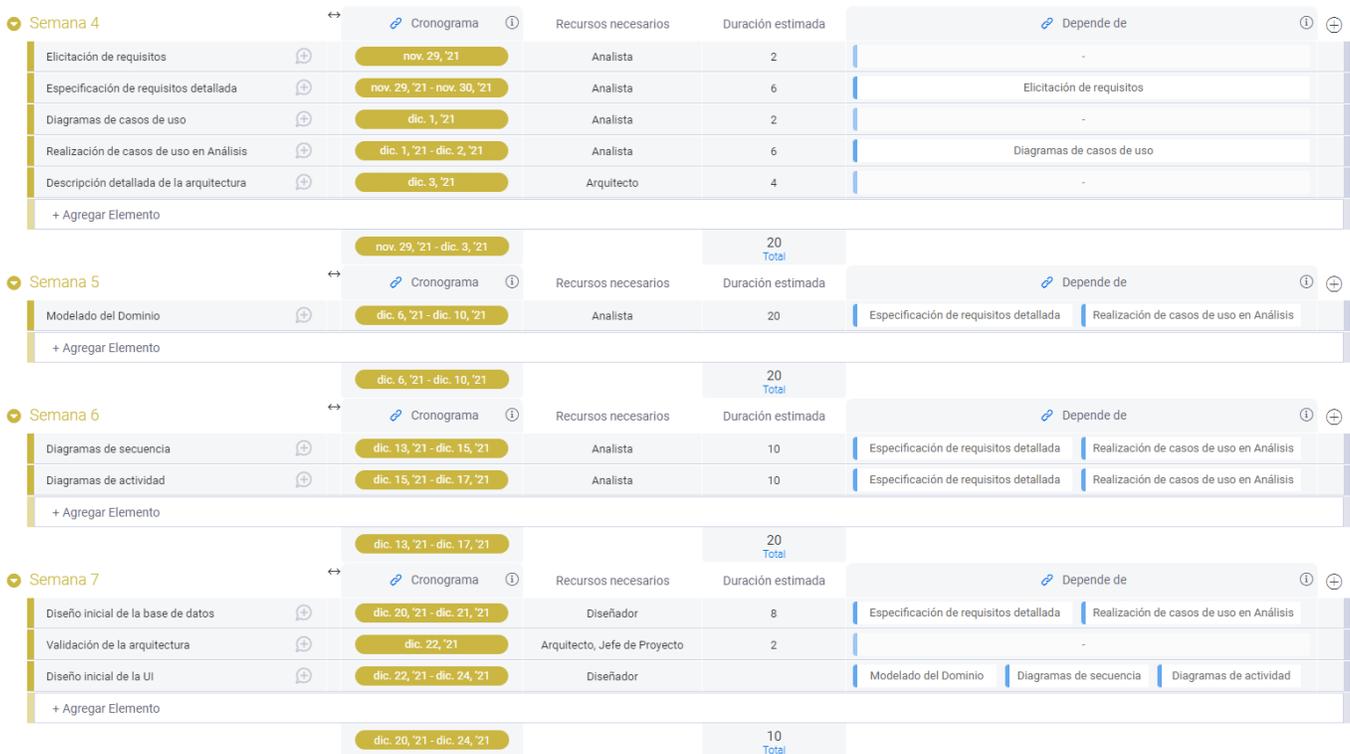


Figura 2.6: Planificación de la Fase de elaboración

FASE DE CONSTRUCCIÓN

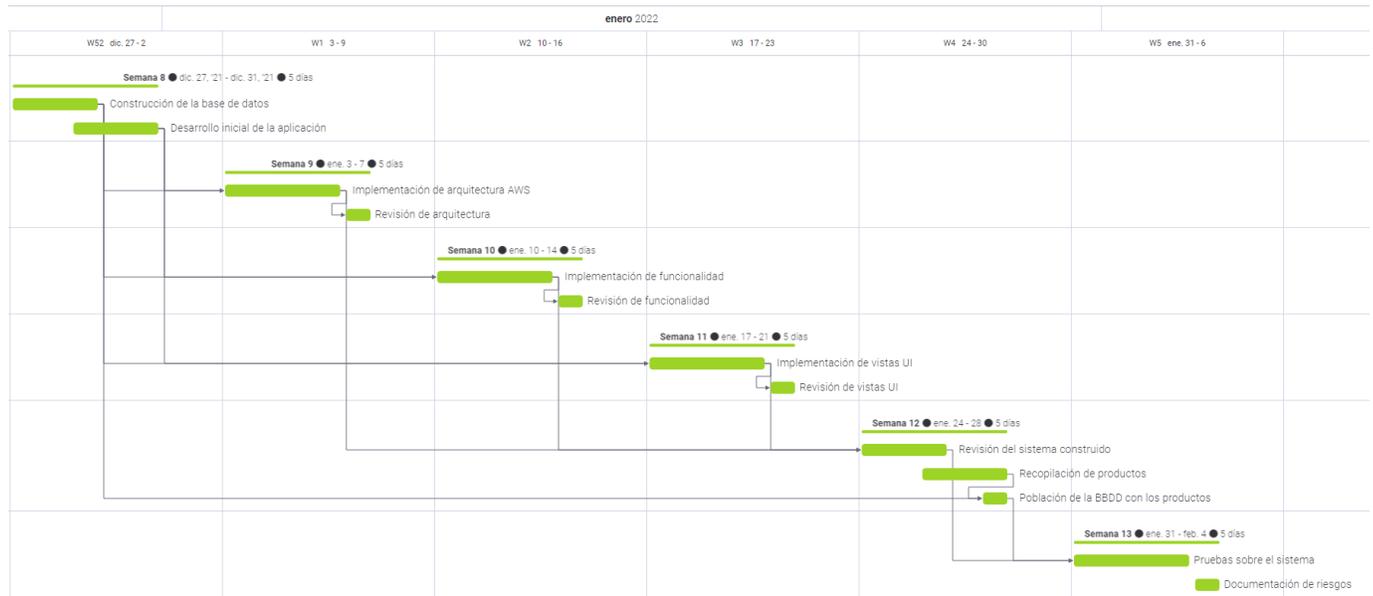


Figura 2.7: Diagrama de Gantt de la Fase de construcción

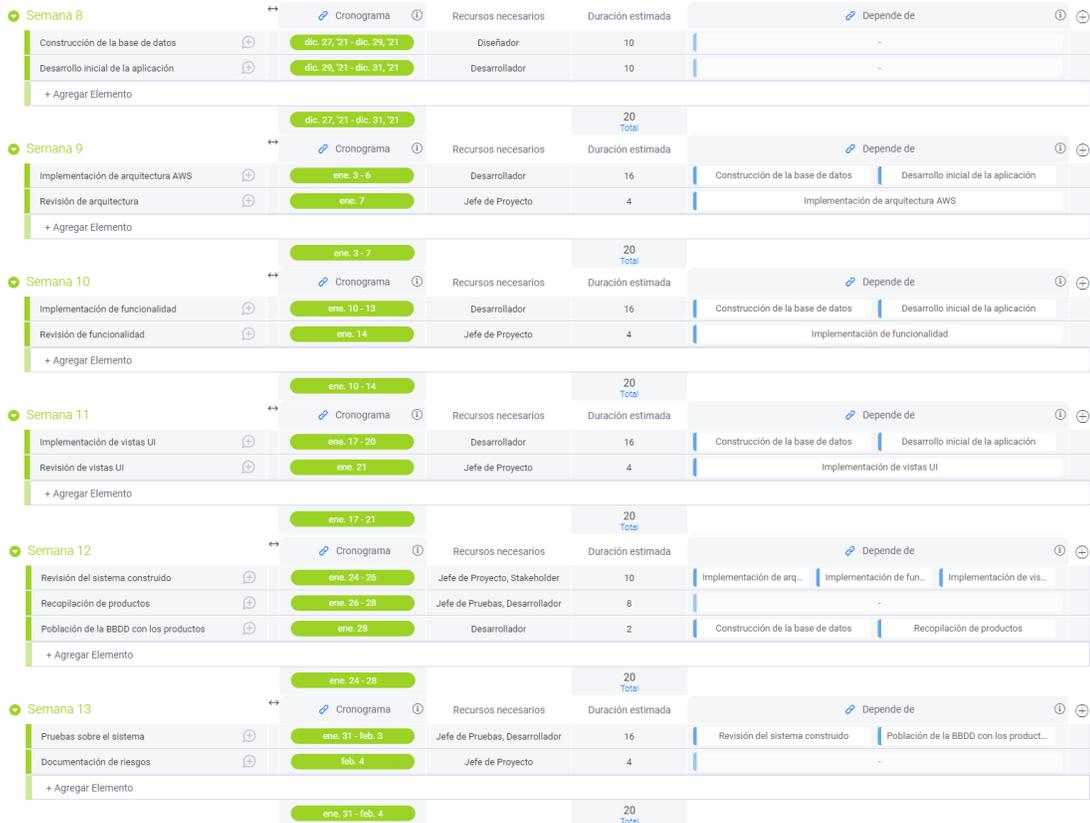


Figura 2.8: Planificación de la Fase de construcción

FASE DE TRANSICIÓN

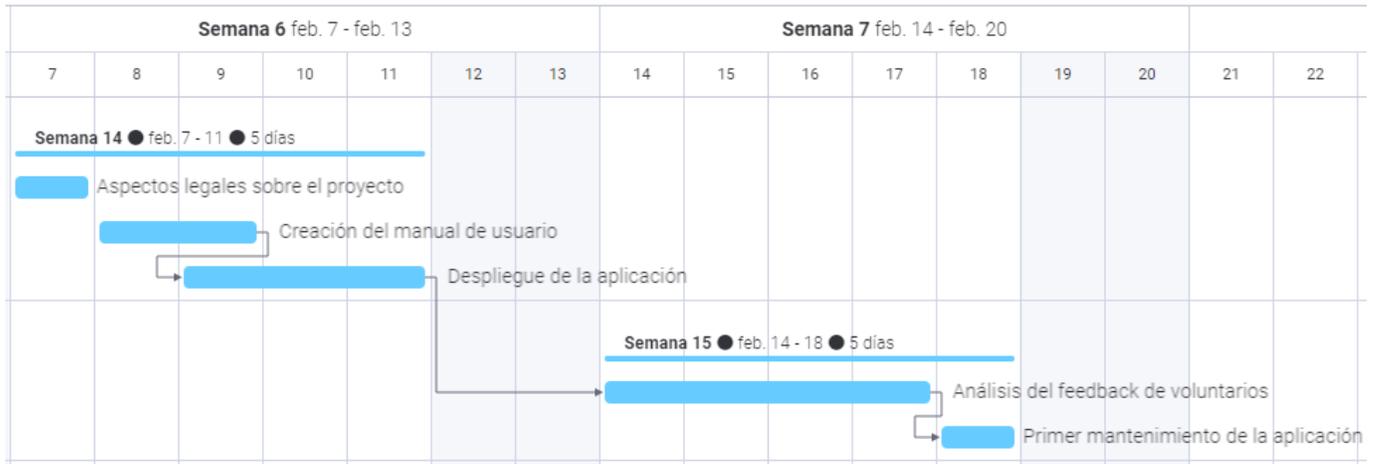


Figura 2.9: Diagrama de Gantt de la Fase de transición



Figura 2.10: Planificación de la Fase de transición

2.4. Plan de gestión de riesgos del proyecto

En cualquier proceso que requiera una planificación previa existen condicionantes que pueden poner en una situación comprometida su correcto desarrollo. Esta posibilidad genera la necesidad de adelantarnos a los posibles eventos que puedan surgir y de disponer de un plan que evite o reduzca las consecuencias negativas que pueda tener su aparición.

El Plan de Gestión de Riesgos pretende definir todas esas posibles adversidades que puedan acontecer, definiendo para ello tanto los detalles del obstáculo o riesgo como los de la solución con la que se tiene la intención de subsanarlo.

Es necesario comenzar por definir qué entendemos por riesgo. Para ello se ha utilizado la comparativa planteada en el libro *Software Project Management* de B. Hughes [5]. En este documento se presentan los riesgos como la combinación de dos definiciones diferentes. La primera, extraída de la guía “PMBOK” [6] considera los riesgos como “una situación o evento desconocido que, en el caso de ocurrir, podría tener un impacto positivo o negativo en los objetivos del proyecto”. Por otro lado, se definen como “la posibilidad de exposición a las consecuencias adversas de un evento futuro”. Ambas, combinadas, nos otorgan una definición concisa de que podemos entender por riesgo. En el presente documento nos acercaremos más a la segunda definición puesto que solo vamos a tener en cuenta aquellas situaciones que en caso de producirse supondrían un obstáculo para alcanzar los objetivos deseados.

Estableciendo la definición, se hace necesario introducir la clasificación de los riesgos que se va a utilizar según los siguientes criterios:

- Probabilidad de ocurrencia: estimación de la probabilidad de que el riesgo mencionado se produzca. Como no se cuenta con experiencia previa en este tipo de proyectos las estimaciones, que se basarán en el conocimiento del equipo sobre su propia forma de trabajo y en las experiencia derivada de la gestión de proyectos más pequeños, pueden resultar inexactas. Por eso la información que se recopile posteriormente en el seguimiento del proyecto a nivel de riesgos será de gran utilidad para proyectos posteriores.
- Impacto generado sobre el proyecto: estimación sobre los daños que la materialización del riesgo pueda ocasionar en el proyecto. Principalmente los daños recaerán en la planificación o en el presupuesto, razones por las cuales aparecerán en este capítulo tanto una planificación y presupuesto iniciales, como un seguimiento de la evolución de los mismos.

En la categorización de los riesgos se ha decidido seguir la siguiente tabla:

Probabilidad de ocurrencia	Descripción
Muy alta	Posibilidades de ocurrencia mayores al 35 %
Alta	Posibilidades de ocurrencia entre el 35 % y el 20 %
Media	Posibilidades de ocurrencia entre el 20 % y el 10 %
Baja	Posibilidades de ocurrencia menores al 10 %

Tabla 2.1: Descripción de las probabilidades de los riesgos

Aunque se ha observado que en otros trabajos como el previamente mencionado [5] se utiliza un rango de posibilidades más flexible, se considera que al estar trabajando en un entorno controlado como es el ámbito académico cualquier riesgo al que se le asigna una probabilidad de ocurrencia superior al 35 % de las ocasiones será considerado como alto.

Para categorizar los riesgos se cuenta con la clasificación planteada en [7] que se basa en Actores, Estructura, Tecnología y Tareas.

Lo siguiente, presentar la tabla que describe como categorizaremos los riesgos según su impacto:

Ámbito	Nivel de impacto	Descripción
Planificación	Muy alto	Acciones cuyo impacto en la planificación pueda retrasar el desarrollo del proyecto el equivalente a más de 16 horas de trabajo
	Alto	Acciones cuyo impacto en la planificación pueda retrasar el desarrollo del proyecto el equivalente a 12 – 8 horas de trabajo
	Medio	Acciones cuyo impacto en la planificación pueda retrasar el desarrollo del proyecto el equivalente a 8 – 4 horas de trabajo
	Bajo	Acciones cuyo impacto en la planificación pueda retrasar el desarrollo del proyecto el equivalente a menos de 4 horas de trabajo
Presupuesto	Muy alto	Acciones cuyo impacto en el presupuesto pueda aumentar los costes en más 20 % del presupuesto inicial
	Alto	Acciones cuyo impacto en el presupuesto pueda aumentar los costes en un 15 % del presupuesto inicial
	Medio	Acciones cuyo impacto en el presupuesto pueda aumentar los costes en un 10 % del presupuesto inicial
	Bajo	Acciones cuyo impacto en el presupuesto pueda aumentar los costes en un 5 % del presupuesto inicial

Tabla 2.2: Categorización del impacto de los riesgos

2.4.1. Gestión de riesgos

A continuación, se enumerarán los riesgos que se han tenido en cuenta en la Fase de inicio del proyecto. Para cada uno de ellos se proporciona una descripción que introduce cuál es la situación que engloba, su categoría, impacto y probabilidad de ocurrencia. También se explicarán las acciones que se van a acometer durante el proyecto para minimizar la exposición a dichos riesgos.

Finalmente se presentará un plan de contingencia que determinará el comportamiento del equipo ante la materialización del riesgo para conseguir que afecte lo menos posible al flujo normal de trabajo.

Riesgo	01 - Planificación no ajustada a la realidad				
Descripción	La poca experiencia con la que se cuenta a la hora de realizar el proyecto puede provocar que las planificaciones no estén ajustadas a la realidad de las capacidades del equipo				
Categoría	Tareas	Probabilidad	Media	Impacto	Medio
Acciones de Mitigación	La planificación se ha elaborado teniendo en cuenta la necesidad de llevar a cabo reajustes durante el proyecto y es por eso que se ha optado por una planificación semanal que garantiza un mayor flexibilidad				
Plan de Contingencia	- Aumentar el número de horas, para subsanar las tareas mal planificadas				

Tabla 2.3: Descripción del riesgo nº1

Riesgo	02 - Retrasos en el desarrollo de las tareas
Descripción	La situación del equipo puede ocasionar que ciertas tareas no puedan ser realizadas dentro de la semana para la que han sido planificadas.
Categoría	Actores Probabilidad Baja Impacto Alto
Acciones de Mitigación	Establecer horarios de trabajo diarios. También se ha contado con días en la semana que en un principio se han planificado como libres pero que pueden utilizarse como días de trabajo si la situación lo requiriese.
Plan de Contingencia	- Priorizar las tareas que cuenten con algún retraso - Aumentar el número de horas que se trabaje durante la semana en cuestión evitando superar las 30 horas semanales

Tabla 2.4: Descripción del riesgo n^o2

Riesgo	03 - Desconocimiento de las tecnologías utilizadas
Descripción	Ciertas tecnologías usadas en el desarrollo resultan totalmente novedosas para los miembros del equipo. Su utilización pueden conllevar imprevistos durante el desarrollo que necesiten tiempo para ser resueltos.
Categoría	Tecnología Probabilidad Alta Impacto Bajo
Acciones de Mitigación	Se han tenido en cuenta espacios de tiempo destinados a que los miembros del equipo se familiaricen con las tecnologías.
Plan de Contingencia	- Reajustar la planificación para encontrar información acerca de la tecnología que pueda resolver los inconvenientes.

Tabla 2.5: Descripción del riesgo n^o3

Riesgo	04 - Fallo en los equipos utilizados para el desarrollo
Descripción	Los equipos informáticos son dispositivos complejos cuyo funcionamiento puede verse comprometido por diversos factores.
Categoría	Tecnología Probabilidad Baja Impacto Muy alto
Acciones de Mitigación	Se van a utilizar sistemas de control de versiones así como almacenamiento en la nube de los documentos para evitar pérdida en caso de fallos.
Plan de Contingencia	- Buscar un equipo de sustitución de forma inmediata. - Recuperar los archivos de los sistemas de almacenamiento en los que se hayan depositado

Tabla 2.6: Descripción del riesgo n^o4

Riesgo	05 - Errores en las conexiones
Descripción	La conexión es un elemento fundamental del desarrollo, especialmente en este caso en el que se van a utilizar servicios en la nube para la mayor parte de componentes. Pueden producirse cortes temporales en la red que imposibiliten el desarrollo de las tareas.
Categoría	Tecnología Probabilidad Baja Impacto Alto
Acciones de Mitigación	Se cuenta con diferentes redes de diferentes proveedores en caso de que alguna falle (por ejemplo la red WIFI o redes móviles 4G)
Plan de Contingencia	- Utilizar una red alternativa a la que está fallando

Tabla 2.7: Descripción del riesgo n°5

Riesgo	06 - Enfermedad de alguno de los miembros del equipo
Descripción	El contexto de pandemia en el que se desarrolla el trabajo aumenta drásticamente las posibilidades de que el equipo caiga enfermo o sea confinado por contacto con algún positivo de COVID-19 u otra enfermedad
Categoría	Tecnología Probabilidad Media Impacto Medio
Acciones de Mitigación	Se han tomado las precauciones sanitarias durante el desarrollo del trabajo que estipula la normativa y protocolos vigentes
Plan de Contingencia	- Proveerse de equipos en caso de confinamiento para poder continuar con el trabajo

Tabla 2.8: Descripción del riesgo n°6

Riesgo	07 - Aparición de aplicaciones equivalentes en el mercado durante el desarrollo
Descripción	El mercado de las aplicaciones móviles está en constante cambio y una aplicación con un concepto similar al planteado puede ser lanzada durante el transcurso del proyecto generando una situación de mercado compleja para el producto
Categoría	Estructura Probabilidad Baja Impacto Muy alto
Acciones de Mitigación	Se han analizado aplicaciones de mercado similares con el objetivo de obtener un factor diferenciador del resto de aplicaciones. En los requisitos iniciales se han planteado ideas de carácter innovador respecto a las aplicaciones ya existentes
Plan de Contingencia	- Establecer nuevos requisitos para diferenciarse de los competidores - Desarrollar la nueva funcionalidad

Tabla 2.9: Descripción del riesgo n°7

Riesgo	08 - Desarrollo incorrecto de artefactos de la metodología propuesta
Descripción	El trabajo semanal y los artefactos generados pueden no cumplir las expectativas de alguno de los miembros del equipo en cuyo caso se podrá sugerir la reedición del contenido de la memoria
Categoría	Tareas Probabilidad Media Impacto Medio
Acciones de Mitigación	Se ha realizado una tarea de documentación previa sobre la metodología y los artefactos que han de generarse.
Plan de Contingencia	- Reestructuración de la planificación con el objetivo de poder seguir las indicaciones que se hayan dado

Tabla 2.10: Descripción del riesgo n°8

Riesgo	09 - La arquitectura no cumple con los estándares derivados de los requisitos
Descripción	El uso de nuevas tecnologías conlleva el riesgo de sobreestimar su alcance. Esto puede derivar en comportamientos y rendimientos inesperados que no satisfagan las condiciones iniciales que se tenían previas al desarrollo del producto.
Categoría	Tecnología Probabilidad Baja Impacto Alto
Acciones de Mitigación	Se ha realizado un estudio previo de alternativas en el mercado para los diversos servicios y tecnologías que se van a usar.
Plan de Contingencia	<ul style="list-style-type: none"> - Reajuste de la funcionalidad - Estudio de alternativas a las tecnologías utilizadas - Incorporación de nuevas tecnologías al proyecto

Tabla 2.11: Descripción del riesgo n°9

Riesgo	10 - Dificultades en el despliegue de la aplicación móvil
Descripción	Los mercados de aplicaciones móviles están restringidos por políticas de usuario estrictas que evitan el mal uso de estas plataformas. Debido a esto las aplicaciones pueden ser rechazadas inicialmente en su puesta en el mercado por el incumplimiento de alguna norma. Esto retrasaría el proyecto ya que sería necesaria una corrección.
Categoría	Tecnología Probabilidad Baja Impacto Medio
Acciones de Mitigación	Se ha tenido en cuenta a la hora de realizar la planificación la necesidad de atender a cuestiones legales de esta índole
Plan de Contingencia	<ul style="list-style-type: none"> - Detectar cuál de las funcionalidades está ocasionando el problema - Modificar esta funcionalidad si fuese posible adaptarla a los requerimientos de la compañía. - En última instancia, eliminar la funcionalidad

Tabla 2.12: Descripción del riesgo n°10

Riesgo	11 - Incapacidad para coordinar a los voluntarios para la carga de datos
Descripción	Con el fin de que las funcionalidades de la aplicación puedan apreciarse de forma completa es necesario un alto volumen de datos inicial que se traduzca en múltiples productos almacenados. Los voluntarios se encargarán de recoger la información de los productos y su carga en la base de datos utilizando un prototipo de la aplicación suministrado por el equipo de desarrollo. Estas tareas requieren de coordinación y pueden no llevarse adelante con la efectividad esperada.
Categoría	Estructura Probabilidad Baja Impacto Alto
Acciones de Mitigación	Planear con antelación los días en los que se vaya a realizar la recogida de información. Dar a los voluntarios instrucciones detalladas de como se va a realizar este proceso.
Plan de Contingencia	- Distribuir la acción en varios días con grupos de personas más pequeños

Tabla 2.13: Descripción del riesgo nº11

Riesgo	12 - Dificultades en la comunicación entre los miembros del equipo por razones de agenda
Descripción	Debido a la situación tanto de la profesora como del alumno es posible que en alguna semana sea inviable la comunicación fluida de los resultados o del estado del proyecto. Esto puede suponer la imposibilidad de avanzar en un cierto aspectos del trabajo, o su retraso con respecto a la planificación inicial.
Categoría	Actores Probabilidad Baja Impacto Bajo
Acciones de Mitigación	Se cuenta con varias vías de comunicación tanto síncrona como asíncrona tales como videoconferencias, reuniones presenciales, correos electrónicos o un chat interno mediante plataformas como Microsoft Teams.
Plan de Contingencia	- En caso de no poder realizar reuniones se utilizarán medios suplementarios como el correo electrónico. - En caso de que uno de los dos integrantes del equipo se encuentre totalmente ocupado se dará prioridad a las tareas que no requieran supervisión o retroalimentación.

Tabla 2.14: Descripción del riesgo nº12

Riesgo	13 - Sobrecostes derivados del mal uso de las herramientas de computación en la nube
Descripción	El proveedor de los servicios que van a usarse para la gestión del servidor cuenta con un sistema de pagos por uso, Dentro de los presupuestos considerados inicialmente se ha realizado una estimación del coste que generará el uso de los servicios necesarios (en su mayor parte gratuitos). Existe la posibilidad de que el desconocimiento por parte del equipo de estos servicios genere costes innecesarios por recursos que no se estén utilizando
Categoría	Estructura Probabilidad Baja Impacto Alto
Acciones de Mitigación	Estas plataformas cuentan con herramientas de gestión de los costes que deben ser consultadas regularmente para detectar posibles anomalías.
Plan de Contingencia	- Sí se detectase un coste derivado de un servicio que no estaba considerado se analizará la necesidad y si es prescindible se eliminarán los recursos que están incrementando la factura

Tabla 2.15: Descripción del riesgo nº13

Riesgo	14 - Dificultades en el desarrollo del sistema propuesto por motivos de conocimiento
Descripción	Las tecnologías que van a utilizarse son totalmente nuevas para el alumno. Lo mismo sucede con el dominio en el que el negocio desarrolla su acción. Esto puede provocar que no se consiga la funcionalidad esperada en el tiempo determinado para ello.
Categoría	Tecnología Probabilidad Alta Impacto Medio
Acciones de Mitigación	Se han tenido en cuenta en la planificación tiempos destinados a la documentación sobre aspectos clave del sistema.
Plan de Contingencia	- Búsqueda de información y alternativas a la funcionalidad que este generando los bloqueos - Cambios en los requisitos en el caso de que la funcionalidad no esté correctamente estimada y su implementación no sea abarcable

Tabla 2.16: Descripción del riesgo nº14

2.5. Seguimiento de la planificación

En este apartado vamos a observar cómo ha evolucionado la planificación detallada en la sección 2.3 como consecuencia de los riesgos de la sección 2.4 que se hayan materializado finalmente.

2.5.1. Cambios sobre la planificación inicial

A continuación se presentarán las correcciones en las fechas de inicio y fin de las diferentes tareas divididas en fases. En los recursos planificados no se han producido cambios porque, como ya se había comentado, solo existe un integrante en el equipo que desempeña los roles además de la tutora, que se encarga de la orientación y supervisión.

Fase de inicio

La fase de inicio se desarrolló de forma correcta en un principio salvo por la parte destinada al estudio de las tecnologías principales que se iban a utilizar en el desarrollo.

Tarea	Fecha de inicio planificada	Fecha de fin planificada	Fecha de inicio real	Fecha de fin real	Motivos del desajuste
Estudio del marco de trabajo Ionic	22/11/2021	24/11/2021	22/11/2021	26/11/2021	RIESGO-3
Estudio de tecnologías AWS	24/11/2021	26/11/2021	26/11/2021	28/11/2021	RIESGO-3

Tabla 2.17: Modificaciones en el calendario para la Fase de Inicio

Fase de elaboración

Durante la fase de elaboración se comenzaron a percibir ciertos desajustes en algunas tareas que pudieron ser corregidos con el trabajo en los días en los que no se había asignado ninguna tarea aunque finalmente al llegar al diseño de la interfaz se produjeron retrasos debidos a las festividades y otros aspectos que retrasaron el final de la fase.

Tarea	Fecha de inicio planificada	Fecha de fin planificada	Fecha de inicio real	Fecha de fin real	Motivos del desajuste
Diagramas de casos de uso	01/12/2021	01/12/2021	01/12/2021	02/12/2021	RIESGO-1
Realización de casos de uso en análisis	01/12/2021	02/12/2021	03/12/2021	06/12/2021	RIESGO-1
Modelado del dominio	06/12/2021	10/12/2021	06/12/2021	12/12/2021	RIESGO-8
Diagramas de secuencia en análisis	13/12/2021	15/12/2021	13/12/2021	17/12/2021	RIESGO-1
Diagramas de actividad	15/12/2021	17/12/2021	-	-	RIESGO-17
Diseño inicial de la interfaz de usuario	22/12/2021	24/12/2021	22/12/2021	29/12/2021	RIESGO-16

Tabla 2.18: Modificaciones en el calendario para la Fase de Elaboración

Fase de construcción

Los mayores desajustes aparecieron durante la Fase de construcción. Hubo diferentes motivos para que una fase planeada con un mes de duración haya terminado abarcando casi cinco meses de desarrollo. Entre ellos se encuentran la falta de tiempo por incorporación a la empresa en la que se han desarrollado las prácticas, previsiones incorrectas del tiempo que iban a tomar ciertas tareas fruto de la inexperiencia del desarrollador y periodos de exámenes en los que no se compaginó eficientemente el desarrollo con el estudio de otras asignaturas.

Tarea	Fecha de inicio planificada	Fecha de fin planificada	Fecha de inicio real	Fecha de fin real	Motivos del desajuste
Construcción de la base de datos	27/12/2021	29/12/2021	02/01/2022	25/03/2022	RIESGO-14 RIESGO-15
Desarrollo inicial de la aplicación	29/12/2021	31/12/2021	04/01/2022	06/01/2022	RIESGO-2
Implementación de la arquitectura AWS	03/01/2022	06/01/2022	06/01/2022	25/03/2022	RIESGO-9 RIESGO-14 RIESGO-15
Revisión de la arquitectura	07/01/2022	07/01/2022	25/03/2022	26/03/2022	RIESGO-2
Implementación de la funcionalidad	10/01/2022	13/01/2022	10/01/2022	12/04/2022	RIESGO-14 RIESGO-15
Revisión de la funcionalidad	14/01/2022	14/01/2022	12/04/2022	12/04/2022	RIESGO-2
Implementación de las vistas UI	17/01/2022	20/01/2022	05/02/2022	12/04/2022	RIESGO-15
Revisión de las vistas UI	21/01/2022	21/01/2022	12/04/2022	20/04/2022	RIESGO-2
Revisión del sistema construido	24/01/2022	26/01/2022	20/04/2022	25/04/2022	RIESGO-2
Recopilación de productos	26/01/2022	28/01/2022	04/01/2022	25/04/2022	RIESGO-2
Población de la BBDD	28/01/2022	28/01/2022	04/01/2022	25/04/2022	RIESGO-2
Pruebas sobre el sistema	31/01/2022	03/02/2022	04/01/2022	29/04/2022	RIESGO-14 RIESGO-15
Documentación de riesgos	04/02/2022	04/02/2022	02/05/2022	02/05/2022	RIESGO-2

Tabla 2.19: Modificaciones en el calendario para la Fase de Construcción

Fase de transición

Afectada por los problemas surgidos durante el desarrollo la Fase de transición también se ha visto notablemente retrasada en el tiempo aunque a nivel de días de duración se ha visto reducida al ser eliminados el despliegue de la aplicación en plataformas de mercado de aplicaciones móviles y los mantenimientos derivados por falta de tiempo.

Tarea	Fecha de inicio planificada	Fecha de fin planificada	Fecha de inicio real	Fecha de fin real	Motivos del desajuste
Aspectos legales sobre el proyecto	07/02/2021	07/02/2021	03/05/2022	03/05/2022	RIESGO-2
Creación del manual de usuario	08/02/2021	09/02/2021	03/05/2022	07/05/2022	RIESGO-2
Despliegue de la aplicación	09/02/2021	11/02/2021	-	-	RIESGO-17
Análisis del feedback de voluntarios	14/02/2021	17/02/2021	10/05/2022	12/05/2022	RIESGO-2
Primer mantenimiento de la aplicación	18/02/2021	18/02/2021	-	-	RIESGO-17

Tabla 2.20: Modificaciones en el calendario para la Fase de Transición

2.5.2. Monitorización de riesgos

En esta sección se aportará información sobre los riesgos que han amenazado la consecución del proyecto y se extraerán conclusiones acerca de qué errores de la planificación han facilitado su aparición. Los riesgos que han acontecido durante el desarrollo y figuraban en el plan de gestión de riesgos son los siguientes:

- **Planificación no ajustada a la realidad (RIESGO-1):** la planificación realizada no ha tenido en cuenta ajustadamente la poca experiencia del alumno en un desarrollo de este alcance.
- **Retrasos en el comienzo de las tareas (RIESGO-2):** han existido algunas tareas cuyo inicio ha tenido lugar más tarde de lo previsto. No se tuvo en cuenta a la hora de planificar el reservar días sin trabajo planificado para propiciar un banco de tiempo que supliese los días en los que no se ha podido trabajar en el producto.
- **Desconocimiento de las tecnologías utilizadas (RIESGO-3):** Se minusvaloraron, cuando se inició el proyecto, las complicaciones que puede tener adquirir un nivel de principiante en ciertas tecnologías. La pronunciada curva de aprendizaje de Angular y AWS conllevó dedicar más tiempo del esperado a recopilar información.
- **Desarrollo incorrecto de artefactos de la metodología (RIESGO-8):** otra cuestión que no se tuvo en cuenta a la hora de planificar las tareas fue la posibilidad de que un artefacto no se generase con la cualidad adecuada en el tiempo programado.
- **La arquitectura no cumple con los estándares previstos (RIESGO-9):** es complicado implementar en el primer desarrollo con una tecnología, un programa que tenga el rendimiento como su máxima. Suele ser uno de los aspectos que se perfeccionan con la práctica y en el caso de la presente aplicación se han necesitado tiempos de estudio de soluciones alternativas para algunos puntos de la aplicación cuyo diseño penalizaba el rendimiento.

En cualquier proyecto aparecen riesgos no tenidos en cuenta a la hora de plantear cómo distribuir las tareas. Estos riesgos son los que más peligro pueden generar ya que no existen planes de prevención que eviten sus daños y una vez se producen deben improvisarse soluciones. Los riesgos de este estilo que se produjeron durante el desarrollo fueron:

- **Contratación del alumno por parte de la empresa de prácticas (RIESGO-15):** un elemento crucial, aunque no el único, para entender por qué el desarrollo se alargó más de lo previsto en el tiempo es la incorporación a la plantilla de la empresa del único miembro desarrollador del equipo, con el correspondiente aumento de horas diarias que era necesario dedicar al trabajo nada más terminar las prácticas en empresa en el mes de febrero. Ya se venían experimentando ciertos retrasos previos por la compaginación que se analiza en el siguiente punto, pero esto requirió un reajuste total del programa que se había planteado para la distribución de las horas.
- **Dificultades para compaginar asignaturas con el desarrollo del producto (RIESGO-16):** cuando se comenzó a planificar las tareas se tuvieron en cuenta las asignaturas que el alumno estaba cursando en el momento y se realizaron previsiones demasiado optimistas de cómo se iban a compaginar las horas de trabajo en otras asignaturas del grado con las que se dedicaban a la creación de la aplicación.
- **Falta de tiempo para el desarrollo de algunas tareas (RIESGO-17):** algunos de los elementos a los que se pretendía dar forma han tenido que ser eliminados, ya que asumirlos excedía el alcance de un trabajo de este tipo. Se han eliminado los aspectos menos relacionados con la Ingeniería de Software y más prescindibles como los que tienen que ver con el despliegue en los mercados de aplicaciones móviles.

2.6. Presupuesto del proyecto

Durante la ejecución de cualquier proyecto debemos contar como un factor determinante del éxito del mismo el cumplimiento de los objetivos económicos que este tuviese a no ser que se desarrolle de forma altruista. En nuestro caso se va a realizar una estimación de los costes derivados de los recursos que se van a utilizar para crear el sistema. En las siguientes secciones presentaremos los recursos y los costes asociados a los mismos. Luego se presentará la planificación de costes que se diseñó en las primeras fases del proyecto y se terminará con la corrección de este presupuesto inicial y las conclusiones extraídas. Se ha calculado el coste proporcional en base a que el trabajo de 300 horas según la planificación determinada equivale a 75 días para aquellos costes que deben ajustarse al periodo de trabajo.

2.6.1. Recursos necesarios

Recursos de personal

Tradicionalmente, los equipos que gestionan desarrollan proyectos *software* están configurados por diferentes miembros que desempeñan distintos roles en la empresa. En nuestro caso el alumno como único integrante del equipo de desarrollo deberá alternarse en el ejercicio de las tareas asignadas a los diferentes roles. Para conocer los gastos asociados a cada posición se ha recurrido al estudio de remuneraciones de la consultora Michael Page [10] para el año de realización de este trabajo tomando los valores de la Comunidad de Madrid y con la experiencia mínima recogida.

- Jefe de Proyecto: 35.000€/año
- Analista: 30.000€/año
- Desarrollador: 21.000€/año

Recursos tecnológicos

- Ionic Framework: plan gratuito (0€)
- AWS Amplify: capa gratuita (0€)
- GitHub: licencia de estudiante (Precio normal para empresas 220€/año)

- Visual Studio Code: *software* libre (0€)
- Astah: licencia proporcionada por la universidad (Precio normal 1588€/año)
- Microsoft Teams: 120€/mes por usuario

Recursos materiales

Existen diferentes gamas para cada uno de los productos necesarios así que se tomará el precio que se estima suficiente para adquirir productos que puedan satisfacer las necesidades del equipo en un desarrollo *software* común. Se ajustarán según los 75 días de trabajo planificados dentro del tiempo de vida total de los productos.

Recurso	Precio(€)/unidad	Tiempo de vida (días)	Coste proporcional (€)
Ordenador	500	1095	34
Monitor	100	730	10
Teclado	50	730	5
Ratón	50	365	10
Dispositivo Móvil	200	365	41
Mesa de Oficina	100	180	41
Total (€)			141

Tabla 2.21: Costes asociados a los recursos materiales utilizados

Recursos de localización

El tiempo planificado para el desarrollo supondrá alquilar una oficina durante al menos tres meses.

- Alquiler Oficina: 500€/mes

Recursos de infraestructura

Para los recursos de infraestructura existen numerosas ofertas que dependen de la situación del mercado. Se ajustarán al tiempo planificado que supondría contratarlos durante tres meses.

- Calefacción: 1500€/año
- Elementos de protección sanitaria: 30€/mes
- Conexión de red: El precio medio de las conexiones consultadas es 50€/mes
- Agua: 20€/mes
- Luz: 150€/MWh con un consumo mensual de los hogares españoles de 270KWh

2.6.2. Presupuesto inicial

Con los precios establecidos anteriormente para cada uno de los elementos va a calcular el presupuesto estimado del proyecto. Para ello se tendrá en cuenta el sueldo anual medio para un desarrollador puesto que se está simulando lo que se presupuestaría en una contratación como autónomo.

El presupuesto calculado puede verse en detalle en la Tabla 2.22

2.6.3. Presupuesto final

Las modificaciones llevadas a cabo tienen que ver con el tiempo de desarrollo. Los recursos de infraestructura se han reajustado a los días de desarrollo que finalmente ha implicado el proyecto (150 días de forma estimada), aunque el coste de los activos materiales se ha mantenido.

También se produjeron algunos costes asociados a servicios de AWS que no se controlaron de la manera esperada y supusieron un leve desvío sobre el coste cero presupuestado para estos recursos. El presupuesto calculado puede verse en detalle en la Tabla 2.22.

Recurso	Coste Base	Coste proporcional	Recurso	Coste Base	Coste proporcional
Recursos de personal					
Personal (media)	21.000€/año	4310€	Personal (media)	21.000€/año	8600€
Recursos tecnológicos					
Ionic Framework	0€	0€	Ionic Framework	0€	0€
AWS Amplify	0€ + variable	0€	AWS Amplify	0€ + variable	10€
GitHub	220€/año	55€	GitHub	220€/año	128€
Visual Studio Code	0€	0€	Visual Studio Code	0€	0€
Astah	1588€/año	397€	Astah	1588€/año	926€
Microsoft Teams	120€/mes	360€	Microsoft Teams	120€/mes	840€
Recursos materiales					
Ordenador	500€/u	34€	Ordenador	500€/u	34€
Monitor	100€/u	10€	Monitor	100€/u	10€
Teclado	50€/u	5€	Teclado	50€/u	5€
Ratón	50€/u	10€	Ratón	50€/u	10€
Dispositivo móvil	200€/u	41€	Dispositivo móvil	200€/u	41€
Mesa de Oficina	100€/u	41€	Mesa de Oficina	100€/u	41€
Recursos de localización					
Alquiler Oficina	500€/mes	1500€	Alquiler Oficina	500€/mes	3500€
Recursos de infraestructura					
Calefacción	1500€/año	308€	Calefacción	1500€/año	620€
Elementos de protección sanitaria	30€/mes	90€	Elementos de protección sanitaria	30€/mes	150€
Conexión de red	50€/mes	150€	Conexión de red	50€/mes	350€
Agua	20€/mes	49€	Agua	20€/mes	99€
Luz	0,15€/KWh	45€	Luz	0,2€/KWh	60€
Coste total presupuestado		7405€	Coste total final		15450€

Tabla 2.22: Comparación entre presupuestos inicial y final del proyecto

Capítulo 3: Análisis

3.1. Introducción

Durante este capítulo se va a profundizar en todos los artefactos que componen la etapa de análisis del proyecto y en la documentación correspondiente a los diagramas iniciales que modelan el problema que se quiere solucionar y la forma en la que se plantea esta solución

3.2. Actores y Roles del sistema

Craig Larman, en su libro “Applying UML and Patterns”, [9] define los actores como elementos con un comportamiento específico que forman parte de los casos de uso de nuestro sistema. En función de la importancia que adquieran dentro de ese caso de uso los actores suelen clasificarse en principales y secundarios. Los diferentes papeles que los actores pueden desempeñar en los casos de uso se conocen como roles.

3.2.1. Actores Principales

El actor principal de nuestro sistema es el usuario. Podemos definir al usuario como el cliente que utiliza la aplicación para subir productos que le interesen de supermercados cercanos y que consume la información de productos generados por sus iguales. Su función principal varía según el rol que esté desempeñando en cada momento. Los roles que puede desempeñar este actor son los siguientes:

- **Productor de contenido:** engloba todo aquello relacionado con la recopilación de información sobre un producto que otros usuarios pueden utilizar en un futuro. Principalemente conlleva añadir productos al catálogo general (o modificar los añadidos previamente si sus detalles cambian), notificar al resto de clientes de la existencia de ofertas sobre algún producto y relacionarlos.
- **Consumidor de contenido:** es el rol que incluye el manejo principal de la aplicación y el uso de las funcionalidades que se alimentan de productos aportados por la comunidad de clientes. Las acciones principales que se engloban en este rol serían: guardar productos, añadir productos a la lista de la compra, observar ofertas o visualizar los detalles de un producto

3.3. Requisitos

En un proyecto real en el que el trabajo se formaliza mediante un contrato entre el cliente y los desarrolladores, este documento será consultado cada vez que surja un conflicto entre ambas partes por la funcionalidad que se esperaba en contraste a la obtenida. El contrato debe ser capaz de resolver estos desacuerdos y por lo tanto, ser lo más concreto y detallado posible.

Para la clasificación de los requisitos del sistema se va a utilizar la planteada en la metodología UP según se recoge en el libro previamente mencionado de Craig Larman [9], que los organiza en las siguientes categorías:

- **Funcionales (*Functional*):** características que implementa el sistema junto con otros elementos como la seguridad.
- **Usabilidad (*Usability*):** engloban todos aquellos que están destinados a mejorar la capacidad de los usuarios de adaptarse al manejo del sistema. Se incluyen en este grupo los relacionados con documentación o ayudas disponibles.
- **Fiabilidad. (*Reliability*):** relacionados con la capacidad de evitar, reducir o sobreponerse a los posibles fallos que se produzcan en el sistema.
- **Rendimiento. (*Performance*):** engloban métricas que determinan el buen comportamiento del sistema en momentos de estrés como pueden ser los tiempos de respuesta o el uso de recursos.
- **Mantenimiento. (*Supportability*):** relacionados con el soporte del sistema.
- **+**: Bajo esta categoría se incluyen todos los requisitos relacionados con las tecnologías de desarrollo, las interfaces de usuario, el hardware y aspectos legales.

A este grupo de requisitos se los conoce por las siglas FURPS+, que recogen la inicial de sus nombres en inglés. Primero se describirán aquellos relacionados con la funcionalidad y la información almacenada por el sistema. A continuación se describirán los requisitos no funcionales, que podrían nominarse mediante las siglas URPS+.

3.3.1. Requisitos funcionales

- RF-1: Registro de usuarios: el sistema debe permitir a los nuevos usuarios registrarse otorgando toda la información detallada en el requisito de información correspondiente [RFI-1]
- RF-2: Confirmación de registro: el sistema incorporará una validación de los usuarios recién registrados mediante un mensaje al correo electrónico utilizado en el registro
- RF-3: Inicio de sesión: el sistema debe permitir a los usuarios registrados iniciar sesión utilizando el correo electrónico y contraseña proporcionados durante el registro.
- RF-4: Funcionalidad restringida: la funcionalidad de los usuarios no identificados quedará completamente restringida a las funcionalidades de registro e inicio de sesión.
- RF-5: Cierre de sesión: el sistema debe ofrecer la posibilidad a los usuarios de cerrar su sesión en el momento en el que lo deseen
- RF-6: Modificación de información: el sistema debe permitir a los usuarios modificar la información que se ha guardado de ellos en el sistema.
- RF-7: Eliminación de información: el sistema debe permitir a los usuarios eliminar la información sobre los mismos que se desee
- RF-8: Eliminación de perfil: el sistema debe permitir a los usuarios eliminar completamente su perfil del sistema junto a toda su información.
- RF-9: Consultar detalles de un producto: el sistema debe permitir a los usuarios obtener la información almacenada sobre un determinado producto. Esta información es la recogida en el requisito de información correspondiente [RFI-2]
- RF-10: Registro de un nuevo producto: el sistema debe implementar un formulario para el registro de nuevos productos en la aplicación provenientes de una tienda de productos alimenticios real utilizando la información recogida en el requisito de información correspondiente [RFI-2].

- RF-11: Búsqueda de un producto: el sistema debe permitir realizar la búsqueda de productos dentro del catálogo de la aplicación en base al escaneo de su código de barras o en diversos criterios como el nombre del producto buscado, la tienda del producto buscado, la categoría del producto buscado y/o las palabras clave relacionadas con el producto buscado.
- RF-12: Resultados de las búsquedas de productos: los resultados disponibles a una determinada búsqueda se mostraran en orden decreciente de coincidencia con el criterio establecido en el caso de que más de un producto coincidiese con el mismo.
- RF-13: Recomendaciones de productos: el sistema debe proporcionar alternativas a un producto cuyos detalles el usuario este consultando.
- RF-14: Valoración de productos: el sistema deberá permitir a los usuarios identificados valorar un producto estableciendo una nota para el mismo entre 1 y 5
- RF-15: Modificación del catálogo de productos: el sistema deberá permitir a los usuarios modificar o eliminar la información de los productos que hayan sido añadidos por el mismo a la aplicación
- RF-16: Modificaciones temporales del precio: el sistema debe proveer a los usuarios de un método de actualización del precio de los productos registrados de forma temporal (oferta).
- RF-17: Reporte de un error en producto: el sistema debe permitir a los usuarios reportar un producto cuya información consideren errónea añadiendo a este reporte la categoría del reporte.
- RF-18: Histórico de precios: el sistema deberá guardar los precios que ha tenido un producto incluyendo y diferenciando los precios regulares y de oferta
- RF-19: Lista de la compra: el sistema deberá implementar una lista de la compra con el precio de los productos seleccionados dentro del catálogo, agrupados por el establecimiento en el que se encuentren
- RF-20: Mejoras de lista de la compra: el sistema deberá recomendar al usuario mejoras en la lista de la compra basadas en los productos que haya seleccionado y las ofertas que se encuentren activas en ese momento en el sistema
- RF-21: Guardado de listas de la compra: el sistema deberá permitir a los usuarios guardar el estado de una lista de la compra por la fecha en la que se haya creado. Se entiende como estado los productos que la componen junto con el número de unidades y precios
- RF-22: Lista de productos de interés: el sistema deberá proveer a los usuarios de una forma de acceso rápido a los productos que les han resultado de interés mediante una lista de productos a la que pueden añadirse o eliminarse productos durante el uso de la aplicación
- RF-23: Mapa de tiendas cercanas: el sistema deberá mostrar al usuario las tiendas que se sitúan en un radio de kilómetros variable según su propia ubicación siempre que estas estén registradas en el sistema

3.3.2. Requisitos funcionales de baja prioridad

- RF-24: Estado de los productos mostrados: el sistema deberá permitir al usuario conocer el estado (ofertado o reportado) de un producto durante la navegación.
- RF-25: Radio de búsqueda de tiendas: el sistema deberá informar al actor del radio concreto en el que ha mostrado las tiendas cercanas en base a su posición
- RF-26: Categorías de productos visibles: el sistema deberá mostrar al usuario la categoría de un producto junto a su nombre durante el uso de la aplicación
- RF-27: Tiendas de los productos visibles: el sistema deberá mostrar al usuario la tienda en la que encontrar un producto junto a su nombre durante el uso de la aplicación
- RF-28: Modo oscuro: el sistema deberá proveer a los usuarios de un modo oscuro que cambie la paleta de colores de la aplicación para los usuarios que así lo deseen

3.3.3. Requisitos funcionales de información

RFI-1: Usuario - La información relacionada con los usuarios que debe mantenerse en el sistema es:

- Nombre de usuario
- Correo electrónico
- Contraseña

RFI-2: Producto - La información relacionada con los productos que debe mantenerse en el sistema es:

- EAN (European Article Number)
- Marca
- Nombre
- Establecimiento en el que se encuentra
- Cadena a la que pertenece el establecimiento, si existiese
- Precio
- Tipo de precio
- Categoría
- Imágenes
- Autor

RFI-3: Ofertas - La información relacionada con los niveles de prestigio que debe mantenerse en el sistema es:

- Producto al que afecta
- Precio por unidad anterior a la oferta
- Precio por unidad con la oferta
- Descripción de la oferta
- Fecha de comienzo
- Fecha de fin

3.3.4. Requisitos no funcionales

RNF-1: El sistema debe ser accesible desde aplicaciones con sistema operativo Android

RNF-2: El sistema deberá utilizar OpenStreetMaps para la funcionalidad que implemente geolocalización y mapas

RNF-3: El sistema deberá garantizar que los nuevos usuarios deben poder comprender y utilizar el 95 % de las funciones pasada la primera media hora de uso

RNF-4: El sistema debe proveer funcionalidad constante las 24 horas al día y los 365 días del año.

RNF-5: El sistema deberá poder arrancar en un tiempo inferior a 7 segundos

RNF-6: El sistema deberá asegurar un tiempo de respuesta nunca superior a 3 segundos

RNF-7: El sistema deberá usar alguna metáfora visual de cara al usuario para identificar los tiempos de procesamiento

RNF-8: El sistema deberá permitir en todo momento la trazabilidad de los datos de los usuarios que puedan ser considerados sensibles

RNF-9: El sistema deberá implementar los procesos de inicio de sesión mediante las tecnologías provistas por Cognito (Amazon Web Services)

- RNF-10: El sistema deberá utilizar para la persistencia DynamoDB (Amazon Web Services)
- RNF-11: El sistema deberá asegurar que los usuarios podrán registrarse desde la app móvil y la app web
- RNF-12: El sistema, cuando se elimine un perfil, deberá eliminar toda la información del mismo manteniendo los productos aportados en un plazo máximo de 24 horas
- RNF-13: El sistema utilizará EAN13 como formato aceptado para los códigos de barras de los productos
- RNF-14: El sistema destacará en su interfaz los productos ofertados y reportados mediante colores distinguibles entre ellos y a los de los productos en estado normal

3.4. Casos de uso

3.4.1. Presentación de los casos de uso

- CU-1: Registrarse
- CU-2: Iniciar sesión
- CU-3: Cerrar sesión
- CU-4: Eliminar perfil
- CU-5: Modificar información de perfil
- CU-6: Visualizar detalles de un producto
- CU-7: Publicar producto
- CU-8: Buscar producto en el catálogo por criterios de búsqueda
- CU-9: Buscar productos en el catálogo por escaneo de código de barras
- CU-10: Consultar lista de la compra
- CU-11: Añadir producto a la lista de la compra
- CU-12: Eliminar producto de la lista de la compra
- CU-13: Guardar la lista de la compra
- CU-14: Modificar cantidad de un producto de la lista de la compra
- CU-15: Modificar información de un producto
- CU-16: Valorar producto
- CU-17: Añadir oferta
- CU-18: Reportar producto
- CU-19: Consultar histórico de precios
- CU-20: Marcar un producto como guardado
- CU-21: Desmarcar un producto como guardado
- CU-22: Consultar lista de productos guardados

3.4.2. Diagrama de casos de uso

En la Figura 3.1 se muestran los casos de uso del sistema (agrupados según funcionalidad para facilitar la comprensión del diagrama)

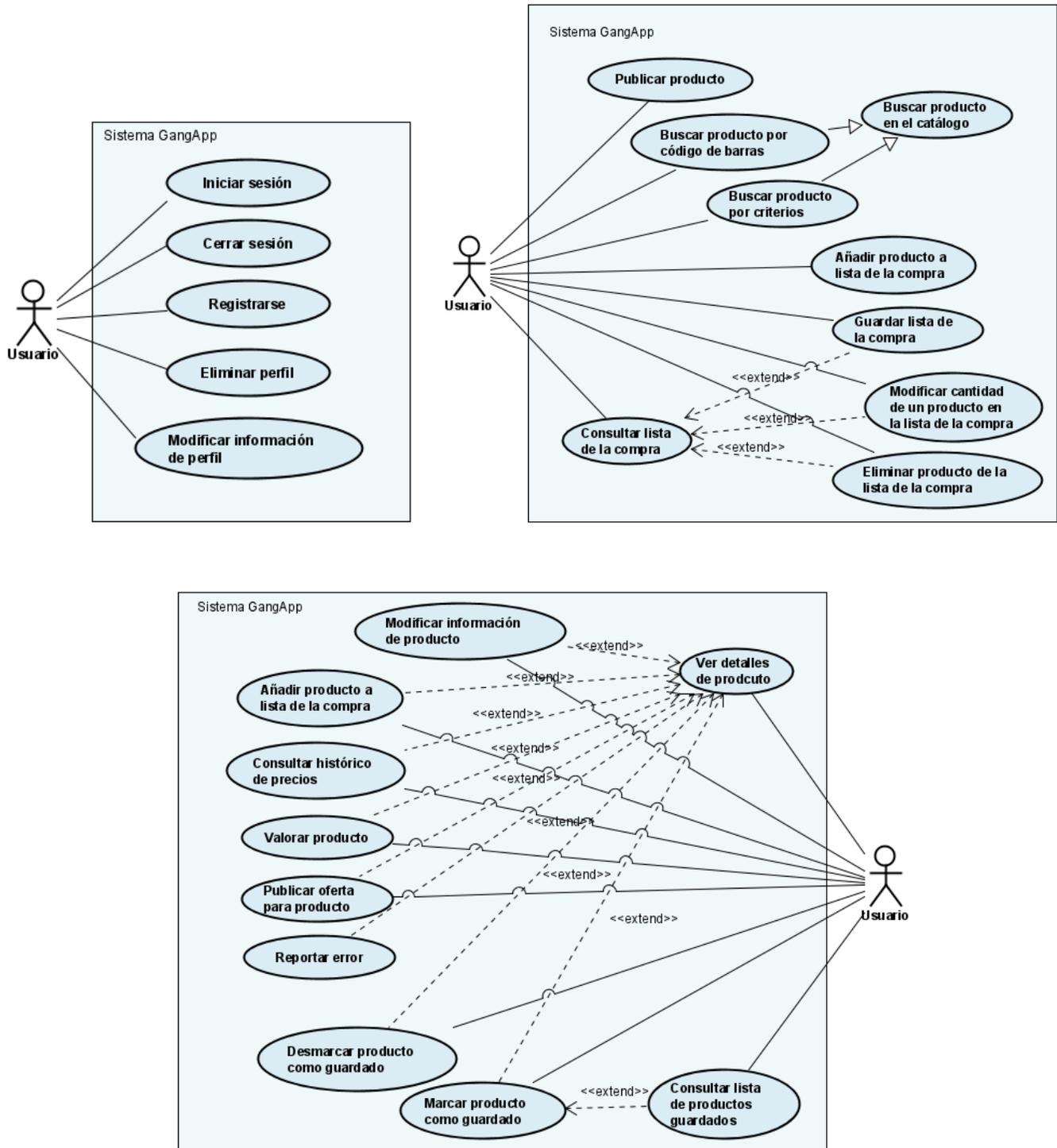


Figura 3.1: Diagramas de casos de uso

3.4.3. Descripción detallada de los casos de uso

CU01 - Registrarse	
Actor	Usuario
Resumen	Un nuevo usuario quiere crear una cuenta en el sistema que le permita identificarse y acceder a la funcionalidad restringida
Secuencia principal	
Precondiciones	El actor ha accedido al sistema mediante la aplicación móvil o web
	<ol style="list-style-type: none"> 1. El actor comienza el proceso de registro 2. El sistema solicita nombre, correo electrónico y contraseña del actor 3. El actor introduce los datos solicitados 4. El sistema valida el formato de los datos solicitados 5. El sistema solicita confirmación para el registro 6. El actor confirma el registro 7. El sistema comprueba que el correo electrónico no ha sido usado en otro registro 8. El sistema informa al usuario del éxito en el registro 9. El caso de uso finaliza
Postcondiciones	El actor queda registrado en el sistema mediante su correo electrónico y cuenta con acceso a toda la funcionalidad
Secuencia alternativas	
3A, 6A	<ol style="list-style-type: none"> 1. El actor cancela el proceso de inicio de sesión y el caso de uso queda sin efecto
Excepciones	
4A	<ol style="list-style-type: none"> 1. El sistema comprueba que el formato de los datos introducidos es incorrecto 2. El sistema informa al actor 3. El caso de uso continua en el paso 3
7A	<ol style="list-style-type: none"> 1. El sistema comprueba que el correo electrónico ya ha sido utilizado en el registro de otro usuario 2. El sistema informa al actor 3. El caso de uso continua en el paso 3

Tabla 3.1: Detalles del caso de uso “Registrarse”

CU02 - Iniciar Sesión	
Actor	Usuario
Resumen	Un usuario pretende utilizar la aplicación y lo primero que debe hacer es identificarse ante el sistema para poder acceder a su funcionalidad
Secuencia principal	
Precondiciones	El actor está previamente registrado en el sistema
	<ol style="list-style-type: none"> 1. El actor comienza el proceso de iniciar sesión 2. El sistema solicita el correo electrónico y la contraseña 3. El actor introduce las credenciales 4. El sistema valida el formato de los datos introducidos 5. El sistema solicita confirmación 6. El actor confirma el inicio de sesión 7. El sistema comprueba que las credenciales se corresponden con un usuario registrado 8. El sistema informa del éxito en el proceso de inicio de sesión 9. El sistema carga la información de sesión del usuario 10. El sistema muestra la pantalla principal de la aplicación 11. El caso de uso finaliza
Postcondiciones	El actor tiene acceso a la funcionalidad del sistema y su información de sesión está cargada en el sistema
Secuencia alternativas	
3A, 6A	<ol style="list-style-type: none"> 1. El actor cancela el proceso de inicio de sesión y el caso de uso queda sin efecto
Excepciones	
4A	<ol style="list-style-type: none"> 1. El sistema comprueba que el formato de los datos introducidos es incorrecto 2. El sistema informa al actor 3. El caso de uso continua en el paso 3
7A	<ol style="list-style-type: none"> 1. El sistema comprueba que las credenciales no se corresponden con ningún usuario registrado 2. El sistema informa al actor 3. El caso de uso continua en el paso 3

Tabla 3.2: Detalles del caso de uso “Iniciar sesión”

CU03 - Cerrar Sesión	
Actor	Usuario
Resumen	Un usuario que previamente ha iniciado sesión quiere cerrar la misma después del uso de la aplicación
Secuencia principal	
Precondiciones	El actor tiene una sesión activa en el sistema
<ol style="list-style-type: none"> 1. El actor solicita cerrar sesión 2. El sistema solicita confirmación del proceso 3. El actor confirma el cierre de sesión 4. El sistema cierra la sesión y elimina la información sobre el usuario que se haya cargado para la misma 5. El caso de uso finaliza 	
Postcondiciones	La sesión del actor ha sido cerrada y no podrá accederse a la funcionalidad hasta el siguiente inicio de sesión
Secuencia alternativas	
3A	1. El actor cancela el proceso de cerrar sesión y el caso de uso queda sin efecto
Excepciones	
No hay	

Tabla 3.3: Detalles del caso de uso “Cerrar sesión”

CU04 - Eliminar perfil	
Actor	Usuario
Resumen	Un usuario quiere eliminar su cuenta del sistema y borrar la información que se haya almacenado sobre él durante el uso del mismo
Secuencia principal	
Precondiciones	El actor está previamente registrado en el sistema y cuenta con una sesión activa
<ol style="list-style-type: none"> 1. El actor selecciona eliminar su perfil 2. El sistema informa de las consecuencias de realizar la acción 3. El sistema solicita confirmación sobre la acción 4. El actor confirma el borrado del perfil 5. El sistema elimina los productos guardados del actor 6. El sistema elimina las listas de la compra generadas por el actor 7. El sistema asigna a los productos creados por el actor un autor genérico 8. El sistema elimina el nombre, correo electrónico y contraseña del actor de los registros 9. El sistema informa del éxito en la operación 10. El caso de uso finaliza 	
Postcondiciones	El actor ya no podrá acceder a la funcionalidad del sistema usando su cuenta
Secuencia alternativas	
4A	1. El actor cancela el proceso de eliminación del perfil y el caso de uso queda sin efecto
Excepciones	
No hay	

Tabla 3.4: Detalles del caso de uso “Eliminar perfil”

CU05 - Modificar perfil	
Actor	Usuario
Resumen	El usuario quiere cambiar el nombre con el que le reconoce el sistema o la contraseña de acceso
Secuencia principal	
Precondiciones	El actor tiene una sesión activa en el sistema
	<ol style="list-style-type: none"> 1. El actor selecciona cambiar su información de perfil 2. El sistema pregunta que dato pretende modificarse 3. El actor selecciona el dato que pretende cambiar 4. El sistema solicita el nuevo valor modificado para el dato 5. El actor introduce el nuevo valor 6. El sistema valida el formato del nuevo valor introducido 7. El sistema solicita confirmación 8. El actor confirma el cambio 9. El sistema guarda la nueva información 10. El sistema informa del éxito en la operación 11. El caso de uso finaliza
Postcondiciones	La cuenta asociada al actor tendrá en su registro en el sistema la información con el dato modificado
Secuencia alternativas	
3A, 5A, 8A	1. El actor cancela el proceso de cambio de información y el caso de uso queda sin efecto
Excepciones	
6A	<ol style="list-style-type: none"> 1. El sistema comprueba que el formato de los datos introducidos es incorrecto 2. El sistema informa al actor 3. El caso de uso continua en el paso 5

Tabla 3.5: Detalles del caso de uso “Modificar perfil”

CU06 - Visualizar detalles de un producto	
Actor	Usuario
Resumen	El actor en su uso de la aplicación ha encontrado un producto del que quiere conocer más información
Secuencia principal	
Precondiciones	El actor se encuentra identificado en el sistema
<ol style="list-style-type: none"> 1. El actor selecciona un producto para consultar sus detalles 2. El sistema recopila la información del producto 3. El sistema recopila las ofertas que afectan al producto 4. El sistema muestra la información al actor 5. El sistema muestra las posibles acciones que se pueden realizar sobre los productos 6. El caso de uso finaliza 	
Postcondiciones	El actor se encuentra visualizando la información recopilada por el sistema sobre el producto y puede realizar ciertas acciones incluidas en el resto de casos de uso descritos que incluyen este caso de uso como precondición
Secuencia alternativas	
Excepciones	

Tabla 3.6: Detalles del caso de uso “Visualizar detalles de un producto”

CU07 - Publicar producto	
Actor	Usuario
Resumen	Un usuario ha decidido que quiere poder interactuar con un cierto producto en el sistema pero considera que este no se encuentra en él, por lo tanto, pretende añadirlo al catálogo de forma que sea accesible para él y para toda la comunidad
Secuencia principal	
Precondiciones	El actor se encuentra identificado y el producto que va a ser añadido no se encuentra en el sistema.
<ol style="list-style-type: none"> 1. El actor selecciona publicar un producto 2. El sistema solicita el escaneo del código de barras del producto 3. El actor escanea el código de barras del producto 4. El sistema solicita el nombre del producto, la marca y el precio 5. El actor introduce los datos solicitados 6. El sistema valida que los datos introducidos son correctos 7. El sistema muestra un mapa con las tiendas cercanas al actor disponibles en la aplicación 8. El actor selecciona la tienda del producto 9. El sistema muestra las cadenas de alimentación disponibles en el sistema 10. El actor selecciona la cadena a la que pertenece la tienda introducida 11. El sistema muestra las categorías disponibles en el sistema 12. El actor selecciona la categoría del producto 13. El sistema solicita fotografías del producto 14. El actor introduce las fotografías en el sistema 15. El sistema muestra productos relacionados con el introducido 16. El sistema solicita elegir aquellos productos cuya relación con el producto a publicar sea confirmada por el actor 17. El actor selecciona los productos 18. El sistema comprueba que la información introducida es correcta 19. El sistema solicita confirmación para publicar el producto 20. El actor confirma la publicación 21. El sistema guarda la información del producto 22. El sistema añade el producto al catálogo 23. El caso de uso finaliza 	
Postcondiciones	El producto ha quedado registrado en el sistema. Sus detalles y funcionalidad relacionada son accesibles para todos los usuarios de la aplicación
Secuencia alternativas	
3A, 5A, 8A, 10A, 12A, 14A, 17A, 20A	El actor cancela el proceso de publicación de un producto y el caso de uso queda sin efecto
Excepciones	
3A	<ol style="list-style-type: none"> 1. El sistema comprueba que ya existe un producto con un código de barra similar 2. El sistema muestra los productos con el mismo código de barras y sus tiendas 3. El actor selecciona el producto mostrado que se corresponda con el que pretende subir 4. Se inicia el caso de uso CU06 - Visualizar detalles de producto

6A	<ol style="list-style-type: none"> 1. El sistema comprueba que el formato de los datos introducidos es incorrecto 2. El sistema informa al actor 3. El caso de uso continua en el paso 4
8A	<ol style="list-style-type: none"> 1. El sistema comprueba que ya existe una combinación con ese código de barras y tienda en el sistema 2. El sistema informa al actor 3. Se inicia el caso de uso CU06 - Visualizar detalles de producto para el producto en cuestión
15A	<ol style="list-style-type: none"> 1. El sistema no cuenta con productos registrados que puedan ser relacionables a aquel que se está tratando de publicar 2. El sistema informa al actor 3. El caso de uso continua en el paso 18
18A	<ol style="list-style-type: none"> 1. El sistema comprueba que alguna información es incorrecta 2. El sistema informa al actor 3. El sistema muestra el dato incorrecto 4. El sistema solicita el dato corregido 5. El caso de uso continua en el paso 18

Tabla 3.7: Detalles del caso de uso ‘Publicar producto’

CU08 - Buscar producto en el catálogo por criterios de búsqueda	
Actor	Usuario
Resumen	Un usuario pretende encontrar un determinado producto en la aplicación utilizando diferentes criterios de búsqueda preestablecidos por la misma
Secuencia principal	
Precondiciones	El actor está previamente registrado en el sistema
<ol style="list-style-type: none"> 1. El actor inicia el proceso de búsqueda de productos 2. El sistema muestra las categorías de productos disponibles 3. El sistema solicita la categoría buscada 4. El actor selecciona una categoría 5. El sistema muestra las cadenas disponibles 6. El sistema solicita la cadena de alimentación buscada 7. El actor introduce la cadena buscada 8. El sistema solicita un texto clave para buscar productos relacionados 9. El actor introduce el texto clave 10. El sistema solicita confirmación para realizar la búsqueda 11. El actor confirma la búsqueda de productos 12. El sistema recopila productos que satisfagan los criterios seleccionados 13. El sistema informa de los resultados obtenidos 14. El sistema muestra los resultados de la búsqueda al actor 15. El caso de uso finaliza 	
Postcondiciones	El actor ha obtenido una lista de productos que satisfagan los criterios de la búsqueda introducida
Secuencia alternativas	
4A, 7A, 9A, 11A	1. El actor cancela el proceso de búsqueda de productos y el caso de uso queda sin efecto
4B, 7B, 9B	<ol style="list-style-type: none"> 1. El actor decide no incluir este criterio en la búsqueda de productos 2. El sistema no incluye la información de este criterio en la búsqueda 3. El caso de uso continúa siguiendo la secuencia normal
Excepciones	
13A	<ol style="list-style-type: none"> 1. El sistema no ha encontrado ningún resultado que satisfaga la búsqueda 2. El sistema informa al actor 3. El caso de uso finaliza

Tabla 3.8: Detalles del caso de uso “Buscar producto en el catálogo por criterios de búsqueda”

CU09 - Buscar producto en el catálogo por escaneo de código de barras	
Actor	Usuario
Resumen	Un usuario pretende encontrar un determinado producto en la aplicación utilizando el código de barras disponible en el envase del mismo
Secuencia principal	
Precondiciones	El actor está previamente registrado en el sistema y realiza el caso de uso desde un dispositivo que cuente con una cámara
<ol style="list-style-type: none"> 1. El actor inicia el proceso de búsqueda de productos 2. El sistema solicita al actor que escanee el código de barras del producto 3. El actor escanea el código de barras del producto 4. El sistema valida la lectura del código realizada 5. El sistema busca los productos en el sistema con el código de barras introducido 6. El sistema informa de los resultados obtenidos 7. El sistema muestra los resultados de la búsqueda al actor 8. El caso de uso finaliza 	
Postcondiciones	El actor ha obtenido una lista de productos con el código de barras escaneado que pueden ser de diferentes supermercados
Secuencia alternativas	
3A	1. El actor cancela el proceso de búsqueda de productos y el caso de uso queda sin efecto
Excepciones	
7A	<ol style="list-style-type: none"> 1. El sistema no ha encontrado ningún resultado que satisfaga la búsqueda 2. El sistema informa al actor 3. El caso de uso finaliza

Tabla 3.9: Detalles del caso de uso "Buscar producto en el catálogo por escaneo de código de barras"

CU10 - Consultar lista de la compra	
Actor	Usuario
Resumen	Un usuario ha decidido consultar que productos ha añadido a su cesta de la compra durante la navegación en la aplicación y los costes asociados a los productos incluidos en ella. También está interesado en productos relacionados con los que ya ha añadido a la cesta y que puedan acabar sustituyendo o añadiendo a los productos de la misma.
Secuencia principal	
Precondiciones	El actor está previamente registrado en el sistema y ha incluido durante el uso de la aplicación algún producto en su lista de la compra
<ol style="list-style-type: none"> 1. El actor selecciona ver su lista de la compra 2. El sistema recopila la información de los productos en la lista de la compra 3. El sistema agrupa los productos guardados según su establecimiento 4. El sistema calcula el coste de los productos en función del precio y unidades de cada uno para cada establecimiento 5. El sistema calcula el precio total para todos los establecimientos 6. El sistema muestra la información calculada al actor 7. El sistema muestra productos relacionados con los de la lista de la compra al actor 8. El sistema pregunta al actor que productos de los recomendados desea añadir a la lista de la compra 9. El actor no selecciona ningún producto 10. El caso de uso finaliza 	
Postcondiciones	El actor tiene a su disposición la lista de la compra con la información actualizada sobre los productos que en ella se encuentran
Secuencia alternativas	
9A	<ol style="list-style-type: none"> 1. El actor selecciona los productos deseados 2. El sistema añade el/los producto/s a la lista de la compra 3. El caso de uso continua en el paso 2
Excepciones	
2A	<ol style="list-style-type: none"> 1. El sistema no ha encontrado ningún producto en la lista de la compra 2. El sistema informa al actor 3. El caso de uso finaliza

Tabla 3.10: Detalles del caso de uso “Consultar lista de la compra”

CU11 - Añadir producto a la lista de la compra	
Actor	Usuario
Resumen	Un usuario durante el uso de la app ha encontrado un producto que desea añadir a su lista de la compra para poder conocer el precio total de esta incluyendo este nuevo producto
Secuencia principal	
Precondiciones	El actor está previamente registrado en el sistema y se ha ejecutado el caso de uso CU06 - Visualizar detalles de un producto
<ol style="list-style-type: none"> 1. El actor selecciona añadir el producto visualizado a su lista de la compra 2. El sistema añade el producto a la lista de la compra 3. El sistema informa al actor 4. El sistema pregunta al actor si desea consultar la lista de la compra 5. El actor responde negativamente 6. El caso de uso finaliza 	
Postcondiciones	La lista de la compra del actor contará con un nuevo producto cuando su precio se vaya a calcular
Secuencia alternativas	
5A	<ol style="list-style-type: none"> 1. El actor decide consultar la lista de la compra 2. Se inicia el caso de uso CU10 - Consultar lista de la compra

Tabla 3.11: Detalles del caso de uso “Añadir producto a la lista de la compra”

CU12 - Eliminar producto de la lista de la compra	
Actor	Usuario
Resumen	Un usuario durante el uso de la app ha decidido que un producto previamente añadido a la lista de la compra ya no es de su interés
Secuencia Principal	
Precondiciones	El actor está previamente registrado en el sistema y se ha ejecutado el caso de uso CU06 -Visualizar detalles de un producto o el CU10 - Consultar lista de la compra
<ol style="list-style-type: none"> 1. El actor selecciona eliminar el producto de su lista de la compra 2. El sistema elimina el producto de la lista de la compra 3. El sistema informa al actor 4. El caso de uso finaliza 	
Postcondiciones	La lista de la compra del actor contará con un producto menos cuando su precio se vaya a calcular
Secuencias Alternativas	
No hay	
Excepciones	
No hay	

Tabla 3.12: Detalles del caso de uso ‘Eliminar producto de la lista de la compra’

CU13 - Guardar lista de la compra	
Actor	Usuario
Resumen	El actor ha decidido guardar la lista de la compra para poder consultarla en el futuro en el estado y con los productos que la lista contiene en ese momento.
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 -Consultar lista de la compra
<ol style="list-style-type: none"> 1. El actor inicia el proceso de guardado de la lista de la compra 2. El sistema solicita confirmación 3. El actor confirma la acción 4. El sistema guarda la información de la lista de la compra registrada por la fecha 5. El sistema informa al actor 6. El caso de uso finaliza 	
Postcondiciones	La lista de la compra del actor podrá ser consultada en un futuro y contendrá la información de los productos, establecimientos, cantidades y precios en el momento en el que fue guardada
Secuencias Alternativas	
3A	El actor decide cancelar el proceso de guardado de la lista de la compra y el caso de uso queda sin efecto
Excepciones	
No hay	

Tabla 3.13: Detalles del caso de uso “Guardar lista de la compra”

CU14 - Modificar cantidad de un producto de la lista de la compra	
Actor	Usuario
Resumen	El actor desea cambiar el número de unidades de un producto que se han añadido a la lista de la compra
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 -Consultar lista de la compra
<ol style="list-style-type: none"> 1. El actor decide modificar las unidades seleccionadas de un producto en la lista de la compra 2. El sistema muestra los productos de la lista 3. El sistema solicita el producto cuyas cantidades desean modificarse 4. El actor selecciona el producto 5. El sistema solicita la nueva cantidad 6. El actor introduce la nueva cantidad de unidades 7. El sistema valida la cantidad 8. El sistema modifica la cantidad de unidades de ese producto en la lista de la compra 9. Se inicia el caso de uso CUXX - Consultar lista de la compra 	
Postcondiciones	La lista de la compra del actor contará con las cantidades actualizadas para el producto sobre el que se ha efectuado el caso de uso
Secuencias Alternativas	
4A, 6A	1. El actor decide cancelar el proceso y el caso de uso queda sin efecto
Excepciones	
7A	<ol style="list-style-type: none"> 1. El sistema comprueba que la cantidad no es correcta o posible 2. El sistema informa al actor 3. El caso de uso continua en el paso 5

Tabla 3.14: Detalles del caso de uso “Modificar cantidad de un producto de la lista de la compra”

CU15 - Modificar información de un producto	
Actor	Usuario
Resumen	El actor desea cambiar la información de un producto debido a que ha encontrado un fallo y esta se encuentra en un estado incorrecto
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU06 - Visualizar detalles de un producto, sobre un producto creado por el mismo
<ol style="list-style-type: none"> 1. El actor decide modificarla información de un producto 2. El sistema solicita el campo que desea cambiarse 3. El actor introduce el campo cuyo valor quiere cambiarse 4. El sistema solicita la nueva información del producto para ese campo 5. El actor introduce la información del producto para ese campo 6. El sistema valida la información para ese campo 7. El sistema pregunta si se desean realizar más cambios 8. El actor responde negativamente 9. El sistema solicita confirmación para realizar los cambios 10. El actor confirma las modificaciones 11. El sistema modifica el producto con las actualizaciones de su información 12. El caso de uso finaliza 	
Postcondiciones	La información sobre el producto estará modificada y este cambio se verá reflejado en el catálogo del sistema
Secuencias Alternativas	
5A, 8A, 10A	1. El actor cancela la modificación y el caso de uso queda sin efecto
8B	<ol style="list-style-type: none"> 1. El actor decide realizar más cambios 2. El caso de uso continua en el paso 2
Excepciones	
6A	<ol style="list-style-type: none"> 1. El sistema comprueba que la información introducida no es correcta 2. El sistema informa al usuario 3. El caso de uso continua en el paso 4

Tabla 3.15: Detalles del caso de uso “Modificar información de un producto”

CU16 - Valorar producto	
Actor	Usuario
Resumen	El actor desea valorar un producto de manera que su opinión sea mostrada en la pantalla del producto y otros usuarios puedan tenerla en cuenta a la hora de elegir que productos les interesan
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 - Visualizar detalles de un producto
<ol style="list-style-type: none"> 1. El actor decide añadir una valoración sobre el producto 2. El sistema solicita la valoración 3. El actor introduce la valoración 4. El sistema valida la valoración 5. El sistema guarda la valoración junto con la fecha de realización 6. El sistema informa al actor del éxito en la operación 7. El caso de uso finaliza 	
Postcondiciones	La valoración del producto mostrada al resto de usuarios tendrá en cuenta la nueva aportación realizada por el actor
Secuencias Alternativas	
3A	1. El actor cancela el proceso de valoración del producto y el caso de uso queda sin efecto
Excepciones	
4A	<ol style="list-style-type: none"> 1. El sistema comprueba que la valoración introducida no es correcta 2. El sistema informa al usuario 3. El caso de uso continua en el paso 4

Tabla 3.16: Detalles del caso de uso “Valorar producto”

CU17 - Añadir oferta	
Actor	Usuario
Resumen	El actor ha encontrado en un establecimiento una oferta que mejora el precio registrado de un producto del catálogo cuyos detalles está visualizando
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 - Visualizar detalles de un producto
<ol style="list-style-type: none"> 1. El actor decide añadir una oferta sobre el producto visualizado 2. El sistema solicita el nuevo precio de la oferta 3. El actor introduce el nuevo precio de la oferta 4. El sistema valida que el precio de la oferta es menor que el precio normal del producto 5. El sistema solicita las fechas de inicio y fin de la oferta 6. El actor introduce las fechas solicitadas 7. El sistema solicita la descripción de la oferta 8. El actor introduce la descripción de la oferta 9. El sistema solicita confirmación para crear la oferta 10. El actor confirma la acción 11. El sistema añade la oferta al registro 12. El sistema añade el nuevo precio al histórico de precios del producto 13. El sistema informa al actor del éxito en la operación 14. El caso de uso finaliza 	
Postcondiciones	El producto cuenta con una nueva oferta que reduce su precio de forma temporal que será visualizable por el resto de usuarios de la aplicación
Secuencias Alternativas	
3A, 6A, 8A, 10A	1. El actor cancela el proceso de añadir una oferta y el caso de uso queda sin efecto
Excepciones	
4A	<ol style="list-style-type: none"> 1. El sistema comprueba que el precio introducido no es correcto 2. El sistema informa al usuario 3. El caso de uso continua en el paso 2

Tabla 3.17: Detalles del caso de uso “Añadir oferta”

CU18 - Reportar producto	
Actor	Usuario
Resumen	El actor ha encontrado un fallo en uno de los productos del catálogo y quiere reportarlo para que sea solucionado a la mayor brevedad posible
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 - Visualizar detalles de un producto
<ol style="list-style-type: none"> 1. El actor inicia el proceso de generación de un reporte sobre el producto visualizado 2. El sistema muestra la información del producto 3. El sistema muestra los motivos de reporte disponibles 4. El sistema solicita el motivo de reporte 5. El actor introduce el motivo del reporte 6. El sistema solicita información extra sobre el reporte 7. El actor introduce la información extra solicitada 8. El sistema solicita confirmación para añadir el reporte 9. El actor confirma la acción 10. El sistema añade el reporte al registro 11. El sistema notifica al autor de la publicación del producto del reporte 12. El sistema informa del éxito al actor 13. El caso de uso finaliza 	
Postcondiciones	El producto cuenta con un reporte lo que significará que su aparición en la aplicación para la comunidad de usuarios se verá reducida hasta que el problema que lo ha generado se solucione
Secuencias Alternativas	
5A, 7A, 9A	1. El actor cancela el proceso de reportar un producto y el caso de uso queda sin efecto
7B	<ol style="list-style-type: none"> 1. El actor decide no aportar información extra al reporte 2. El sistema omite este paso en la creación del reporte 3. El caso de uso continúa en el paso 8
Excepciones	
No hay	

Tabla 3.18: Detalles del caso de uso "Reportar producto"

CU19 - Consultar histórico de precios	
Actor	Usuario
Resumen	El actor quiere conocer la evolución del precio de un determinado producto a lo largo del tiempo debido a la existencia de ofertas o a cambios en el precio normal del producto
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 - Visualizar detalles de un producto
	<ol style="list-style-type: none"> 1. El actor selecciona consultar el histórico de precios del producto 2. El sistema recopila los precios normales del producto y sus fechas 3. El sistema recopila los precios de oferta del producto y sus fechas 4. El sistema genera el histórico de precios 5. El sistema muestra el histórico de precios al usuario 6. El caso de uso finaliza
Postcondiciones	El actor tendrá disponible un gráfico en el que se puedan apreciar los diferentes precios que ha tenido el producto elegido a lo largo del tiempo que este producto ha estado en la aplicación
Secuencias Alternativas	
	No hay
Excepciones	
3A	<ol style="list-style-type: none"> 1. El sistema no cuenta con registros de ofertas para el producto 2. El sistema informa al actor 3. El caso de uso continúa en el paso 4

Tabla 3.19: Detalles del caso de uso “Consultar histórico de precios”

CU20 - Marcar producto como guardado	
Actor	Usuario
Resumen	El actor quiere poder acceder rápidamente a un producto de su interés para poder ver sus detalles sin tener que realizar una búsqueda en un futuro así que decide marcarlo como guardado
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 - Visualizar detalles de un producto
<ol style="list-style-type: none"> 1. El actor selecciona guardar el producto 2. El sistema actualiza la lista de productos guardados del actor con el nuevo producto 3. El sistema informa al actor 4. El caso de uso finaliza 	
Postcondiciones	La lista de productos guardados del actor incluye el nuevo producto
Secuencias Alternativas	
No hay	
Excepciones	
No hay	

Tabla 3.20: Detalles del caso de uso “Marcar producto como guardado”

CU21 - Desmarcar producto como guardado	
Actor	Usuario
Resumen	El actor ya no está interesado en un producto de los que decidió guardar y quiere eliminarlo de la lista
Secuencia Principal	
Precondiciones	El actor se encuentra identificado en el sistema y ha realizado el caso de uso CU10 - Visualizar detalles de un producto o el caso de uso CU22 - Consultar lista de productos guardados
<ol style="list-style-type: none"> 1. El actor selecciona desmarcar el producto como guardado 2. El sistema actualiza la lista de productos guardados del actor eliminando el producto 3. El sistema informa al usuario 4. El caso de uso finaliza 	
Postcondiciones	La lista de productos guardados del actor ya no incluye el nuevo producto
Secuencias Alternativas	
No hay	
Excepciones	
No hay	

Tabla 3.21: Detalles del caso de uso “Desmarcar producto como guardado”

CU22 - Consultar lista de productos guardados	
Actor	Usuario
Resumen	Un usuario en el uso de la aplicación ha ido guardando productos que le interesaría tener a mano en un futuro confeccionando una lista de productos guardados y decide consultarla para observar sus precios y otras características de interés. En esta lista se incluirán también los productos que el actor haya publicado en el sistema.
Secuencia Principal	
Precondiciones	El actor está previamente registrado en el sistema y ha incluido durante el uso de la aplicación algún producto en su lista de productos guardados
<ol style="list-style-type: none"> 1. El actor selecciona ver su lista de productos guardados 2. El sistema recopila la información de los productos creados por el usuario 3. El sistema recopila la información de los productos guardados por el usuario 4. El sistema muestra la información de los productos recopilados 5. El caso de uso finaliza 	
Postcondiciones	El actor cuenta con una serie de productos que ha guardado o creado previamente cuya información es accesible sin necesidad de ejecutar una búsqueda o encontrarlos en el catálogo
Secuencias Alternativas	
No hay	
Excepciones	
2A	<ol style="list-style-type: none"> 1. El sistema no ha encontrado ningún producto creado por el usuario 2. El sistema informa al usuario 3. El caso de uso continúa en el paso
3A	<ol style="list-style-type: none"> 1. El sistema no ha encontrado ningún producto guardado por el usuario 2. El sistema informa al usuario 3. El caso de uso continúa en el paso 4

Tabla 3.22: Detalles del caso de uso “Consultar lista de productos guardados”

3.4.4. Matriz de correspondencia de requisitos y casos de uso

En la Figura 3.2 se muestra la correspondencia de los requisitos funcionales implementados por los diferentes casos de uso

CU/RF	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	RF11	RF12	RF13	RF14	RF15	RF16	RF17	RF18	RF19	RF20	RF21	RF22	RF23
CU1	■	■		■																			
CU2			■	■																			
CU3					■																		
CU4							■	■															
CU5						■																	
CU6									■				■										
CU7										■													■
CU8											■	■											
CU9											■	■											
CU10																				■	■		
CU11																				■			
CU12																				■			
CU13																						■	
CU14																				■			
CU15															■								
CU16														■									
CU17																■							
CU18																	■						
CU19																		■					
CU20																						■	
CU21																						■	
CU22																						■	

Figura 3.2: Matriz de correspondencia de requisitos y casos de uso

3.5. Realización en Análisis de los casos de uso

En todos los casos de uso se han omitido las secuencias alternativas de cancelar y la referencia a que previo a cada caso de uso es necesario haber efectuado el de Iniciar Sesión por motivos de claridad en los diagramas.

Además, en los diagramas de hace referencia a los casos de uso auxiliares que por su sencillez se han excluido de esta sección. Se pueden encontrar en el repositorio descrito en el Anexo I. Estructura del repositorio en el apartado de Anexos.

CU-01: Registrarse

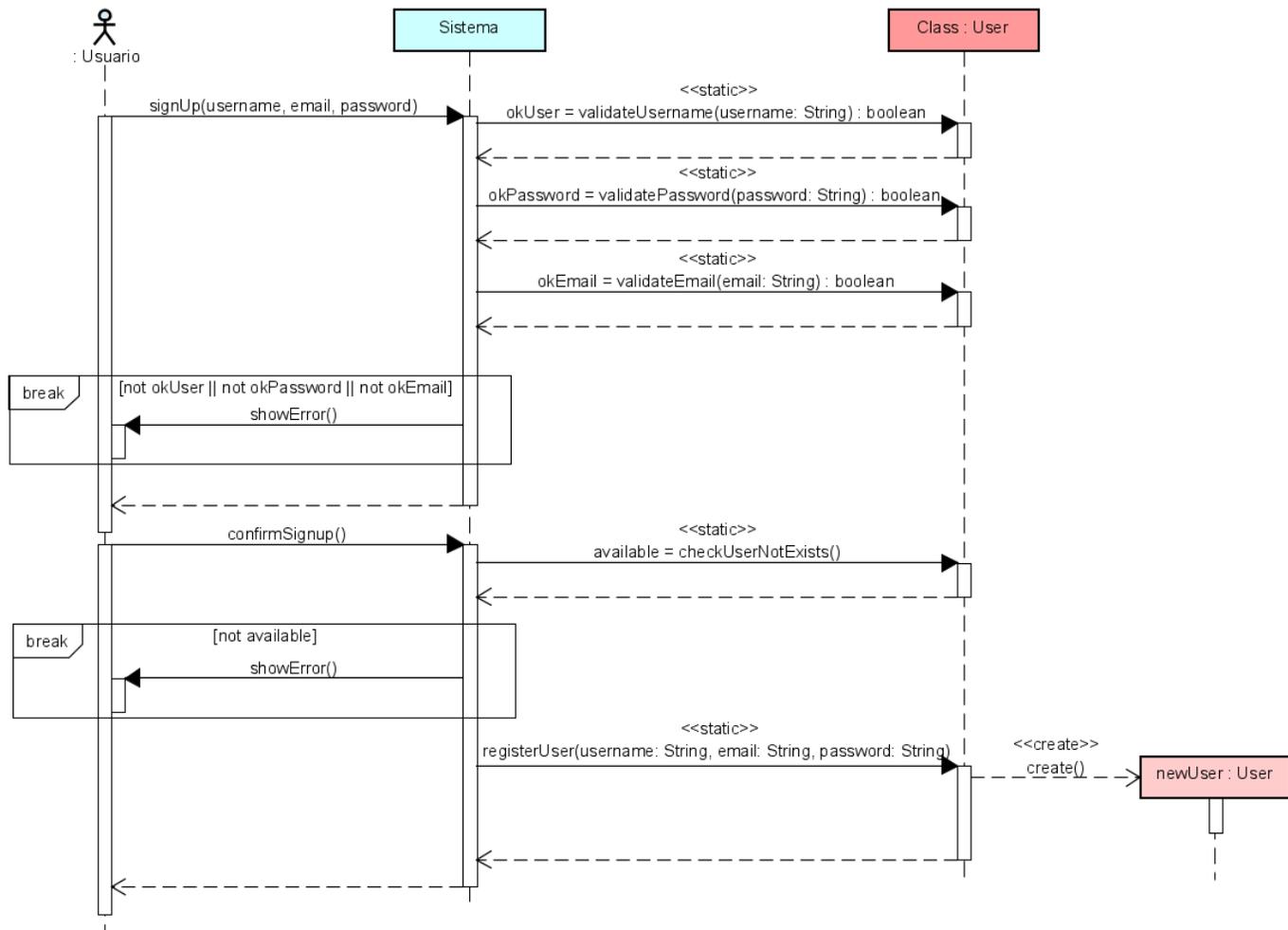


Figura 3.3: Diagrama de secuencia del caso de uso “Registrarse”

CU-02: Iniciar sesión

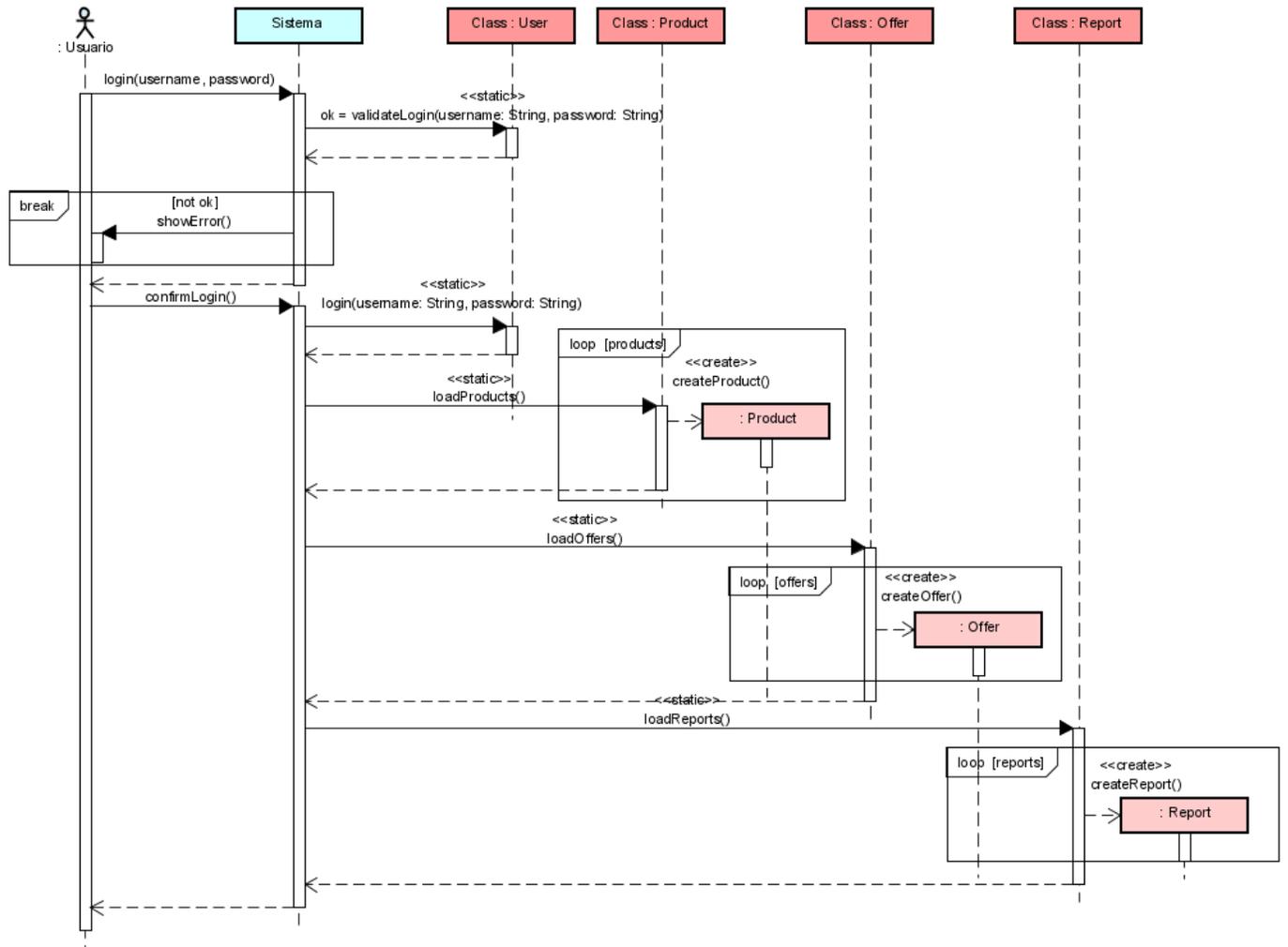


Figura 3.4: Diagrama de secuencia del caso de uso “Iniciar Sesión”

CU-03: Cerrar sesión

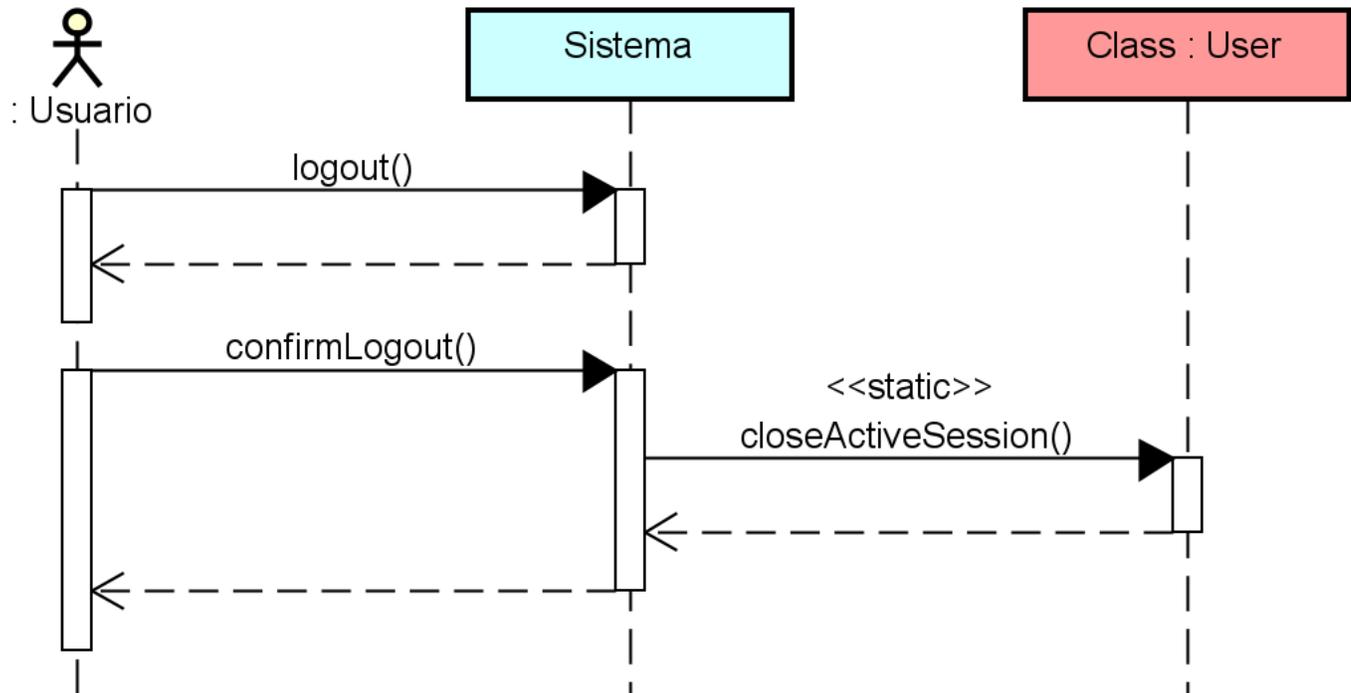


Figura 3.5: Diagrama de secuencia del caso de uso "Cerrar sesión"

CU-04: Eliminar perfil

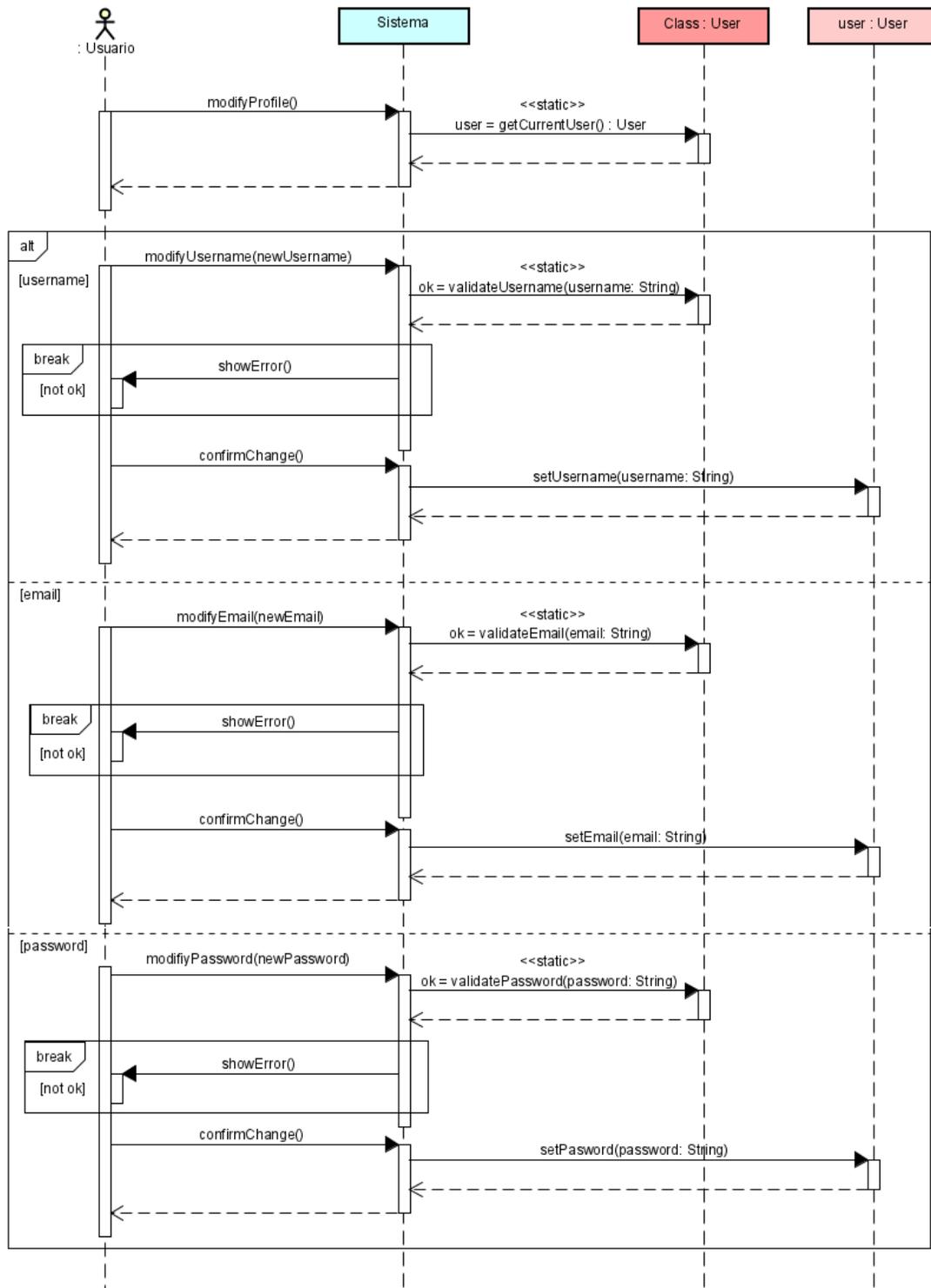


Figura 3.6: Diagrama de secuencia del caso de uso “Eliminar perfil”

CU-05: Modificar información de perfil

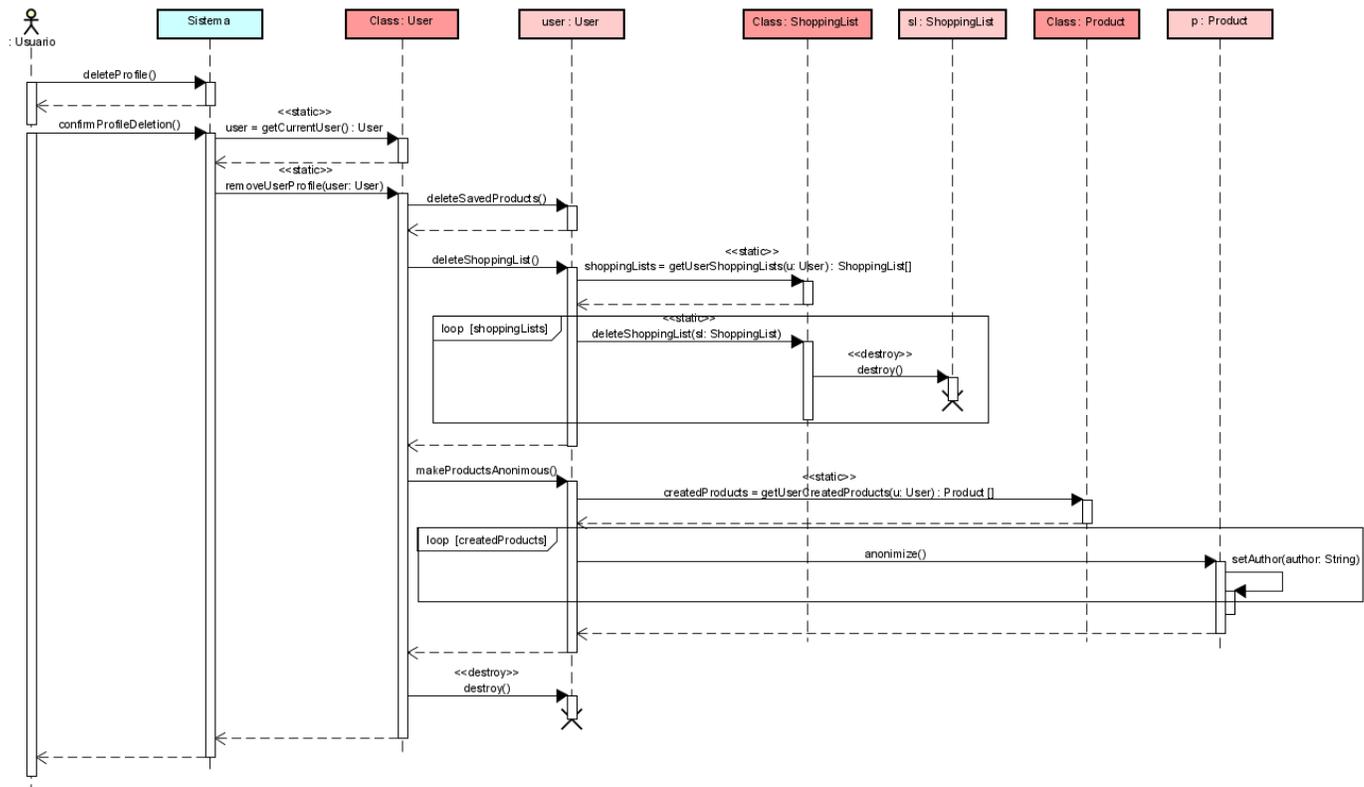


Figura 3.7: Diagrama de secuencia del caso de uso “Modificar información de perfil”

CU-06: Visualizar detalles de un producto

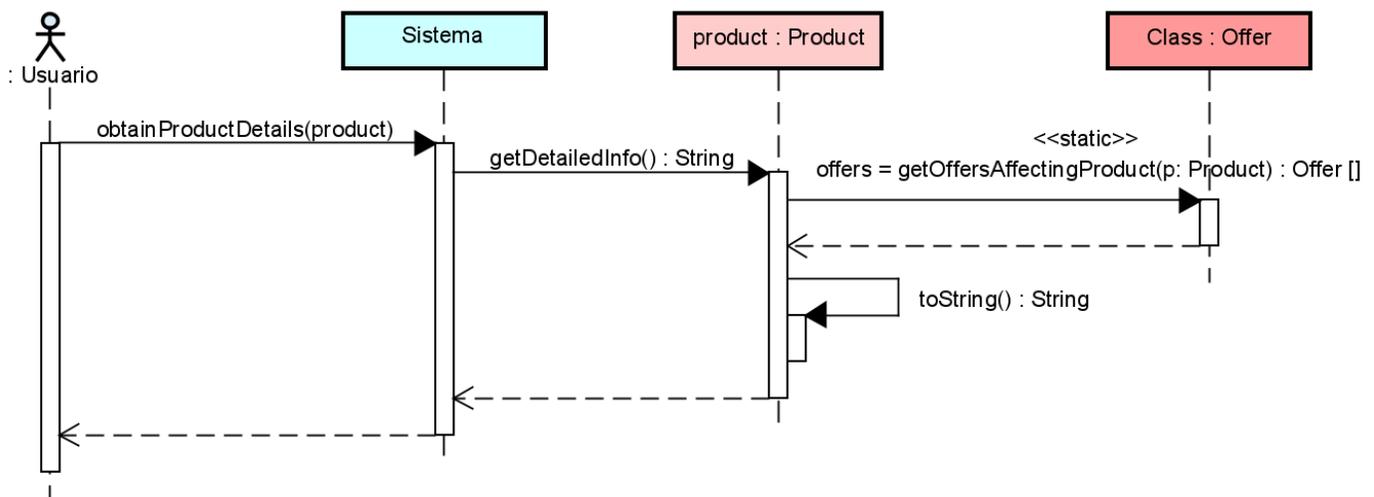


Figura 3.8: Diagrama de secuencia del caso de uso “Visualizar detalles de un producto”

CU-07: Publicar producto

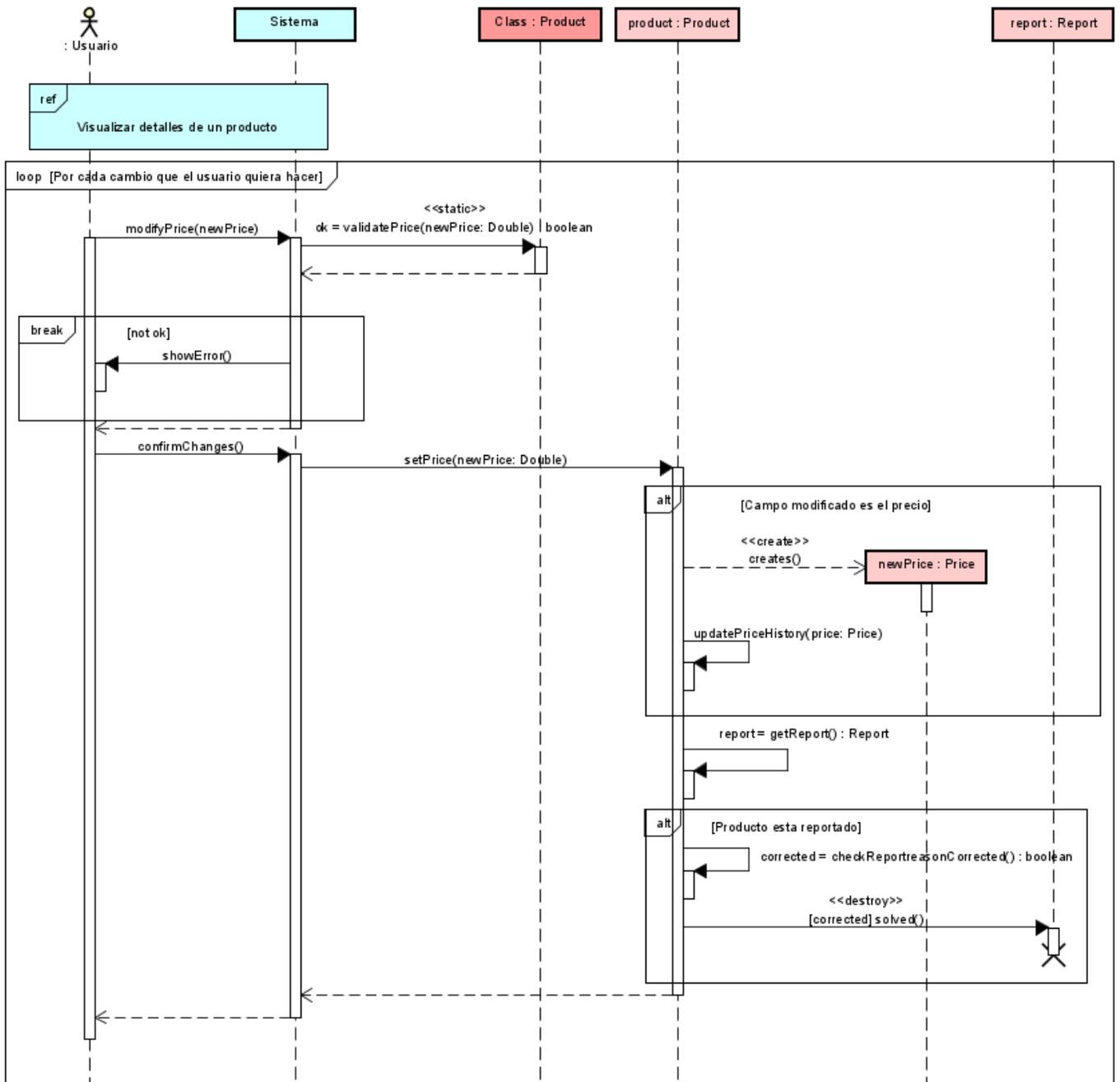


Figura 3.9: Diagrama de secuencia del caso de uso “Publicar producto”

CU-08: Buscar producto en el catálogo por criterios de búsqueda

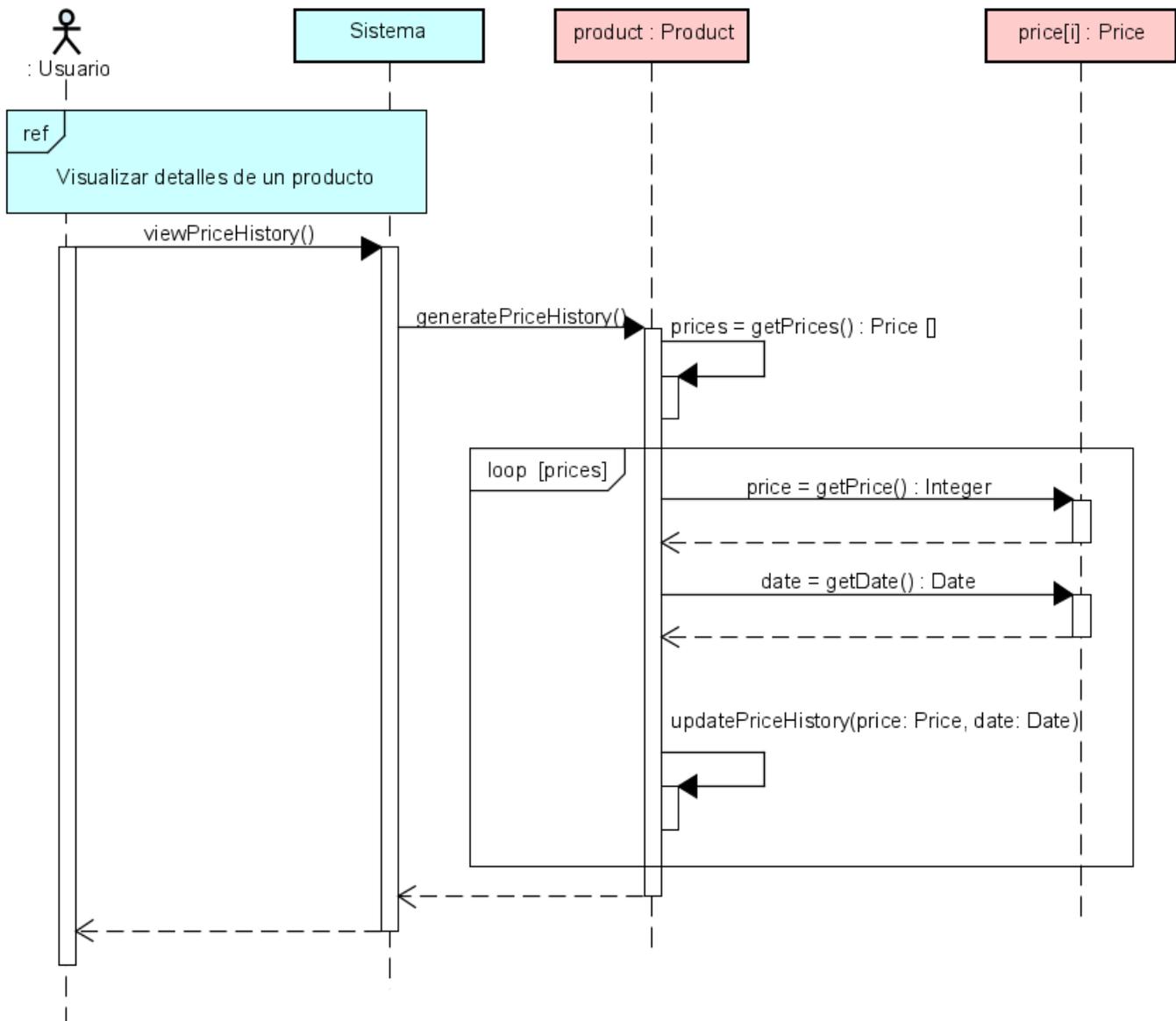


Figura 3.10: Diagrama de secuencia del caso de uso "Buscar producto en el catálogo por criterios de búsqueda"

CU-09: Buscar productos en el catálogo por escaneo de código de barras

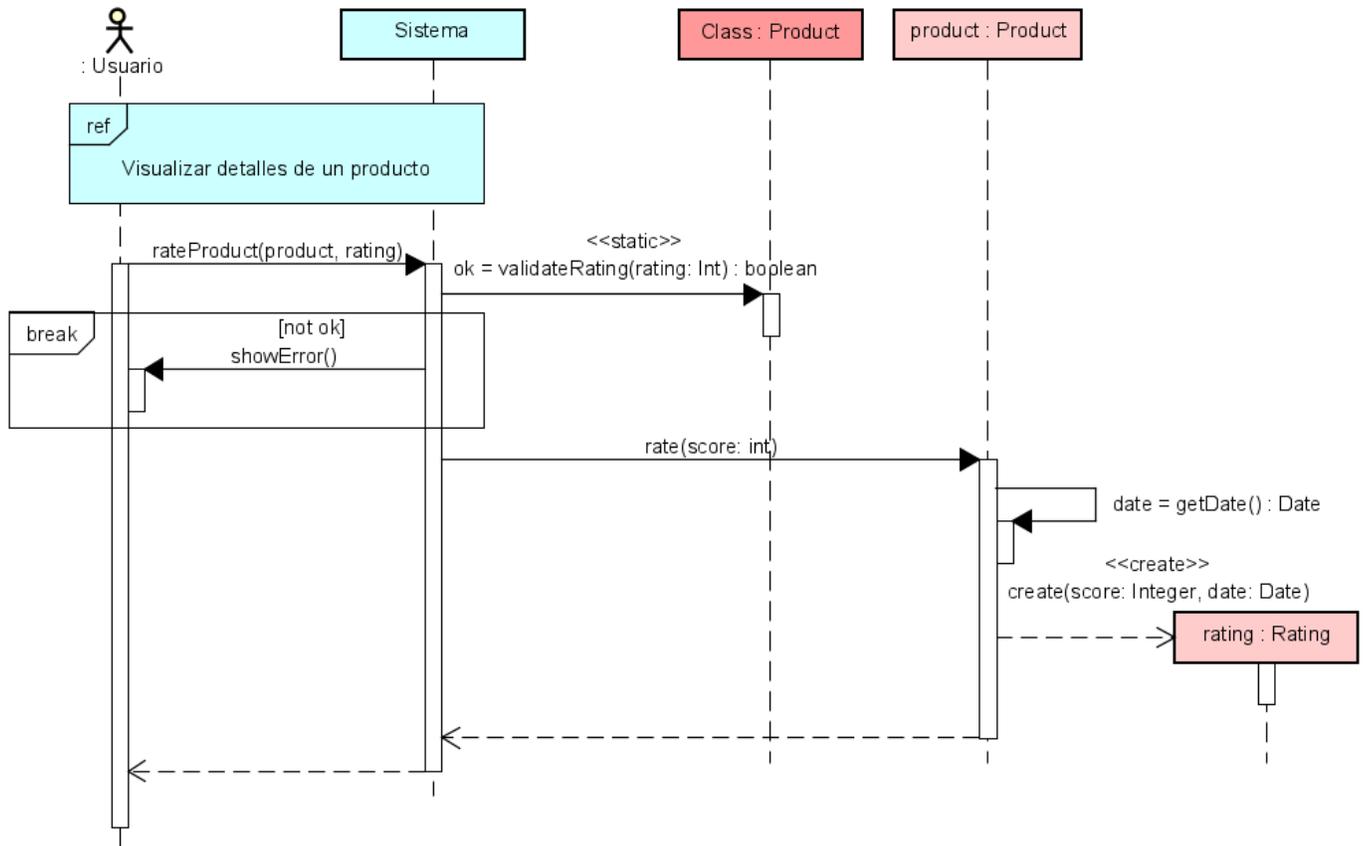


Figura 3.11: Diagrama de secuencia del caso de uso “Buscar productos en el catálogo por escaneo de código de barras”

CU-10: Consultar lista de la compra

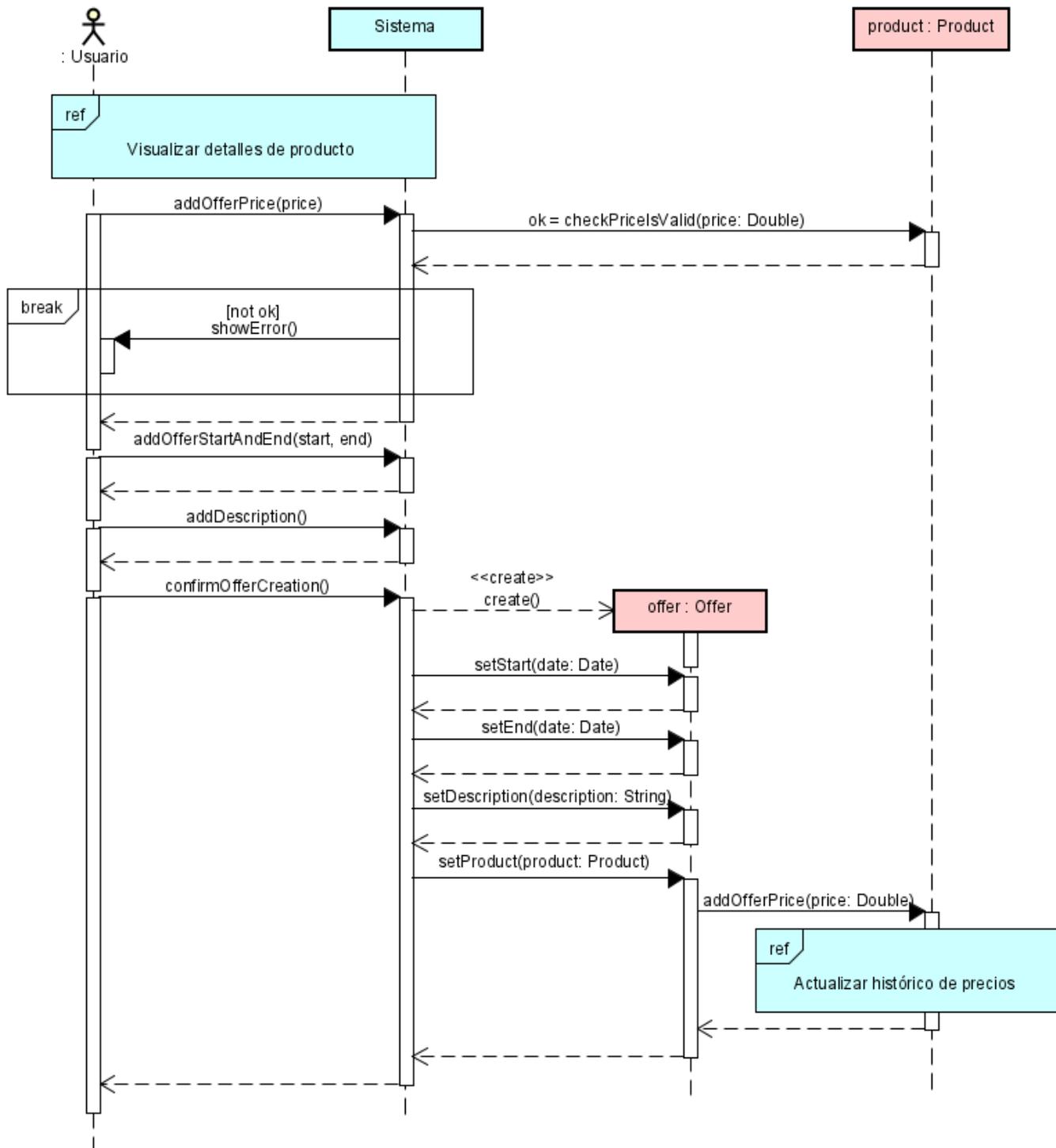


Figura 3.12: Diagrama de secuencia del caso de uso “Consultar lista de la compra”

CU-11: Añadir producto a la lista de la compra

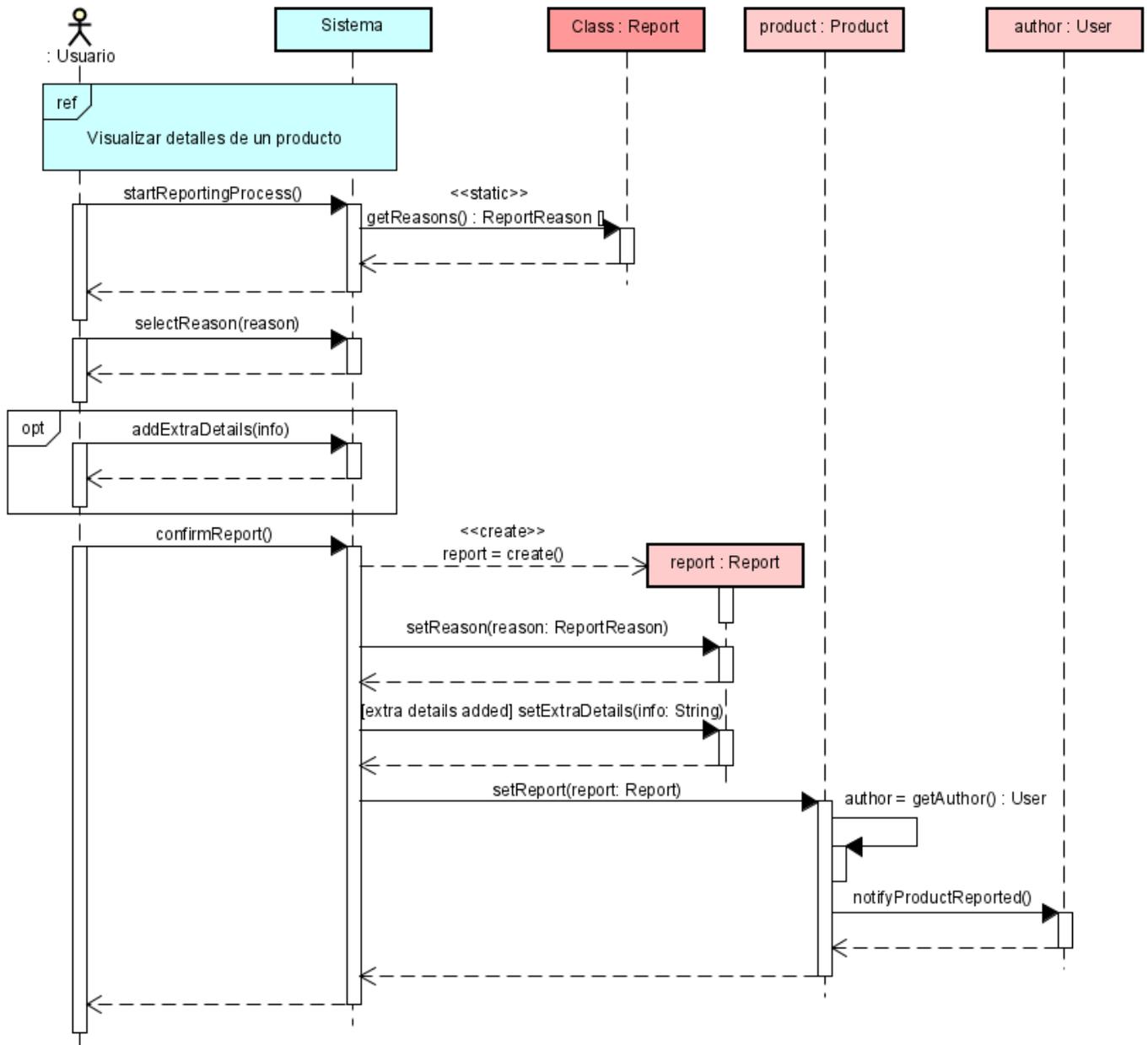


Figura 3.13: Diagrama de secuencia del caso de uso “Añadir producto a la lista de la compra”

CU-12: Eliminar producto de la lista de la compra

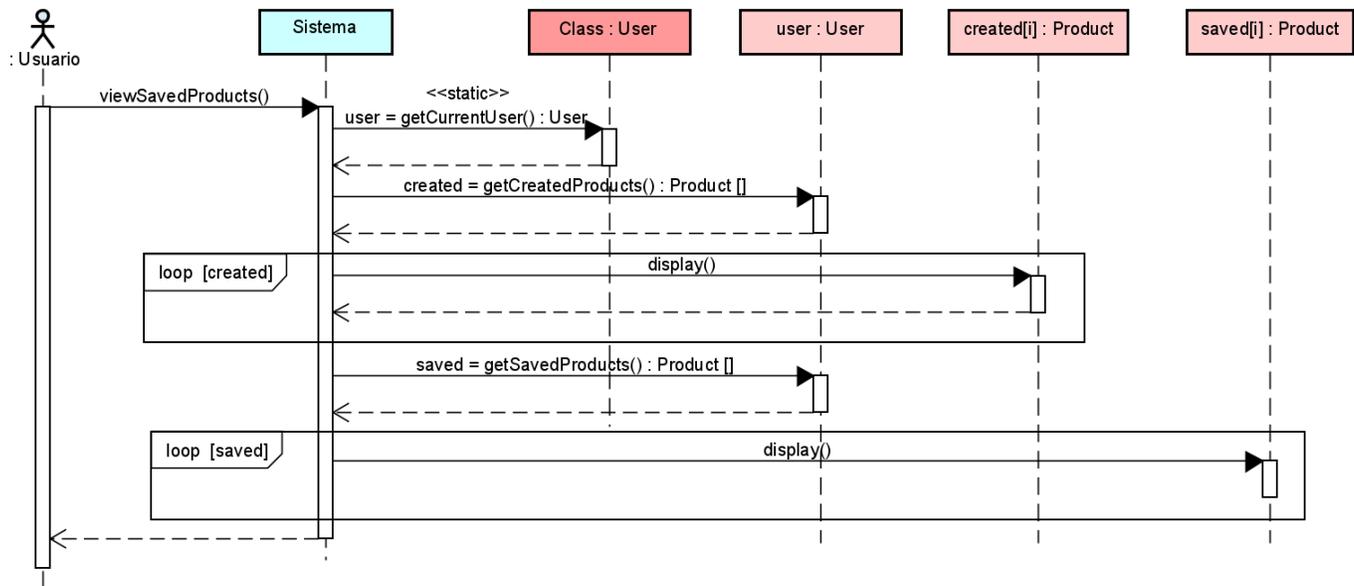


Figura 3.14: Diagrama de secuencia del caso de uso "Eliminar producto de la lista de la compra"

CU-13: Guardar la lista de la compra

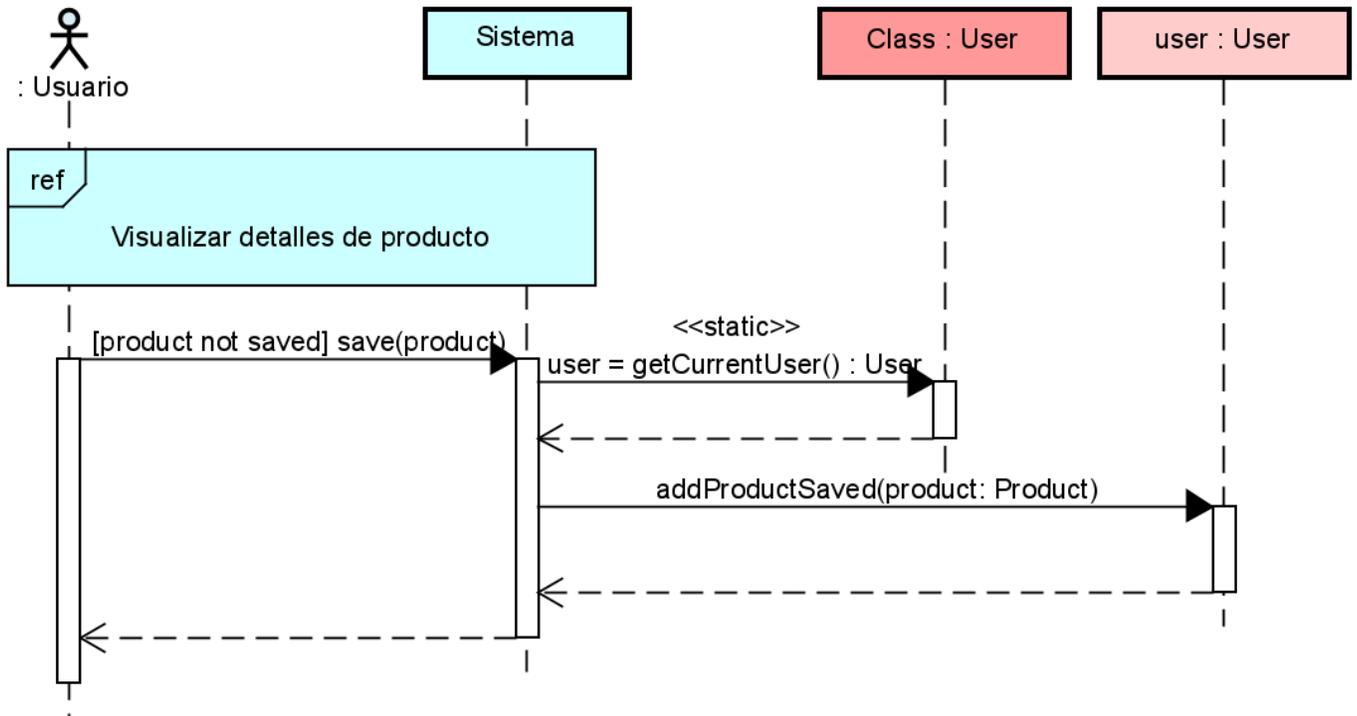


Figura 3.15: Diagrama de secuencia del caso de uso “Guardar la lista de la compra”

CU-14: Modificar cantidad de un producto de la lista de la compra

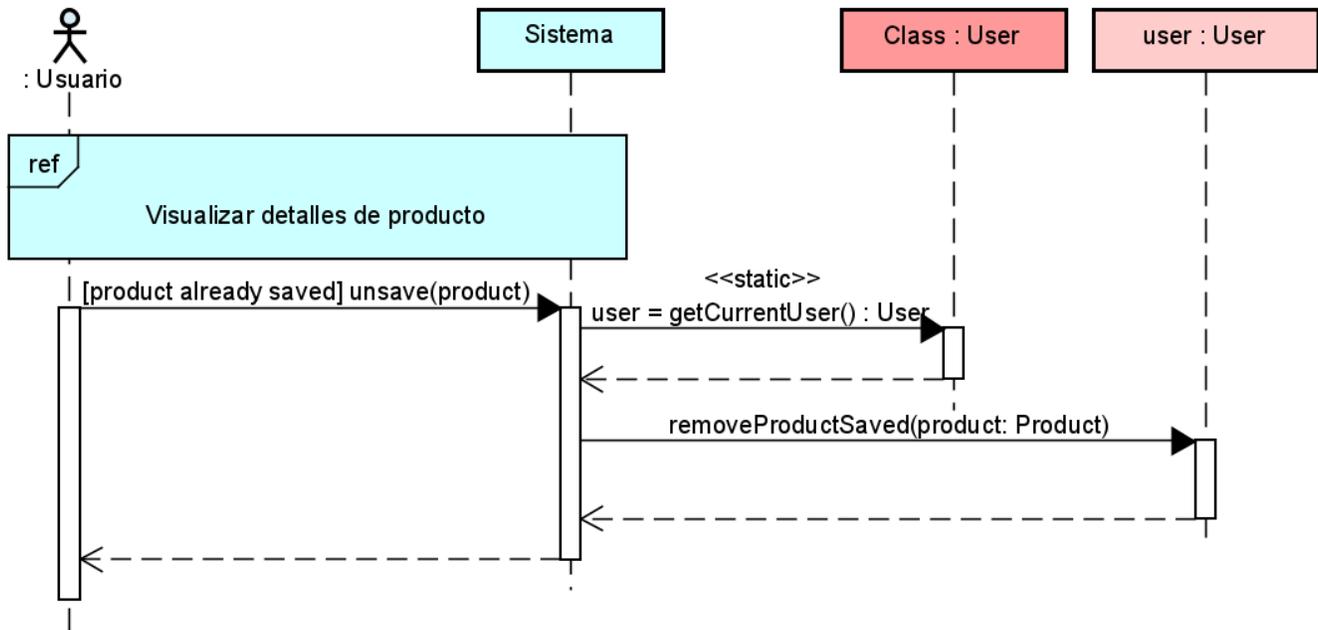


Figura 3.16: Diagrama de secuencia del caso de uso “Modificar cantidad de un producto de la lista de la compra”

CU-15: Modificar información de un producto

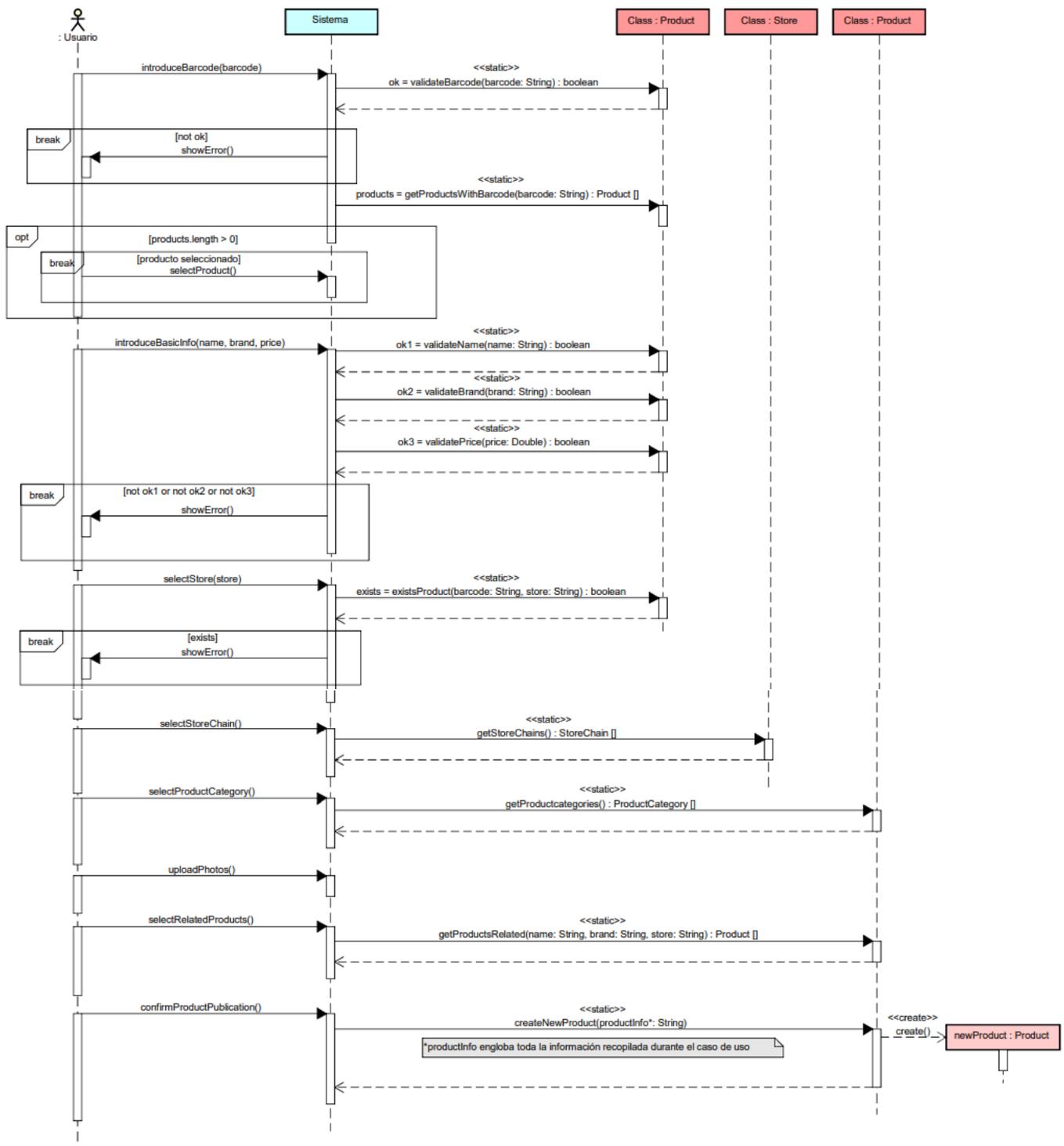


Figura 3.17: Diagrama de secuencia del caso de uso “Modificar información de un producto”

CU-16: Valorar producto

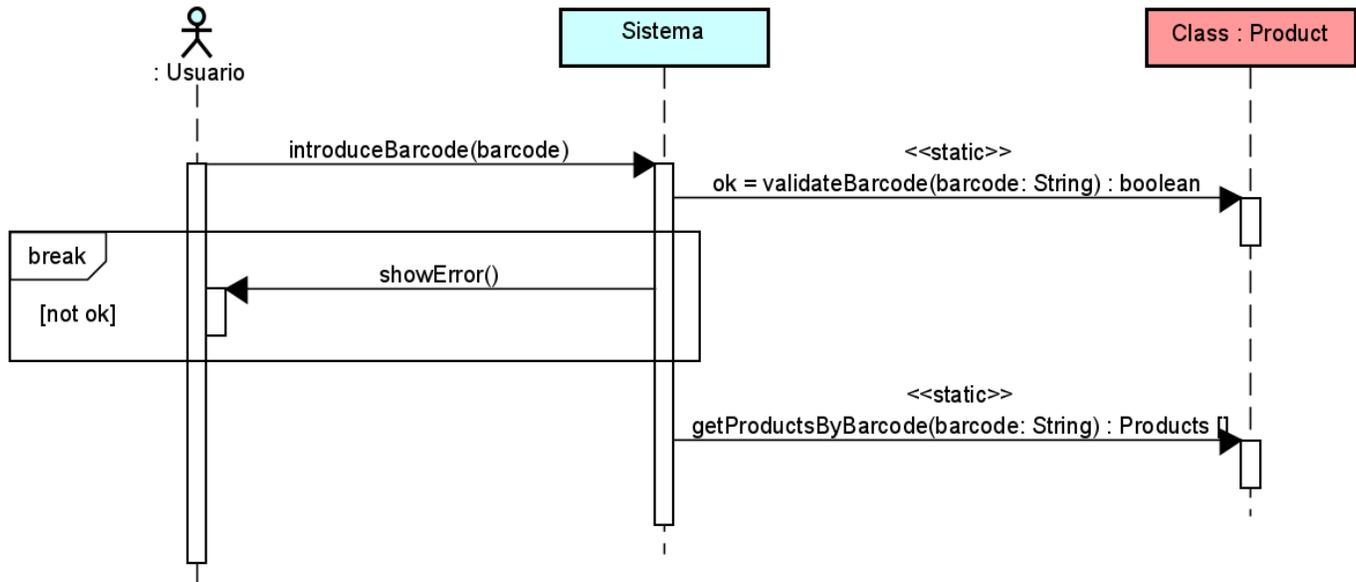


Figura 3.18: Diagrama de secuencia del caso de uso "Valorar producto"

CU-17: Añadir oferta

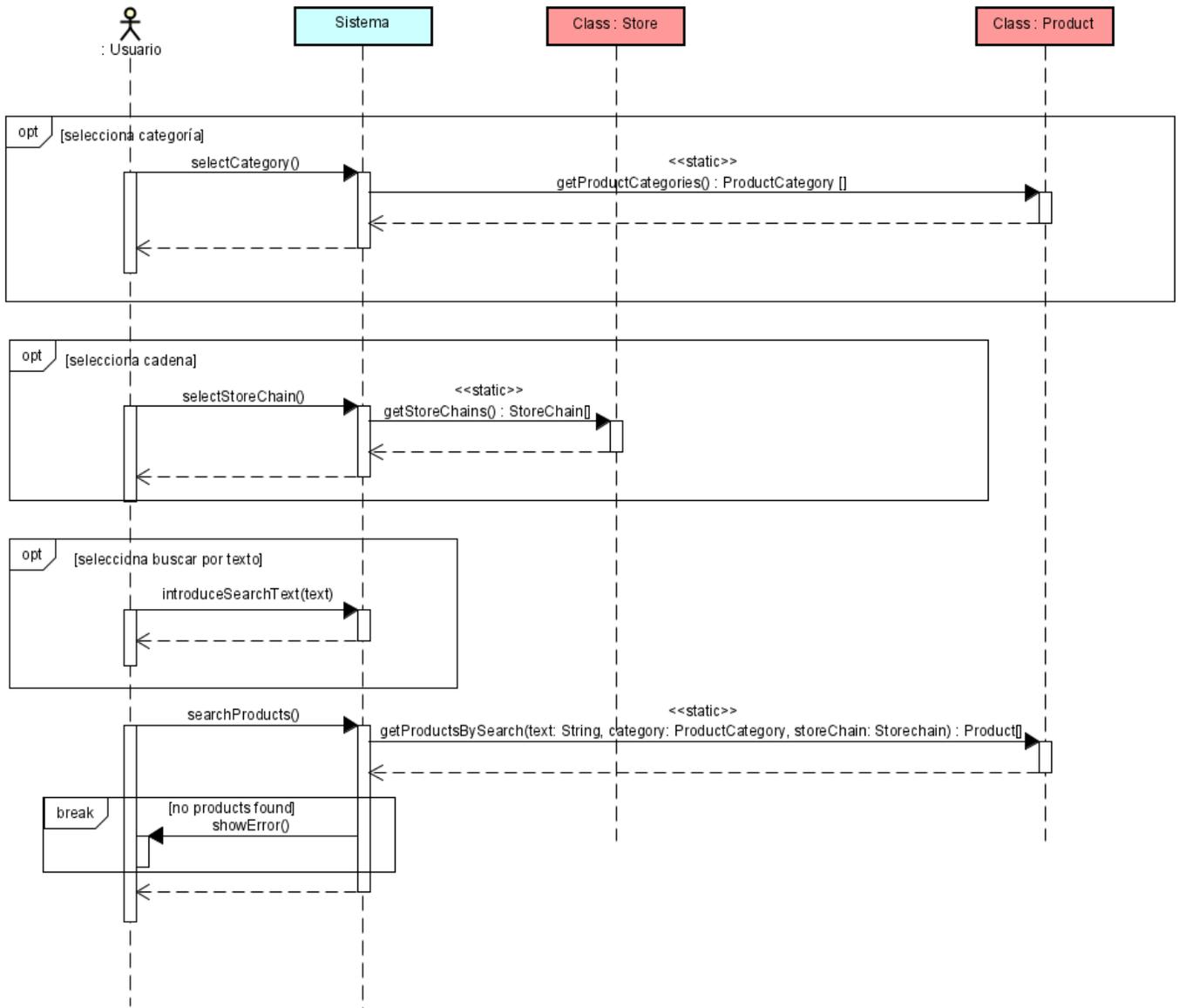


Figura 3.19: Diagrama de secuencia del caso de uso “Añadir oferta”

CU-18: Reportar producto

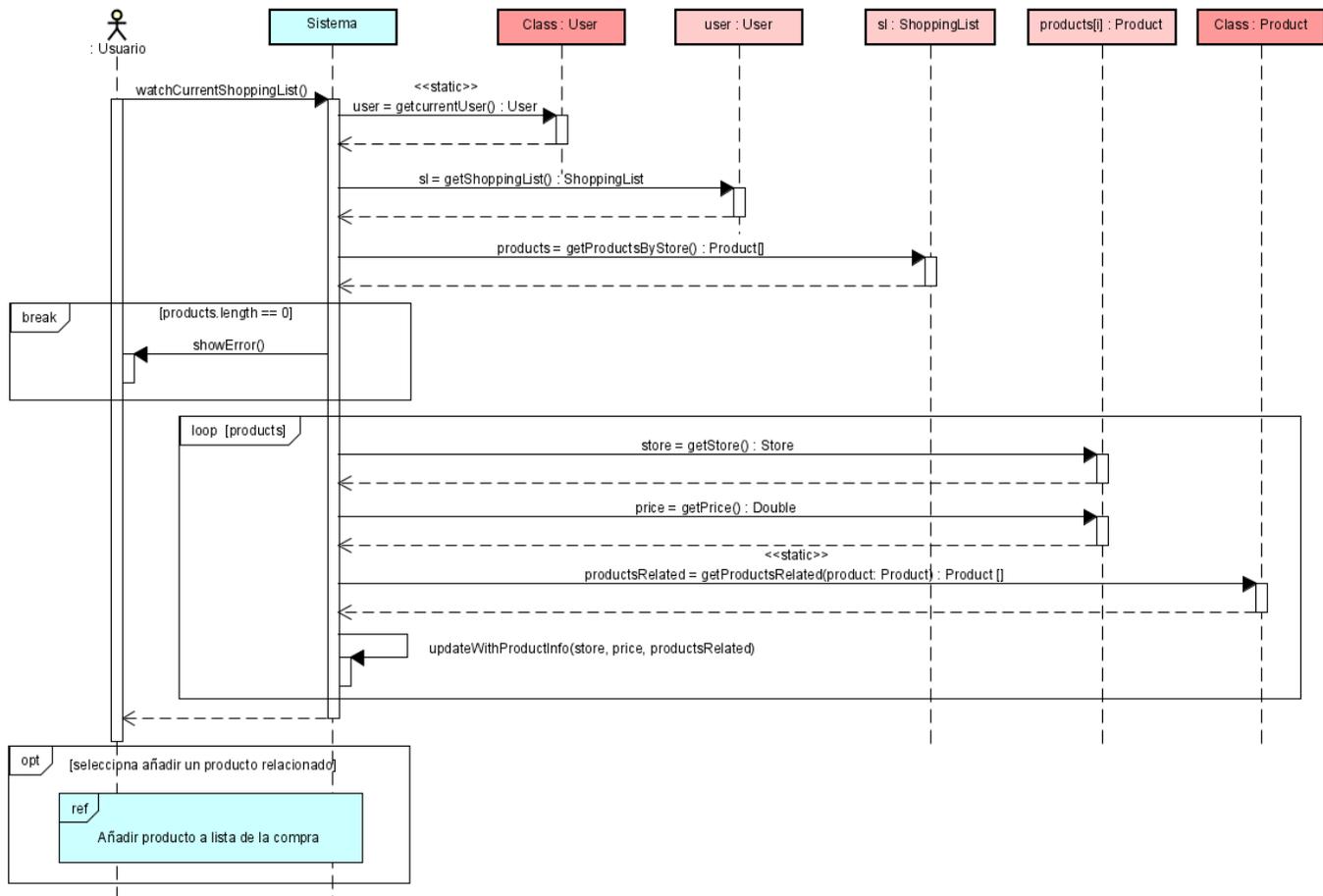


Figura 3.20: Diagrama de secuencia del caso de uso “Reportar producto”

CU-19: Consultar histórico de precios

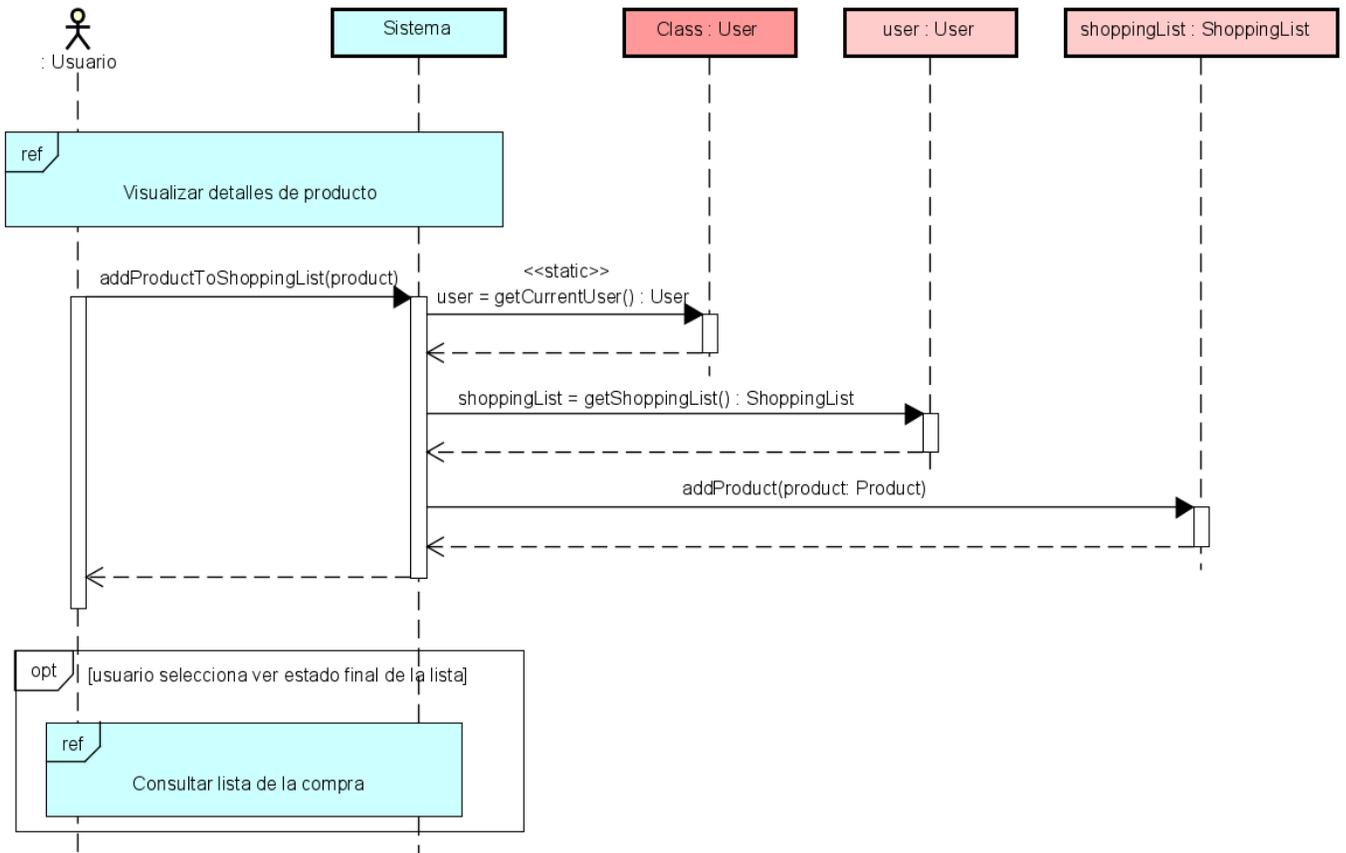


Figura 3.21: Diagrama de secuencia del caso de uso “Consultar histórico de precios”

CU-20: Marcar un producto como guardado

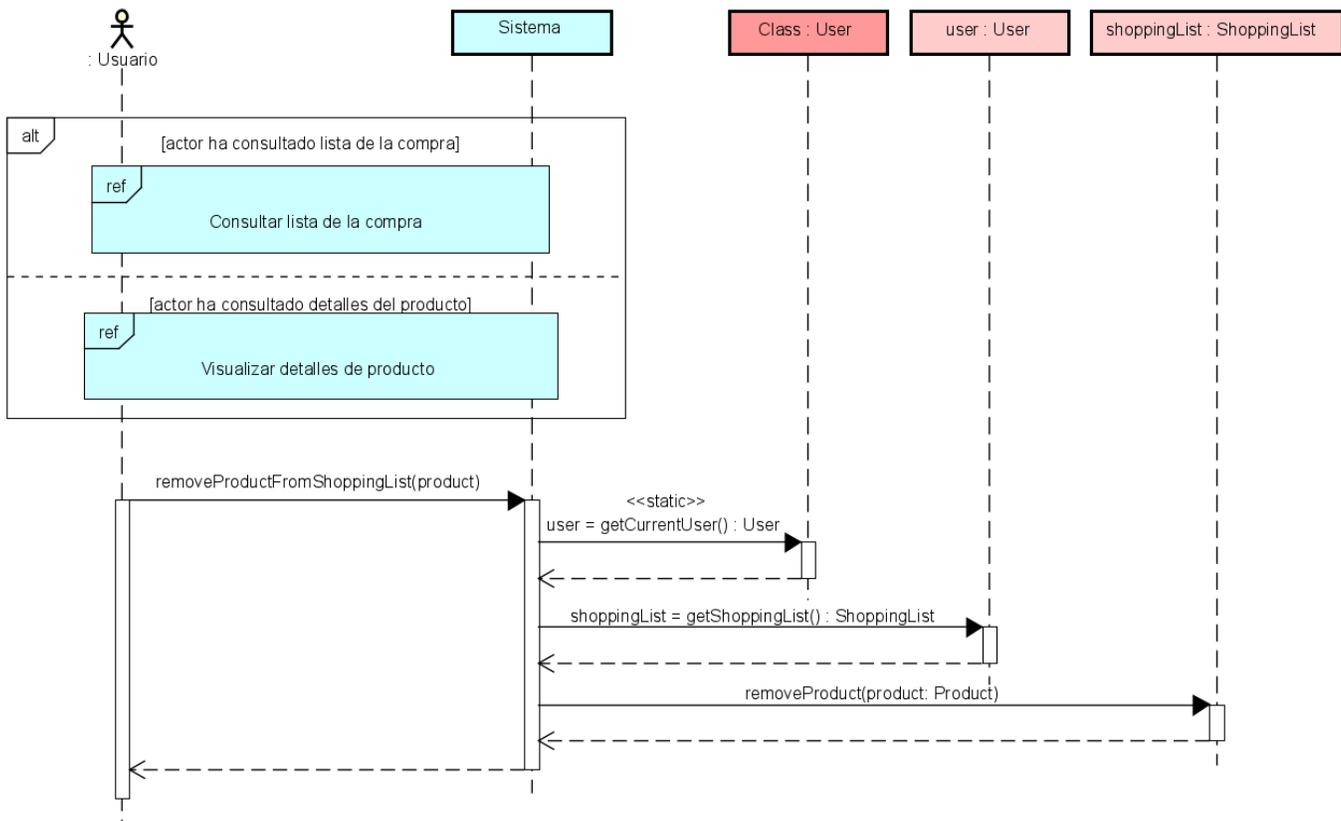


Figura 3.22: Diagrama de secuencia del caso de uso “Marcar un producto como guardado”

CU-21: Desmarcar un producto como guardado

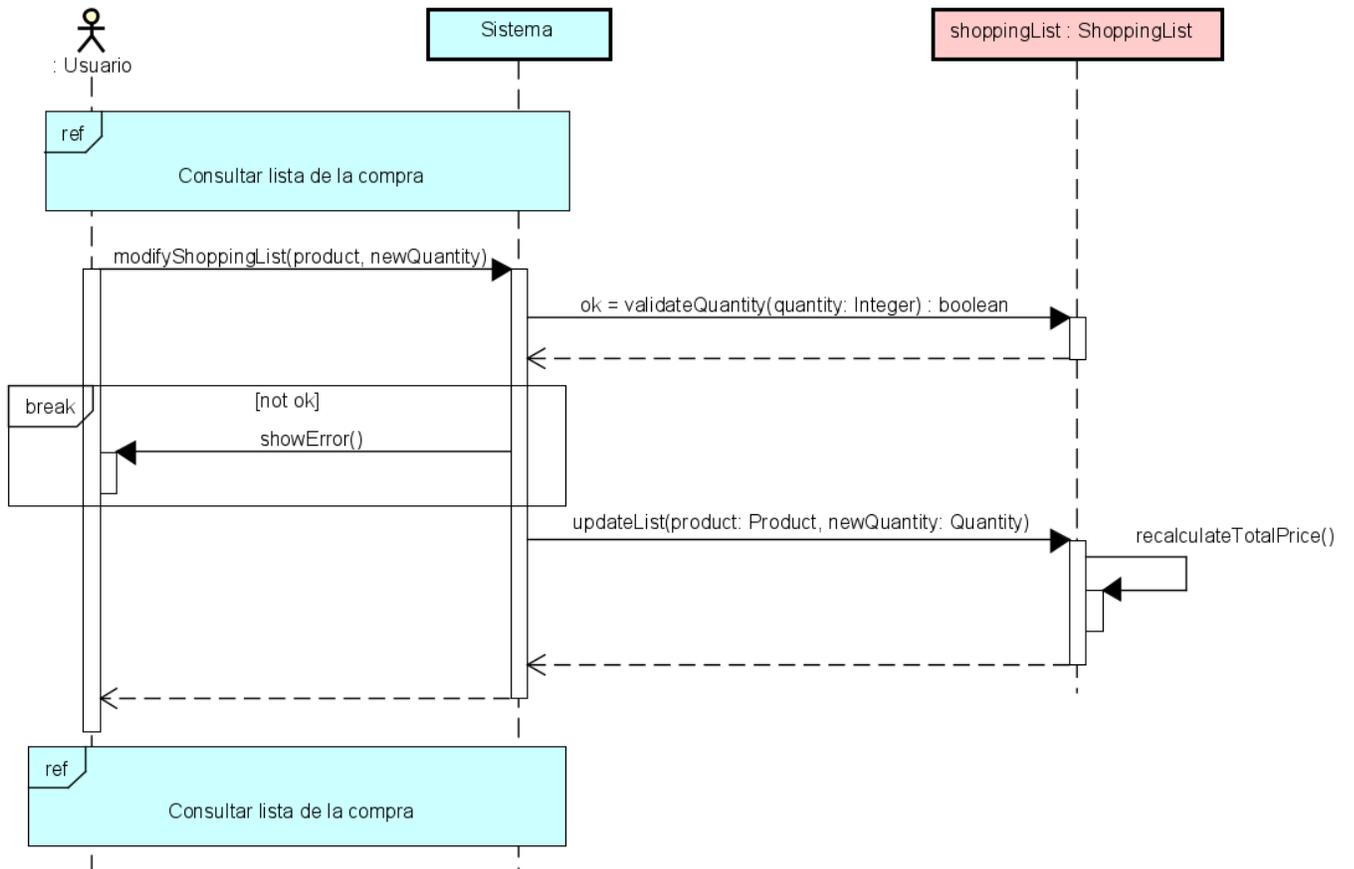


Figura 3.23: Diagrama de secuencia del caso de uso “Desmarcar un producto como guardado”

CU-22: Consultar lista de productos guardados

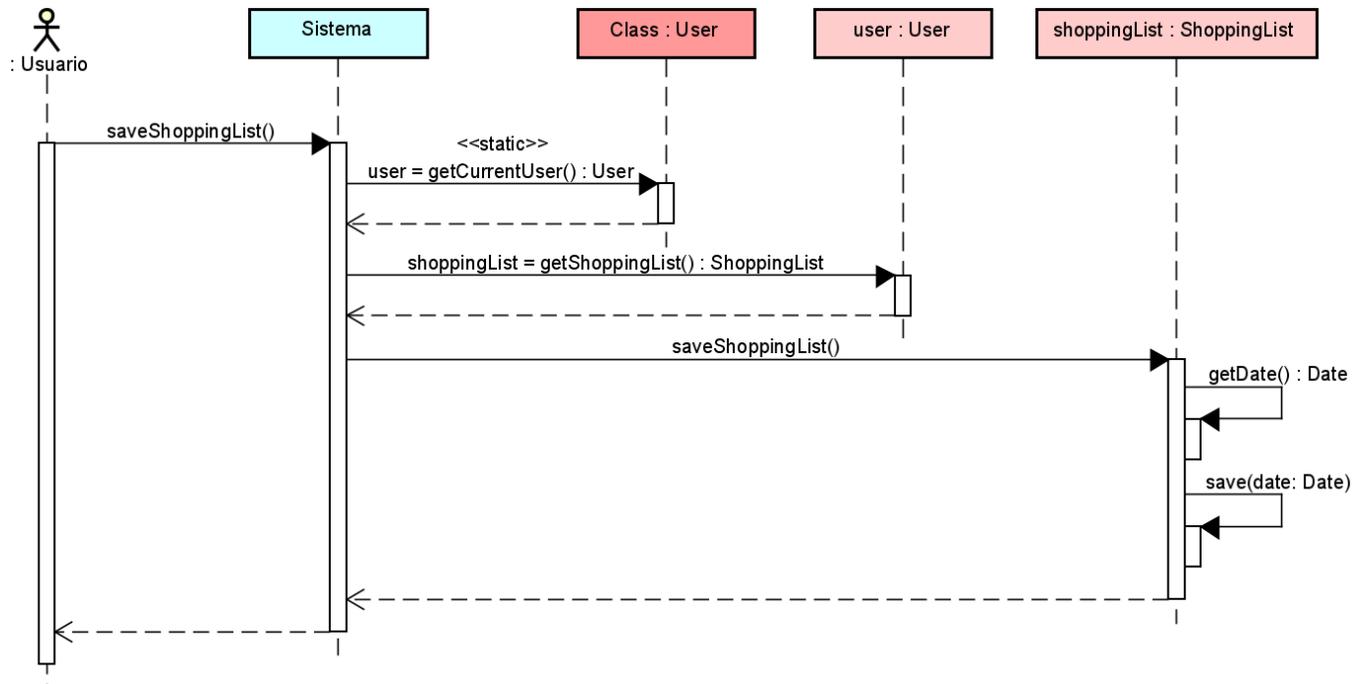


Figura 3.24: Diagrama de secuencia del caso de uso "Consultar lista de productos guardados"

Capítulo 4: Diseño

4.1. Arquitectura del sistema

La arquitectura propuesta para la aplicación se basa en tres capas que englobarán la interfaz, los controladores y el modelo. En la aplicación de este tipo de arquitecturas han surgido patrones que recogen una forma estandarizada de resolver los problemas que puedan surgir.

4.1.1. Patrón arquitectónico MVVM

La mayoría de marcos de trabajo actuales para el desarrollo de aplicaciones que están basados en el patrón de diseño MVC (“Model-View-Controller”). Este patrón nos permite separar la información y esquemas que constituyen el modelo de nuestro sistema, de las representaciones que estos tendrán en las vistas y de los controladores que escucharán los eventos que en ellas se produzcan. El patrón, que se detallará a continuación, es una parte fundamental del *framework* Angular y el uso del mismo requiere la utilización de esta arquitectura.

Existen numerosas implementaciones del patrón MVC partiendo de la descripción básica que hemos realizado en el párrafo anterior. En todas ellas es común que se siga un flujo de actividad como el descrito en la Figura 4.1. Podemos apreciar como el usuario realiza una petición al controlador de una vista, que realiza las modificaciones necesarias del modelo de datos. A continuación, el controlador notificará a la vista el cambio en el modelo de datos y esta presentará los nuevos cambios al usuario.

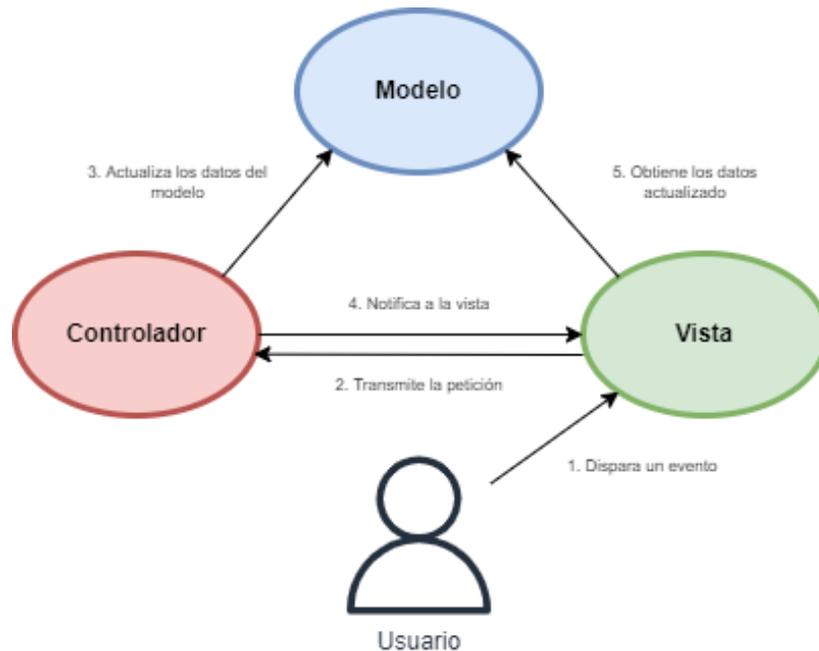


Figura 4.1: Representación gráfica del patrón MVC Pasivo

Podremos encontrar dos implementaciones principales del patrón conocidas como MVC Activo y MVC Pasivo diferenciadas por la capacidad del Modelo de reconocer a la vista y poder comunicarse con ella. No es objetivo de este estudio profundizar en los detalles de las diferentes ventajas e inconvenientes que surgen de las modificaciones de este patrón.

El marco de trabajo utilizado para el desarrollo de este TFG se basa en una modificación del MVC que permite dividir totalmente las implementaciones de la vista y el modelo para así conseguir que ambos puedan funcionar con independencia de los lenguajes utilizados en cada una de ellas. Esta modificación se conoce como MVVM (“Model-View-View Model”) y está específicamente diseñada para interfaces de usuario orientadas a eventos, grupo en el que se incluyen la mayoría de las aplicaciones web, podemos apreciar una comparativa en el estudio de Gloria Arcos-Medina, et al [4].

El patrón MVVM se compone de los siguientes elementos:

- **Modelo:** en este concepto agrupamos aquellos elementos del sistema en desarrollo relacionados con la lógica y los datos del negocio
- **Vista:** en este concepto se agrupan todos los componentes que realizan la función de interfaz con el usuario y que recibirán los eventos que este produzca, mostrando de manera visual los cambios que hayan detonado
- **Modelo de Vista:** con este concepto abstracto se pretende representar la capacidad del patrón de conectar modelo y vistas manteniendo la independencia de ambos. Basa su funcionamiento en la existencia de un enlace o “binder” de datos que utilizará para poder comunicar las actualizaciones entre las otras dos entidades.
- **Enlace Modelo Vista:** es una construcción utilizada para evitar incluir código en las vistas que obtenga la información del modelo lo que aumentaría el grado de dependencia entre ambos. Se implementa mediante sentencias declarativas que se traducirán en un flujo de datos entre modelo y vista, que puede ser unidireccional o bidireccional.

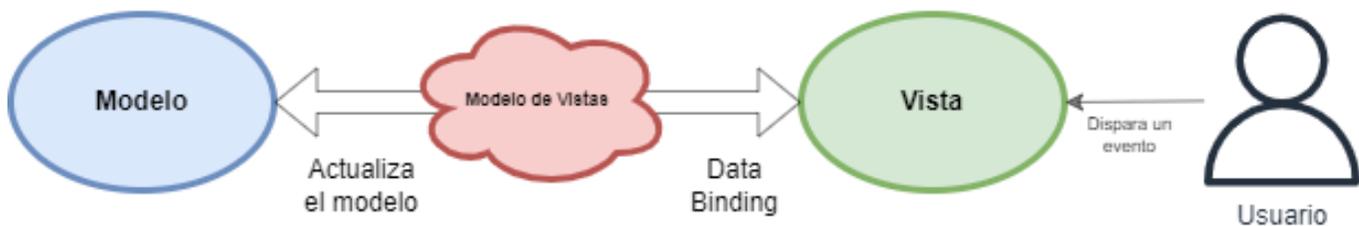


Figura 4.2: Representación gráfica del patrón MVVM

4.1.2. Arquitectura del cliente

Para describir la arquitectura del cliente se van a emplear los diagramas propios de los estilos de documentación *Decomposition Style* y *Uses Style*. Forman parte del modelo de vistas “4+1” que nos permite representar y documentar arquitecturas *software* desde puntos de vistas diferentes, concurrentes y que informan sobre diferentes aspectos del sistema.

En las Figuras 4.3, 4.4, 4.5 y 4.6 se puede apreciar la arquitectura del cliente descrita utilizando este tipo de diagramas.

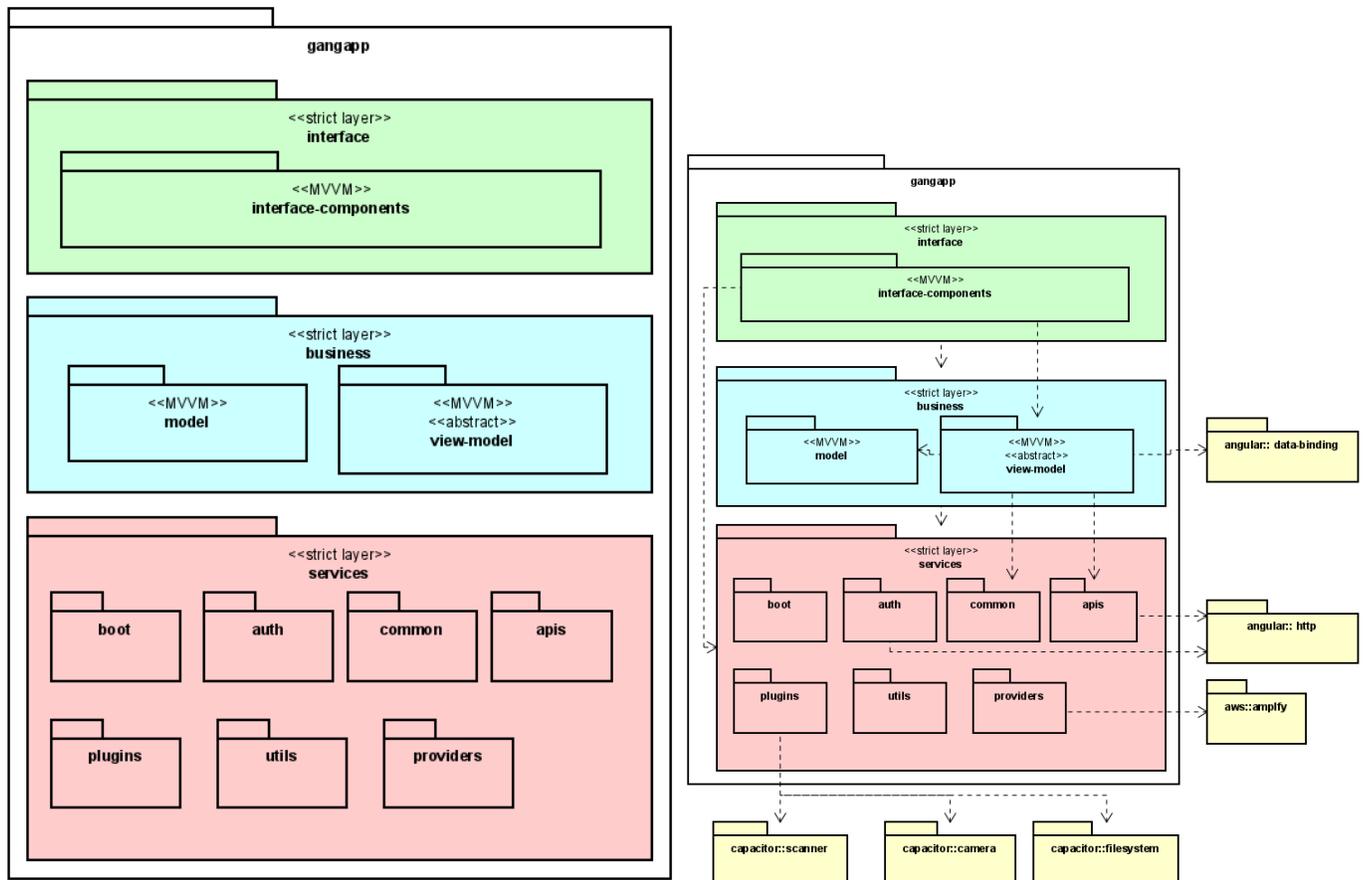


Figura 4.3: *Modules Style* y *Uses Style* general de la aplicación

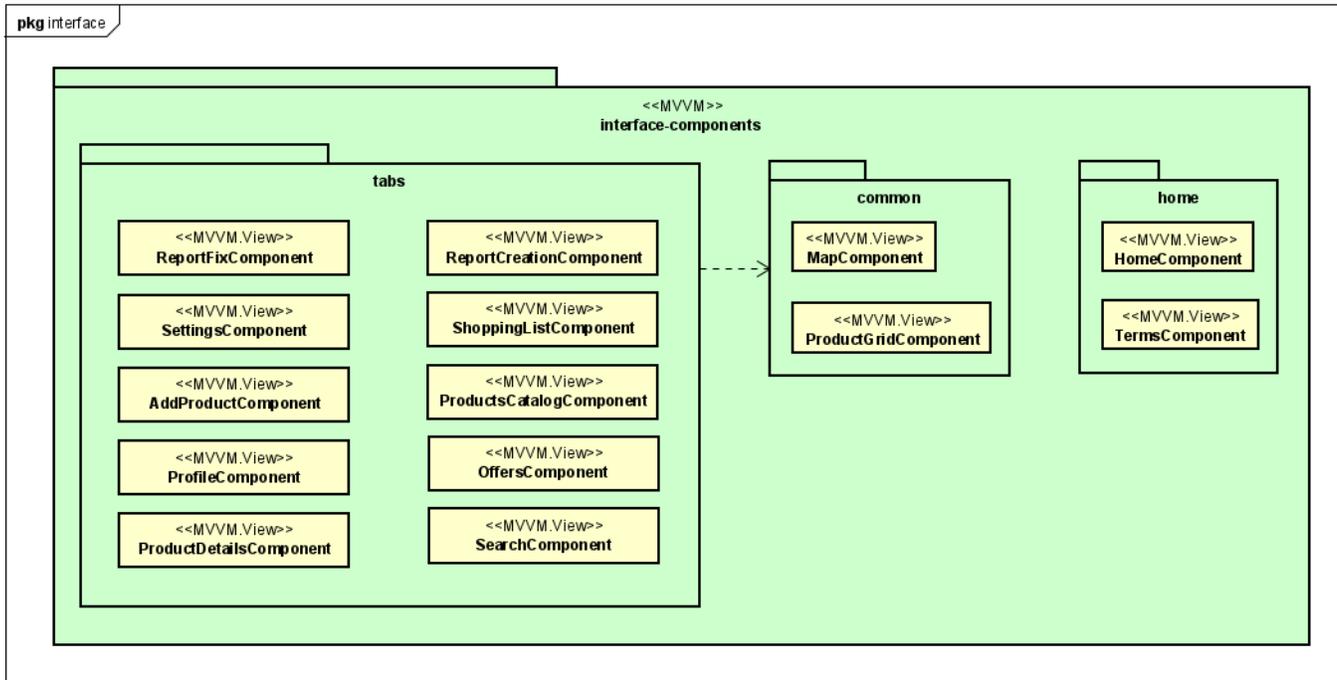


Figura 4.4: Diagramas de la arquitectura del paquete de interfaz

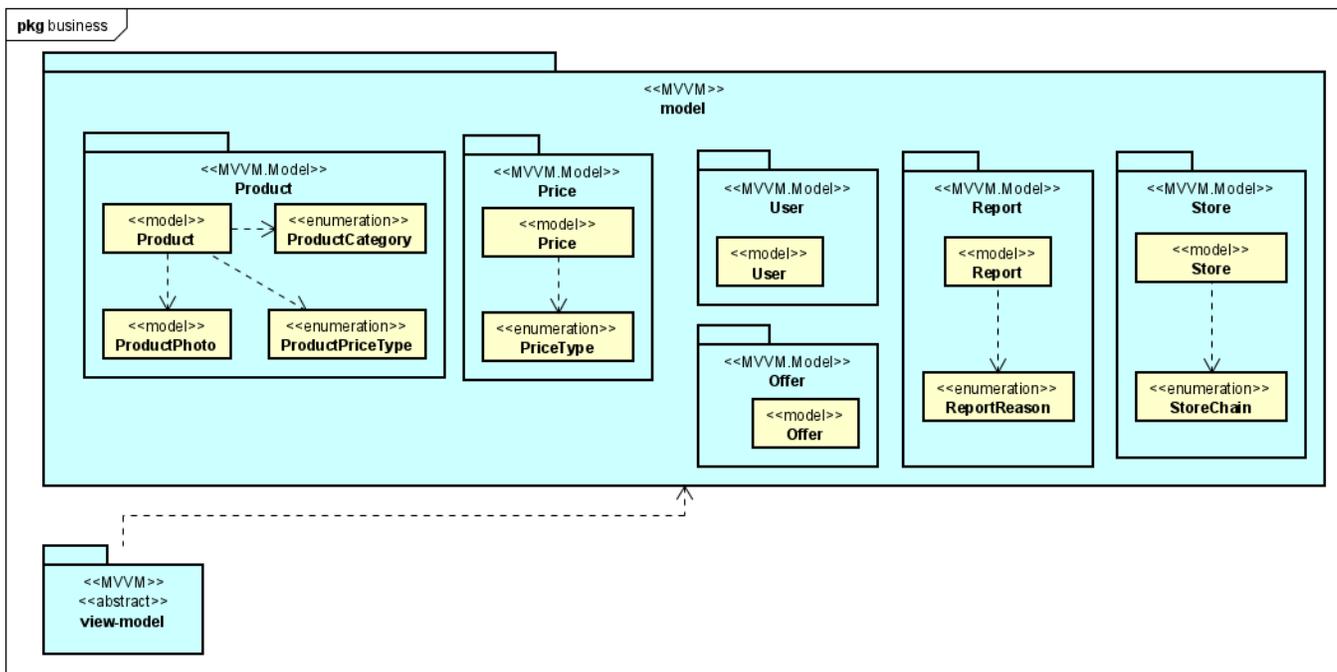


Figura 4.5: Diagramas de la arquitectura del paquete de negocio

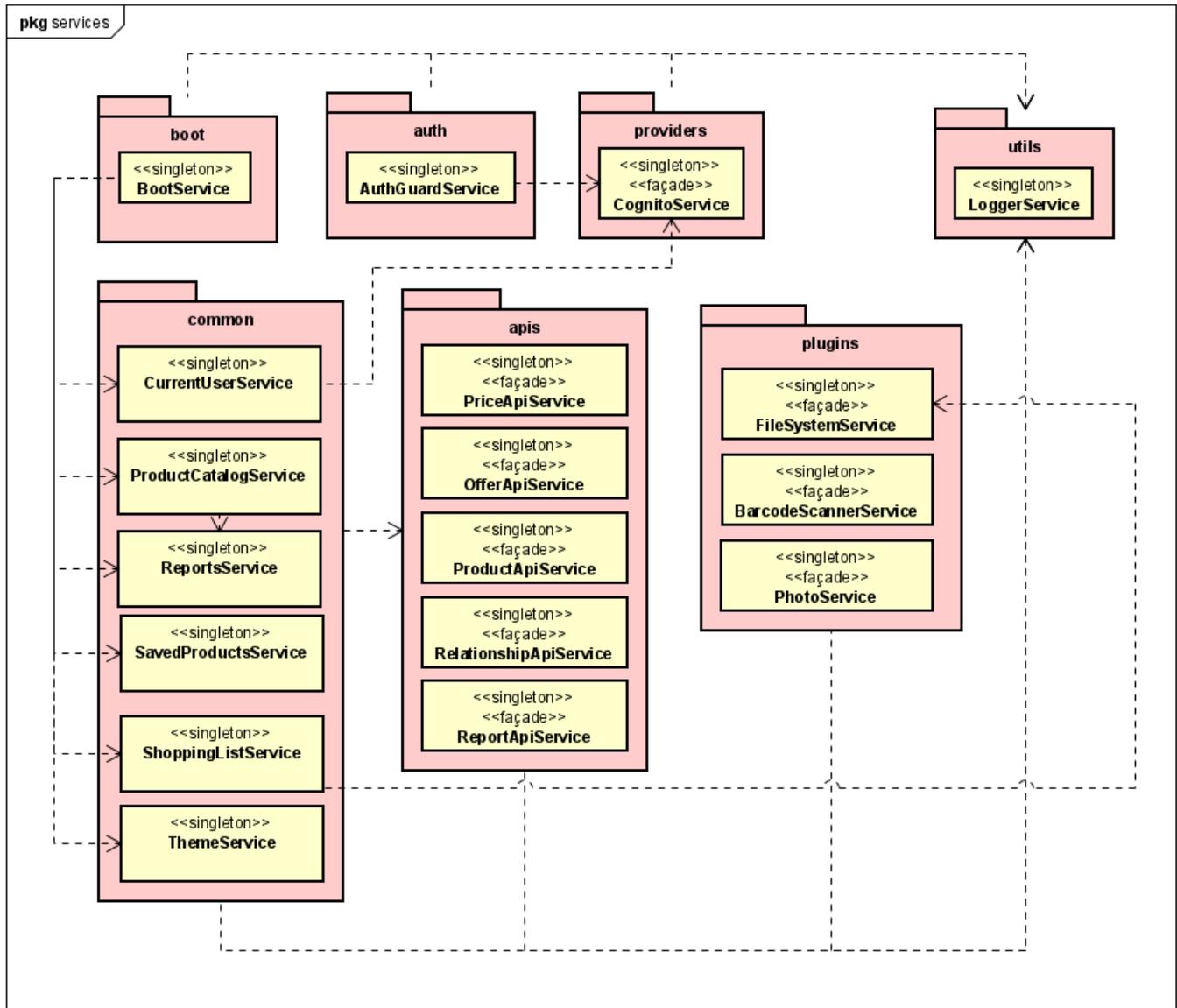


Figura 4.6: Diagramas de la arquitectura del paquete de servicios

4.1.3. Arquitectura del servidor

Este apartado no se adaptará a los esquemas clásicos de documentación de una arquitectura de *software* que se han seguido en el apartado anterior. La principal razón es que para esta aplicación nos hemos basado en una arquitectura *serverless* empleando un serie de servicios predeterminados ofrecidos por Amazon sobre cuyo diseño no se tiene control.

La arquitectura del servidor se basa en cinco servicios de AWS principalmente: Amazon Cognito, AWS Lambda, Amazon API Gateway, Amazon DynamoDB y Amazon S3 Bucket.

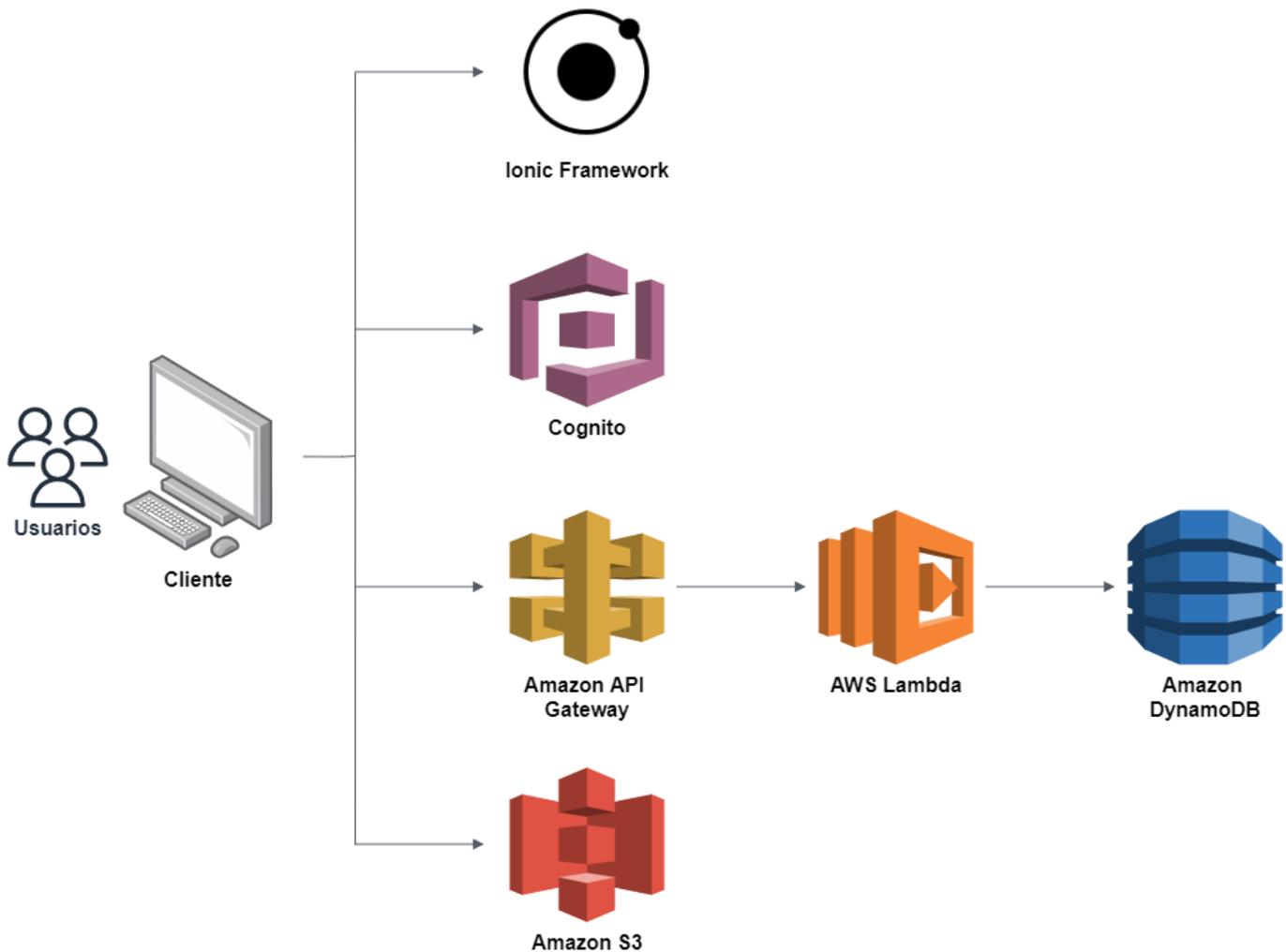


Figura 4.7: Diagrama que representa la arquitectura básica de la aplicación

Mediante estos servicios se ha implementado un sistema de persistencia basado en microservicios de estructura sencilla. El funcionamiento, cuyo diagrama se encuentra en la Figura 4.8, se basa en un flujo de peticiones y respuestas. El *frontend* comienza realizando una petición a uno de los “endpoints” de la API de la aplicación que redirigen la petición a la función Lambda que corresponda. Esta función se encarga de recibir los datos de la petición y en base a estos realizar diferentes solicitudes de datos a la base de datos NoSQL DynamoDB. La base de datos recupera la información solicitada en formato JSON, y se crea una respuesta que se envía de vuelta entre los diferentes servicios hasta llegar al frontal de la aplicación que presentará la información.

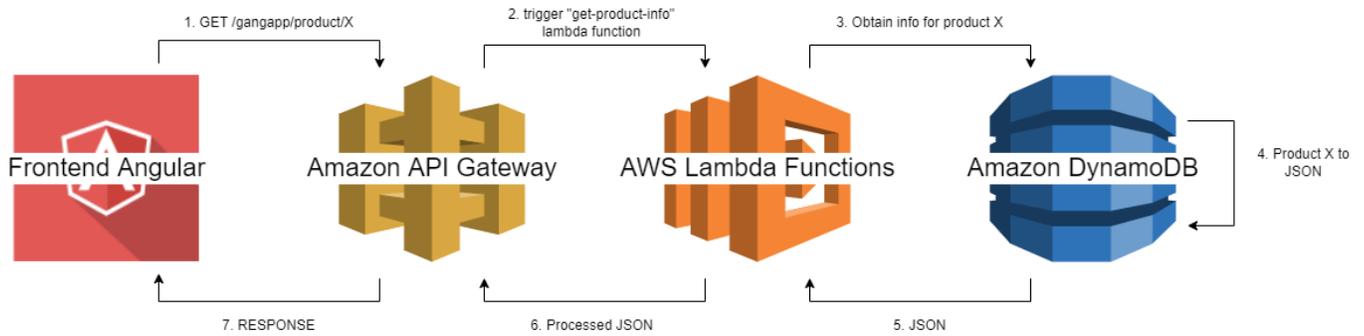


Figura 4.8: Flujo básico de peticiones a la persistencia de la aplicación

Existe una API secundaria para el almacenamiento de archivos binarios. La razón de separarla de la presentada en el anterior párrafo es la dificultad que supone compaginar el acceso a DynamoDB y S3 (que es el sistema de almacenamiento para estos archivos) en las mismas APIs. Este recurso está destinado específicamente a la subida y recuperación de los archivos multimedia (fotografías en este caso) que los clientes de la aplicación generan durante el uso. El flujo es mucho más sencillo que el anterior ya que sólo se necesita realizar una simple petición a la API y esta conecta directamente con el almacenamiento estático tanto para almacenar nuevos archivos como para recuperar los ya existentes.

A continuación se va a detallar el API REST finalmente utilizado. Los servicios AWS que se han utilizado para la implementación se detallarán en la sección 5.1.5 puesto que no son parte de la etapa de diseño.

API REST

En esta sección se detallará la API REST que gestiona las peticiones a los servicios de AWS encargados de la persistencia de los datos de la aplicación.

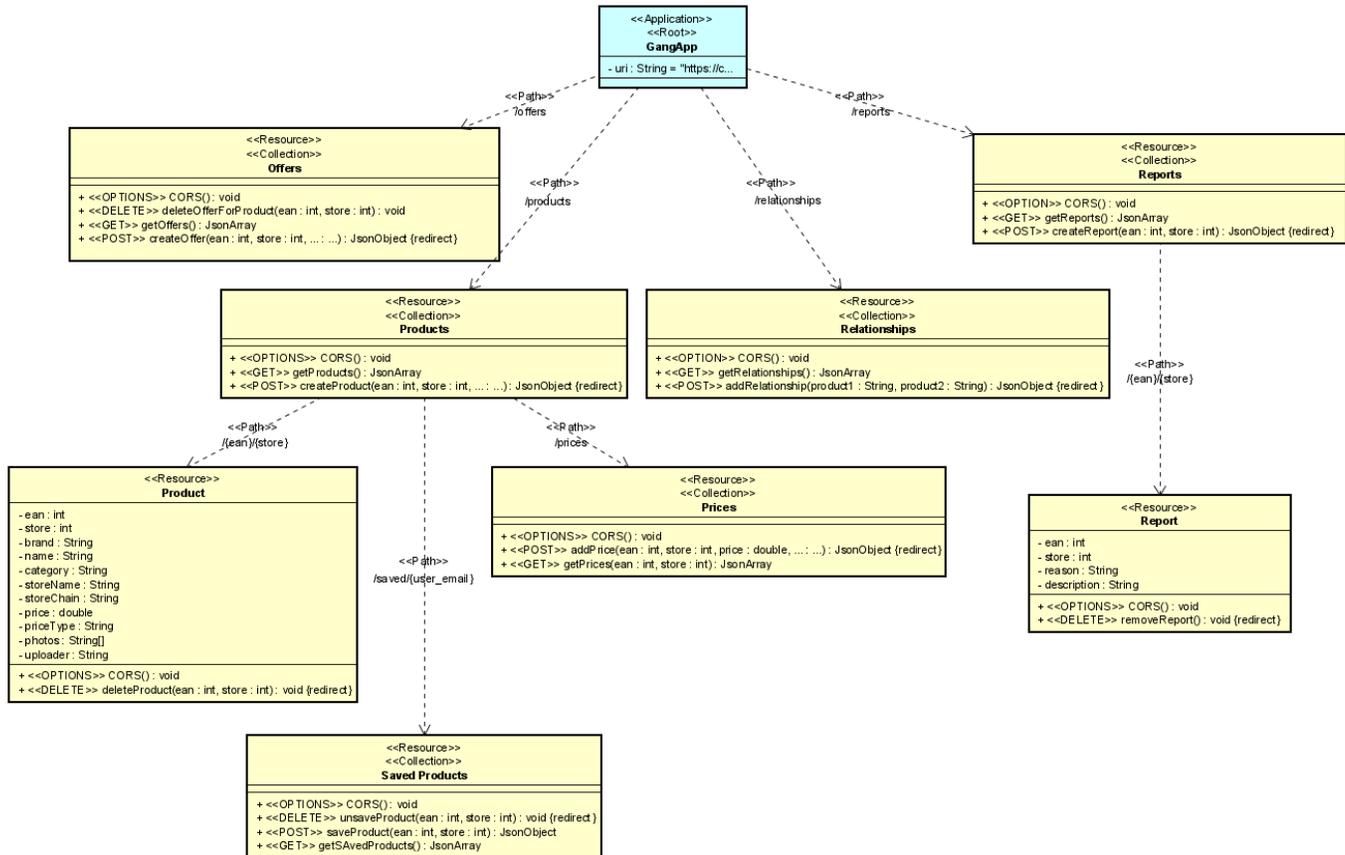


Figura 4.9: Diagrama de recursos de la API REST

4.2. Otros patrones presentes en la aplicación

A continuación se explicarán los patrones que se han aplicado en el desarrollo del producto bien por decisión explícita de los desarrolladores o por ser un aspecto clave de alguno de los *frameworks* y servicios utilizados, salvo el patrón MVVM ya descrito en la sección 4.1.1.

4.2.1. Patrón *Facade*

Las comunicaciones entre los subsistemas aumentan considerablemente la complejidad de los mismos. Las diferentes clases de estos subsistemas pueden interactuar y proveerse mutuamente de servicios. Para simplificar el uso de los sistemas, mejorar la mantenibilidad del código y ofrecer un único punto de entrada a la funcionalidad contamos con el patrón Fachada [11]. Es un patrón de diseño estructural que consiste en la implementación de una interfaz que unifique las existentes en el subsistema. Todos los servicios que pretendan utilizar la funcionalidad provista por el subsistema deberán utilizar este punto común.

El ejemplo de uso de este patrón se puede observar en la capa de servicios del frontal de la aplicación. Se han creado una serie de clases que ofrecen métodos sencillos para el trabajo con las funcionalidades provistas por los dispositivos móviles recogidas en la librería Capacitor [15] y otros paquetes auxiliares. De esta manera, si una clase quiere tomar una fotografía no necesita conocer como (ni con qué paquete) se ha implementado la funcionalidad para poder utilizarla.

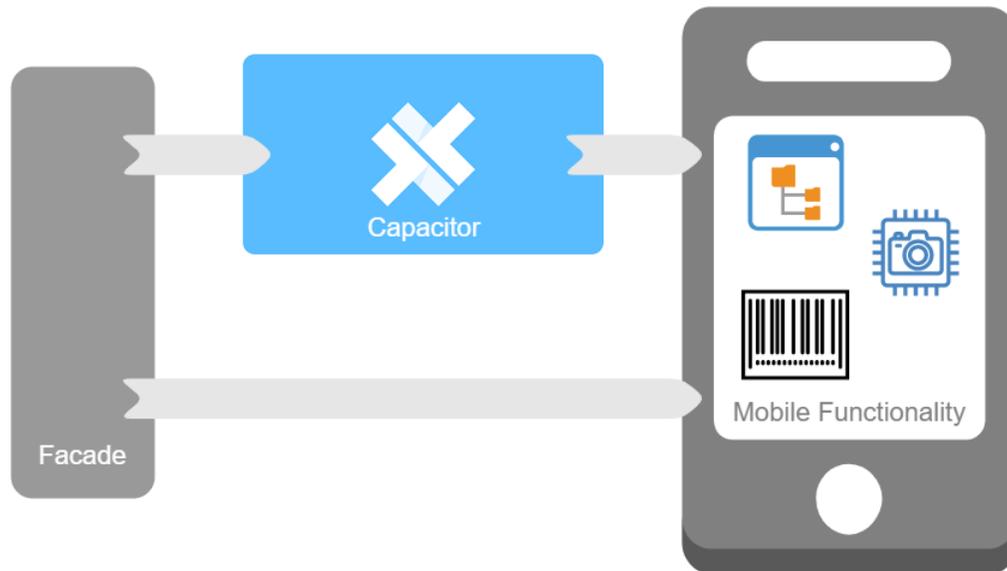


Figura 4.10: Esquema del uso del patrón Fachada en la aplicación

4.2.2. Patrón *Inversion of Control*

Este patrón, muy utilizado en el desarrollo de *frameworks*, se utiliza para alterar el flujo normal de ejecución de los programas. Se basa en evitar realizar las funciones directamente y otorgar a los programas una forma de decidir cómo realizar una determinada tarea. Existen múltiples formas de implementar esta inversión de control con patrones conocidos como “Factory” o “Dependency Injection”. Es en este último en el que nos vamos a centrar.

La inyección de dependencias es un aspecto clave de Angular. Es una característica principal de los componentes. Cuando vayamos a crear nuestros propios componentes los declararemos (servicios generalmente) y será el propio Angular quien se encargue de instanciar y pasarle estos servicios a los diferentes componentes. Es interesante destacar que Angular mantiene una lista o “pool” de componentes y utiliza un patrón *Singleton* (si no indicamos lo contrario) de forma que sólo debe instanciarlos una vez, este patrón vamos a analizarlo en detalle a continuación.

4.2.3. Patrón *Singleton*

En ciertas ocasiones queremos que una clase sólo se instancie una vez y todas las clases que necesiten de los servicios de esta clase usen esa única instancia. El patrón creacional “Singleton” nos permite obtener este comportamiento en la instanciación de los objetos.

Cómo ejemplo de esto tenemos el conjunto de Servicios mantenido por Angular durante la ejecución de una aplicación web. Angular nos facilita así el traspaso de información de unos componentes a otros usando servicios comunes sin tener que preocuparnos de que la instancia del servicio que este usando un componente sea la misma instancia que la que usará otro componente en el futuro.

En nuestro caso es especialmente útil en la utilización del catálogo de la aplicación. Recuperamos en el arranque toda la información sobre los productos almacenados y durante el uso de la aplicación cualquier componente que utilice

este servicio de catálogo tiene acceso a los productos del sistema sin que sea necesario volver a solicitar la información a la base de datos.

4.2.4. Patrón DAO

Para solucionar el problema que la relación entre la lógica del negocio y el almacenamiento en persistencia conlleva, se utiliza el patrón “Data Access Object”. Como se expone en el estudio sobre patrones de diseño realizado por D. Matic, H. Kegalj y D. Butorac (2003) [3] tener una lógica de persistencia incluida en los objetos del negocio facilitará que ante cualquier cambio tanto en el propio negocio como en la forma de persistir los datos requiera un cambio en la aplicación, esto es grave para productos en fase de desarrollo pero crítico para aquellos en producción. Para evitar esta situación pretendemos desacoplar la lógica de acceso a los sistemas de almacenamiento de información de los objetos que van a ser almacenados mediante un objeto intermedio que realiza las traducciones entre las dos capas: negocio y persistencia. En la Figura 4.11 se pueden apreciar los diferentes elementos del patrón en un esquema inspirado en el estudio mencionado previamente.

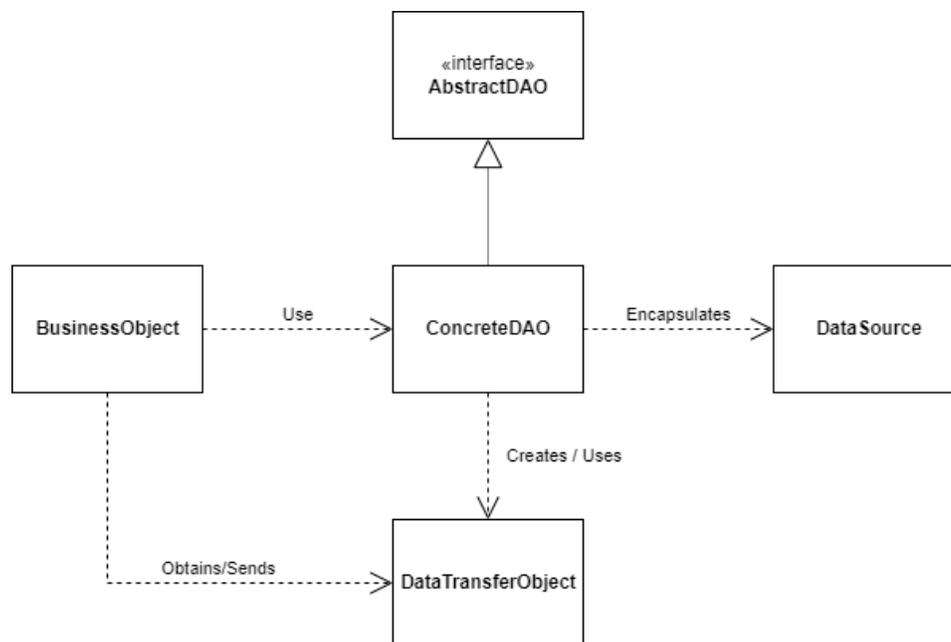


Figura 4.11: Ejemplificación del patrón DAO

Esos objetos conocidos comunmente como “DAO” implementan una interfaz que unifica la forma de acceder a los datos y mediante un objeto de un formato preestablecido conocido como “Data Transfer Object” o “DTO” permiten el intercambio de información desacoplando totalmente las bases de datos utilizadas de la información almacenada y facilitando en gran medida situaciones como cambios en los gestores de la persistencia.

Para la implementación de este patrón en el sistema que puede observarse en el control de versiones (Anexo 7.3) se han utilizado servicios de Angular como clases DAO que acceden a la persistencia personalizadas para los diferentes objetos del modelo y empleando JSON como DTO, utilizando como inspiración los ejemplos recogidos en los recursos web [26, 14].

4.3. Modelo de datos

Ya se ha presentado el servicio que vamos a utilizar para la persistencia, AWS DynamoDB que como se ha comentado es una base de datos de tipo “clave-valor”. Durante la carrera se han obtenido unas buenas bases en el análisis, diseño, desarrollo, documnetación y pruebas de bases de datos relacionales o SQL. Es por esto, que aún teniendo las grandes ventajas de las bases de datos NoSQL como DynamoDB se ha reflejado en la persistencia un esquema propio de un sistema de almacenamiento del primer tipo.

Los diagramas necesarios para la documentación de bases de datos no relacionales no están igual de especificados que para aquellas que sí los son. Por esta razón y tras consultar el artículo *NoSQL Database Design Using UML Conceptual Data Model Based on Peter Chen’s Framework* [8] se ha optado por el diagrama presente en la Figura 4.12

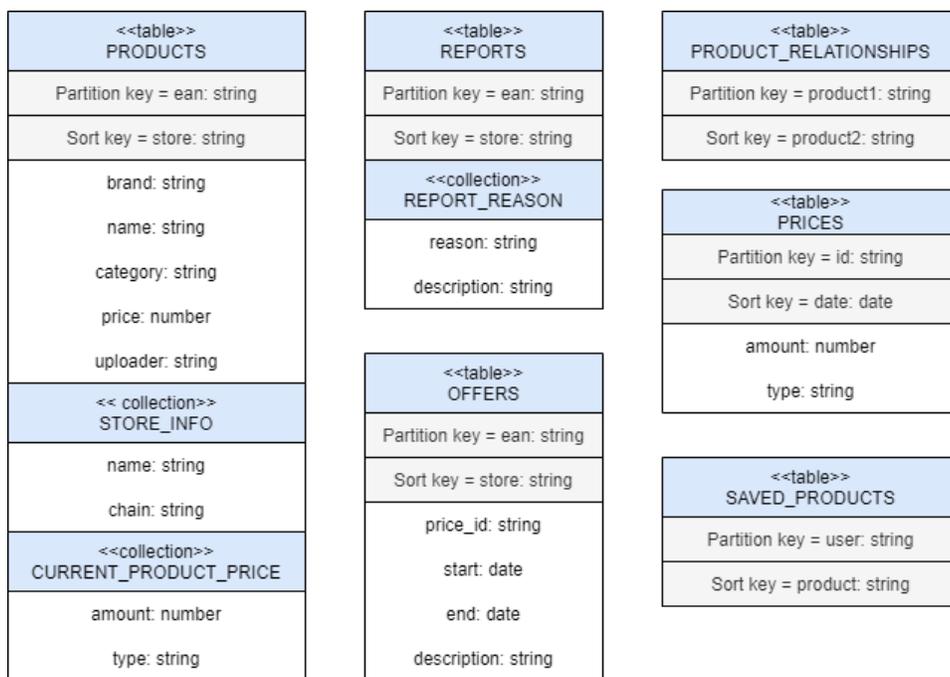


Figura 4.12: Modelo lógico de datos

4.4. Clases de diseño detalladas

A continuación se detallan en las figuras 4.13, 4.14 y 4.15 los detalles de las clases en la etapa de diseño.

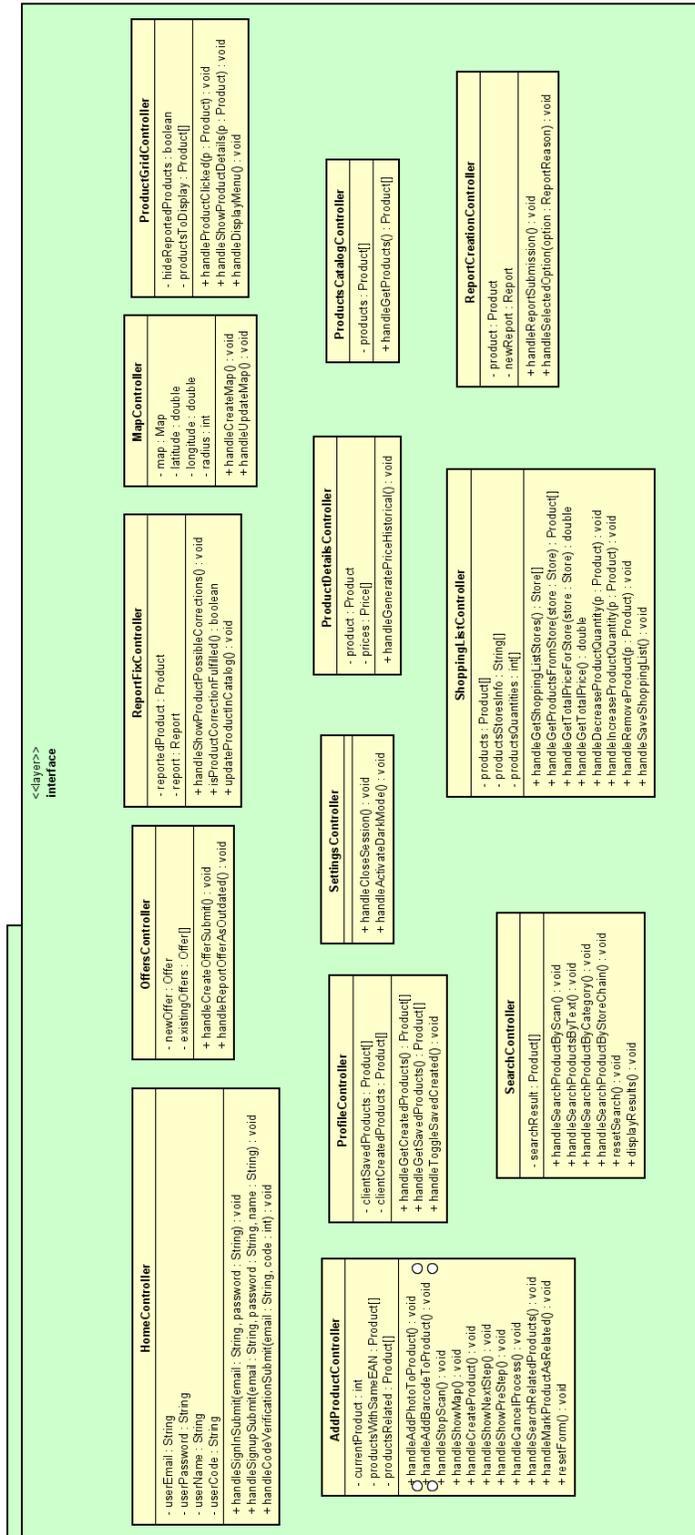


Figura 4.13: Clases de diseño detalladas para la capa de Interfaz

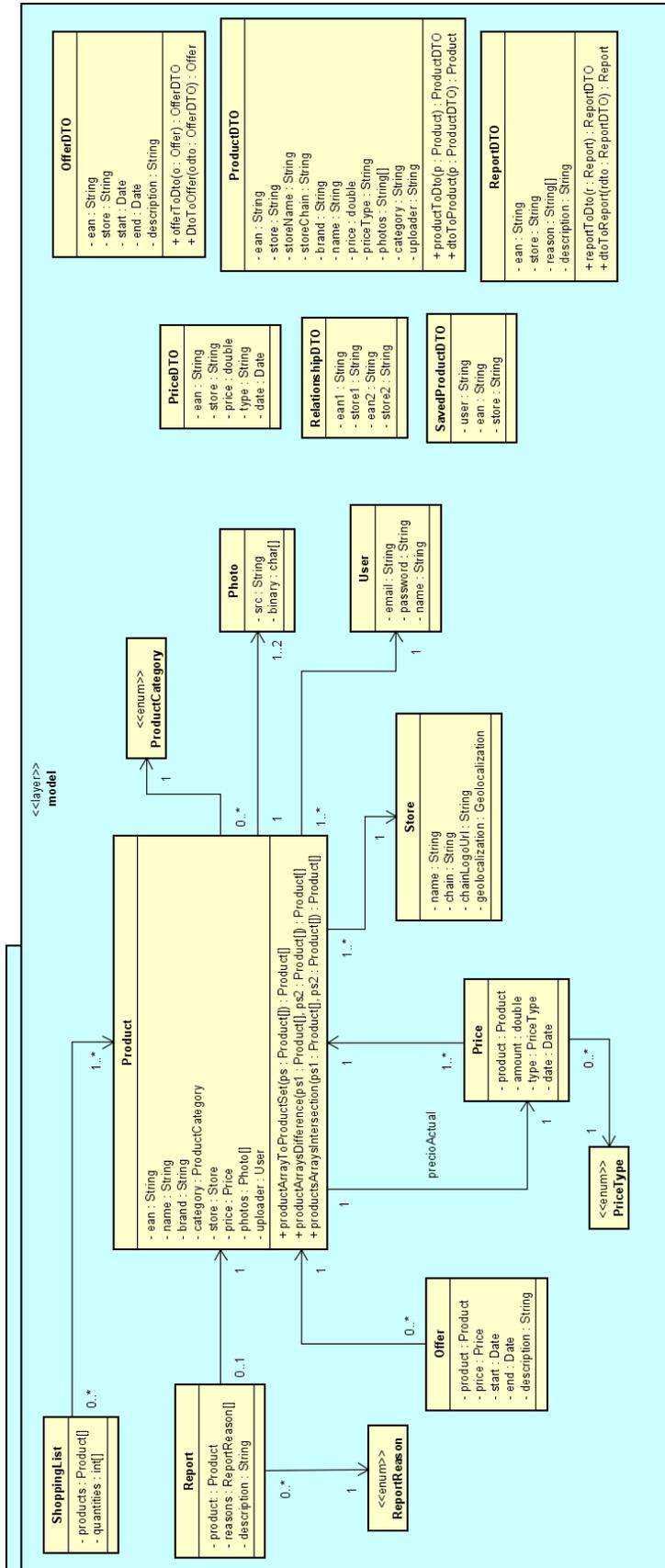


Figura 4.14: Clases de diseño detalladas para la capa de Modelo

4.5. Privacidad desde el diseño y por defecto

La privacidad está adquiriendo un rol cada vez más importante en todos los procesos que conllevan el tratamiento de datos con carácter personal. Por esta razón la tendencia es tener siempre presente durante la ejecución de un proyecto los aspectos de privacidad y seguridad de los datos.

La privacidad desde el diseño y por defecto (“Privacy by Design”) es un concepto recogido en la RGPD que incita a diseñar paralelamente los aspectos funcionales de la aplicación y los aspectos de tratamiento de los datos de los potenciales clientes.

Las características fundamentales de este principio extraídas de análisis como el de Grupo Ático 34 [12] y en base a la información recogida en la guía de protección de datos publicada por la AEPD en 2020 [2] son:

- **Preventivo, no correctivo; proactivo, no reactivo:** las medidas deben orientarse primordialmente a la reducción o eliminación de los riesgos, no a su posible mitigación
- **Privacidad de forma predeterminada:** el usuario no debe solicitar que se tenga en cuenta su privacidad
- **Privacidad como parte del diseño:** debe considerarse un elemento crucial en cualquier desarrollo que se lleve a cabo
- **Funcionalidad completa:** la privacidad no debe interferir en la operatividad de las aplicaciones
- **Seguridad de extremo a extremo:** durante todo el ciclo de tratamiento de datos se debe garantizar la seguridad e integridad en la protección de los mismos
- **Visibilidad y transparencia:** item Respeto por los usuarios: Para mantener la confianza de los usuarios en el sistema se debe seguir un modelo transparente de tratamiento ofreciendo facilidades para su verificación y trazabilidad
- **Respeto por la privacidad de los usuarios:** es necesario situar al usuario en el centro del diseño realizado según este principio.

Se describirán a continuación las medidas llevadas a cabo para garantizar la aplicación de estos principios

Políticas de usuario

Previamente al registro de cualquier usuario en la aplicación este debe leer y aceptar las condiciones de uso de la aplicación. En estas condiciones se describe cómo van a ser tratados los datos de carácter personal recopilados y aquellas acciones que los usuarios pueden llevar adelante para ejercer sus derechos en relación con la privacidad como, por ejemplo, el “derecho al olvido”. También se describen los métodos de contacto con los administradores de la aplicación para posibles reclamaciones o dudas con respecto a estas condiciones o a su aplicación.

En estas políticas se recogen, también, los términos que el usuario debe respetar en el uso de la aplicación precisamente destinados a poder garantizar la seguridad de sus datos.

Tratamiento de los datos recogidos de los usuarios

En todo momento se ha buscado reducir los datos necesarios para el funcionamiento de la aplicación que a los cruciales, ya que cuantos menos datos sea necesario almacenar, menores riesgos se van a asumir. Finalmente sólo se han almacenado datos personales considerados de carácter identificativo según [13], como el correo electrónico, para los inicios de sesión.

Capítulo 5: Implementación

5.1. Tecnologías utilizadas

5.1.1. Generación de recursos y documentación visual

Esta sección recoge las tecnologías utilizadas para la creación de las figuras utilizadas en esta documentación.

Diagrams.net

Esta aplicación web (previamente conocida como “Draw.io”) recoge una serie de componentes visuales y conectores que pueden ser utilizados para crear esquemas de documentación de diferentes tecnologías y productos *software*.

Su primera ventaja es la capacidad de personalización que ofrece gracias a sus plantillas. Estas plantillas predefinidas de diferentes clases (incluyendo diagramas UML2.0 con una interfaz moderna) nos permite tener un punto de partida para el recurso gráfico que queremos construir. Podemos ver la clasificación de algunos de estos esquemas en la captura de la izquierda de la Figura 5.1

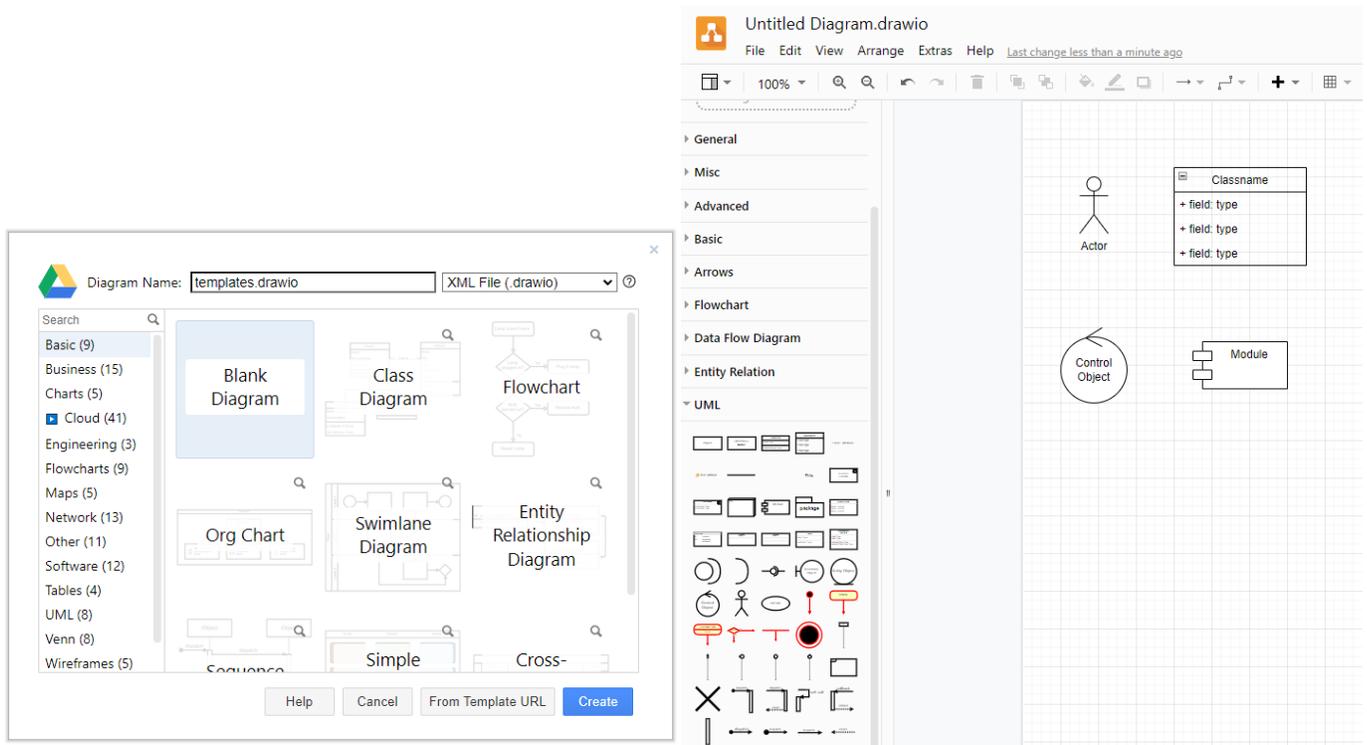


Figura 5.1: Capturas de pantalla de la aplicación “Diagrams.net”

La segunda de las ventajas está relacionada con su integración en plataformas de almacenamiento en la nube como Google Drive. Todos los archivos pueden ser creados en una carpeta virtual dentro de este sistema y podremos recuperarlos desde ahí en el caso de querer modificarlos en el futuro.

Finalmente podremos utilizar una serie de iconos de diferentes categorías o incluso importar nuestras propias imágenes para construir los diagramas deseados, tal y como se presenta en la captura a la derecha de la Figura 5.1

Astah

Junto con Visual Paradigm, Astah es la herramienta de modelado de productos *software* empleado durante la carrera. Con ella se han generado la mayoría de artefactos de las diferentes etapas del desarrollo. Como ya se ha mencionado, es una herramienta muy conocida y no se va a profundizar más en su uso. En cualquier caso y para cualquier duda, se puede consultar su página web [16] donde podemos encontrar documentación y ejemplos de uso.

Monday.com

Es una herramienta de colaboración diseñada para la gestión de equipos. Permite crear tableros y diagramas para empresas de multitud de ámbitos mediante una interfaz de usuario muy visual y sencilla en el manejo. El uso que se le ha dado no tiene que ver con la gestión de equipos sino con la planificación puesto que este es el programa con el que se han generado las planificaciones de la sección 2.3.2

5.1.2. Creación de la memoria

Esta sección recogerá los programas utilizados para realizar el documento de memoria asociado a este trabajo de fin de grado.

LaTeX

Se trata de un sistema que incluye el programa procesador de textos del mismo nombre y todas las herramientas necesarias para la gestión de paquetes y la compilación de los archivos, basado en la creación de textos con un lenguaje de marcado de etiquetas que luego se compila para generar documentos como PDF. Su uso está extendido para la creación de libros, artículos académicos y cualquier publicación que requiera elementos de complejidad como bibliografías, tablas, fórmulas matemáticas, etc.

Es un *software* de licencia libre de gran profundidad cuya principal ventaja es la cantidad de paquetes creados por la comunidad de los que se disponen. Además de estos paquetes que pueden incluirse en las distribuciones que estamos usando, cualquier elemento o etiqueta puede ser configurado para ajustarse a las necesidades de la persona que lo esté utilizando.

Otra de sus claves es la capacidad para separar el contenido de lo que quiere presentarse, del estilo en el que va a ser presentado, pudiendo aplicarse un estilo global a los elementos de un documento de forma sencilla y rápida una vez se haya terminado de redactar toda la información referente a la materia tratada.

En el caso de este trabajo, para el alumno, que no había tenido contacto previo con esta tecnología, ha sido de gran ayuda para la gestión de los índices, las bibliografías, las figuras y las tablas que se han requerido.

OverLeaf

Como se ha mencionado en la sección anterior, LaTeX es una tecnología que conlleva cierta complejidad y la aplicación web que se va a presentar en esta sección, OverLeaf, es un editor de textos online que simplifica la escritura

de documentos, con un funcionamiento similar al que tiene un entorno de desarrollo integrado.

Las funcionalidades principales que ofrece este sistema son:

- Gestión de los paquetes adicionales que se usan en el documento.
- Detección de errores y alertas antes de la compilación para su corrección temprana.
- Gestión sencilla de la estructura de los archivos destinados a mostrarse como figuras u otros recursos en el documento final.
- Permite añadir diferentes tipos de archivos como los “.bib”, que contienen la bases de datos para la bibliografía.
- Autocompletado de las etiquetas en diferentes idiomas según la preferencia del usuario.
- Compilación sencilla del documento (evitando tener que realizar este proceso mediante la línea de comandos).
- Visualización del resultado y de los posibles errores que se hayan encontrado durante la compilación mediante un registro de errores detallado que indica el lugar del error en el documento.
- Numerosas plantillas que facilitan la creación de documentos de diferentes tipos.
- Integración con sistemas de control de versiones basados en *Git* como GitHub o *GitLab*.

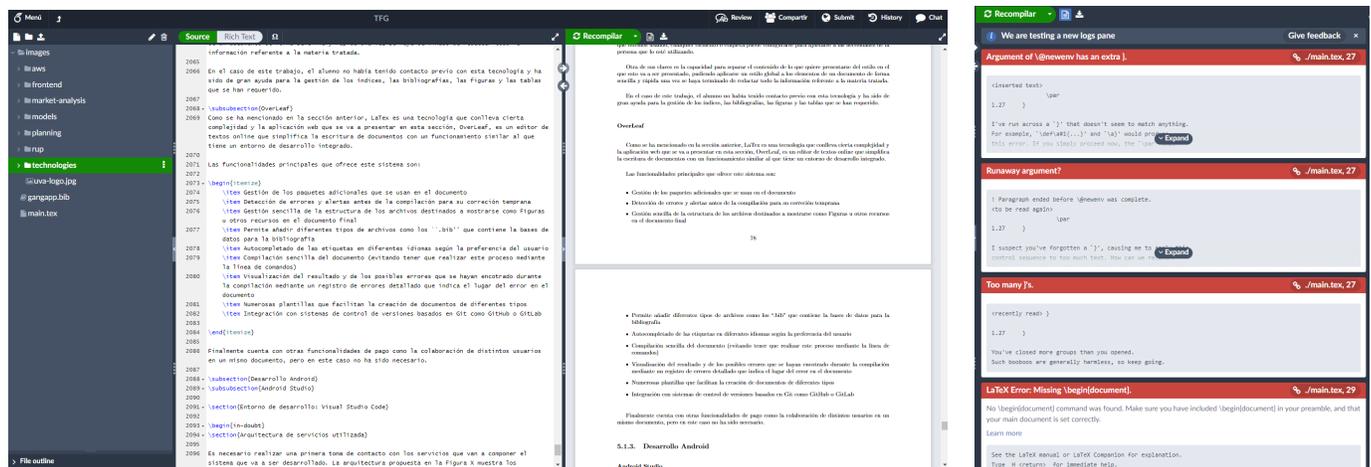


Figura 5.2: Interfaz de usuario de la aplicación Overleaf

Finalmente cuenta con otras funcionalidades de pago como la colaboración de distintos usuarios en un mismo documento, aunque en este caso no haya sido necesario utilizarlas.

5.1.3. Desarrollo de aplicación móvil

A continuación se presentarán el *software* que se ha utilizado en el ajuste de las aplicaciones para cada uno de los dispositivos. En nuestro caso la aplicación solo se ha probado para dispositivos Android así que exclusivamente utilizaremos tecnologías dentro de este ecosistema.

Android Studio

Es el entorno de desarrollo integrado oficial para Android. Esta herramienta creada por Google contiene todo lo necesario para el desarrollo, pruebas y despliegue de las aplicaciones en este sistema operativo para dispositivos móviles.

La funcionalidad más destacable o que más importancia ha cobrado durante el desarrollo, es la posibilidad de lanzar las aplicaciones en desarrollo en emuladores de dispositivos con diferentes características tanto al nivel de las prestaciones como al de las dimensiones, pudiendo probar tanto el rendimiento como la flexibilidad de la interfaz de usuario para adaptarse a los diferentes modelos en el mercado.

Con estos emuladores tenemos control total sobre la ejecución de la aplicación y de los logs durante su uso que son muy útiles para labores de depuración del código.

La simulación de diferentes dispositivos dentro de un PC puede traer consigo tiempos de arranque o de lanzamiento de las aplicaciones muy elevados que retrasan el desarrollo. Por esta razón se ha utilizado una funcionalidad complementaria a estos que consiste en el uso del propio dispositivo móvil del desarrollador como entorno para las pruebas. Esta forma de desarrollar la aplicación es mucho más rápida ya que sólo necesitamos conectar el ordenador y el teléfono mediante un cable USB y podremos probar el producto. Todos los detalles sobre cómo configurar el dispositivo propio para el desarrollo así como otras informaciones de interés sobre la plataforma pueden encontrarse en su documentación oficial [18].

Ampliamente conocida, no se ha requerido un trabajo con ella en profundidad, razón por la que no se va a entrar más detalles sobre su uso.

5.1.4. Ionic Framework

Como complemento a Angular, *framework* de desarrollo web previamente explicado en 5.2, se va a utilizar Ionic. Constituye una categoría por sí mismo ya que ha sido utilizado tanto para UX/UI como para el desarrollo multiplataforma

La propia compañía describe la tecnología como “un kit de código abierto para el desarrollo de aplicaciones nativas y web, multiplataforma y de alta calidad”. Obviando el marketing de la definición queda claro que este *framework* basado en el lenguaje JavaScript está destinado a crear aplicaciones que pueden ser ejecutadas en el navegador y en dispositivos móviles de forma sencilla.

Ionic ofrece una serie de paquetes que absorben toda la complejidad de tener que construir y desplegar una aplicación en Android, IOS, Web o escritorio, entre otras. Las herramientas que ofrece están pensadas para desarrollar siguiendo una serie de principios y utilizando unos componentes predefinidos, que están optimizados para crear aplicaciones escalables y con tiempos de respuesta bajos.

El flujo principal de desarrollo se basa en:

1. Crear el proyecto utilizando los paquetes Ionic.
2. Desarrollar la aplicación como si se estuviese creando para entorno web.
3. Compilar la aplicación generando los ejecutables para cada una de las posibles plataformas que se deseen.
4. Reajustar la aplicación utilizando el *software* específico de cada plataforma.
5. Recompilar con este *software* específico y desplegar en el dispositivo.

Para los dos últimos pasos el *framework* ofrece una cierta integración con Android Studio (tecnología descrita en la sección 5.1.3) que ha sido muy útil durante el desarrollo de la aplicación.

Además de las ventajas previamente mencionadas, nos proporciona una capa de trabajo que se ejecuta por encima de los *framework* de desarrollo web clásicos como *React* o *Angular* y añade una serie de componentes reutilizables para implementar las diferentes funcionalidades de nuestra aplicación. Toda la aplicación se ha desarrollado utilizando estos componentes personalizables. En la Figura 5.3 podemos observar ejemplos de algunos de sus componentes recogidos en su página de documentación [19]

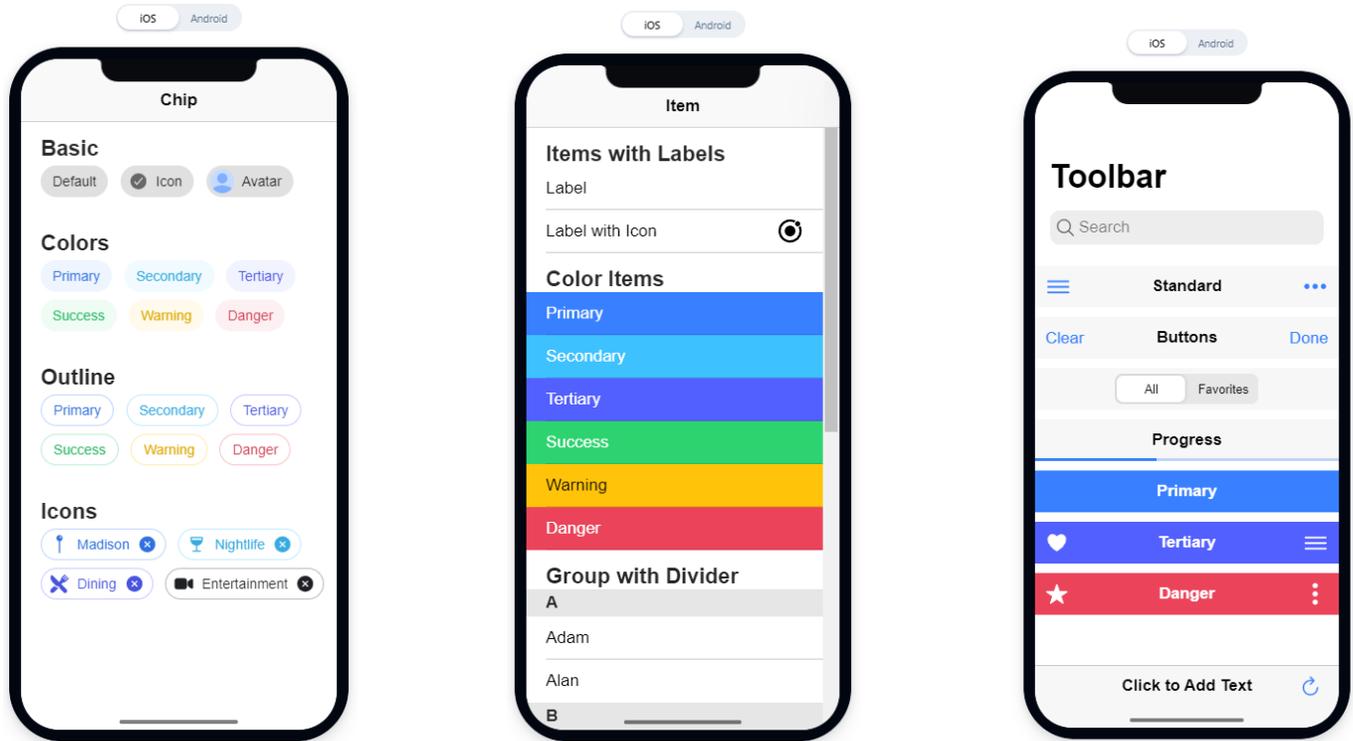


Figura 5.3: Ejemplos de componentes Ionic

5.1.5. Servicios AWS

Los servicios de persistencia de la aplicación se han implementado mediante los servicios de computación en la nube de la empresa Amazon conocidos como “Amazon Web Services”. En los siguientes apartados se van a presentar los servicios utilizados, el motivo de su uso y las configuraciones aplicadas sobre los mismos para construir el sistema.

Amazon Cognito

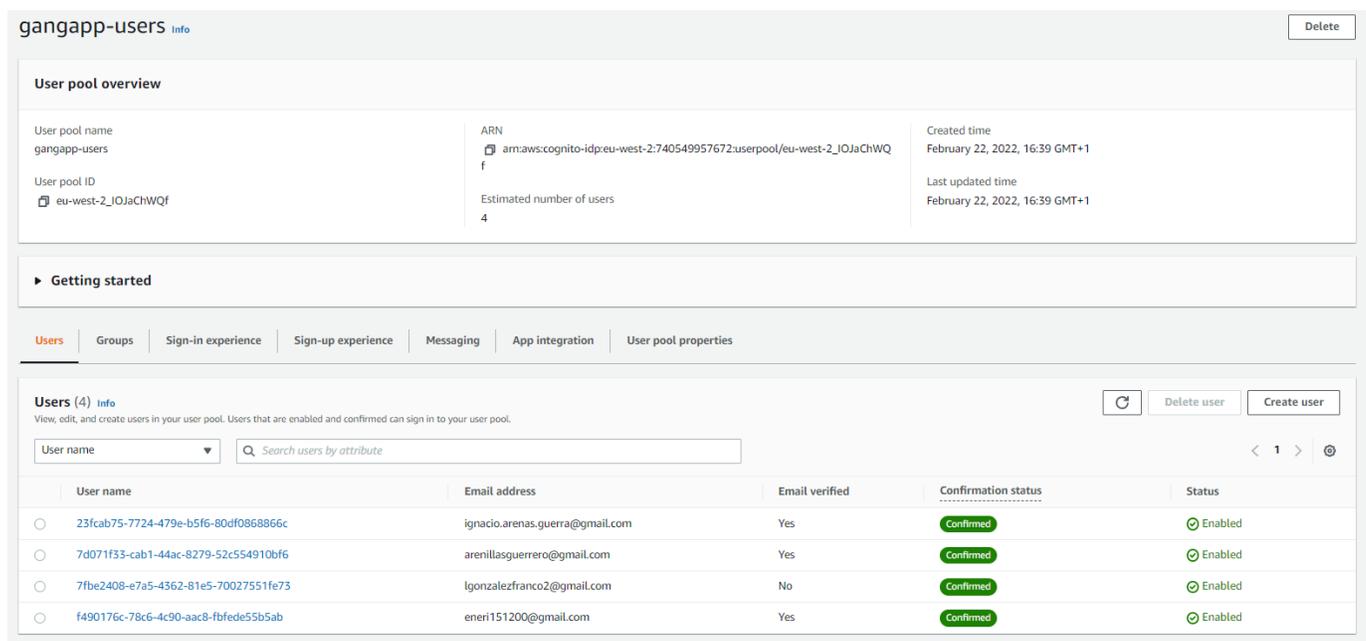
La solución proporcionada por AWS para el control de los usuarios de las arquitecturas basadas en sus servicios es Amazon Cognito [28]. Este servicio está destinado a proveer de un sistema de inicio de sesión estandarizado, rápido y seguro para las aplicaciones. Solo se requiere la personalización del servicio y algunos ajustes para conectarla a la aplicación desarrollada mediante la librería de JavaScript “Amplify” desarrollada por Amazon.

Para la gestión del servicio por los usuarios nos ofrece la creación de una “User Pool”. Las configuraciones aplicadas son las siguientes:

- El atributo utilizado para el inicio de sesión es el correo electrónico. Esta configuración aplica automáticamente la validación de los formatos de correo electrónico aceptados de forma que ya no es necesario validarlos en el cliente.

- Las condiciones de seguridad de las contraseñas son contener un número, un carácter especial, una mayúscula y una minúscula. También serán validadas en la llamada al servicio.
- No se utiliza un factor de autenticación múltiple ya que la información que almacena el sistema no es de carácter personal.
- La recuperación de cuenta se lleva a cabo exclusivamente utilizando el correo electrónico.
- La verificación de la cuenta se realiza mediante un correo electrónico con un código.
- Los atributos requeridos para efectuar un registro son el correo electrónico y el nombre del usuario.
- El envío de los mensajes de verificación se realiza mediante el propio servicio de Cognito, que nos permite enviar hasta 50 correos de verificación diarios siendo lo cuál más que suficiente para el volumen de usuarios estimado durante el desarrollo. Si quisieramos desplegar el sistema en producción podríamos usar servicios alternativos como Amazon SES.

Una vez aplicadas las configuraciones el servicio despliega el elemento que se acaba de crear. En este momento podremos gestionar los usuarios desde la interfaz de la consola de AWS. En la Figura 5.4 podemos apreciar las diferentes opciones que podemos realizar sobre los usuarios registrados como consultar el estado en el proceso de registro en el que se encuentran o bloquear usuarios maliciosos si fuera necesario.



The screenshot shows the AWS IAM console interface for the 'gangapp-users' user pool. It includes a 'User pool overview' section with details like ARN, Created time, and Estimated number of users. Below that is a 'Getting started' section with navigation tabs for 'Users', 'Groups', 'Sign-in experience', 'Sign-up experience', 'Messaging', 'App integration', and 'User pool properties'. The 'Users' tab is active, showing a list of 4 users with columns for User name, Email address, Email verified, Confirmation status, and Status.

User name	Email address	Email verified	Confirmation status	Status
23fcab75-7724-479e-b5f6-80df0868866c	ignacio.arenas.guerra@gmail.com	Yes	Confirmed	Enabled
7d071f33-cab1-44ac-8279-52c554910bf6	arenillasguerrero@gmail.com	Yes	Confirmed	Enabled
7fbc2408-e7a5-4362-81e5-70027551fe73	lgonzalezfranco2@gmail.com	No	Confirmed	Enabled
f490176c-78c6-4c90-aac8-fbfede55b5ab	eneri151200@gmail.com	Yes	Confirmed	Enabled

Figura 5.4: Capturas de la consola de administración AWS para el servicio Amazon Cognito

Amazon API Gateway

Mediante este servicio podemos crear puntos de acceso de diferentes tipos como API RESTful o API WebSocket. Además de la creación, API Gateway [27] permite gestionar todos los aspectos relacionados con el acceso, publicación, mantenimiento y monitorización. También se encarga de tareas de alta complejidad como el balanceo de carga o la compatibilidad CORS, ofreciendo una interfaz sencilla manejable desde una consola central como el resto de servicios AWS.

Sin entrar en detalles de la navegación de los menús, el flujo de trabajo con el que se ha creado la API que puede verse en la Figura 5.5 es el siguiente:

1. Crearemos el recurso especificando la ruta del mismo. Podemos utilizar parámetros variables como “store” o “ean” que podrán ser utilizados en el procesamiento de las peticiones. Por defecto se nos generará un método “OPTIONS” que según el [24] nos permitirá realizar una petición sobre información para la comunicación con el recurso.
2. Crearemos los métodos HTTP que necesitemos, tendremos que especificar la función que procesará la información obtenida del mensaje en este punto. Las funciones Lambda encargadas de este tratamiento de la información se explican en la sección 5.1.5.
3. Para evitar problemas con la política CORS (“Cross-Origin Resource Sharing”) deberemos especificar las cabeceras de las respuestas permitidas junto con los métodos y los orígenes.
4. Crearemos una nueva etapa de despliegue si no existiese ya alguna. Las etapas de despliegue nos permitirán trabajar en diferentes funcionalidades de forma simultánea. Podremos desplegar en una las partes más estables de la API mientras que podremos crear otra de pruebas para añadir nuevas funcionalidades
5. Por último, desplegaremos en la etapa elegida los cambios realizados. Tras unos segundos, los cambios serán efectivos y podremos realizar peticiones a los nuevos recursos

El diseño de la API utilizada se ha explicado en la sección 4.1.3

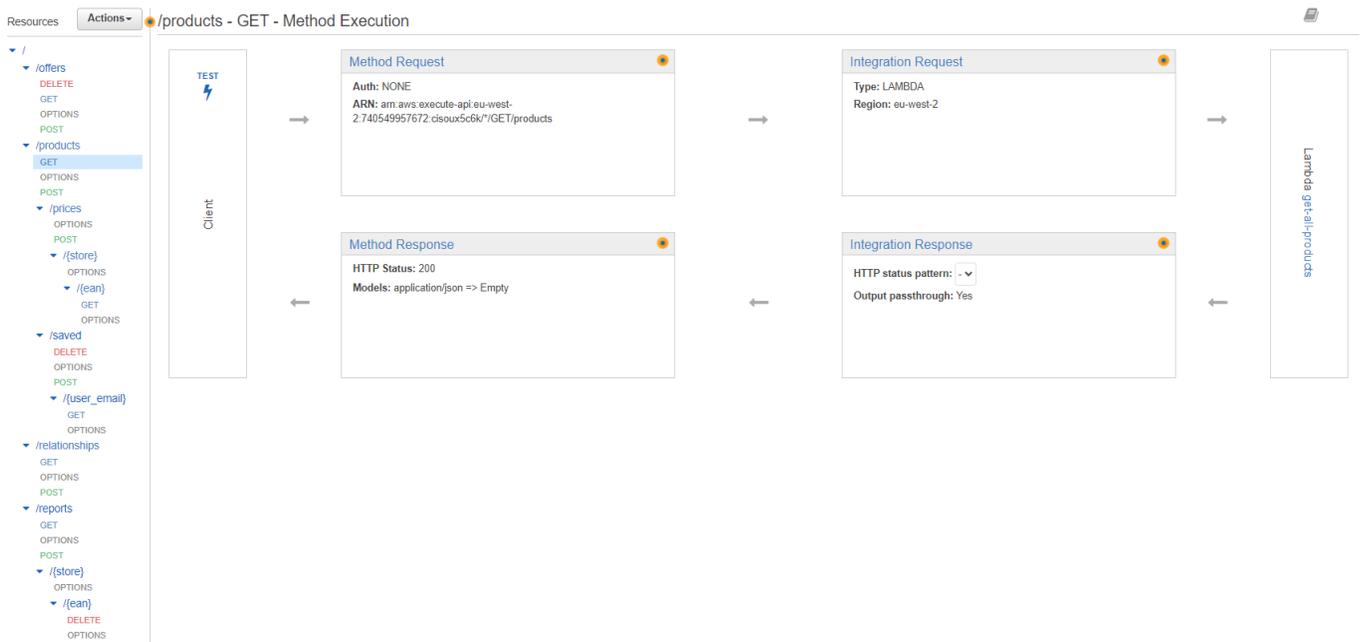


Figura 5.5: Capturas de la consola de administración AWS para el servicio Amazon API Gateway

AWS Lambda

Mediante este servicio se nos ofrece la posibilidad de ejecutar código sin entrar en detalles de la infraestructura subyacente que lo ejecuta. Las funciones “Lambda” [31] nos permiten establecer eventos de entrada que lanzarán ciertos fragmentos de código y producir eventos de salida en otros servicios. Con esto podremos crear un circuito de procesamiento de la información escalable. Al igual que el resto de servicios, AWS establece una fachada sencilla sobre elementos de complejidad y nos permite manejar las funciones desde su consola.

En nuestro caso utilizamos este servicio para recoger la información de las peticiones realizadas a la API y trasladarla a los servicios de persistencia DynamoDB y Amazon S3 que se explicarán, respectivamente, en las secciones 5.1.5 y 5.1.5.

En la Figura 5.6 podemos ver un ejemplo de la función que obtiene y procesa la creación de nuevos productos. Se puede apreciar como el propio editor de la web nos permite ejecutar el código, realizar pruebas y depurar.

```

Code source Info
File Edit Find View Go Tools Window Test Deploy
Go to Anything (Ctrl-P) index.js Execution results x
Environment
  create-new-product
  index.js
1 | var AWS = require("aws-sdk");
2
3 | var dynamoDb = new AWS.DynamoDB({
4 |   region: "eu-west-2",
5 |   apiVersion: "2012-08-10",
6 | });
7
8 | exports.handler = (event, context, callback) => {
9 |   var params = {
10 |     TableName: "products",
11 |     Item: {
12 |       ean: { S: "" + event.ean },
13 |       brand: { S: "" + event.brand },
14 |       name: { S: "" + event.name },
15 |       category: { S: "" + event.category },
16 |       store: { S: "" + event.store },
17 |       storeName: { S: "" + event.storeName },
18 |       storeChain: { S: "" + event.storeChain },
19 |       price: { N: event.price.toString() },
20 |       priceType: { S: "" + event.priceType },
21 |       photos: { SS: event.photos },
22 |       uploader: { S: "" + event.uploader }
23 |     },
24 |   };
25
26 |   dynamoDb.putItem(params, function (err, data) {
27 |     if (err) {
28 |       console.log("Error while saving product", err);
29 |       callback(err);
30 |     } else {
31 |       console.log("Product saved successfully", data);
32 |       callback(null, data);
33 |     }
34 |   });
35 | };

```

Figura 5.6: Capturas de la consola de administración AWS para el servicio AWS Lambda

Amazon DynamoDB

Se trata de una base de datos NoSQL de tipo clave-valor que nos permite el almacenamiento de documentos en formato JSON variable. Este servicio nos otorga la flexibilidad necesaria para poder crear una aplicación en un desarrollo iterativo ya que tolera perfectamente cambios en los esquemas de los datos. Además, DynamoDB [29] está configurado para ser un servicio de alto rendimiento y con conectividad al resto de servicios de AWS.

La mayor parte de la persistencia de este proyecto se ha implementado utilizando DynamoDB. La creación de tablas se realiza mediante el menú de la consola y sólo es necesario especificar las siguientes configuraciones:

- Nombre de la tabla que se quiere crear.
- Clave de partición y clave de ordenación (siendo esta opcional). Este par de claves identifican los documentos almacenados en la tabla.
- Establecemos las capacidades de lectura y escritura. Como el volumen de datos manejado no es muy alto y en este trabajo se busca exprimir al máximo la capa gratuita de servicios se utilizan los valores por defecto.
- Podemos establecer índices secundarios para facilitar las búsquedas de documentos, esto no es necesario en nuestro caso.
- Clase de la tabla. Existen tablas optimizadas mediante inteligencia artificial útiles si el caso de uso implementado exige pocos accesos a la persistencia y un gran volumen de datos. Como no es este nuestro caso, se opta por el tipo clásico.

Con este sencillo proceso ya tendremos una tabla lista para gestionar documentos. Para poder manejar los datos almacenados podremos cargarlos de manera manual o utilizando el kit de desarrollo incluido con las funciones “Lambda” que puede visualizarse en el código de la Figura 5.6.

Amazon S3

El último servicio a destacar son los “Buckets” [30]. Son unidades de almacenamiento de información de gran tamaño de forma segura y escalable. El principal uso que se le ha dado a este servicio en el proyecto, es el de almacenar los contenidos multimedia (en este caso fotografías) asociados a los productos.

No exige ninguna configuración aparte de la creación de una unidad de almacenamiento y la especificación del acceso en el caso de que se quieran restringir a ciertos usuarios acreditados.

5.2. Implementación del patrón MVVM. Angular Framework

El desarrollo de aplicaciones web en la actualidad se fundamenta principalmente en una serie de herramientas que abstraen todos los aspectos no relacionados directamente con el propio código como puedan ser el rendimiento o el despliegue. Estas herramientas se conocen comunmente como *frameworks*. Para la construcción del sistema expuesto en este estudio se ha utilizado una de estas herramientas conocida como Angular. A continuación se va a explicar la implementación que esta tecnología hace del patrón MVVM descrito anteriormente en la sección 4.1.1.

Los tres ejes principales de angular son los *Components* (componentes), las *Templates* (plantillas) y *Dependency Injection* (inyección de dependencias). El patrón de inversión de dependencias ya ha sido explicado en la sección 4.2.2, explicaremos ahora como implementar el MVVM mediante componentes y plantillas.

Templates

Las plantillas son la implementación de las vistas. Utilizan el estandarizado lenguaje de marcado HTML para construir las interfaces del sistema reforzado con ciertas sentencias especiales propias de Angular destinadas a facilitar el desarrollo. A este documento se le incluyen las sentencias que proporciona el “Data Binder” del *framework* para conectar cada vista con el modelo de datos de la aplicación.

Componentes (“Components”)

Construcción básica sobre la que se sustenta esta herramienta. Engloban una plantilla de las descritas en el párrafo anterior junto con el estilo y un pequeño controlador de vista que define la funcionalidad principal. A esto hay que añadir un selector que nos permitirá identificar este elemento en el sistema.

El ciclo de vida de un componente se basa en una serie de eventos que provocarán las actualizaciones de las plantillas. El *framework* utiliza estos eventos para notificar a los controladores de las vistas de los cambios que hayan podido producirse, ya sea por acciones directas o indirectas del usuario (entendiendo como ejemplo de acción indirecta la destrucción de un componente debido a la navegación del usuario en la web o la creación del siguiente componente cuya vista va a ser mostrada).

En la Figura 5.7 podemos encontrar las etapas del ciclo de vida de un componente Angular y el orden en el que son ejecutadas. Pero, aunque configuran una parte fundamental del *framework*, no es objetivo del presente trabajo describir en detalle en que consisten cada una de ellas.



Figura 5.7: “Hooks” del ciclo de vida de un componente Angular

La principal ventaja de estos componentes es la reutilización. Podemos crear varias instancias de un mismo componente en nuestra aplicación reduciendo la cantidad de código duplicado utilizando los selectores mencionados previamente.

Construcción del MVVM mediante componentes Angular

En la Figura 5.8 se presenta un diagrama de la implementación del patrón arquitectónico MVVM utilizando componentes, junto con la capa que contiene las entidades del modelo. Podemos identificar como las plantillas y sus estilos se implementan en la Vista, las clases del negocio constituirían el Modelo, y el Modelo de Vistas se implementa mediante los componentes formados por las plantillas, el controlador de vista asociado y la lógica de flujo de datos.

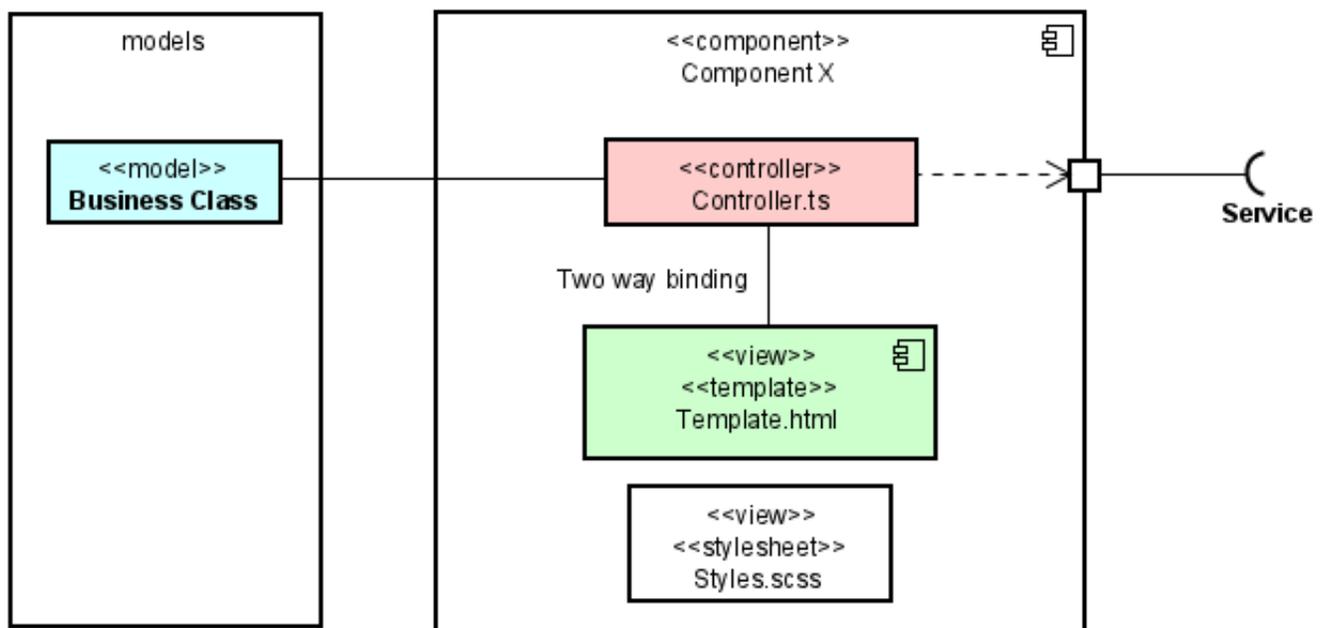


Figura 5.8: Patrón MVVM aplicado a un componente Angular

5.3. Tecnologías de pruebas

El paso siguiente es analizar que tecnologías se han empleado para realizar las pruebas. Aparte de los paquetes de pruebas integrados con Angular han sido necesarias otras herramientas para realizar las pruebas sobre las API en las

que se basa el sistema. Para estas pruebas se ha utilizado el programa *Postman*.

5.3.1. *Postman*

Esta tecnología, que comenzó como una extensión del navegador Chrome, y que ahora cuenta ya con su propia aplicación de escritorio, nos acompañará en el desarrollo de las diferentes APIs ofreciendo un entorno personalizable de trabajo en el que podemos realizar todo tipo de tareas sobre las peticiones. Nos permitirá tanto automatizarlas y encadenarlas como analizar las peticiones y respuestas recibidas.

Para este desarrollo, el principal uso que se le ha conferido es realizar las diferentes pruebas sobre AWS API Gateway y comprobar que los datos se almacenan y se recuperan de forma correcta. *Postman* permite organizar de forma sencilla las peticiones en colecciones así como guardar cierta información en variables para no tener que repetirla de unas a otras. En la Figura 5.9 podemos observar como se estructuran algunas de las peticiones con las que se ha probado la aplicación.

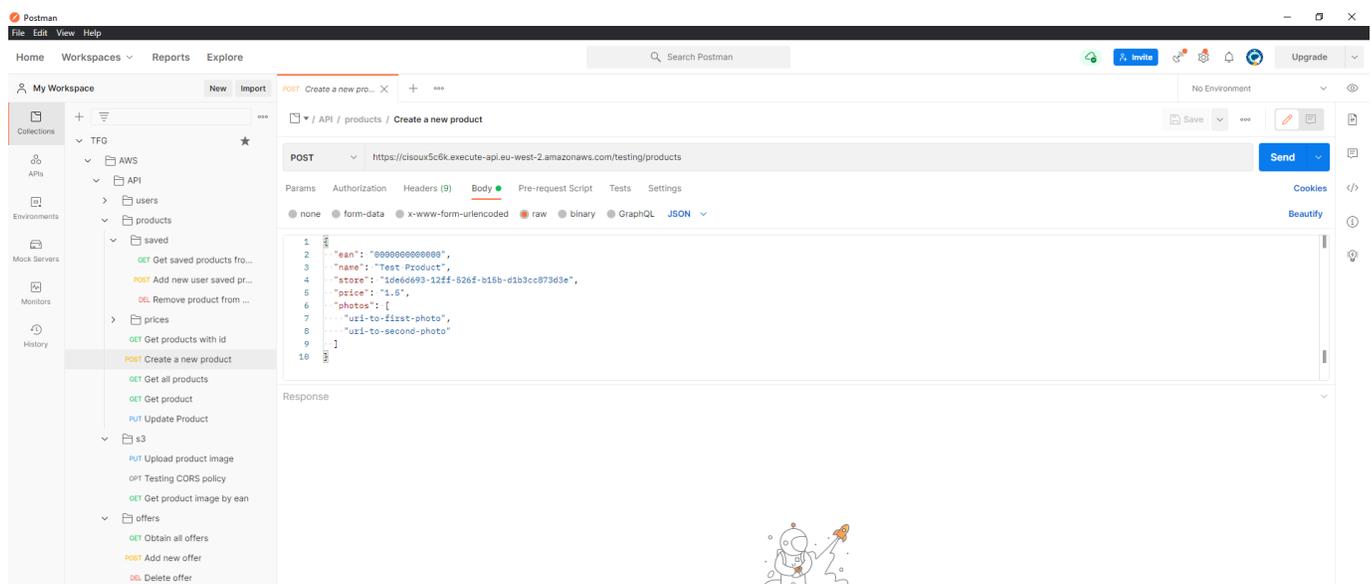


Figura 5.9: Interfaz de usuario de la aplicación *Postman*

5.4. Entorno de desarrollo. Visual Studio Code

Se ha utilizado el editor de código Visual Studio Code para el desarrollo de la aplicación angular. Es un entorno gratuito proporcionado por Microsoft que cuenta con múltiples extensiones que permiten integrarlo con algunos de los servicios utilizados como por ejemplo GitHub, descrito en la siguiente sección.

5.5. Control de versiones. Git Windows y GitHub

Git es un *software* de control de versiones (VCS) creado por Linus Torvalds. Los VCS son herramientas utilizadas para el mantenimiento de los archivos que componen la aplicación y sistemas como GitHub permiten utilizar esta herramienta de manera online y colaborativa mediante repositorios. La estructura del repositorio en el sistema de control de versiones será similar a la puede encontrarse en el Anexo I. Estructura del repositorio

Capítulo 6: Pruebas

6.1. Ejecución de las pruebas

Existen numerosos tipos de pruebas posibles para comprobar que el *software* desarrollado cumple las especificaciones y objetivos deseados. Según la forma de ejecución podemos encontrar dos tipos de pruebas:

- Pruebas automatizadas: programadas utilizando *software* específico para ello. Permiten ejecutar gran cantidad de pruebas de manera rápida pero la creación de estos conjuntos de pruebas conlleva una gran complejidad y dominio de las herramientas necesarias.
- Pruebas manuales: llevadas a cabo por los desarrolladores ejecutando diferentes partes del sistema y comprobando la respuesta obtenida. Se desarrollan de forma más lenta que las automatizadas, pero no requieren de programas complejos para su estructuración.

Las pruebas se dividen a su vez en funcionales y no funcionales, pudiendo apreciarse la relación de esta división con la división de los requisitos que se plantea durante la fase de análisis del desarrollo. Clasificaremos como pruebas funcionales aquellas destinadas a evaluar que se ha conseguido desarrollar la funcionalidad requerida, y como no funcionales las pruebas destinadas a medir que la aplicación cumple con requisitos de seguridad o escalabilidad, entre otros. En nuestro caso vamos a centrarnos en las pruebas funcionales.

Existen muchos tipos de pruebas funcionales en función de los objetivos que se desee conseguir o de las partes de la aplicación que se desee probar. Algunos de los grupos más conocidos y utilizados son:

- Pruebas unitarias: mediante las que se busca evaluar cada una de las piezas o componentes del software de forma aislada para detectar fallos en su funcionamiento que puedan suponer problemas de mayor escala al integrarse con el resto de elementos del sistema.
- Pruebas de integración: nos permiten evaluar partes del sistema de forma conjunta observando los posibles fallos que puedan darse al trabajar de forma conjunta, en las comunicaciones o el traspaso de datos.
- Pruebas de aceptación: el sistema se prueba en escenarios similares a la realidad del entorno en el que el *software* será desplegado. Con estas pruebas el cliente determinará su satisfacción con los resultados que ofrece el sistema.

En el presente proyecto existen numerosas partes que podrían ser probadas poniendo en práctica las diferentes técnicas descritas anteriormente. Contamos con librerías de test unitarios para Angular como *Jasmine* o *Karma* y con herramientas para elaborar peticiones web a las API de la aplicación como *Postman*, descrita en la sección 5.3.1. En algunos de los servicios utilizados como Amazon Lambda o Amazon API Gateway también se ofrecen pequeños entornos de prueba útiles para detectar fallos durante el desarrollo.

Como no es objetivo del proyecto dominar ningún *framework* de pruebas unitarias para Angular, y dadas las restricciones de tiempo con las que se desarrolla el proyecto, se ha decidido probar la aplicación de forma integrada utilizando pruebas manuales usando la aplicación final. Las pruebas realizadas se consideran de “caja negra” puesto que el código no se visualizará durante su ejecución.

6.2. Batería de pruebas

A continuación se presentan las pruebas, dirigidas por los casos de uso, que se van a realizar. Se describe el procedimiento que se llevará a cabo y el resultado que se considerará satisfactorio.

CP1	Registro en el sistema
Entrada	Con la sesión cerrada y en la pantalla principal (Inicio de sesión) cambiamos a la pantalla de Registro. Añadimos el nombre “Test”, el correo del alumno y “123,,Abc” como contraseña y se envía el formulario.
Resultado Esperado	En el correo utilizado se ha recibido el código numérico de verificación y la aplicación muestra la ventana de validación del código.

Tabla 6.1: Caso de prueba 1

CP2	Verificación de la cuenta registrada en el sistema
Entrada	Se debe realizar a continuación del proceso de registro. Se consulta el código recibido en el correo y se introduce en el formulario. Se envía el formulario.
Resultado Esperado	Se ha obtenido acceso a la funcionalidad de la aplicación y se muestran las imágenes de guía de uso del sistema.

Tabla 6.2: Caso de prueba 2

CP3	Registro sin aceptar los términos y condiciones del sistema
Entrada	Con la sesión cerrada y en la pantalla principal (Gestión de sesión) cambiamos a la opción de registro. Añadimos el nombre “Test”, el correo del alumno y “123,,Abc” como contraseña, nos saltamos la opción de aceptar los términos de uso y se envía el formulario
Resultado Esperado	El sistema no lleva a cabo el registro y se informa al usuario de la situación

Tabla 6.3: Caso de prueba 3

CP4	Inicio de sesión en el sistema de manera correcta
Entrada	En la pantalla principal y seleccionando la opción de inicio de sesión se introduce un correo electrónico ya registrado y verificado mediante código numérico en el sistema. También se introduce la contraseña usada en el registro y se envía el formulario.
Resultado Esperado	El sistema permite acceder a la funcionalidad total de la aplicación al usuario.

Tabla 6.4: Caso de prueba 4

CP5	Inicio de sesión en el sistema con credenciales incorrectas
Entrada	En la pantalla principal y seleccionando la opción de inicio de sesión se introduce un correo electrónico ya registrado y verificado mediante código numérico en el sistema. Se introduce una contraseña incorrecta y se envía el formulario
Resultado Esperado	El sistema impide el acceso e informa del problema mediante un mensaje.

Tabla 6.5: Caso de prueba 5

CP6	Publicación de un producto nuevo en el sistema.
Entrada	Se inicia el proceso de publicación de un producto. Se escanea su código de barras, se introduce su nombre, marca y precio. En el mapa, se busca y selecciona la tienda a la que pertenece y se selecciona la cadena a la que pertenece la tienda. Se selecciona la categoría. Se añaden dos fotos. Sin seleccionar ningún producto para relacionarlo se envía el formulario.
Resultado Esperado	En la pantalla del catálogo y la de productos creados se puede encontrar el producto. Este producto no tiene la opción de menú de guardar ya que está sustituida por un mensaje que informa de que el propio usuario es el autor de la publicación

Tabla 6.6: Caso de prueba 6

CP7	Cerrar sesión
Entrada	En la pantalla de perfil, se abre el menú superior de opciones y se hace click en el icono con la salida.
Resultado Esperado	El sistema cierra la sesión del usuario y este ya no tiene acceso a la funcionalidad.

Tabla 6.7: Caso de prueba 7

CP8	Buscar tiendas en las que añadir un producto aparte de las mostradas inicialmente
Entrada	Durante el proceso de publicación de un producto en el apartado de selección de tienda en el mapa se busca por el mapa un punto alejado del original y se toca la pantalla.
Resultado Esperado	Tras un periodo de carga se muestra un área de búsqueda nueva y las tiendas que se hayan encontrado en el interior mediante iconos distinguibles.

Tabla 6.8: Caso de prueba 8

CP9	Publicación de un producto con el mismo código de barras y distinta tienda
Entrada	Utilizando un producto se inicia el proceso de publicación. En el primer paso, se escanea el código de barras que ya debe existir en el sistema. Posteriormente se seleccionará una tienda diferente de las mostradas.
Resultado Esperado	Se muestra el producto del sistema con el mismo código de barras y su tienda. Se permite continuar con el proceso de publicación sin permitir repetir esta combinación mostrada.

Tabla 6.9: Caso de prueba 9

CP10	Publicación de un producto ya existente en el sistema
Entrada	Utilizando un producto se inicia el proceso de publicación. En el primer paso, se escanea el código de barras que ya debe existir en el sistema. Posteriormente se seleccionará una tienda que haya sido mostrada como que ya cuenta con ese producto en el sistema.
Resultado Esperado	Se muestra el producto del sistema con el mismo código de barras y su tienda. No se permite continuar con el proceso de publicación informando al usuario mediante un mensaje.

Tabla 6.10: Caso de prueba 10

CP11	Modificar el precio de un producto creado por el usuario
Entrada	Se escoge un producto creado por el usuario. Se despliega el menú tocándolo en la pantalla y se selecciona la opción de editar sus propiedades. En el formulario que aparece se debe seleccionar un precio diferente al ya existente y enviar.
Resultado Esperado	El producto se muestra en el catálogo con un precio diferente al anterior y si se consulta su histórico de precios mediante el menú se puede observar una nueva entrada.

Tabla 6.11: Caso de prueba 11

CP12	Guardar un producto en lista de la compra
Entrada	Se escoge un producto subido o no por el usuario. Se despliega el menú tocándolo en la pantalla y se selecciona la opción de guardar en la lista de la compra.
Resultado Esperado	En la pantalla de la lista de la compra se podrá ver el nuevo producto y el precio de la lista actualizado tanto para el supermercado del producto como en total. En el menú se habrá sustituido la opción de guardar por la de retirar de la lista

Tabla 6.12: Caso de prueba 12

CP13	Retirar un producto de la lista de la compra
Entrada	Se escoge un producto subido o no por el usuario. Se despliega el menú tocándolo en la pantalla y se selecciona la opción de retirar en la lista de la compra.
Resultado Esperado	En la pantalla de la lista de la compra ya no se podrá ver el producto y el precio de la lista actualizado tanto para el supermercado del producto como en total. En el menú se habrá sustituido la opción de retirar por la de guardar en la lista.

Tabla 6.13: Caso de prueba 13

CP14	Consultar la guía interna de la aplicación
Entrada	En la pantalla de perfil, se abre el menú superior de opciones y se hace click en el icono con la interrogación.
Resultado Esperado	El sistema mostrará las imágenes que muestran como se usa la aplicación.

Tabla 6.14: Caso de prueba 14

CP15	Guardar un producto en la lista de productos guardados
Entrada	Se escoge un producto no subido por el usuario. Se despliega el menú tocándolo en la pantalla y se selecciona la opción de guardar.
Resultado Esperado	En la pantalla de productos guardados se podrá ver el nuevo producto. En el menú se habrá sustituido la opción de guardar por la de retirar de productos guardados.

Tabla 6.15: Caso de prueba 15

CP16	Reportar un producto
Entrada	Se selecciona un producto no subido por el usuario. Se despliega el menú tocándolo en la pantalla y se selecciona la opción de reportar. En el formulario que aparece se selecciona el motivo del reporte y se introduce una descripción.
Resultado Esperado	El producto no se podrá encontrar en la pantalla del catálogo hasta que sea corregido por la persona que realizó la publicación

Tabla 6.16: Caso de prueba 16

CP17	Corregir reporte de un producto
Entrada	El usuario despliega el menú de un producto creado por él mismo que esté en estado reportado. En base al motivo del reporte escogido se corrige la información necesaria y se envía el formulario.
Resultado Esperado	El producto aparece en la pantalla del catálogo y si su precio ha sido modificado, figurará en el histórico de precios del producto.

Tabla 6.17: Caso de prueba 17

CP18	Cancelar proceso de publicar un producto en el momento de la confirmación
Entrada	Durante la publicación de un producto y tras introducir toda la información necesaria en el proceso de confirmación se pulsa el botón de cancelar.
Resultado Esperado	No existe ningún producto en el sistema creado con la información introducida y si se repite el proceso el formulario no habrá guardado ninguna de las informaciones introducidas en el proceso anterior.

Tabla 6.18: Caso de prueba 18

CP19	Buscar por un criterio que tenga productos registrados en el sistema
Entrada	En la pantalla de búsqueda de productos seleccionamos la categoría a la que se haya añadido un producto previamente de forma correcta.
Resultado Esperado	El sistema mostrará los productos de la categoría en una lista con toda la información de estos y permite desplegar sus menús.

Tabla 6.19: Caso de prueba 19

CP20	Buscar por una combinación de criterios que tenga productos registrados en el sistema
Entrada	En la pantalla de búsqueda de productos seleccionamos la categoría y la tienda a las que se hayan añadido un producto previamente de forma correcta.
Resultado Esperado	El sistema mostrará los productos que coinciden con ambos criterios en una lista con toda la información de estos y permite desplegar sus menús.

Tabla 6.20: Caso de prueba 20

CP21	Buscar un producto existente en el sistema escaneando su código de barras
Entrada	En la pantalla de búsqueda de productos seleccionamos la opción de escanear un código de barras y escaneamos un producto que previamente se haya registrado en el sistema.
Resultado Esperado	El sistema mostrará los productos que tienen ese código de barras en las diferentes tiendas mediante una lista con toda la información de estos y permite desplegar sus menús. No existirá ninguna tienda repetida en la lista.

Tabla 6.21: Caso de prueba 21

CP22	Buscar un producto inexistente en el sistema escaneando su código de barras
Entrada	En la pantalla de búsqueda de productos seleccionamos la opción de escanear un código de barras y escaneamos un producto que no se haya registrado en el sistema previamente.
Resultado Esperado	El sistema mostrará un mensaje informando de que la búsqueda no ha encontrado ningún resultado.

Tabla 6.22: Caso de prueba 22

CP23	Buscar por un criterio que no cuente con productos registrados en el sistema
Entrada	En la pantalla de búsqueda de productos seleccionamos la categoría a la que no se haya añadido un producto previamente.
Resultado Esperado	El sistema mostrará un mensaje informando de que la búsqueda no ha encontrado ningún resultado.

Tabla 6.23: Caso de prueba 23

CP24	Cambiar de tema entre modo oscuro y modo claro
Entrada	En la pantalla de perfil, se despliega el menú superior y se selecciona la opción con el icono del sol/luna.
Resultado Esperado	El tema de la aplicación ha cambiado de claro a oscuro o viceversa. Los colores de toda la aplicación han cambiado pero la funcionalidad se mantiene igual.

Tabla 6.24: Caso de prueba 24

CP25	Recordar dispositivos con sesión activa
Entrada	Ya identificados en la aplicación, la cerramos completamente (eliminándola de los procesos en segundo plano) y la volvemos a abrir.
Resultado Esperado	No es necesario realizar de nuevo el proceso de inicio de sesión y se cuenta con toda la funcionalidad.

Tabla 6.25: Caso de prueba 25

CP26	Retirar un producto de la lista de productos guardados
Entrada	Se escoge un producto no creado por el usuario. Se despliega el menú tocándolo en la pantalla y se selecciona la opción de guardar.
Resultado Esperado	En la pantalla de productos guardados se podrá ver el nuevo producto. En el menú se habrá sustituido la opción de retirar por la de guardar en productos guardados.

Tabla 6.26: Caso de prueba 26

6.3. Resultados de las pruebas

A continuación van a detallarse los resultados de las pruebas que han obtenido resultados diferentes a los esperados y se detallará si la funcionalidad ha sido corregida o modificada.

CP1	Registro en el sistema
Salida obtenida	La pantalla de validación del código no se muestra correctamente.
Resultado final	Corregido.
Detalles	Tras la corrección la ventana se despliega correctamente y permite realizar un registro normal.

Tabla 6.27: Resultados de la ejecución del Caso de prueba 1

CP3	Registro sin aceptar los términos y condiciones del sistema
Salida obtenida	El sistema no permite ejecutar el registro pero no se da información sobre el problema.
Resultado final	Modificado.
Detalles	Se ha añadido un elemento visual que indica que el formulario está deshabilitado hasta que el usuario rellene todos los campos.

Tabla 6.28: Resultados de la ejecución del Caso de prueba 3

CP6	Publicación de un producto nuevo en el sistema
Salida obtenida	La ubicación inicial que muestra el mapa no se corresponde con la geolocalización del usuario de forma precisa y dificulta encontrar la tienda buscada.
Resultado final	Corregido.
Detalles	Tras la corrección se ha aumentado la precisión y el mapa se ubica más cerca de la localización del usuario aunque no de forma completamente acertada.

Tabla 6.29: Resultados de la ejecución del Caso de prueba 6

CP9	Publicación de un producto con el mismo código de barras y distinta tienda
Salida obtenida	El sistema permite crear un producto con mismo código y cadena siempre que la tienda sea diferente.
Resultado final	Modificado.
Detalles	El comportamiento esperado es no poder incluir productos de la misma cadena por que no deberían variar en el precio por eso se han eliminado del selector aquellas cadenas que generarían estas repeticiones.

Tabla 6.30: Resultados de la ejecución del Caso de prueba 9

CP16	Reportar un producto
Salida obtenida	El producto reportado sigue apareciendo como disponible en el catálogo hasta que se recargue la página para el usuario.
Resultado final	Corregido.
Detalles	Para el usuario que genera el reporte este producto ya no aparece como disponible en el catálogo aunque sí para el resto de usuarios hasta que recarguen la aplicación. No se ha implementado ningún mecanismo para tener en cuenta estas situaciones de concurrencia.

Tabla 6.31: Resultados de la ejecución del Caso de prueba 16

CP18	Cancelar proceso de publicar un producto en el momento de la confirmación
Salida obtenida	El formulario mantiene la tienda seleccionada para futuros procesos de publicación.
Resultado final	Corregido.
Detalles	Toda la información introducida se elimina una vez cancelado el proceso.

Tabla 6.32: Resultados de la ejecución del Caso de prueba 18

CP24	Cambiar de tema entre modo oscuro y modo claro
Salida obtenida	Aunque el cambio de tema se efectúa correctamente en el modo oscuro se han detectado comportamientos no controlados al usar la aplicación
Resultado final	Corregido.
Detalles	Todos los procesos afectados que se han detectado han sido corregidos.

Tabla 6.33: Resultados de la ejecución del Caso de prueba 24

Capítulo 7: Conclusiones y trabajo futuro

7.1. Objetivos conseguidos

7.1.1. Objetivos relacionados con las tecnologías

- Adquirido un conocimiento básico del funcionamiento del *framework* Angular y del desarrollo web en general (maquetación, diseño UX/UI, procesamiento en el cliente, etc)
- Toma de contacto con la utilización de la API Overpass de Open Street Maps y su integración con la librería Leaflet para la gestión de los mapas de la aplicación
- Adquirido conocimiento sobre bases de datos NoSQL mediante la implementación de una base de datos utilizando la tecnología DynamoDB
- Gestión de una API REST desarrollada utilizando el servicio Amazon API Gateway
- Gestión de usuarios mediante la tecnología Amazon Cognito y mayor conocimiento en temas de protección de los datos de los mismos
- Orquestación de múltiples servicios de un proveedor de servicios en la nube (Amazon AWS) para la creación de un circuito de procesamiento de información
- Desarrollo de una aplicación móvil utilizando el *framework* Angular junto con el kit de desarrollo Ionic utilizando múltiples paquetes de externos para implementar funcionalidad compleja como el escáner y las fotografías
- Depuración de errores en sistemas móviles mediante los emuladores proporcionados por la herramienta Android Studio y utilizando un dispositivo real

7.1.2. Objetivos relacionados con el negocio

- Conocimiento de la etiquetación de los productos y de los diferentes códigos numéricos utilizados para su identificación en las diversas cadenas de productos de supermercados existentes en España.
- Desarrollo de un catálogo común actualizado por y para los usuarios mediante un sistema de autorregulación basado en reportes
- Representación de productos en diferentes estados basándose en las ofertas o los reportes existentes
- Representación de la evolución del precio de los productos atendiendo a diversos factores como correcciones de los propios autores, correcciones motivadas por reportes u ofertas
- Elaboración de una lista de la compra actualizable de manera sencilla
- Implementación de un motor de búsqueda simple en el conjunto de productos del catálogo basándose en las combinaciones de diferentes criterios.
- Creación de un proceso para añadir productos al catálogo personalizable con elementos multimedia

7.2. Valoración personal

El proyecto ha supuesto la oportunidad de investigar tecnologías nuevas que nos llamaban la atención por su popularización en los últimos años. Hemos tenido la ocasión de poner en práctica ese conocimiento, fruto del estudio previo con una idea, desde nuestra perspectiva, muy interesante, experimentando de primera mano la complejidad de realizar un proyecto partiendo de cero. Este desarrollo me ha permitido poner en práctica el conocimiento obtenido en muchas de las asignaturas cursadas durante el grado en un entorno todavía académico pero mucho más cerca de la experiencia real de la profesión.

El proyecto en sí ha planteado muchos retos que resolver tanto tecnológicos como personales. En lo tecnológico, terminé el proyecto con una base de tecnologías que me permitirán desarrollarme como profesional. En lo personal, la compaginación del proyecto con un puesto de trabajo me ha permitido adquirir habilidades blandas muy importantes como la gestión del tiempo y la organización. Considero el proyecto como un éxito y estoy satisfecho con el producto resultante.

7.3. Futuras líneas de desarrollo

Existen una serie de funcionalidades que se han valorado antes, durante y después del desarrollo como posibles continuaciones del trabajo realizado. A continuación se presentan estas ideas que podría ser interesante incluir si se desarrollasen nuevas versiones de la aplicación

- Implementar notificaciones a los usuarios informándoles de los eventos que sucedan en el sistema y que les afecten, como reportes sobre los propios productos u ofertas sobre los productos guardados.
- Añadir soporte para dispositivos móviles con sistema operativo IOS.
- Modificar las ofertas para añadir distintos tipos en función de las diferentes promociones que puedan existir en las tiendas como, por ejemplo, las segundas unidades a mitad de precio.
- Implementar la verificación por parte de terceros de un reporte publicado por un usuario antes de proceder a eliminar temporalmente el producto del catálogo.
- Permitir guardar una lista de la compra como imagen en el dispositivo para facilitar el acceso, favoreciendo así el poder compartirla.
- Desplegar la aplicación en el mercado de aplicaciones de los dispositivos Android.
- Crear un rol de comerciante que permita a ciertos usuarios publicar productos sobre los que tienen total control en cuestión de ofertas.
- Añadir los gastos en combustible que supone llegar a cada una de las tiendas en la lista de la compra y ofrecer productos similares a los de una tienda en la tienda más cercana con el fin de optimizar costes.

Bibliografía

- [1] Angelo Fernandes Cécile Péraire Mike Edwards. *The IBM Rational Unified Process for System z*. IBM, 2007. (Última consulta: 20/04/2022).
- [2] Corporativo. «Protección de datos: Guía para el ciudadano». En: *Agencia Española de Protección de Datos* (2020). (Última consulta: 01/05/2022).
- [3] Dino Butorac Danijel Matic Hrvoje Kegalj. «Data access architecture in object-oriented applications using design patterns». En: *Journal of information and organizational sciences* (2003). (Última consulta: 04/04/2022).
- [4] Javier Vallejo Gloria Arcos-Medina Jorge Menéndez. «Comparative Study of Performance and Productivity of MVC and MVVM design patterns». En: *Simposio Iberoamericano en Programación Informática* (2017). (Última consulta: 23/04/2022).
- [5] Bob Hughes. *Software Project Management*. McGraw Hill, 2009. (Último consulta: 15/03/2022).
- [6] Project Management Institute. *Software Project Management*. Project Management Institute, 2017. (Último consulta: 23/04/2022).
- [7] Janne Ropponen Kalle Lyytinen Lars Mathiassen. «A Framework for Software Risk Management». En: *Journal of Information Technology* (1996). (Última consulta: 28/02/2022).
- [8] Hoekyung Jung Kwangchul Shin Chulhyun Hwang. «NoSQL Database Design Using UML Conceptual Data Model Based on Peter Chen's Framework». En: *International Journal of Applied Engineering Research* (2017). (Última consulta: 01/05/2022).
- [9] Craig Larman. *Applying UML and patterns: An introduction to object-oriented analysis and design*. Upper Saddle River, N.J: Prentice Hall PTR, 1998. (Última consulta: 28/04/2022).
- [10] Michael Page. «Estudio de remuneración 2022». En: (2022). (Último consulta: 23/04/2022).
- [11] Douglas C. Schmidt. «Wrapper Facade. A Structural Pattern for Encapsulating Functions within Classes». En: *C++ Report magazine* (1999). (Última consulta: 19/04/2022).

Webgrafía

- [12] Grupo Ático 34. *Privacidad desde el diseño y por defecto ¿Qué es?* <https://protecciondatos-lopdp.com/empresas/privacy-by-design/>. 2022. (Último Acceso: 17/05/2022).
- [13] Grupo Adaptalia. *Categorías de datos personales RGPD*. <https://www.grupoadaptalia.es/blog/categorias-de-datos-personales-rgpd/>. 2021. (Último Acceso: 17/05/2022).
- [14] Baeldung. *The DAO Pattern in Java*. baeldung.com/java-dao-pattern. 2021. (Último Acceso: 01/05/2022).
- [15] Capacitor. *Cross-platform Native Runtime for Web Apps*. <https://capacitorjs.com/docs>. Capacitor, 2022. (Último Acceso: 15/04/2022).
- [16] Inc. Change Vision. *Astah: Premier Diagramming, Modeling Software & Tools*. <https://astah.net/>. 2022. (Último Acceso: 06/05/2022).
- [17] Desconocido. *¿Qué es RUP?* <https://rupingsw.wordpress.com/>. 2016. (Último Acceso: 19/04/2022).
- [18] Google Developers. *Documentación oficial de la plataforma Android Studio*. <https://developer.android.com/>. 2022. (Último Acceso: 21/05/2022).
- [19] Ionic. *Ionic Framework Docs*. <https://ionicframework.com/docs>. 2022. (Último Acceso: 23/05/2022).
- [20] *Meet Material Design 3*. <https://m3.material.io/>. Google, 2022. (Último Acceso: 22/05/2022).
- [21] Mercadona. *Aplicación oficial de la cadena de supermercados “Mercadona”*. <https://play.google.com/store/apps/details?id=es.mercadona.tienda&hl=es&gl=US>. 2022. (Último Acceso: 04/05/2022).
- [22] Pepper. *Aplicación social de publicación de ofertas “Chollómetro”*. <https://play.google.com/store/apps/details?id=com.chollometro&hl=es&gl=US>. 2022. (Último Acceso: 04/05/2022).
- [23] Pesades. *Rational Unified Process (RUP)*. <https://es.scribd.com/document/266908850/MetodoPesadesRUP>. 2016. (Último Acceso: 19/04/2022).
- [24] et al R. Fielding. *Hypertext transfer Protocol – HTTP/1.1*. <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>. 1999. (Último Acceso: 21/05/2022).
- [25] Soysuper.com. *Aplicación de comparación de precios de supermercados “SoySuper”*. <https://play.google.com/store/apps/details?id=com.soysuper.soysuper&hl=es&gl=US>. 2022. (Último Acceso: 04/05/2022).
- [26] Khalil Stemmler. *Implementing DTOs, Mappers & the Repository Pattern using the Sequelize ORM [with Examples] - DDD w/ TypeScript*. <https://khalilstemmler.com/articles/typescript-domain-driven-design/repository-dto-mapper/>. 2022. (Último Acceso: 01/05/2022).
- [27] *What is Amazon API Gateway?* <https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>. Amazon Web Services, 2022. (Último Acceso: 22/05/2022).
- [28] *What is Amazon Cognito?* <https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>. Amazon Web Services, 2022. (Último Acceso: 22/05/2022).
- [29] *What is Amazon DynamoDB?* <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>. Amazon Web Services, 2022. (Último Acceso: 22/05/2022).
- [30] *What is Amazon S3?* <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>. Amazon Web Services, 2022. (Último Acceso: 22/05/2022).
- [31] *What is AWS Lambda?* <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>. Amazon Web Services, 2022. (Último Acceso: 22/05/2022).

Anexo I: Estructura del repositorio

El repositorio accesible en <https://gitlab.inf.uva.es/ignaren/tfg-ignacio-arenas-guerra/-/tree/main/> se estructura de la siguiente forma:

- **Instalador de la aplicación:** archivo llamado *gangapp.apk*.
- **Software desarrollado:** todo el código fuente del sistema se encuentra en la carpeta *GangApp*.
- **Documentación:** los documentos UML generados se encuentran en la carpeta *Diagramas*. En el archivo *gangapp.astah* podremos encontrar los siguientes diagramas:
 - Diagrama de Casos de uso
 - Modelo de Dominio
 - Diagramas de realización de los Casos de uso en Análisis
 - Diagramas de Diseño de la Arquitectura general y por capas
 - Diagramas de clases en Diseño
 - Diagramas de despliegue
 - Diagrama de recursos de la API REST

Además se añade el archivo *gangapp.drawio* con el Modelo lógico de datos.

Anexo II: Manual de instalación

Opción Principal

Los pasos a seguir para la instalación de la aplicación del proyecto en un dispositivo móvil con sistema Android son los siguientes:

1. Descargar el archivo “.apk” del repositorio al teléfono en el que se quiera desplegar la aplicación (se encuentra en <https://gitlab.inf.uva.es/ignaren/tfg-ignacio-arenas-guerra/-/blob/main/gangapp.apk>).
2. Permitir la instalación de aplicaciones por parte de fuentes desconocidas. Este proceso puede ser diferente en función de la marca del dispositivo pero es necesario para poder instalar una aplicación fuera del mercado oficial. Se recomienda volver a restringir esta opción una vez instalada la aplicación.
3. Otorgar los permisos necesarios y lanzar la aplicación. Puede que estos pasos se inviertan en el orden en función del modelo de dispositivo.

Opción Alternativa

En el caso de no querer añadir la aplicación a un dispositivo móvil se puede seguir este proceso alternativo basado en el emulador de Android Studio aunque puede provocar que cierta funcionalidad como el escáner o la cámara no funcione correctamente.

1. Instalar Android Studio. Cualquier versión de las que tienen soporte actualmente permite instalar aplicaciones de la manera descrita en esta guía aunque se sugiere la utilización de la distribución “Artic Fox” versión 2020.3.1
2. Mediante la opción “AVD Manager” que se encuentra en el menú de ejecución, seleccionar un modelo de dispositivo a emular e instalar una distribución de Android en el caso de que no se tenga instalado. La Figura 7.1 muestra el menú en el que se realiza el paso explicado
3. Lanzar el emulador
4. Descargar el archivo “.apk” en el ordenador en el que se está realizando el proceso
5. Arrastrar la aplicación al navegador como se muestra en la Figura 7.2
6. Lanzar la aplicación, nos pedirá los permisos antes o después al igual que en el proceso descrito anteriormente

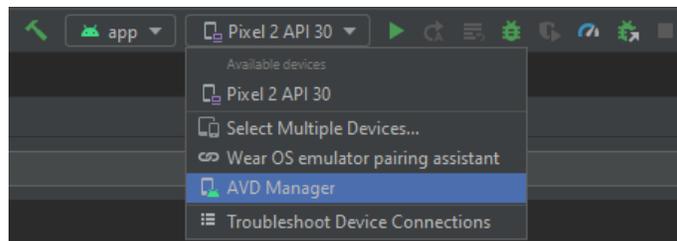


Figura 7.1: Menú de instalación de emuladores de Android Studio

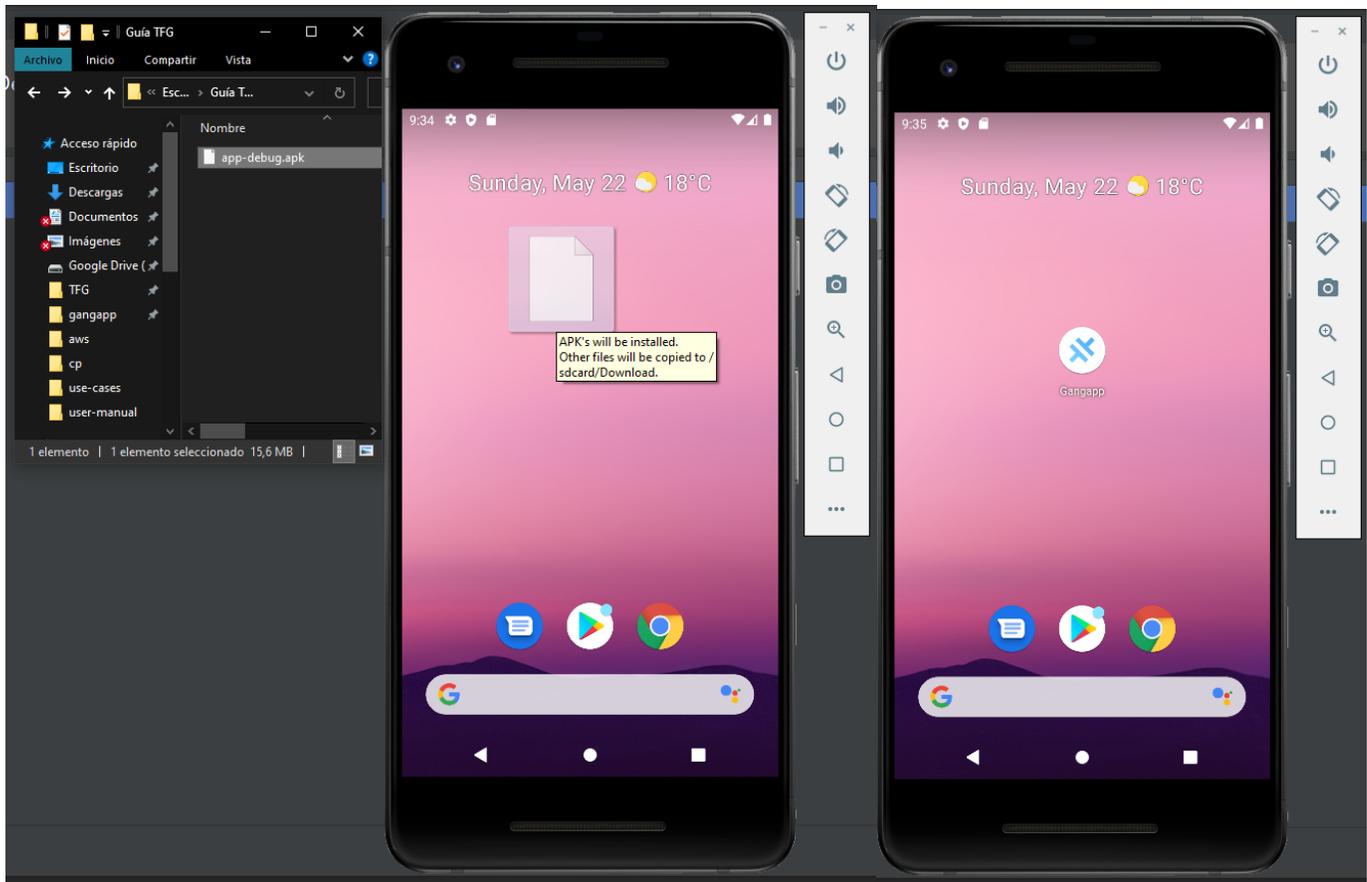


Figura 7.2: Instalación de la aplicación en el emulador

Anexo III: Manual de uso

Este anexo contiene las instrucciones para el uso del sistema. Se presentará toda la funcionalidad implementada mediante imágenes ilustrativas capturadas de la propia aplicación. Los datos con los que se ejemplifica son reales y han sido introducidos en la aplicación con el objetivo de realizar las pruebas finales y la presente guía.

Acceso a la aplicación

En la Figura 7.3 podemos ver las pantallas de registro e inicio de sesión. En el registro deberemos cumplimentar el formulario, leer los términos de uso de la aplicación y aceptarlos para poder continuar. Tras esto se nos enviará un correo de confirmación con un código que deberemos introducir. Una vez realizado este proceso la sesión quedará iniciada automáticamente. En futuros accesos deberemos introducir nuestras credenciales siempre y cuando utilicemos un dispositivo nuevo, ya que la aplicación es capaz de recordar los dispositivos.

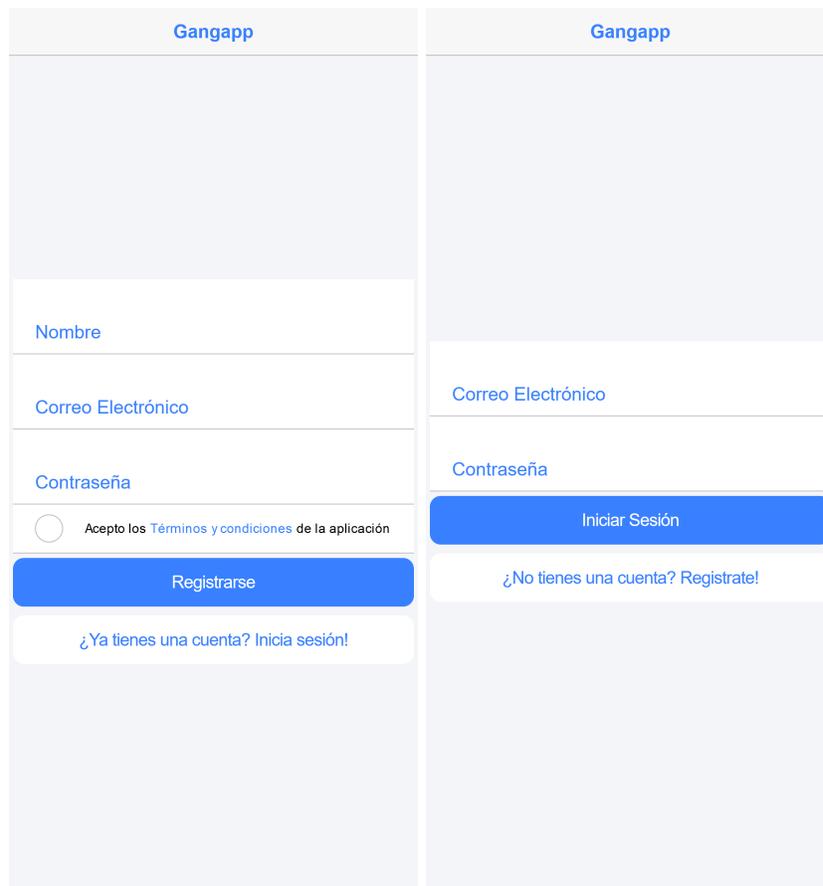


Figura 7.3: Acceso a la aplicación

Navegación por la aplicación

La aplicación cuenta con un menú inferior basado en pestañas para facilitar el acceso a las funcionalidades principales. En la Figura 7.4 podemos apreciar que existen cinco apartados: catálogo de productos, añadir nuevo producto, buscar productos, gestionar perfil y lista de la compra.

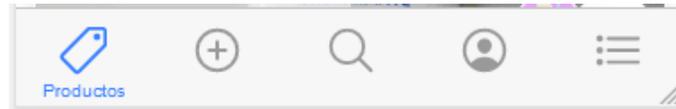


Figura 7.4: Menú de navegación

Catálogo de productos

En la Figura 7.5 se presenta la página inicial de arranque de la aplicación. Esta página está formada por el catálogo de productos sobre el que navegaremos para encontrar los productos más interesantes. Tocando sobre la tarjeta del producto se desplegará un menú con las acciones que podremos llevar a cabo. Para facilitar la comparativa se incluye una imagen de la versión en modo oscuro de este mismo apartado, que se detallará en la sección 7.3

Las opciones incluyen: guardado/desguardado del producto, visualización de sus detalles y añadir/quitar de la lista de la compra y reportar/corregir un producto (en función de si el producto es de publicación propia o ajena)

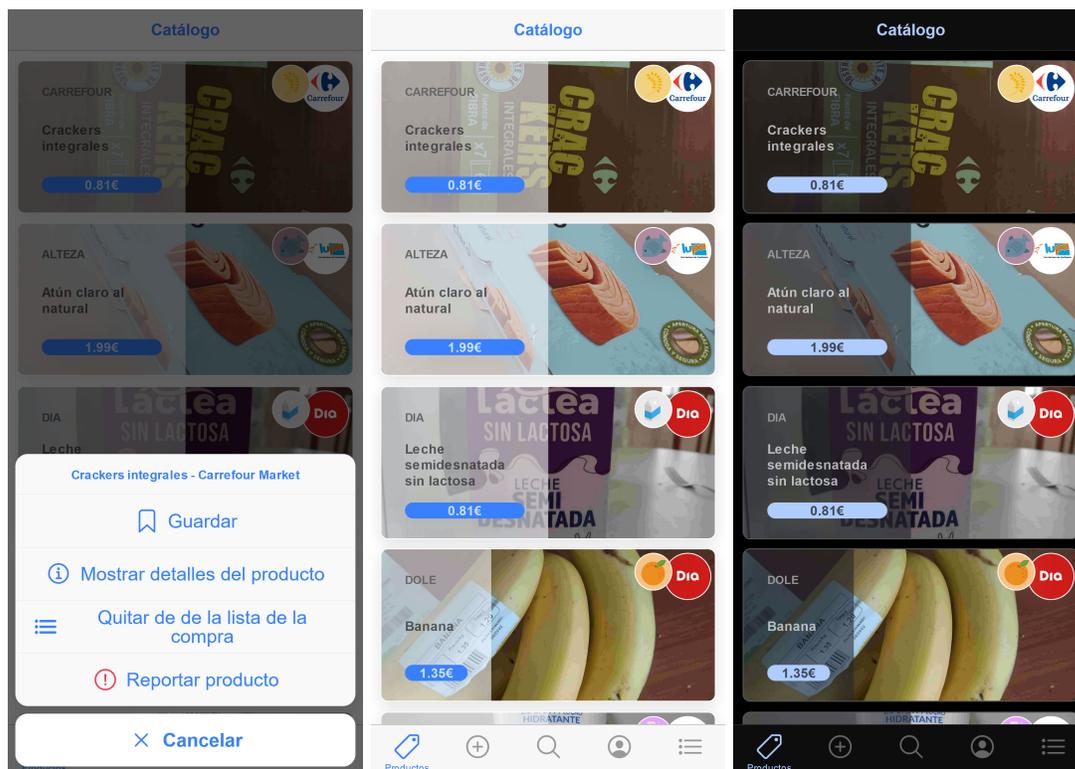


Figura 7.5: Catálogo de la aplicación

Añadir productos nuevos al catálogo

Situándonos en la pestaña correcta al usar el menú de navegación podremos comenzar el proceso de añadir un nuevo producto. Para ello hay que seguir una serie de pasos que describiremos a continuación, siendo los más relevantes los mostrados en la Figura 7.6. Los pasos en cuestión son los que a continuación se detallan:

1. Escaneo del código de barras del producto. Debemos pulsar el botón para iniciar el escaneo y situar el código de barras del producto frente a la cámara de forma visible.
2. Añadir información básica del producto como su nombre, marca y precio.
3. Iniciar la selección de tienda tocando el botón con el símbolo de mapa. Una vez dentro del propio mapa deberemos tocar la ubicación dónde queremos buscar las tiendas y se nos desplegarán aquellas que se ubiquen en un radio específico. Elegiremos la tienda deseada y su cadena, si es que fuese el caso.
4. Seleccionar la categoría del producto entre las propuestas
5. Añadir fotografías al producto pulsando el botón con el símbolo de la cámara. Podremos añadir un máximo de dos fotografías. Si el resultado de una fotografía no se considera satisfactorio, podremos borrarla mediante un toque en ella.
6. Seleccionar productos relacionados con el que estamos creando y finalizar el proceso

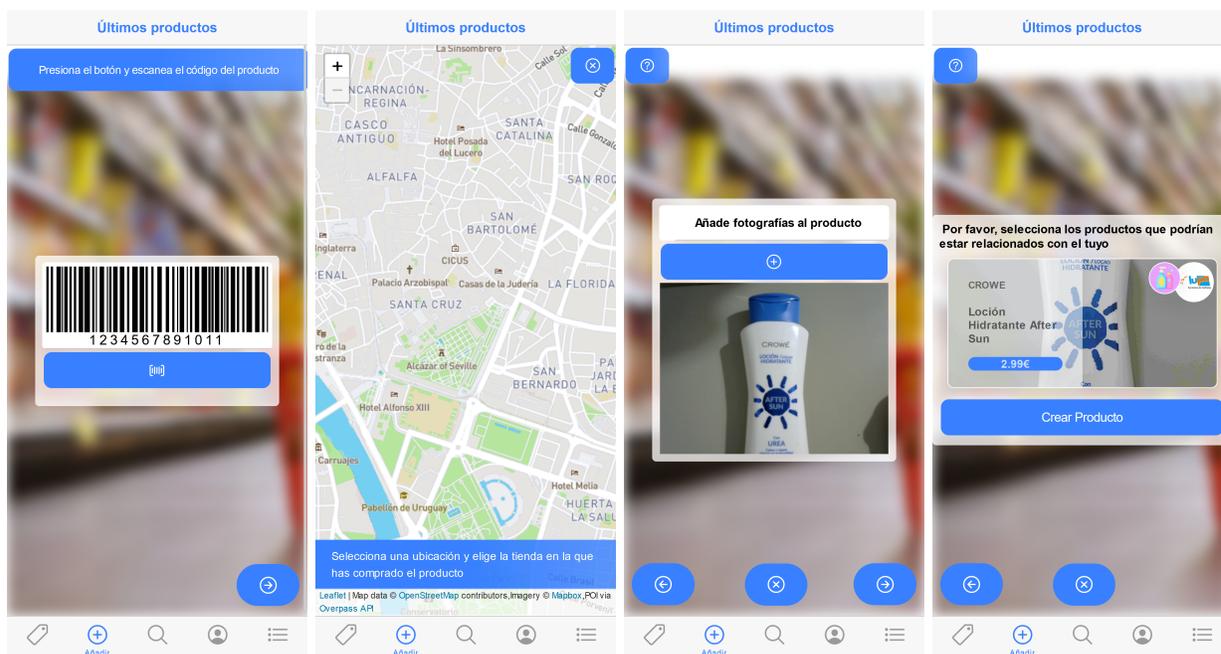


Figura 7.6: Pasos para añadir un producto al sistema

Búsqueda de productos en la aplicación

Avanzando a la siguiente pestaña del menú inferior tendremos acceso a un menú de búsqueda de productos. Tenemos múltiples opciones para buscar productos algunas de ellas combinables. Utilizando los iconos de la parte superior podremos buscar diferentes combinaciones de tipo de producto y cadena en la que se encuentra.

En el ejemplo de la Figura 7.7 se han utilizado los criterios de “Panadería y bollería” junto con la cadena “Carrefour” lo que nos ha mostrado el único producto que satisface estos criterios. También podremos utilizar una búsqueda por texto si conocemos el nombre del producto buscado. Pulsando en el botón del código de barras podremos efectuar una búsqueda rápida para saber si un producto en concreto se encuentra registrado.

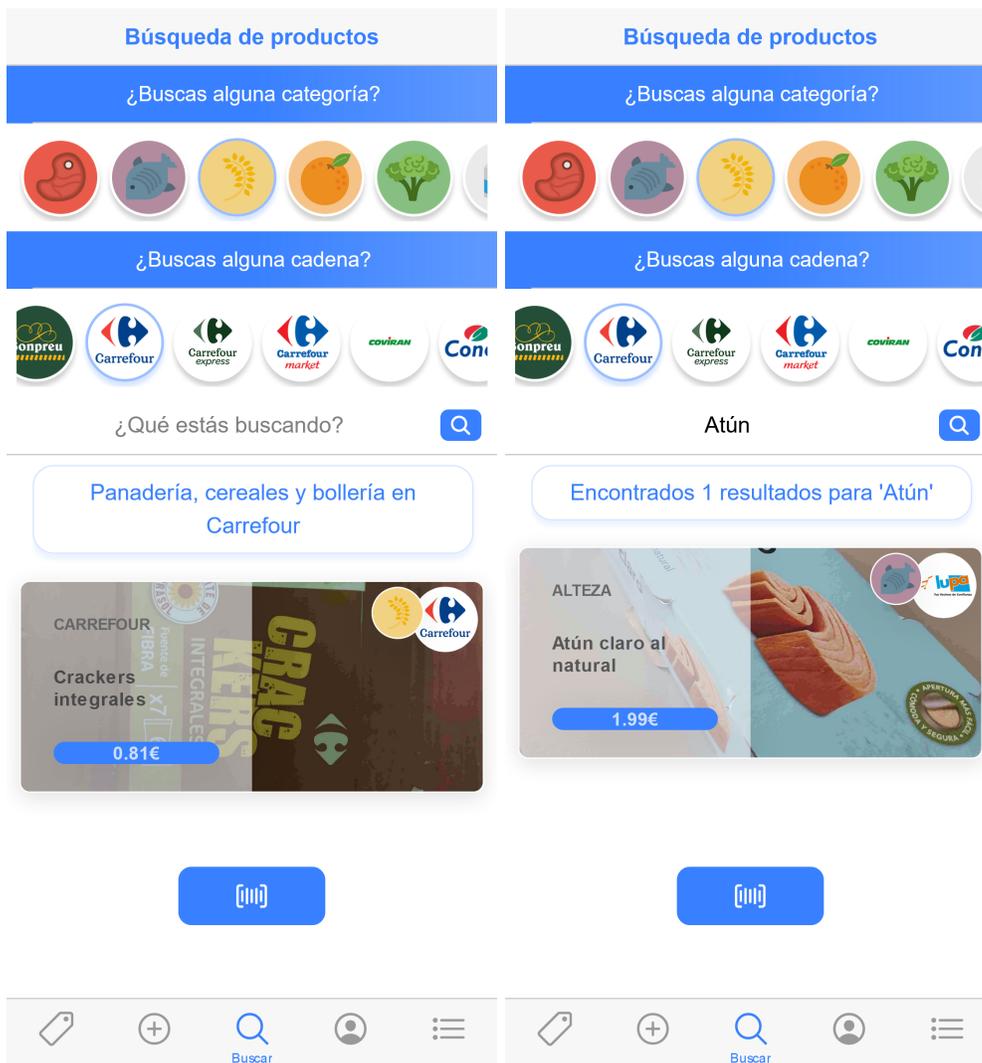


Figura 7.7: Búsquedas de productos aplicando diferentes criterios

Gestión del perfil del usuario

La siguiente pestaña, Figura 7.8, nos da acceso a dos listas principales de productos relacionados con el usuario: productos creados y productos guardados. Además nos permite utilizar un menú con algunas opciones interesantes para el usuario: el cierre de sesión, el cambio entre temas claro/oscuro y el cambio de información del perfil.

Esta sección se divide en dos subgrupos de productos. El subgrupo identificado mediante un icono de un marca-páginas se corresponde con la lista de productos guardados a la que podremos añadir productos mediante el menú desplegable. El icono del lapicero y el papel nos dará acceso a la lista de productos que hemos creado. Podemos ver que existe un producto resaltado mediante el color rojo, indicándonos que existe un problema con el mismo que debemos corregir aunque este proceso se detallará en la sección 7.3.

Desde esta sección también tenemos acceso a las opciones de menú de los productos de las que hemos hablado anteriormente.

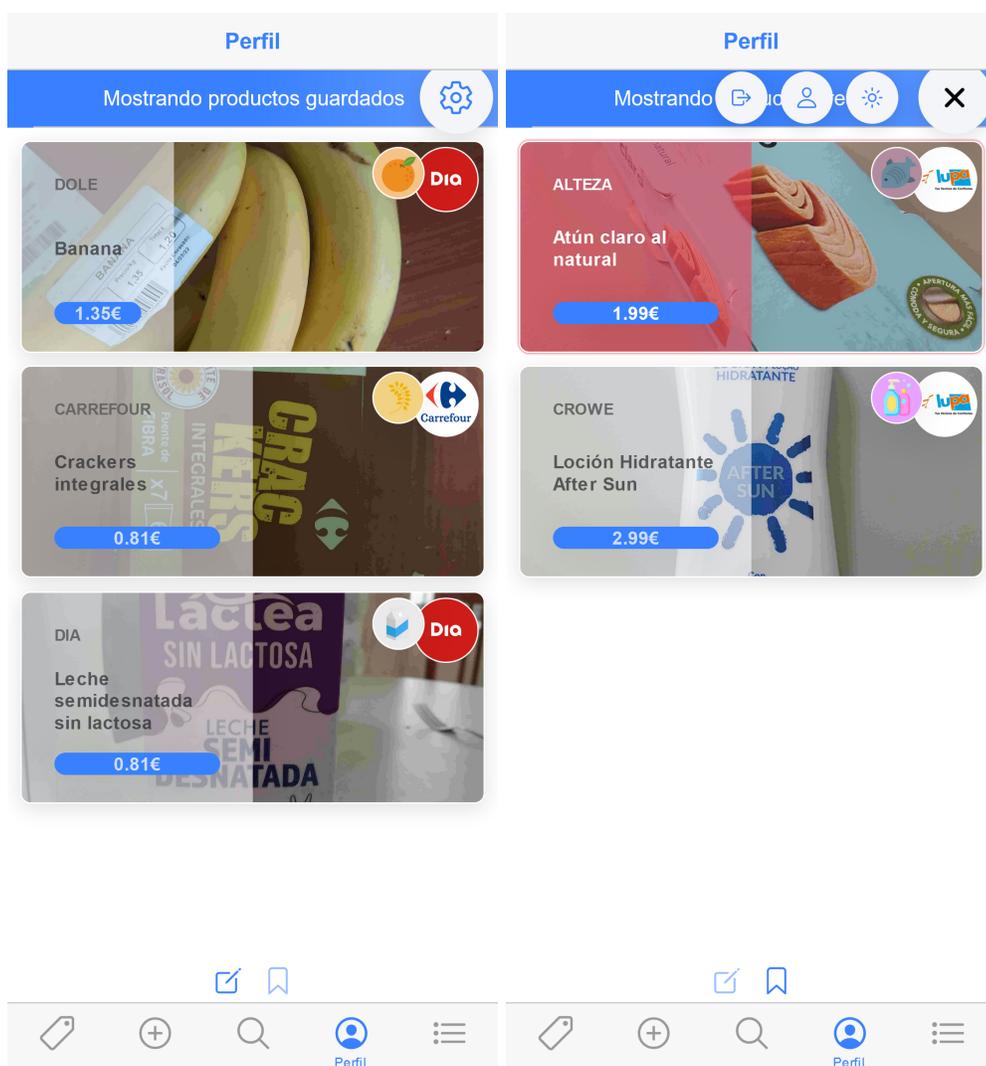


Figura 7.8: Acciones disponibles en la pestaña de gestión de perfil

Listas de la compra

En la última pestaña del manú podemos encontrar toda la funcionalidad asociada con las listas de la compra y los productos que añadimos a ella. Se nos presenta la información de manera visualmente atractiva para poder ver el precio total de los productos que hemos seleccionado y se nos desglosan los costes también para cada una de las tiendas como podemos ver en la Figura 7.9.

En esta pestaña la funcionalidad se obtiene al arrastrar el dedo por la pantalla sobre los diferentes elementos. Mediante este gesto podremos cambiar las cantidades de un producto, eliminarlo rápidamente de la lista o guardar en el dispositivo la lista de la compra actual (incluyendo los precios de los productos en el momento del guardado).

Las listas de la compra guardadas aparecen por fechas en la parte inferior y tocándolas podremos ver que productos contenían. También mediante el empleo táctil podremos eliminar aquellas que ya no necesitamos.

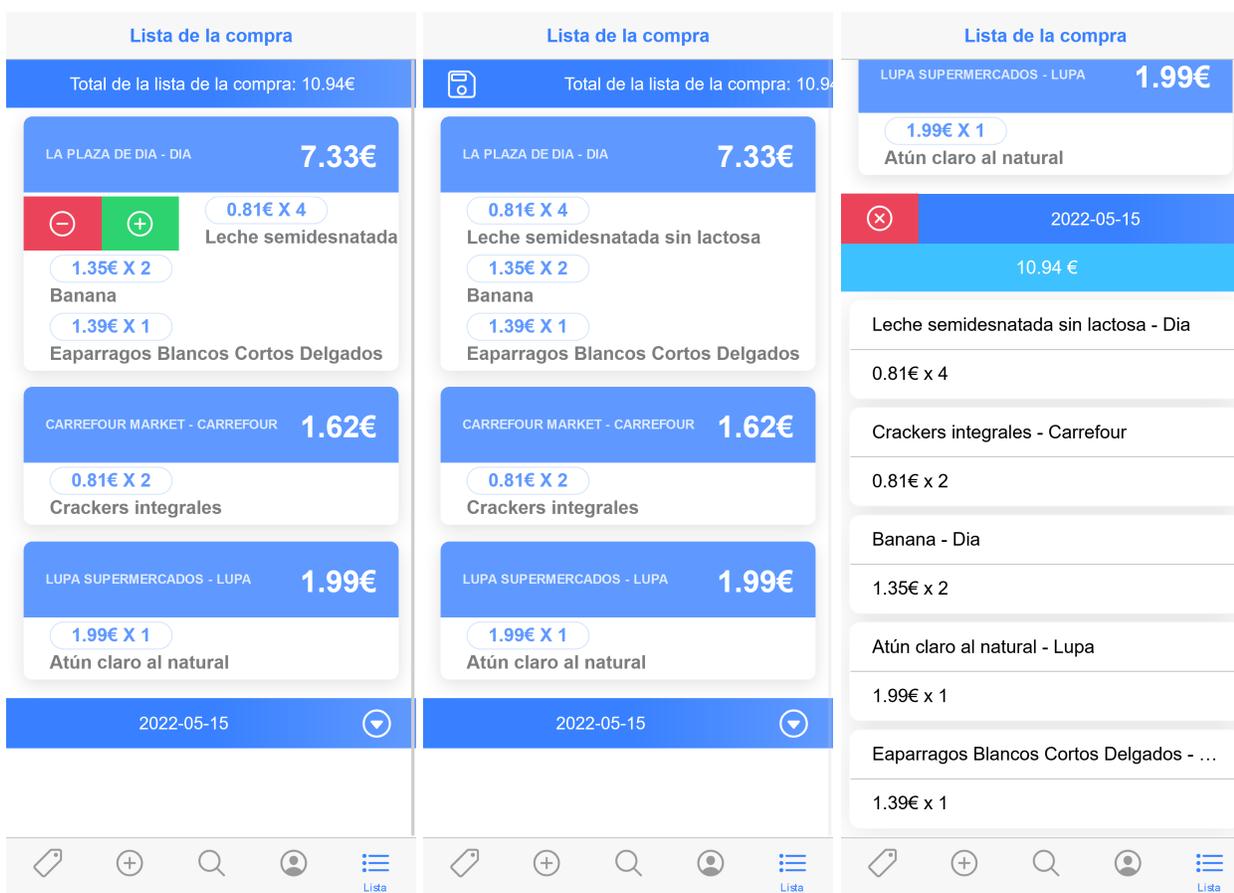


Figura 7.9: Listas de la compra actuales y guardadas

Detalles de un producto

En esta sección presentada en la Figura 7.10, accesible desde el menú desplegable, podemos encontrar una breve presentación del producto y su información en la parte superior. Desplazando la vista hacia abajo podemos encontrar un apartado para añadir ofertas sobre el producto completando el formulario con la información necesaria. Si ya existiese alguna oferta sobre el producto tendríamos sus detalles en esta ventana y podríamos marcarla como completada si la fecha de fin ya se ha completado.

En la zona inferior de la vista se puede apreciar un gráfico en el que se presenta el precio y sus oscilaciones desde que el producto fue añadido al sistema. Cuando se completa el formulario de ofertas el precio de oferta se añade a este histórico.



Figura 7.10: Visualización de los detalles de un producto seleccionado

Funcionalidad sobre los reportes

Los productos con errores son reportados mediante el menú desplegable presentado en las anteriores secciones. En la Figura 7.11 podemos ver las vistas en las que se añade la información para crear un reporte y para corregir un producto reportado.

Para crear un reporte tras identificar los fallos en un producto deberemos seleccionarlos de la lista de posibles razones y añadir, si lo creemos necesario, una descripción del reporte para ayudar a la solución del problema. Con el formulario completo, procederemos a enviarlo mediante el botón correspondiente.

Para solucionar el reporte el usuario deberá acceder a la pestaña determinada para ello que le mostrará un formulario personalizado en función de las razones elegidas por el usuario reportador. Una vez solucionados los problemas, actualizaremos el producto y volverá a estar disponible.

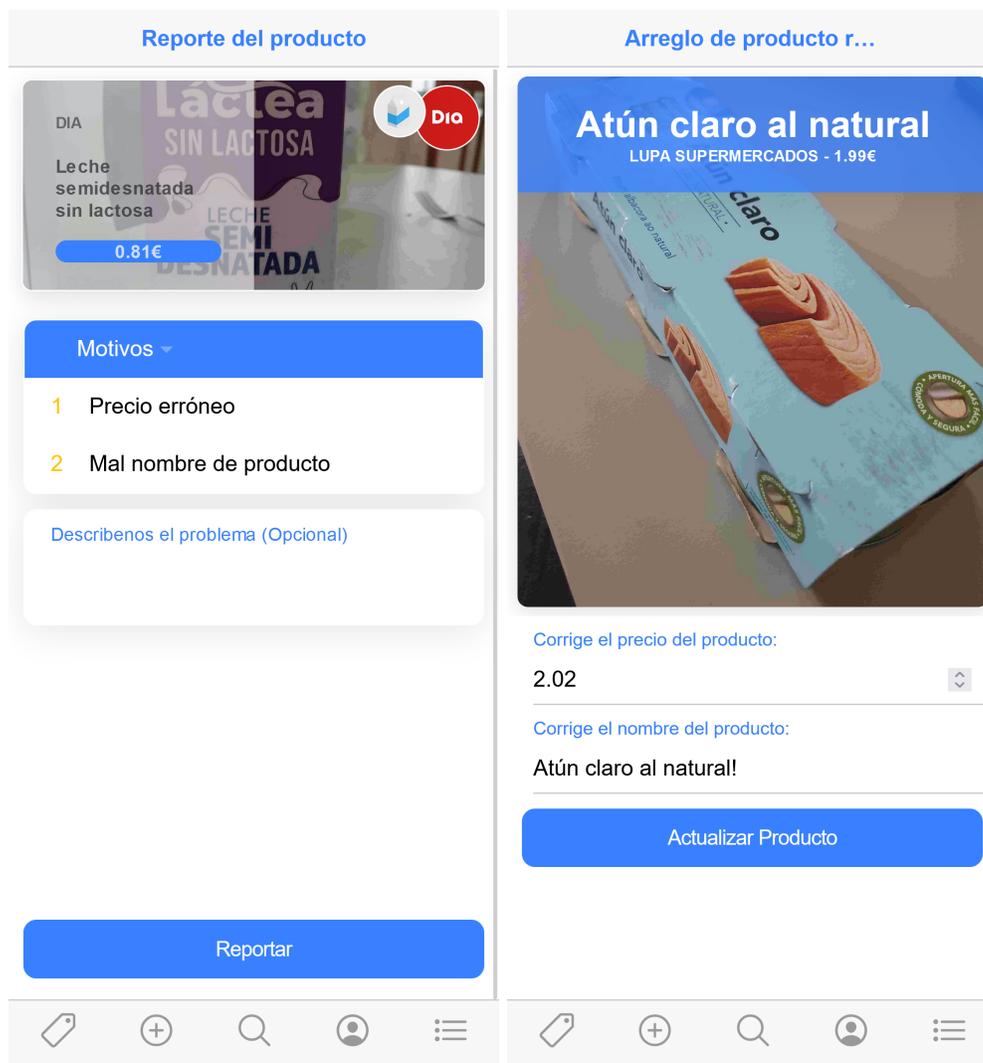


Figura 7.11: Funcionalidad asociada a los reportes