



TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Aplicación para la gestión de una tienda minorista

Autor

Julia María Cano Flores

Directores

María Luisa Rodríguez Almendros



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, febrero de 2024

Aplicación para la gestión de una tienda minorista

Julia María Cano Flores

Palabras clave: minorista, base de datos, software, requisitos

Resumen

El desarrollo de este Trabajo Fin de Grado (TFG) tiene como finalidad principal simplificar y mejorar la eficiencia de la gestión de negocios minoristas textiles. Está diseñado para optimizar el proceso de seguimiento de ventas mediante una base de datos que registre todas las transacciones realizadas. El sistema no solo registra compras y devoluciones, sino que también incorpora una característica distintiva: la gestión de préstamos. Esta funcionalidad es particularmente útil en localidades pequeñas, donde las relaciones de confianza entre comerciantes y clientes permiten prácticas como llevarse productos sin un desembolso inicial. Los clientes pueden probar los artículos antes de comprometerse a la compra o optar por la devolución sin costos adicionales.

El software proporcionará una interfaz de usuario intuitiva compuesta de múltiples pantallas, entre las que podemos destacar un inventario actualizado en tiempo real, la visualización de los artículos en venta o una lista de clientes habituales del negocio. Además, para impulsar la toma de buenas decisiones estratégicas, el programa generará gráficos analíticos que reflejarán el progreso económico del negocio.

Este proyecto surge de la necesidad evidente de una solución tecnológica adaptada a los requisitos específicos de los negocios minoristas textiles. Hasta el momento, son negocios que no se han tenido en cuenta debido a su bajo impacto y su pequeño tamaño. Sin embargo, debemos de mirar por llevar la tecnología a todos los ámbitos. Esta aplicación busca llenar ese vacío y convertirse en una herramienta útil para comerciantes minoristas.

Application for the management of a retail store

Julia María Cano Flores

Keywords: retailer, database, software, requirements

Abstract

The main purpose of the development of this Final Degree Project (FDP) is to simplify and improve the efficiency of textile retail business management. It is designed to optimize the sales tracking process through a database that records all transactions made. The system not only records purchases and returns but also incorporates a distinctive feature: loan management. This functionality is particularly useful in small localities, where trust relationships between merchants and customers allow practices such as taking products without an initial outlay. Customers can test items before committing to purchase or opt for a return without additional costs.

The software will provide an intuitive user interface composed of multiple screens, among which we can highlight an inventory updated in real-time, the display of items for sale, or a list of regular customers of the business. In addition, to drive good strategic decision-making, the program will generate analytical charts that will reflect the economic progress of the business.

This project arises from the clear need for a technological solution adapted to the specific requirements of textiles retail businesses. So far, they are businesses that have not been considered due to their low impact and small size. However, we must look to bring technology to all areas. This application seeks to fill that void and become a useful tool for retail merchants.

Yo, **Julia María Cano Flores**, alumna de la titulación INGENIERÍA INFORMÁTICA de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77649643x, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Julia María Cano Flores

Granada a 2 de febrero de 2024 .

D. **María Luisa Rodríguez Almendros**, Profesora del Área de Ingeniería del Software del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Aplicación para la gestión de una tienda minorista***, ha sido realizado bajo su supervisión por **Julia María Cano Flores**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 2 de febrero de 2024 .

Los directores:

María Luisa Rodríguez Almendros

Agradecimientos

En primer lugar, quiero dar mis más sinceros agradecimientos a mi familia, en especial a mis padres. Gracias por enseñarme el significado de sacrificio, perseverancia y constancia. Gracias por confiar en mí, por apoyarme en mis decisiones. Gracias por los ánimos cuando me sentía insegura y por los elogios al alcanzar una meta. Gracias por estar ahí siempre, por nunca abandonar. Durante estos cuatro años he crecido mucho, no solo académicamente. Me siento más válida, más resolutiva, más segura, más tolerante, más adaptativa. Me siento más. Quería terminar esta etapa devolviéndoles una pequeña parte de todo lo que me han aportado, por eso, este trabajo de fin de grado es para ellos. En concreto, para mi padre, propietario de un pequeño negocio minorista. Ayudando en la tienda rápidamente pude ver la necesidad de introducir la informática y este proyecto ha sido una oportunidad perfecta para llevar a cabo mis ideas.

Por otro lado, también quería agradecer a mis amigos: a aquellos que conocí gracias a la carrera, los que conocí durante el Erasmus, los que conocí un día cualquiera y los que llevo conociendo toda una vida. Son los que han hecho que esta experiencia universitaria sea inmejorable. Gracias por acompañarme en esta etapa y regalarme grandes momentos, pero sobre todo por hacer de cualquier ciudad, un hogar para mí.

Por último, agradecer a la Universidad de Granada, a los profesores y a mis compañeros, por todas las facilidades que me han proporcionado para finalizar la carrera de informática. En especial, me gustaría agradecer a mi tutora María Luisa Rodríguez, por su gran atención y dedicación. Este proyecto no hubiera sido lo mismo sin sus anotaciones y su conocimiento experto. He adquirido grandes aprendizajes con ella y estoy orgullosa del proyecto que hemos logrado.

Gracias.

Índice general

1. Introducción	15
1.1. Motivación	15
1.2. Objetivos del proyecto	16
1.3. Estructura del documento	17
2. Planificación	18
2.1. Fases	18
2.1.1. Fase inicial	18
2.1.2. Fase de investigación	18
2.1.3. Fase de análisis	19
2.1.4. Fase de diseño	19
2.1.5. Fase de desarrollo	19
2.1.6. Redacción de la memoria	20
2.1.7. Diagrama de Gantt	20
2.2. Presupuesto	22
2.2.1. Recursos	22
2.2.2. Costes	22
3. Estado del arte	25
3.1. Definición de un negocio minorista	25
3.2. Estudio de los pequeños negocios mixtos	28
3.3. Estudio de comerciantes minoristas	30
3.4. Estudio de usabilidad y accesibilidad	31
3.4.1. Estudio de usabilidad	31
3.4.2. Estudio de accesibilidad	31
3.5. Legislación actual de protección de datos	33
3.6. Aplicaciones similares y comparativas	34
3.6.1. myGESTIÓN	34
3.6.2. Clover	35
4. Análisis de tecnologías a utilizar	37
4.1. Dispositivo destino para la aplicación	37
4.2. Tecnología de documentación	37

4.2.1.	Descripción de LaTeX	37
4.3.	Tecnología para la realización de los diagramas de casos de uso	38
4.3.1.	Descripción de Visual Paradigm	38
4.4.	Tecnología para el diseño de los bocetos	39
4.4.1.	Descripción de Goodnotes	39
4.5.	Metodología de desarrollo de la aplicación	40
4.5.1.	Descripción de metodología iterativa	40
4.6.	Tecnología de soporte de la base de datos	41
4.6.1.	Descripción de Supabase	41
4.7.	Framework de desarrollo	42
4.7.1.	Descripción de Flutter	42
5.	Análisis de la aplicación	44
5.1.	Especificación de requisitos	44
5.1.1.	Requisitos funcionales	44
5.1.2.	Requisitos no funcionales	62
5.2.	Modelos de caso de uso	63
5.2.1.	Diagrama de casos de uso	63
5.2.2.	Descripción de casos de uso	67
5.3.	Modelo conceptual	69
5.4.	Modelos de comportamiento	70
5.4.1.	Apertura y cierre de sesión	70
6.	Diseño	71
6.1.	Diagrama de la arquitectura del sistema	71
6.2.	Diseño de la interfaz gráfica	72
6.2.1.	Pantalla de inicio de sesión	73
6.2.2.	Pantalla principal	74
6.2.3.	Pantalla de clientes	75
6.2.4.	Pantalla de artículos	76
6.2.5.	Pantalla de movimientos	77
6.2.6.	Pantalla de gráficos y cierre de sesión	78
7.	División en iteraciones	79
7.1.	Primera iteración	79
7.1.1.	Alcance de la iteración	79
7.1.2.	Implementación	80
7.1.3.	Pruebas de funcionalidad	85
7.1.4.	Revisión de la iteración	86
7.1.5.	Plan para la próxima iteración	87
7.2.	Segunda iteración	88
7.2.1.	Alcance de la iteración	88
7.2.2.	Implementación	88
7.2.3.	Pruebas de funcionalidad	95

7.2.4.	Revisión de la iteración	95
7.2.5.	Plan para la próxima iteración	96
7.3.	Tercera iteración	96
7.3.1.	Alcance de la iteración	96
7.3.2.	Pantalla de gráficos	96
7.3.3.	Mejora de la interfaz gráfica y la accesibilidad	97
7.3.4.	Colores de la aplicación	98
7.3.5.	Pruebas de funcionalidad	110
7.3.6.	Revisión de la iteración	110
8.	Base de datos e implementación	111
8.1.	Creación y modificaciones de la base de datos	111
8.1.1.	Creación de la base de datos	111
8.1.2.	Tablas de la base de datos	112
8.1.3.	Modificaciones intermedias de la base de datos	113
8.1.4.	Diagrama final del diseño de la base de datos	113
8.2.	Diagrama de clases	113
9.	Conclusiones y trabajos futuros	125
9.1.	Conclusiones	125
9.2.	Trabajos futuros	127
A.	Descripción de casos de uso	128
B.	Diagramas de secuencia	156
B.0.1.	Apertura y cierra de sesión	156
B.0.2.	Gestión de artículos	157
B.0.3.	Gestión de inventario	161
B.0.4.	Gestión de clientes	162
B.0.5.	Gestión de movimientos	166
B.0.6.	Gestión de resúmenes y gráficas	171
	Referencias	173

Índice de figuras

2.1. Diagrama de Gantt	21
3.1. Interfaz gráfica de myGestión	35
3.2. Interfaz gráfica de Clover	35
4.1. Logo LaTeX	38
4.2. Logo Visual Paradigm	39
4.3. Logo GoodNotes	40
4.4. Metodología iterativa	41
4.5. Logo Supabase	42
4.6. Logo Flutter	43
5.1. Diagrama de paquetes de casos de uso	63
5.2. Casos de uso del inicio y cierre de sesión	64
5.3. Casos de uso de la gestión de artículos	64
5.4. Casos de uso de la gestión de clientes	65
5.5. Casos de uso de la gestión de movimientos	65
5.6. Casos de uso de la gestión de gráficos y resúmenes	66
5.7. Modelo conceptual	69
5.8. Diagrama de secuencia de inicio de sesión	70
5.9. Diagrama de secuencia de cierre de sesión	70
6.1. Diagrama de la arquitectura del sistema.	71
6.2. Diseño de la pantalla de inicio de sesión.	73
6.3. Diseño de la pantalla principal.	74
6.4. Diseño de la pantalla clientes.	75
6.5. Diseño de la pantalla artículos.	76
6.6. Diseño de la pantalla movimientos.	77
6.7. Diseño de la pantalla gráficos.	78
7.1. Interfaz de usuario de inicio de sesión.	80
7.2. Interfaz de usuario de la pantalla principal.	80
7.3. Interfaz de usuario de la pantalla de visualización de clientes.	81
7.4. Interfaz de usuario de la pantalla de nuevo cliente.	81

7.5. Interfaz de usuario de la pantalla de editar cliente.	82
7.6. Interfaz de usuario de la pantalla de visualizar datos de un cliente.	82
7.7. Interfaz de usuario de la pantalla de visualizar las categorías.	83
7.8. Interfaz de usuario de la pantalla de añadir nueva categoría.	83
7.9. Interfaz de usuario de la pantalla de editar una categoría.	84
7.10. Interfaz de usuario de la pantalla de visualizar artículos.	84
7.11. Interfaz de usuario de la pantalla de añadir un nuevo artículo.	85
7.12. Interfaz de usuario de la pantalla de editar un artículo existente.	85
7.13. Interfaz de usuario de la pantalla principal.	88
7.14. Interfaz de usuario de la pantalla de nueva venta.	89
7.15. Interfaz de usuario de la pantalla de nuevo préstamo.	90
7.16. Interfaz de usuario de la pantalla de visualización de movimientos.	90
7.17. Interfaz de usuario de la pantalla de detalles de una venta.	91
7.18. Interfaz de usuario de la pantalla de devolución de una venta.	92
7.19. Interfaz de usuario de la pantalla de una venta con devolución hecha.	92
7.20. Interfaz de usuario de la pantalla los detalles de un préstamo.	93
7.21. Interfaz de usuario de la pantalla de compra a partir de un préstamo.	93
7.22. Interfaz de usuario de la pantalla de detalles de una devolución.	94
7.23. Interfaz de usuario que muestra el cambio de la gestión de las tallas.	94
7.24. Pantalla de gráficos en modo Mensual.	97
7.25. Pantalla de gráficos en modo Anual.	97
7.26. Gama de colores azules de Flutter.	98
7.27. Gama de colores amarillos de Flutter.	98
7.28. Pantalla de inicio de sesión.	99
7.29. Cierre de sesión.	99
7.30. Pantalla principal.	100
7.31. Pantalla de nueva venta.	101
7.32. Pantalla de nuevo préstamo.	101
7.33. Pantalla de visualización de clientes.	102
7.34. Pantalla de visualización de los datos de un cliente.	103
7.35. Pantalla de edición de los datos de un cliente.	103
7.36. Pantalla de registro de un nuevo cliente.	103
7.37. Pantalla de visualización de las categorías.	104
7.38. Pantalla de edición de las propiedades de una categoría.	104
7.39. Pantalla de registro de una nueva categoría.	105
7.40. Pantalla de visualización de los artículos de una categoría.	105
7.41. Pantalla de visualización de los datos de un artículo.	106
7.42. Pantalla de edición de los datos de un artículo.	106
7.43. Pantalla de registro de un nuevo artículo.	106

7.44. Pantalla de visualización la lista de renovación del stock. . . .	107
7.45. Pantalla de visualización los movimientos de la tienda. . . .	108
7.46. Pantalla de visualización de los detalles de un movimiento. . .	109
7.47. Pantalla de gráficos en modo mensual.	110
8.1. Instancia de la base de datos.	111
8.2. Diagrama de la base de datos.	113
8.3. Diagrama de clases.	114
B.1. Diagrama de secuencia de inicio de sesión	156
B.2. Diagrama de secuencia de cierre de sesión	157
B.3. Diagrama de secuencia de un nuevo artículo	157
B.4. Diagrama de secuencia de edición de un artículo	158
B.5. Diagrama de secuencia de eliminación de un artículo	158
B.6. Diagrama de secuencia de visualización de datos de un artículo	159
B.7. Diagrama de secuencia de búsqueda de un artículo	160
B.8. Diagrama de secuencia de categorización de un artículo . . .	160
B.9. Diagrama de secuencia de visualización de la lista de artículos	161
B.10. Diagrama de secuencia de actualización de inventario	161
B.11. Diagrama de secuencia de visualización de la lista de renova- ción de artículos	162
B.12. Diagrama de secuencia de registro de un cliente	162
B.13. Diagrama de secuencia de edición de los datos de un cliente .	163
B.14. Diagrama de secuencia de eliminación de un cliente	163
B.15. Diagrama de secuencia de visualización de los datos de un cliente	164
B.16. Diagrama de secuencia de visualización de la lista de clientes	164
B.17. Diagrama de secuencia de búsqueda de un cliente	165
B.18. Diagrama de secuencia de filtrado de clientes	165
B.19. Diagrama de secuencia de nueva venta	166
B.20. Diagrama de secuencia de nuevo préstamo	167
B.21. Diagrama de secuencia de nueva devolución	167
B.22. Diagrama de secuencia de eliminación de un movimiento . . .	168
B.23. Diagrama de secuencia de visualización de datos de un movi- miento	169
B.24. Diagrama de secuencia de visualización la lista de movimientos	169
B.25. Diagrama de secuencia del filtrado de movimientos	170
B.26. Diagrama de secuencia de búsqueda de movimientos	170
B.27. Diagrama de secuencia de generación de una compra desde un préstamo	171
B.28. Diagrama de secuencia de visualización de la caja diaria . . .	171
B.29. Diagrama de secuencia de visualización de gráficos	172

Índice de cuadros

2.1. Distribución de horas por fase	20
2.2. Presupuesto total	24
3.1. Descripción de la tabla	36
5.1. RF1	45
5.2. RF2	46
5.3. RF3	46
5.4. RF4	47
5.5. RF5	47
5.6. RF6	48
5.7. RF7	48
5.8. RF8	49
5.9. RF9	49
5.10. RF10	50
5.11. RF11	50
5.12. RF12	51
5.13. RF13	51
5.14. RF14	52
5.15. RF15	52
5.16. RF16	53
5.17. RF17	53
5.18. RF18	54
5.19. RF19	55
5.20. RF20	56
5.21. RF21	57
5.22. RF22	58
5.23. RF23	58
5.24. RF24	59
5.25. RF25	59
5.26. RF26	60
5.27. RF27	60
5.28. RF28	61

5.29. RF29	61
5.30. RNF1	62
5.31. RNF2	62
5.32. RNF3	62
5.33. RNF4	62
5.34. Caso de uso - Inicio de sesión	67
5.35. Caso de uso - Cierre de sesión	68
A.1. Caso de uso - Inicio de sesión	128
A.2. Caso de uso - Cierre de sesión	129
A.3. Caso de uso - Introducción de un nuevo artículo	130
A.4. Caso de uso - Edición de un artículo existente	131
A.5. Caso de uso - Eliminación de un artículo	132
A.6. Caso de uso - Visualización de los datos de un artículo	133
A.7. Caso de uso - Búsqueda de un artículo por nombre	134
A.8. Caso de uso - Categorización de un artículo	135
A.9. Caso de uso - Visualización de la lista de los artículos existentes	136
A.10. Caso de uso - Visualización de lista de renovación de artículos	137
A.11. Caso de uso - Registro de un nuevo cliente habitual	138
A.12. Caso de uso - Edición de los datos de un cliente existente	139
A.13. Caso de uso - Eliminación de un cliente	140
A.14. Caso de uso - Visualización de los datos de un cliente	141
A.15. Caso de uso - Visualización de la lista de clientes existentes	142
A.16. Caso de uso - Búsqueda de un cliente por nombre	143
A.17. Caso de uso - Filtrado de clientes con préstamos	144
A.18. Caso de uso - Introducción de una nueva venta	145
A.19. Caso de uso - Introducción de un nuevo préstamo	146
A.20. Caso de uso - Introducción de una devolución	147
A.21. Caso de uso - Eliminación de un movimiento	148
A.22. Caso de uso - Visualización de los datos de un movimiento	149
A.23. Caso de uso - Visualización de la lista de movimientos existentes	150
A.24. Caso de uso - Filtrado de movimientos según su tipo	151
A.25. Caso de uso - Búsqueda de un movimiento por fecha o cliente	152
A.26. Caso de uso - Generación de una compra a partir de un préstamo	153
A.27. Caso de uso - Visualización de la caja diaria	154
A.28. Caso de uso - Visualización de gráficas	155

Capítulo 1

Introducción

1.1. Motivación

Se entiende como negocio minorista toda empresa de comercio que adquiere mercancías por cuenta propia, y las revende directamente al consumidor final. En negocios minoristas pequeños podemos encontrar un trato más cercano con los clientes y una forma algo distinta de gestionar un comercio.

Al ser nacida y criada en un pueblo, la mayoría de negocios con los que he crecido eran comercios minoristas pequeños. Además, mi padre es propietario de una tienda y he podido conocer de primera mano cuál es el funcionamiento real de este tipo de negocios. Tras una vida tratando con ellos y una entrevista realizada a mi padre, he encontrado la necesidad de informatizar este trabajo con el objetivo de conseguir optimizar el rendimiento y llevar un seguimiento. En la actualidad, muchos comerciantes minoristas no disponen de grandes tecnologías y recurren a métodos tradicionales basados en papel para el seguimiento de su negocio. Esto tiene un riesgo alto de pérdida de datos o cometer errores. Además, el proceso de recuperación y análisis de datos se convierten en tareas muy tediosas, ya que la falta de sistemas automatizados impide un filtrado eficiente de información. Todo esto ralentiza el trabajo de forma significativa, haciendo que la productividad del comerciante sea menor y, por tanto, la evolución del negocio no sea óptima.

En el mercado actual no encontramos aplicaciones que pongan solución al problema de los negocios minoristas pequeños. Las soluciones software existentes tienden a enfocarse en entidades de mayor escala. Como hemos visto, los negocios minoristas tienen singularidades que conllevan establecer requisitos específicos y tratar su gestión de forma independiente a la gestión

de un gran negocio. Por ello, es crucial el desarrollo de un software que satisfaga las necesidades de los pequeños comerciantes y les permita conducir su negocio a una evolución óptima.

1.2. Objetivos del proyecto

El objetivo general de este proyecto es proporcionar una solución informática analizando los procedimientos típicos de un comercio minorista pequeño que actualmente se realizan de forma manual y desarrollando una aplicación que los automatice. Para conseguir dicho objetivo, podemos diferenciar una serie de objetivos específicos:

- **Objetivo 1: Análisis de los comercios minoristas**
 - **Descripción:** Se debe de realizar una búsqueda de información sobre los negocios minoristas con el objetivo de entender mejor cuáles son sus características.
- **Objetivo 2: Estudio de aplicaciones similares**
 - **Descripción:** Análisis del mercado actual de aplicaciones similares para identificar funcionalidades innovadoras que implementar en nuestra aplicación.
- **Objetivo 3: Análisis de usabilidad y accesibilidad**
 - **Descripción:** Estudiar soluciones accesibles para conseguir una aplicación fácil de usar para la mayoría de usuarios.
- **Objetivo 4: Análisis de las tecnologías a utilizar**
 - **Descripción:** Investigar sobre cuáles son las mejores tecnologías para llevar a cabo el proyecto.
- **Objetivo 5: Desarrollo e implementación de la aplicación**
 - **Descripción:** Analizar, desarrollar y probar la aplicación con el objetivo de conseguir los mejores resultados.
- **Objetivo 6: Validación de la aplicación por usuarios reales**
 - **Descripción:** Poner a prueba la aplicación en un entorno real para ver si cumple las expectativas del usuario.

1.3. Estructura del documento

Esta sección se ofrece una visión de la estructura y organización del documento. Se compone de las siguientes partes:

- **Resumen:** Un resumen donde se exponen las principales funcionalidades del proyecto y sus objetivos.
- **Introducción:** En este apartado se exponen los principales objetivos que se pretenden cumplir durante el desarrollo del proyecto. Además, se explica la motivación por la que se empezó a idear dicho proyecto a modo de introducción.
- **Planificación:** Especificación de la planificación del proyecto, donde se muestran las distintas fases y la estimación presupuestaria del mismo.
- **Estado del arte:** Investigación sobre los principales temas que se tratan en el proyecto como los negocios minoristas, los negocios mixtos y los comerciantes minoristas. Además, se tratan temas legales como la protección de datos. Con esta investigación, entramos en contexto para posteriormente entender los requisitos del proyecto.
- **Análisis de tecnologías a utilizar:** Enumeración de tecnologías utilizadas para el desarrollo del proyecto. Para cada tecnología, se explica porqué se ha utilizado y se citan sus principales características.
- **Análisis de la aplicación:** Obtención y explicación de los requisitos del sistema. Posteriormente se amplía esta información desarrollando los casos de uso, el modelo conceptual y los diagramas de secuencia del sistema.
- **Diseño:** Descripción de la arquitectura del sistema, el diseño de las clases necesarias para el programa y la interfaz de usuario.
- **Implementación de la aplicación:** La implementación de la aplicación se desarrollará en con una metodología ágil, por lo que este capítulo se dividirá en sus iteraciones.
- **Bibliografía:** Recopilación de todas las fuentes de información utilizadas durante la realización del proyecto.

Capítulo 2

Planificación

2.1. Fases

La planificación del desarrollo del proyecto se ha dividido en las fases que vamos a ver a continuación.

2.1.1. Fase inicial

En esta etapa es donde se definen las bases del proyecto. Se expone la motivación que nos conduce a llevar el proyecto a cabo, se identifican las necesidades y se establecen unos objetivos principales que deberán de cumplirse a lo largo del desarrollo del proyecto. Esta información podemos verla en el capítulo 1 (ver página) de este documento.

2.1.2. Fase de investigación

Durante esta fase, se recopila información para entender el entorno donde se pondrá en práctica la aplicación. Para ello, se investigarán cuestiones como la definición de negocio minorista, un estudio de los pequeños negocios mixtos y un estudio de los comerciantes minoristas. Además, con el objetivo de que la aplicación cumpla los términos legales, también se realizará un estudio sobre usabilidad, accesibilidad y la legislación de protección de datos.

Para finalizar esta fase de investigación y entender la verdadera necesidad de desarrollar esta aplicación, se realizará una comparación de aplicaciones similares en el mercado.

2.1.3. Fase de análisis

En esta fase encontraremos 2 apartados:

- **Análisis de las tecnologías:** En este apartado encontraremos una lista de las tecnologías que se han utilizado para el desarrollo del proyecto, además de una explicación de porqué han sido seleccionadas y cuáles son sus principales características.
- **Análisis de la aplicación:** Durante este apartado se estudian los requisitos que deberá de cumplir la aplicación para satisfacer las necesidades del cliente. Para una mayor especificación y claridad, se desarrollarán los casos de uso y los diagramas de secuencia del sistema. Para entender como se relacionan las entidades, también se hará un modelo conceptual.

2.1.4. Fase de diseño

En esta fase, se especifica más en detalle cómo va a construirse la aplicación mediante un diagrama de la arquitectura del sistema y los bocetos de la interfaz gráfica del usuario. Los bocetos se harán pantalla a pantalla y organizados por los mismos grupos que los requisitos para una mayor claridad.

2.1.5. Fase de desarrollo

Esta es la fase donde se construye el producto que hemos definido en la fase de análisis y diseño. Será la etapa más larga ya que se transforman todos los planes y diseños previos en código, dando como resultado un entregable final funcional.

Esta fase se dividirá en tres iteraciones, ya que la metodología utilizada para la implementación de la aplicación es una metodología ágil. En cada iteración se escogerá el alcance de la iteración, es decir, los requisitos que se van a llevar a cabo, se realizará la implementación de estos, se testeará y se le expondrá al usuario para que pueda validarlo. Con la validación de usuario se obtendrá información de mejoras que deberán de ser aplicadas en las siguientes iteraciones.

Tras finalizar las tres iteraciones, se obtendrá un entregable final funcional que se le proporcionará al cliente para que pueda disfrutar de este.

2.1.6. Redacción de la memoria

Con el objetivo de documentar el desarrollo del proyecto, se va a plasmar toda la información relevante en este documento. No se ha incluido en ninguna fase ya que la documentación se irá haciendo a lo largo de todo el desarrollo del proyecto.

2.1.7. Diagrama de Gantt

En la siguiente tabla podemos ver cuál ha sido la duración en horas de cada una de las tareas que se exponen en el diagrama de Gantt. El diagrama representa la planificación temporal de las tareas del proyecto. Cada celda representa una semana del mes. En la tabla se muestran las horas y en el diagrama el momento en el que se realizó.

Fase	Horas
Motivación	2
Objetivos del proyecto	3
Estructura del documento	2
Definición de un negocio minorista	10
Estudio de los pequeños negocios mixtos	10
Estudio de comerciantes minoristas	10
Estudio de usabilidad y accesibilidad	12
Legislación actual de protección de datos	5
Aplicaciones similares y comparativas	15
Análisis de tecnologías a utilizar	10
Especificación de requisitos	15
Modelos de casos de uso	20
Modelo conceptual	3
Modelos de comportamiento	20
Diagrama de arquitectura del sistema	1
Diseño de la interfaz gráfica	10
Primera iteración	40
Segunda iteración	45
Tercera iteración	30
Creación y modificaciones de la base de datos	15
Redacción de la memoria	40
Total	308

Cuadro 2.1: Distribución de horas por fase

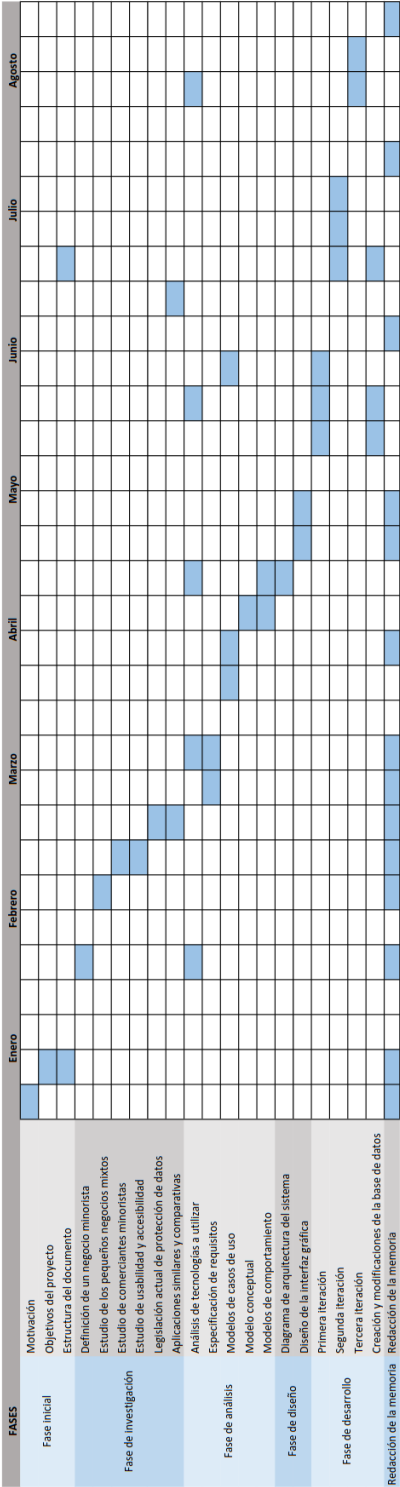


Figura 2.1: Diagrama de Gantt

2.2. Presupuesto

2.2.1. Recursos

En este apartado se va a exponer los recursos hardware y software que se van a utilizar para el desarrollo del proyecto.

2.2.1.1. Hardware

Los componentes hardware que se van a utilizar para llevar a cabo el proyecto son:

- **Ordenador:** MSI Prestige 15 A10SC.

2.2.1.2. Software

Las herramientas software que se van a utilizar para llevar a cabo el proyecto son:

- **Sistema Operativo:** Ubuntu 20.04
- **Lenguaje de programación:** Flutter
- **IDE:** Visual Studio Code
- **Diseño de diagramas UML:** Visual Paradigm
- **Sistema de composición de texto:** LaTeX
- **Editor de texto:** TeXstudio

2.2.2. Costes

En esta sección vamos a analizar el coste total del proyecto. Vamos a distinguir distintos apartados.

2.2.2.1. Licencias

Un tipo de coste que conlleva el desarrollo de un proyecto es la adquisición de las licencias necesarias para la producción del mismo. Las licencias que se van a utilizar en este proyecto son de software libre y por tanto son gratuitas. Esto significa que no supondrán ningún coste adicional. Las licencias que se utilizarán son las siguientes:

- **Ubuntu 20.04:** GNU General Public Licence (GPL).
- **Flutter:** Licencia BSD.
- **Visual Paradigm:** Licencia adquirida por usos académicos.
- **LaTeX:** LaTeX Project Public License (LPPL).

2.2.2.2. Recursos materiales

El único recurso material que se va utilizar para el desarrollo del proyecto es el ordenador personal.

El periodo de amortización común para los ordenadores y equipos informáticos es de 3 a 5 años. Realizaremos la media y utilizaremos un periodo de 4 años para los cálculos. Sabiendo que el equipo costó 1400€, se amortizará 350€ al año. Como la duración del proyecto es de 5 meses, el coste final de los recursos materiales será de 145'83€

2.2.2.3. Recursos humanos

En esta sección se incluyen los gastos por la contratación de personal. Este proyecto solamente lo va a desarrollar una persona, bajo la titulación de programador senior.

En la actualidad, un programador senior recibe una media de 22.000€ anuales. Durante los 5 meses que dura el proyecto, se estima un salario de 9166'66€.

2.2.2.4. Otros

Este apartado engloba costes indirectos como los gastos debidos a la localización para trabajar, los gastos de transporte, conexión a Internet, etc. Este gasto se suele aproximar a un 10 % de los gastos de recursos humanos. Por tanto, la cantidad estipulada para este apartado sería de 916'66€.

2.2.2.5. Total

Descripción	Coste mensual	Coste total
Licencias	0€	0€
Recursos materiales	29'17€	145'83€
Recursos humanos	1833'33€	9166'66€
Otros	183'33€	916'66€
Total	2045,83€	10.229'15€

Cuadro 2.2: Presupuesto total

Capítulo 3

Estado del arte

3.1. Definición de un negocio minorista

La distribución comercial minorista es uno de los sectores de mayor relevancia y dinamismo de nuestra economía. Por tanto, es esencial conocer lo que son, a qué se dedican y cómo se gestionan.

Un negocio minorista es una empresa que vende productos o servicios directamente a consumidores finales. Los minoristas actúan como intermediarios entre los fabricantes o mayoristas y el mercado de consumo. Sin los negocios minoristas, los productos fabricados a grande escala nunca llegarían a ser consumidos por los clientes finales. El comerciante ofrece un asesoramiento especializado sobre los productos y servicio de su tienda, así como un trato personalizado. (García, 2003)

Además, el comerciante tiene la responsabilidad de conocer cuál es la cantidad de producto que debe comprar para asegurarse de vender lo máximo posible y conseguir con ello un beneficio económico. En algunas ocasiones, también se encarga de establecer los precios de los productos de su tienda. Para ello, se debe tener en cuenta el precio base del producto y se le añade un porcentaje para cubrir los gastos y conseguir beneficios. La otra posibilidad es que los precios de venta al mercado de consumo estén fijados en el etiquetado por los fabricantes. En ese caso, el comerciante no deberá modificar el precio previamente estimado.

Podemos clasificar los negocios minoristas según varios criterios, lo cual nos ayudará a entender los tipos que hay y qué funciones tienen cada uno.

Clasificación de un negocio minorista según su forma de venta (Vargas Belmonte, 2023):

- Comercio tradicional: La mercancía no está a disposición del comprador. El dependiente deberá de proporcionar los artículos al cliente. Podemos identificar tres elementos en este tipo de comercio: el almacén (donde se almacenan los productos), el mostrador (donde se atiende a los clientes) y el vendedor (la persona encargada de asesorar al cliente en sus decisiones de compra y proporcionar el producto al comprador). Ejemplo: carnicerías, farmacias, mercerías, etc.
- Comercio de libre servicio: El consumidor tiene libertad para moverse por el espacio de la tienda y elegir los productos a su gusto. En este tipo de comercio el cliente tiene contacto directo con la mercancía, sin intervención del vendedor. Ejemplos: supermercados, autoservicios, etc.
- Comercio mixto: Combina los dos tipos anteriores. El cliente tiene a su disposición la mercancía de la tienda y además el vendedor asesora al comprador sobre sus decisiones dentro de la tienda. Ejemplos: librerías, grandes almacenes, etc.
- Venta sin establecimiento comercial:
 - Venta automática: El comprador selecciona un artículo, lo paga y lo recibe en el momento. Ejemplo: Máquina expendedora.
 - Venta ambulante: Se realiza en rastros o mercadillos.
 - Venta a distancia: El comprador adquiere el producto o servicio a través de un medio de comunicación. Ejemplo: Venta por teléfono, venta por Internet, etc.

Clasificación de un negocio minorista según su agregación (Wikipedia, 2024):

- Comercio independiente o pequeño comercio: Se trata de la tradicional tienda de barrio caracterizada por sus pequeñas dimensiones y por su sistema de venta a través de un mostrador. Funciona de forma autónoma, independiente de otros comercios de su gremio o zona.
- Comercio asociado o comercio integrado: Son tiendas que se localizan en el mismo local, como los pequeños establecimientos de alimentación que se agrupan en mercados. Los centros comerciales surgen del desarrollo de estas pequeñas asociaciones.

- Gran distribución: Grandes empresas que actúan al mismo tiempo como mayoristas y minoristas. Generalmente, son grandes multinacionales. EN este tipo se encuentran los hipermercados, con las marcas blancas.
- Franquicia: Tiendas que forman parte de una cadena. Tienen el mismo nombre e imagen y venden productos similares en diferentes ubicaciones.

Con esta clasificación, hemos podido ver que los negocios minoristas recogen un amplio rango de negocios de distinta escala y distinta gestión interna. En concreto, nos centraremos en el estudio de los pequeños comercios mixtos. Entraremos en más detalle en el próximo punto.

3.2. Estudio de los pequeños negocios mixtos

Estos pequeños negocios se caracterizan por el asesoramiento acerca de los productos o servicios que disponen. Uno de los motivos por lo que los consumidores escogen este tipo de tiendas es este, porque les permite tomar una mejor decisión en sus compras. Además, suelen ser negocios menos masificados donde la relación con el cliente toma importancia. (Vargas Belmonte, 2023)

Con respecto a la variedad de productos que ofrece un pequeño negocio, es esencial que se especialice y se eliminen los artículos que no sean rentables, así como que se adapte a las características y demandas de sus clientes. Hoy en día, existe una gran competencia con las grandes superficies y la compra online. El comerciante debe ser consciente de esa amenaza y debe ser capaz de distinguirse de las grandes empresas ofreciendo a los clientes unos servicios que no puedan encontrar en otros establecimientos.

Los pequeños negocios desarrollan sus propias técnicas de promoción para incentivar y fidelizar clientes. Es su forma de combatir la fuerte competencia entre los negocios y marcas. Estas técnicas se suelen basar en entregar regalos con compras, más cantidad de productos o reducir los precios de estos mediante descuentos. Son acciones que provocan interés en los clientes o consumidores y que les motiva a comprar los productos. Para mantener la fidelidad de los clientes, algunos negocios crean formas de promoción basadas en tarjetas de fidelidad, cupones, concursos, etc. Son técnicas tradicionales y comúnmente utilizadas pero dan los resultados esperados. (Gago Muñiz, 2023)

Las formas de pago también pueden ser algo distintas en este tipo de negocios. Como la relación que se establece con el cliente es fuerte, algunas tiendas fían sus productos a los clientes para que puedan probarlos tranquilamente en sus casas, creando confianza y esperando que sean lo suficientemente honestos para devolverlos o pagarlos tras la prueba.

La publicidad de estas tiendas suele producirse de forma natural cuando un cliente queda satisfecho y la recomienda a su círculo cercano. Por tanto, una vez más, es crucial el trato que el cliente recibe en este tipo de negocios. Además, se pueden incentivar a nuevos clientes a comprar mediante una página web o las redes sociales. Estas son las técnicas más utilizadas en este tipo de negocios. No necesitan grandes campañas publicitarias puesto que el alcance de un pequeño negocio se limita a los ciudadanos de la localidad donde esté situada la tienda.

Los pequeños negocios tienden a situarse en pequeñas localidades donde la actividad comercial tradicional es favorecida. Los ciudadanos de estas localidades tienen menos facilidades para ir a centros comerciales o grandes franquicias, ya que estas se sitúan en las ciudades. Los pequeños negocios ofrecen los servicios que la población necesita para una vida plena, en la comodidad de su localidad. La facilidad e inmediatez de adquisición de los productos visitando estos pequeños negocios es lo que hace que estas tiendas funcionen bien en este tipo de localidades. (Dueñas Nogueras, 2014)

Este proyecto en concreto va destinado al negocio minorista de mi padre. Es una tienda de una pequeña pedanía de Lorca en la que ofrece ropa y calzado para todas las edades, complementos, material de costura y textiles para el hogar. Es una tienda sin empleados, donde el único responsable del funcionamiento de la tienda es mi padre.

3.3. Estudio de comerciantes minoristas

Un estudio realizado por Harvard Business Review demuestra que los emprendedores de 45 años tienen un 85 % más de probabilidades de éxito. Dice que el factor clave de este éxito se debe a la experiencia laboral de los emprendedores de edad media.

Para ser capaz de gestionar y administrar un negocio propio es necesario tener unas habilidades y conocimientos que no solo se adquieren recopilando información de los libros, es necesaria tener cierta experiencia en el sector. (Autonomosyemprendedores, 2024)

De este estudio podemos saber que la edad media de los comerciantes minoristas exitosos será superior a los 40 años. En la actualidad, la brecha digital preocupa al 54 % de los mayores de 50 años y la cifra asciende hasta el 76 % en el caso de las personas de más de 80 años, según datos del Observatorio Sénior de 65YMÁS. Por tanto, podemos concluir que algunos comerciantes minoristas sufrirán la becha digital y tendrán dificultades para aprender a usar las nuevas tecnologías. (BBVA, 2024)

La brecha digital es la desigualdad en el acceso a Internet y las TIC. Hay tres tipos de brecha digital: brecha de acceso (hace referencia a las posibilidades que tienen las personas de acceder a este recurso), brecha de uso (hace referencia a la falta de competencias digitales que impide el manejo de la tecnología) y brecha de calidad de uso (cuando se poseen las competencias digitales para manejarse en Internet, pero no los conocimientos para hacer un buen uso de la red y sacarle el mayor partido posible). La brecha digital se puede dar por la situación geográfica, la diferencia de género, la edad y el nivel económico, entre otros. (Iberdrola, Año de Publicación)

Como se ha comentado anteriormente, los negocios pequeños suelen darse en pequeñas localidades, donde existe un ambiente más rural y, por tanto, un menor acceso a las TIC. Esto sumado a la edad media de los comerciantes minoristas, nos permite saber que va a existir una evidente brecha digital que habrá de tener en cuenta a la hora de desarrollar una aplicación.

3.4. Estudio de usabilidad y accesibilidad

3.4.1. Estudio de usabilidad

La usabilidad se define formalmente como la medida en que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico. (ISO, 2019)

Cuando hablamos de qué tan buena es una herramienta o sistema, pensamos en los siguientes aspectos (Fernández Casado, 2018):

- **Efectividad:** Se refiere a la precisión y completitud con las que los usuarios logran ciertos objetivos.
- **Eficiencia:** Esta dimensión mide los recursos expendidos en relación con la precisión y completitud con las que los usuarios logran esos objetivos. Los recursos pueden incluir el tiempo y el esfuerzo físico o mental.
- **Satisfacción:** Engloba la comodidad y aceptabilidad del trabajo para el usuario, considerando también cómo el diseño afecta a la experiencia del usuario en términos de satisfacción.

Mejorar la usabilidad implica entender las necesidades de los usuarios, sus comportamientos y preferencias, y diseñar los productos de manera que faciliten la interacción entre el usuario y el producto.

3.4.2. Estudio de accesibilidad

La accesibilidad en el contexto de diseño web, tecnologías de la información y comunicación se refiere a la capacidad de los sistemas para ser usados por personas con la más amplia gama de capacidades posible. Formalmente, se define como la usabilidad de un producto, servicio, entorno o instalación por personas con la más amplia gama de capacidades”. (ISO, 2008)

Esta definición abarca varios aspectos clave (W3C, 2023):

- **Diseño universal:** La accesibilidad se logra a menudo a través del diseño universal, que es la práctica de crear productos que sean utilizables por todas las personas, en la mayor medida posible, sin necesidad de adaptación o diseño especializado.

- **Cumplimiento de estándares:** Existen estándares y directrices, como las Pautas de Accesibilidad para el Contenido Web (WCAG), que proporcionan criterios específicos para crear contenido web accesible para personas con discapacidades.
- **Inclusión de capacidades diversas:** La accesibilidad tiene en cuenta una amplia gama de discapacidades.
- **Adaptabilidad:** Capacidad de adaptar y personalizar la interfaz de usuario y el contenido para satisfacer las necesidades individuales de los usuarios.
- **Compatibilidad con Ayudas Técnicas:** Los productos o servicios accesibles deben ser compatibles con las ayudas técnicas utilizadas por las personas con discapacidades, como lectores de pantalla, sistemas de amplificación de sonido, teclados especializados, entre otros.

La accesibilidad se fundamenta en el principio de proporcionar un acceso igualitario a la información y a las funcionalidades de un sistema o servicio. Este compromiso no solo facilita la interacción de las personas con discapacidades, sino que también realza la experiencia general de todos los usuarios. En otras palabras, adoptar medidas de accesibilidad no se limita a cumplir con un estándar ético de inclusión; es una práctica que mejora la interfaz y la usabilidad de manera integral, lo que resulta en un diseño más efectivo y universal. (Fernández Casado, 2021)

3.5. Legislación actual de protección de datos

La ley de protección de datos existe para proteger y controlar los datos personales de los usuarios, un derecho fundamental que tienen todas las personas. Esta ley evita que nuestros datos personales sean usados para vulnerar nuestra intimidad u otros derechos fundamentales o libertades. En España esta ley lleva vigente desde el año 2000. (del Estado, 2018)

Dentro de los datos personales hay distintos niveles de seguridad: bajo, medio y alto. Cada uno de estos niveles tiene distintas medidas de seguridad. Las principales medidas de la ley de protección de datos en España son:

- Solo se deben recoger aquellos datos necesarios para el funcionamiento de la aplicación.
- Se debe avisar a la persona cuáles son los datos que van a ser recogidos y con qué fin van a usarse.
- La persona tiene derecho a dar y deshacer el consentimiento de esos datos si existe una causa justificada.
- Los datos de medio y alto nivel no deben ser recogidos a menos que sea estrictamente necesario. Información como la religión o ideología son datos que no deberían de pedirse.
- La persona que interactúe con datos personales protegidos bajo esta ley deberá de cumplir el secreto profesional.
- Se deben tomar medidas técnicas y organizativas para garantizar la seguridad de los datos en todo momento.

En nuestro caso concreto, los datos que vamos a pedir a los clientes son únicamente tres campos: el nombre, los apellidos y un número de teléfono, este último de carácter opcional. Son los datos estrictamente necesarios para identificar un cliente. No se pedirán datos innecesarios, como bien postula la ley de protección de datos.

Puesto que los clientes pueden tomar prestados artículos de la tienda, será necesario un número de teléfono para llamar en caso de que se olviden de devolver o comprar dichos artículos. Si el cliente no quiere dar el número de teléfono o no dispone de uno de ellos, bastará con el nombre y los apellidos.

Estos datos se almacenarán en una base de datos protegida de manera que se pueda garantizar la seguridad de dichos datos.

3.6. Aplicaciones similares y comparativas

Una de las principales razones para desarrollar esta aplicación es la inexistencia de aplicaciones dedicadas a solucionar los problemas de los pequeños negocios minoristas. Sin embargo, analizaremos a continuación las características de dos de las aplicaciones de gestión de negocios minoristas que hay actualmente en el mercado, para así observar la necesidad de desarrollar una nueva aplicación con unos requisitos distintos.

La aplicación más similar que he podido encontrar es myGESTIÓN (myGestión, 2024), el resto de aplicaciones incorporan un punto de venta online que permite a los clientes comprar desde sus casas (Clover, 2024). Este punto de venta no sería útil en localizaciones donde el conocimiento tecnológico medio de la población es bajo. Por tanto, se trata de una funcionalidad que complica la aplicación y hace que sea más difícil de usar. Buscamos conseguir una aplicación sencilla que cumpla las necesidades básicas de gestión, siendo así más usable y accesible. Por lo que, aquellas funciones extra que implementan las aplicaciones del mercado actual para negocios más grandes, serán un inconveniente a la hora de establecer un entorno de uso sencillo. Además, un punto importante que no incorporan las aplicaciones de gestión actuales es el sistema de préstamos. Este es un sistema comúnmente utilizado en pequeñas localidades que sí se tendrá en cuenta en el desarrollo de nuestra aplicación.

3.6.1. myGESTIÓN

myGESTION ofrece una variedad de funcionalidades, incluyendo la gestión de clientes, de artículos y de pedidos de clientes, así como la gestión de ventas, albaranes, facturas y empleados. También incluye un cuadro de mando. Sin embargo, presenta algunos problemas, como la falta de un sistema de préstamos, el exceso de funcionalidades no aplicables a este tipo de negocios y que es un aplicación de pago.

A continuación se muestra una ilustración que permite ver cómo es la interfaz gráfica de myGESTIÓN.



Figura 3.1: Interfaz gráfica de myGestión

3.6.2. Clover

Clover se centra en la gestión de empleados, clientes y puntos de venta online, además de proporcionar supervisión de inventario. Al igual que myGESTION, Clover enfrenta problemas relacionados con la falta de un sistema de préstamos y tiene funcionalidades que no son necesarias. Además, también es una aplicación de pago.

A continuación se muestra una ilustración que permite ver cómo es la interfaz gráfica de Clover.

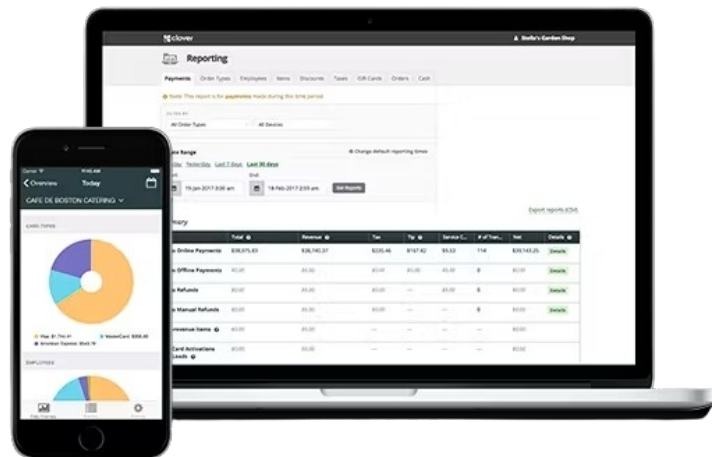


Figura 3.2: Interfaz gráfica de Clover

Nombre de la app	Características principales	Problemas	Plataformas disponibles
myGESTIÓN	<ul style="list-style-type: none"> ■ Gestión de clientes. ■ Gestión de artículos. ■ Gestión de pedidos de clientes. ■ Gestión de ventas, albaranes, facturas. ■ Gestión de empleados y fichaje. ■ Cuadro de mando. 	<ul style="list-style-type: none"> ■ No tiene en cuenta el sistema de préstamos. ■ Excedente de funcionalidades, no son necesarios pedidos de clientes ni empleados. ■ De pago. 	<ul style="list-style-type: none"> ■ Android
Clover	<ul style="list-style-type: none"> ■ Supervisión de inventario. ■ Gestión de empleados. ■ Punto de venta online. ■ Gestión de clientes. 	<ul style="list-style-type: none"> ■ No tiene en cuenta el sistema de préstamos. ■ No necesita gestionar empleados. ■ No es necesario un punto de venta online, los clientes no lo usarían. ■ De pago. 	<ul style="list-style-type: none"> ■ Android ■ iOS

Cuadro 3.1: Descripción de la tabla

Capítulo 4

Análisis de tecnologías a utilizar

4.1. Dispositivo destino para la aplicación

La aplicación está pensada para ser ejecutada en un dispositivo tipo **tablet**. De esta forma conseguimos una pantalla lo suficientemente grande como para visualizar y utilizar la aplicación cómodamente. Si fuera necesario, una tablet también permite conectar un teclado para introducir datos más rápidamente o simular el funcionamiento de un ordenador. Además es un dispositivo altamente portable, lo que permite usar la aplicación en cualquier lugar. Por todos estos motivos, se trata del dispositivo más apropiado.

4.2. Tecnología de documentación

Para documentar se va a utilizar la herramienta de **LaTeX** ya que es bastante útil a la hora de mantener un documento bien estructurado y con el mismo formato. Además, es una herramienta que se suele utilizar para realizar documentos técnicos y es interesante aprenderla para futuros proyectos.

4.2.1. Descripción de LaTeX

LaTeX es un sistema de composición de textos de alta calidad, muy utilizado para la creación de documentos científicos y técnicos debido a su gran capacidad para manejar fórmulas matemáticas, referencias cruzadas y bibliografías, entre otras. Permite establecer un buen formato y dar una buena estructura al documento de forma sencilla. (LaTeX, 2024)

Algunas de sus principales características es:

- **Documentos estructurados:** Permite estructurar documentos de manera lógica con capítulos, secciones y subsecciones, lo que resulta ideal para documentos largos y complejos.
- **Gestión de referencias y citas:** Integra sistemas para la gestión automática de citas y bibliografías como BibTeX y Biber.
- **Sistema de tipografía:** LaTeX permite la tipografía de calidad editorial con características profesionales.
- **Licencia de software libre:** Es un software de código abierto, lo que significa que es gratuito y se puede modificar y distribuir.



Figura 4.1: Logo LaTeX

4.3. Tecnología para la realización de los diagramas de casos de uso

Para realizar los diagramas de casos de uso se ha utilizado **Visual Paradigm**. He escogido esta herramienta ya que hemos trabajado anteriormente con ella, conozco su funcionamiento y es muy útil. Se requiere de una licencia para utilizarla, pero la UGR la proporciona gratuitamente a sus estudiantes.

4.3.1. Descripción de Visual Paradigm

Visual Paradigm es una herramienta de modelado de software. La plataforma ofrece modelado de negocio completo, soporte para una amplia gama de diagramas, automatización de flujos de trabajo, y colaboración en equipo. Incluye un repositorio en la nube y facilita la creación de prototipos, maquetas y wireframes. (Paradigm, 2024)

Algunos de los diagramas más destacados que ofrece son:

- Diagramas de clase UML.
- Diagramas de casos de uso UML.
- Diagramas de secuencia UML.
- Diagramas de Entidad-Relación (ER).
- Diagramas de arquitectura de software.



Figura 4.2: Logo Visual Paradigm

4.4. Tecnología para el diseño de los bocetos

Para el diseño de los bocetos he utilizado **Goodnotes**, una aplicación para escribir en el Ipad. Me ha parecido la forma más sencilla y rápida de hacer unos bocetos. Además, al ser dibujado, hay total libertad para expresar el diseño tal y como se desea hacer. Es un diseño minimalista, pero se puede observar de una manera muy visual cuál será la interfaz de usuario de la aplicación a desarrollar.

4.4.1. Descripción de Goodnotes

Goodnotes es una aplicación de toma de notas digital que se destaca por combinar tecnología de inteligencia artificial con funcionalidades tradicionales para crear una experiencia de usuario avanzada y versátil. (GoodNotes, 2024)

- **Herramientas de escritura y dibujo flexibles:** Permite combinar texto escrito a mano y tecleado en una sola página.
- **Búsqueda potente:** Permite buscar y encontrar rápidamente texto escrito a mano.
- **Personalización y organización:** Ofrece opciones para personalizar plantillas.

- **Gestos intuitivos con el lápiz:** Incluye gestos como "Scribble to Erase" para borrar rápidamente.



Figura 4.3: Logo GoodNotes

4.5. Metodología de desarrollo de la aplicación

Se utilizará una **metodología iterativa**. El desarrollo de la aplicación se dividirá en varias iteraciones, cada una con sus requisitos correspondientes y su entregable final. De esta forma, se realizará una revisión constante de aquello que se está construyendo con el objetivo de no desviarnos de las necesidades del usuario y consiguiendo una mayor adaptación a lo largo del proyecto.

4.5.1. Descripción de metodología iterativa

Una metodología iterativa es un enfoque de desarrollo de proyectos en ciclos repetitivos, permitiendo la adaptación y mejora continua basada en el feedback después de cada iteración. Es especialmente útil en el desarrollo de software para responder a cambios y resolver problemas rápidamente. (Highsmith, 2004)

Las principales características son:

- **Iteraciones:** El desarrollo se divide en iteraciones, cada una resultando en una versión mejorada del producto, permitiendo ajustes regulares según sea necesario.
- **Feedback continuo:** Al final de cada iteración se evalúa el progreso y se recopila feedback para guiar el desarrollo futuro.

- **Flexibilidad en la planificación:** Permite ajustes en el plan basados en el aprendizaje del proyecto y los cambios de requisitos.
- **Entregables frecuentes:** Proporciona versiones funcionales del producto en etapas tempranas, lo que ayuda a verificar la viabilidad y la dirección del proyecto.
- **Mejora continua:** La naturaleza iterativa fomenta la revisión y mejora constante del producto.

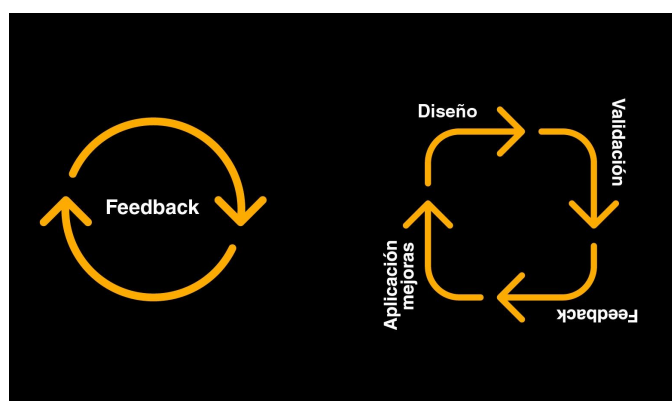


Figura 4.4: Metodología iterativa

4.6. Tecnología de soporte de la base de datos

La tecnología utilizada para la creación y utilización de la base de datos del proyecto es **Supabase**. He escogido esta alternativa ya que proporciona una opción de base de datos en línea, que evita tener que instalar un propio servidor. Además, las bases de datos son relacionales lo que permite una mejor ordenación y acceso de la información.

4.6.1. Descripción de Supabase

Supabase es una plataforma de desarrollo backend como servicio diseñada para ayudar a los desarrolladores a construir aplicaciones rápidamente. Es una alternativa de código abierto a Firebase. Sus principales características por las que destaca son: (Supabase, 2024)

- **Base de datos PostgreSQL:** Es una base de datos relacional, robusta y escalable.

- **API RESTful generada automáticamente:** Se genera automáticamente una API basada en la estructura de la base de datos. Esto permite interactuar con la información de forma sencilla.
- **Autenticación y autorización:** Incluye un sistema de autenticación completo que permite iniciar sesión de múltiples formas. Además, proporciona un control de acceso a las tablas para garantizar la seguridad.
- **Subscripciones en tiempo real:** Permite agregar funcionalidades en tiempo real a sus aplicaciones a través de subscripciones a eventos.
- **Integraciones y extensibilidad:** Se integra fácilmente con otras herramientas y servicios populares.



Figura 4.5: Logo Supabase

4.7. Framework de desarrollo

El framework de desarrollo que he escogido para implementar la aplicación ha sido **Flutter**. El principal motivo por el que he escogido esta tecnología es por ser multiplataforma, lo que permite trasladar la aplicación a android, ios o web. Esto amplía la compatibilidad de la app con diferentes dispositivos. Además, tiene multitud de widgets que te permiten desarrollar una aplicación de forma más eficiente. El lenguaje de programación utilizado es **Dart**.

4.7.1. Descripción de Flutter

Flutter permite construir aplicaciones nativas de alta calidad para múltiples plataformas utilizando una única base de código. **Dart** es el lenguaje de programación utilizado por Flutter. Es un lenguaje sencillo de aprender y usar. Las principales características de Flutter son: (CeroIdeas, 2024)

- **UI Nativa y rápida:** Flutter utiliza su propio motor de renderizado para dibujar widgets directamente en la pantalla. Con esto se logra una sensación nativa en cada plataforma.

- **Desarrollo con un solo código base:** El código se escribe en Dart y se adapta a las distintas plataformas.
- **Hot reload:** Permite ver los cambios en el código de inmediato, sin tener que reiniciar la ejecución.
- **Extensa biblioteca de widgets:** Ofrece una amplia variedad de widgets personalizables para construir interfaces de usuario complejas y atractivas.
- **Alto rendimiento:** Flutter ofrece un rendimiento cercano al nativo.



Figura 4.6: Logo Flutter

Capítulo 5

Análisis de la aplicación

5.1. Especificación de requisitos

Dentro del desarrollo de un proyecto, la fase de especificación de requisitos cumple un rol fundamental, ya que será aquí donde se establezcan las características esenciales y las expectativas que debe cumplir el sistema.

Esta sección está dedicada a la identificación y descripción detallada de los requisitos. Para conseguir este fin, es necesario preguntar al cliente y stakeholders sobre las necesidades o limitaciones que esperan del software.

Cada requisito se identificará de forma única mediante un código que indica el tipo de requisito y su número correspondiente. Mediante esta codificación podremos hacer referencia a los requisitos de forma sencilla a lo largo de este documento.

5.1.1. Requisitos funcionales

Los requisitos funcionales son aquellos que definen las tareas específicas que el sistema debe ser capaz de realizar. Podemos ordenar estos requisitos en 6 grupos según la función que habilitan:

- **Grupo 1:** Apertura y cierre de sesión - En este grupo se encuentran los requisitos funcionales relacionados con el inicio y cierre de sesión.
- **Grupo 2:** Gestión de artículos - En este grupo se encuentran los requisitos funcionales relacionados con los artículos de la tienda.
- **Grupo 3:** Gestión de inventario - En este grupo se encuentran los requisitos funcionales encargados de realizar la actualización del inventario.

- **Grupo 4:** Gestión de clientes - En este grupo se encuentran los requisitos funcionales relacionados con los clientes de la tienda.
- **Grupo 5:** Gestión de movimientos - En este grupo se encuentran los requisitos funcionales relacionados con las ventas, préstamos y devoluciones.
- **Grupo 6:** Gestión de resúmenes y gráficas - En este grupo se encuentran los requisitos funcionales relacionados con la visualización y realización de gráficas.

A continuación se especificarán los requisitos funcionales del proyecto clasificándolos en las secciones anteriormente mencionadas.

5.1.1.1. Apertura y cierre de sesión

RF1	Inicio de sesión
Prioridad	Alta
Descripción	El sistema debe permitir al usuario iniciar sesión de forma segura.
Criterios de aceptación	<ol style="list-style-type: none"> 1. La pantalla de inicio de sesión debe pedir el nombre de usuario y la contraseña. 2. Verificar que los campos no estén vacíos. 3. Verificar que la información introducida es correcta y autentica al usuario. 4. Si la información es incorrecta, mostrar un mensaje de error. 5. Tras el inicio de sesión, redirigir al usuario a la pantalla principal.

Cuadro 5.1: RF1

RF2	Cierre de sesión
Prioridad	Alta
Descripción	El sistema debe permitir al usuario cerrar sesión de forma segura.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para cerrar la sesión. 2. Tras el cierre de sesión, se debe volver a la pantalla de inicio de sesión.

Cuadro 5.2: RF2

5.1.1.2. Gestión de artículos

RF3	Introducción de un nuevo artículo
Prioridad	Alta
Descripción	El sistema debe permitir registrar un nuevo artículo al negocio.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para añadir un nuevo artículo. 2. Verificar que los campos obligatorios no están vacíos. 3. Verificar que la información introducida tenga un formato correcto. 4. Tras guardar un nuevo artículo, se debe actualizar la lista de artículos existente.

Cuadro 5.3: RF3

RF4	Edición de un artículo existente
Prioridad	Alta
Descripción	El sistema debe permitir editar los datos de un artículo existente.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para editar un artículo existente. 2. Verificar que la nueva información introducida tenga un formato correcto. 3. Tras guardar los cambios, se deben actualizar los campos en la base de datos.

Cuadro 5.4: RF4

RF5	Eliminación de un artículo
Prioridad	Alta
Descripción	El sistema debe permitir eliminar un artículo existente.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para eliminar un artículo existente. 2. Mandar un mensaje de confirmación antes de la eliminación. 3. Verificar que el artículo no esté vinculado a movimientos existentes en la base de datos. 4. Tras eliminar el artículo correspondiente, actualizar la lista de artículos existentes.

Cuadro 5.5: RF5

RF6	Visualización de los datos de un artículo
Prioridad	Alta
Descripción	El sistema debe permitir visualizar los datos de un artículo.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para visualizar un artículo existente. 2. El sistema debe mostrar toda la información almacenada sobre un artículo.

Cuadro 5.6: RF6

RF7	Búsqueda de un artículo por nombre
Prioridad	Alta
Descripción	El sistema debe permitir buscar un artículo por nombre.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para buscar un artículo existente. 2. El usuario debe introducir el nombre correcto que identifica el artículo. 3. El sistema debe mostrar al usuario el/los artículos que están relacionados con la búsqueda.

Cuadro 5.7: RF7

RF8	Categorización de un artículo
Prioridad	Alta
Descripción	El sistema debe ser capaz de filtrar los artículos según su categoría.
Criterios de aceptación	<ol style="list-style-type: none"> 1. El usuario debe establecer la categoría del artículo de forma correcta. 2. El sistema debe ser capaz de mostrar solo los artículos de la categoría especificada.

Cuadro 5.8: RF8

RF9	Visualización de la lista de artículos existentes
Prioridad	Alta
Descripción	El sistema debe ser capaz de mostrar una lista con los artículos del negocio.
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe ser capaz de mostrar los artículos existentes en la base de datos en forma de lista.

Cuadro 5.9: RF9

5.1.1.3. Gestión de inventario

RF10	Actualización del inventario de forma automática
Prioridad	Alta
Descripción	El sistema debe ser capaz de actualizar el inventario de los artículos del negocio en tiempo real.
Criterios de aceptación	<ol style="list-style-type: none">1. El sistema debe ser capaz de actualizar las cantidades de los artículos existentes en base a los movimientos (ventas, devoluciones o préstamos) del negocio.

Cuadro 5.10: RF10

RF11	Visualización de lista de renovación de artículos
Prioridad	Media
Descripción	El sistema debe ser capaz de realizar una lista con los artículos que lleguen a una cantidad crítica para indicar al comerciante que debe comprar más stock.
Criterios de aceptación	<ol style="list-style-type: none">1. Debe existir una opción visible para visualizar la lista.2. El sistema debe introducir el artículo en la lista cada vez que la cantidad actual sea igual o inferior a la cantidad mínima establecida por el comerciante.

Cuadro 5.11: RF11

5.1.1.4. Gestión de clientes

RF12	Registro de un nuevo cliente habitual
Prioridad	Alta
Descripción	El sistema debe permitir añadir un nuevo cliente al sistema.
Criterios de aceptación	<ol style="list-style-type: none">1. Debe existir una opción visible para añadir un nuevo cliente.2. Verificar que los campos obligatorios no estén vacíos.3. Verificar que la información proporcionada tenga un formato correcto.4. Añadir el nuevo cliente a la base de datos y actualizar la lista de clientes existentes.

Cuadro 5.12: RF12

RF13	Edición de los datos de un cliente existente
Prioridad	Alta
Descripción	El sistema debe permitir editar los datos de un cliente existente.
Criterios de aceptación	<ol style="list-style-type: none">1. Debe existir una opción visible para editar los datos de un cliente.2. Verificar que la nueva información proporcionada tenga un formato correcto.3. Tras confirmar los cambios, editar los campos modificados en la base de datos.

Cuadro 5.13: RF13

RF14	Eliminación de un cliente existente
Prioridad	Alta
Descripción	El sistema debe permitir eliminar un cliente existente.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para eliminar un cliente. 2. Mandar un mensaje de confirmación antes de la eliminación. 3. Verificar que el cliente no esté vinculado a ningún movimiento activo. 4. Tras eliminar al cliente, actualizar la lista de clientes existentes.

Cuadro 5.14: RF14

RF15	Visualización de los datos de un cliente
Prioridad	Alta
Descripción	El sistema debe permitir visualizar los datos de un cliente existente.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para visualizar los datos de un cliente. 2. El sistema debe mostrar toda la información almacenada sobre un cliente.

Cuadro 5.15: RF15

RF16	Visualización de la lista de clientes existentes
Prioridad	Alta
Descripción	El sistema debe ser capaz de mostrar una lista con los clientes habituales del negocio.
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe ser capaz de mostrar los clientes registrados en la base de datos en forma de lista.

Cuadro 5.16: RF16

RF17	Búsqueda de un cliente por nombre
Prioridad	Alta
Descripción	El sistema debe ser capaz de buscar un cliente habitual por su nombre.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para buscar un cliente. 2. El usuario debe introducir el nombre que identifica al cliente. 3. El sistema debe mostrar a el/los clientes que estén relacionados con la búsqueda.

Cuadro 5.17: RF17

RF18	Filtrado de clientes con préstamos
Prioridad	Alta
Descripción	El sistema debe ser capaz de filtrar la lista completa de clientes registrados, dejando solo aquellos clientes que tengan préstamos.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para filtrar clientes por préstamos. 2. El sistema debe ser capaz de mostrar una lista de clientes que tienen artículos prestados del negocio.

Cuadro 5.18: RF18

5.1.1.5. Gestión de movimientos

RF19	Introducción de una nueva venta
Prioridad	Alta
Descripción	El sistema debe ser capaz añadir una nueva venta al sistema.
Criterios de aceptación	<ol style="list-style-type: none">1. Debe existir una opción visible para añadir una nueva venta.2. El comerciante deberá añadir los artículos y especificar la cantidad de estos.3. El sistema deberá calcular los precios parciales y el precio total de la venta.4. Se debe especificar el método de pago antes de la confirmación de la venta.5. Se puede asignar un cliente a la venta de forma opcional.6. Actualizar el inventario tras la confirmación de la venta.7. Añadir el nuevo movimiento a la base de datos y actualizar la lista de movimientos existente.

Cuadro 5.19: RF19

RF20	Introducción de un nuevo préstamo
Prioridad	Alta
Descripción	El sistema debe ser capaz añadir un nuevo préstamo.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para añadir un nuevo préstamo. 2. El comerciante deberá añadir los artículos y especificar la cantidad de estos. 3. El sistema deberá calcular los precios parciales y el precio total del préstamo. 4. Se debe asignar un cliente de forma obligatoria para vincularlo con el préstamo. 5. Actualizar el inventario tras la confirmación del préstamo. 6. Añadir el nuevo movimiento a la base de datos y actualizar la lista de movimientos existente.

Cuadro 5.20: RF20

RF21	Introducción de una devolución
Prioridad	Alta
Descripción	El sistema debe ser capaz de registrar una devolución de una venta.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Tras encontrar la venta que queremos devolver en la lista de movimientos existentes, debe existir una opción visible para realizar la devolución. 2. Indicar los productos que se van a devolver (puede ser la venta integra o una parte de esta). 3. Indicar el método de devolución de dinero que se va a utilizar (devolución en metálico o dinero a favor en próximas compras). 4. Actualizar el inventario tras la devolución. 5. Añadir la devolución a la base de datos y actualizar la lista de movimientos.

Cuadro 5.21: RF21

RF22	Eliminación de un movimiento
Prioridad	Alta
Descripción	El sistema debe ser capaz eliminar un movimiento (venta, préstamo o devolución).
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para eliminar el movimiento deseado. 2. Mandar un mensaje de confirmación antes de eliminar el movimiento. 3. Tras eliminar el movimiento, actualizar la base de datos y la lista de movimientos existentes.

Cuadro 5.22: RF22

RF23	Visualización de los datos de un movimiento
Prioridad	Alta
Descripción	El sistema debe ser capaz mostrar los datos de un movimiento.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para visualizar los datos del movimiento deseado. 2. El sistema debe ser capaz de mostrar toda la información almacenada de un movimiento.

Cuadro 5.23: RF23

RF24	Visualización de la lista de movimientos existentes
Prioridad	Alta
Descripción	El sistema debe ser capaz de mostrar una lista con los movimientos efectuados en el negocio.
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe ser capaz de mostrar los movimientos registrados en la base de datos en forma de lista.

Cuadro 5.24: RF24

RF25	Filtrado de los movimientos según su tipo
Prioridad	Alta
Descripción	El sistema debe ser capaz de filtrar los movimientos efectuados en el negocio según si son ventas, préstamos o devoluciones.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para filtrar los movimientos por tipo. 2. El sistema debe ser capaz de filtrar la lista de movimientos existentes, dejando solo aquellos movimientos que nos interesen, ya sean ventas, préstamos o devoluciones.

Cuadro 5.25: RF25

RF26	Búsqueda de movimientos cliente
Prioridad	Alta
Descripción	El sistema debe ser capaz de buscar un movimiento específico por cliente asociado.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para buscar el movimiento deseado. 2. El usuario debe introducir el nombre del cliente a buscar de forma correcta. 3. El sistema debe de mostrar los movimientos relacionados con la búsqueda.

Cuadro 5.26: RF26

RF27	Generación de una compra a partir de un préstamo
Prioridad	Alta
Descripción	El sistema debe ser capaz de generar un movimiento de venta a partir de un préstamo.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para comprar los artículos deseados de un préstamo. 2. El usuario debe marcar aquellos artículos que el cliente desea comprar. 3. El sistema debe eliminar el movimiento de tipo préstamo y generar a partir de los artículos indicados un movimiento de tipo venta.

Cuadro 5.27: RF27

5.1.1.6. Gestión de resúmenes y gráficas

RF28	Visualización de la caja diaria
Prioridad	Alta
Descripción	El sistema debe ser capaz de mostrar en tiempo real la cantidad de dinero que se gana o se pierde en base a los movimientos del negocio que se efectúan durante el día.
Criterios de aceptación	<ol style="list-style-type: none"> 1. El sistema debe de mostrar la cantidad total ganada o perdida hasta el momento realizando la sumatoria de las cantidades de los movimientos efectuados en ese mismo día.

Cuadro 5.28: RF28

RF29	Visualización de gráficas
Prioridad	Media
Descripción	El sistema debe ser capaz de mostrar gráficas temporales que reflejen el progreso económico del negocio durante un mes o año.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Debe existir una opción visible para ver las gráficas. 2. El usuario debe elegir si ver la gráfica mensual o anual. 3. El sistema debe ser capaz de mostrar una gráfica que refleje el progreso económico del negocio en base a los movimientos efectuados durante el mes o año.

Cuadro 5.29: RF29

5.1.2. Requisitos no funcionales

Los requisitos no funcionales son los que aportan calidad al sistema. Los requisitos no funcionales que deberemos de tener en cuenta a la hora de desarrollar la aplicación son los siguientes.

RNF1	Seguridad
Prioridad	Alta
Descripción	El sistema debe proteger los datos que el encargado introduzca, especialmente los datos de los clientes.

Cuadro 5.30: RNF1

RNF2	Compatibilidad
Prioridad	Baja
Descripción	El software debe estar disponible en varios sistemas operativos (Android e IOS).

Cuadro 5.31: RNF2

RNF3	Usabilidad
Prioridad	Media
Descripción	El sistema debe ser sencillo para que los usuarios puedan interactuar con este sin problema.

Cuadro 5.32: RNF3

RNF4	Accesibilidad
Prioridad	Media
Descripción	El sistema debe ser utilizable por el mayor número de personas, incluidas las personas con discapacidades.

Cuadro 5.33: RNF4

5.2. Modelos de caso de uso

5.2.1. Diagrama de casos de uso

5.2.1.1. Actores

La aplicación va dirigida a facilitar y automatizar el trabajo de un encargado de un pequeño negocio. Por tanto, el principal actor será dicho encargado. Podemos diferenciar dos actores que interaccionarán con el sistema:

- **Encargado:** El comerciante propietario de la tienda minorista.
- **Cliente:** La persona que compra en el negocio y se registra en el sistema.

5.2.1.2. Diagramas de casos de uso

En esta sección se mostrarán los distintos casos de uso divididos en paquetes para conseguir una visión más clara y ordenada de estos.

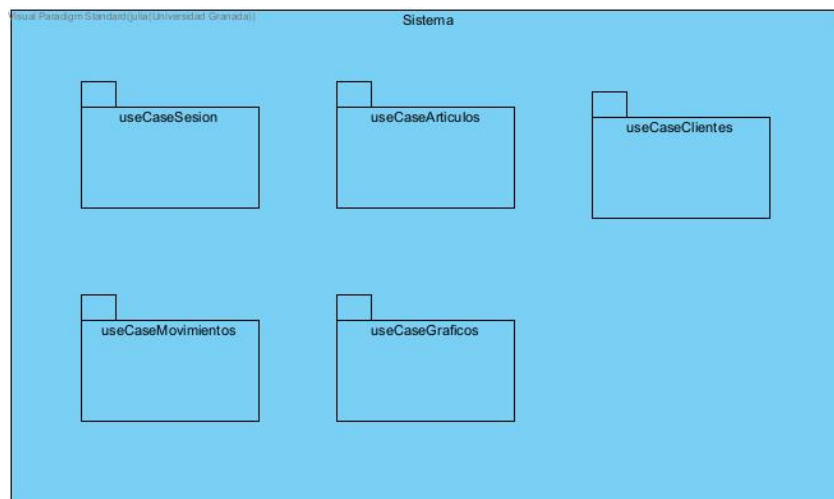


Figura 5.1: Diagrama de paquetes de casos de uso

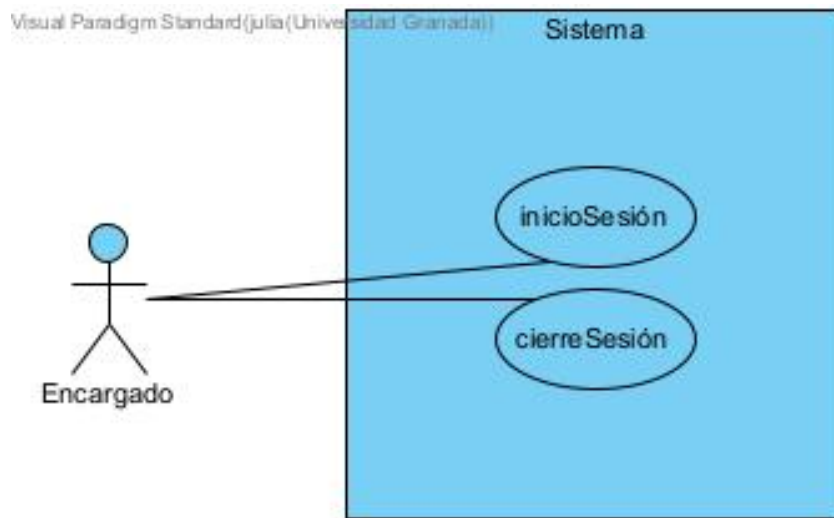


Figura 5.2: Casos de uso del inicio y cierre de sesión

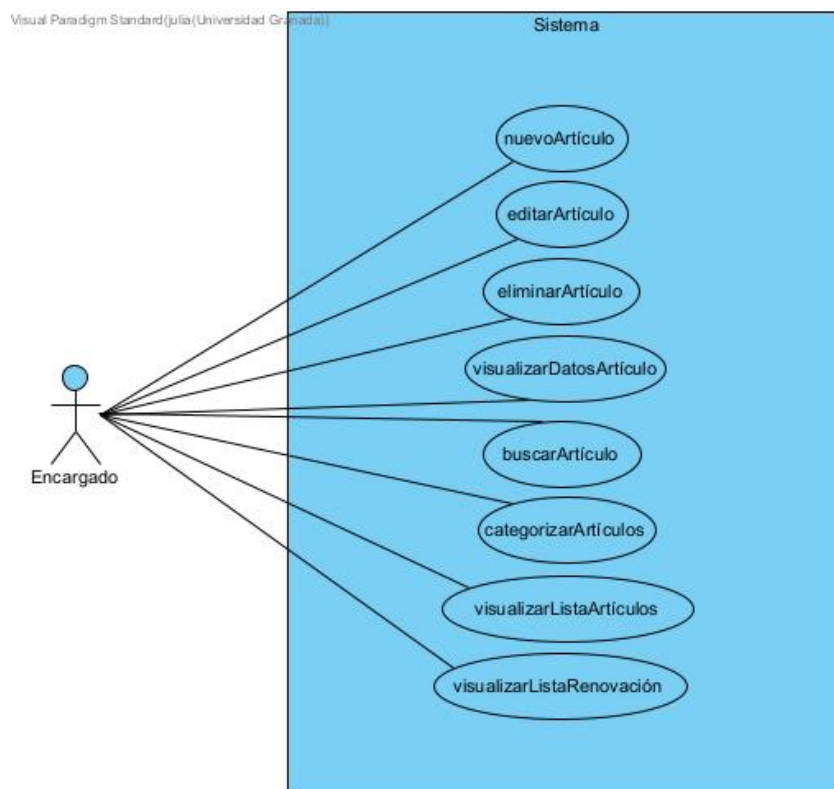


Figura 5.3: Casos de uso de la gestión de artículos

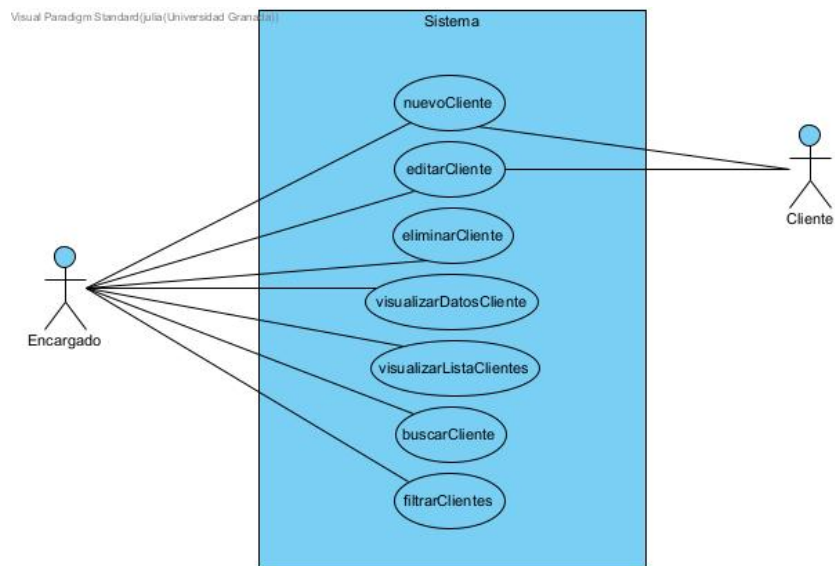


Figura 5.4: Casos de uso de la gestión de clientes

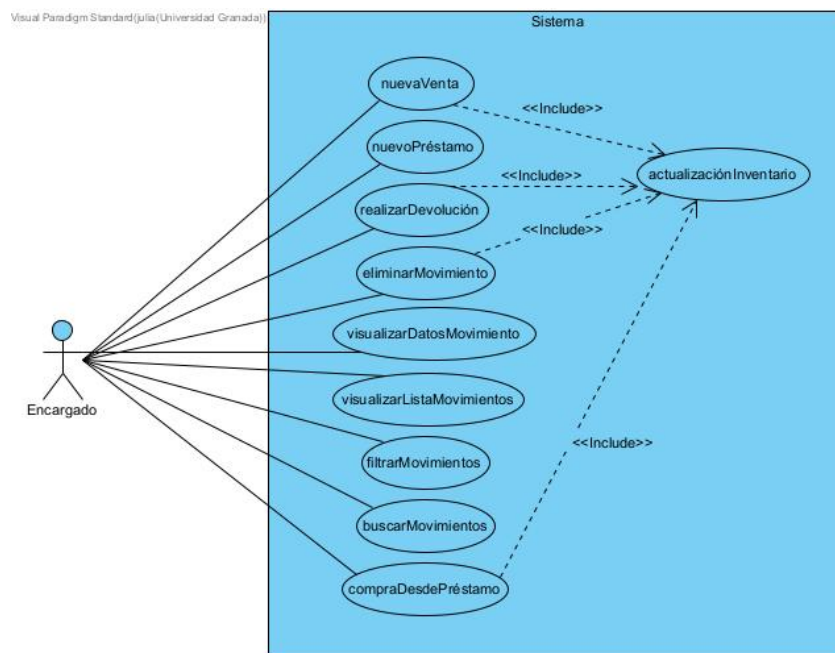


Figura 5.5: Casos de uso de la gestión de movimientos

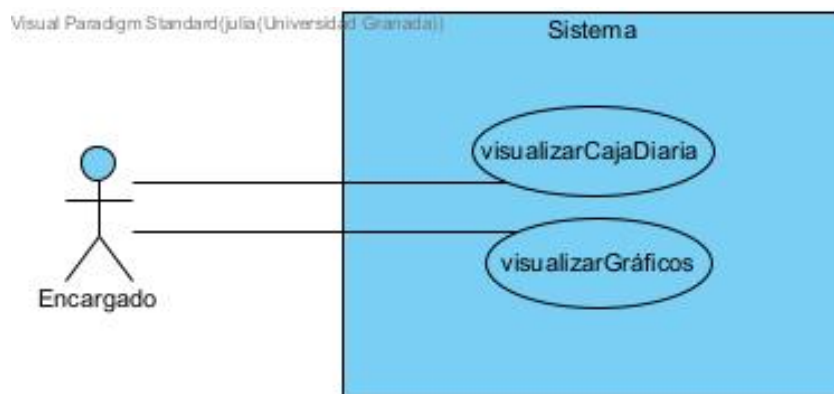


Figura 5.6: Casos de uso de la gestión de gráficos y resúmenes

5.2.2. Descripción de casos de uso

En este apartado podremos ver una descripción detallada de los casos de uso de la aplicación, los cuales representan las principales interacciones entre el usuario y el sistema. A raíz de estos casos de uso, podremos identificar las principales funciones de la aplicación.

Caso de Uso	inicioSesión		CU1
Actores	Encargado		
Tipo	Obligatorio		
Referencias	RF1		
Precondición	El encargado debe estar registrado en el sistema.		
Postcondición	El encargado entra en el sistema.		
Autor	Julia Cano		
Fecha	02/03/2024	Versión	1.0

Propósito	El encargado debe ser capaz de acceder al sistema.
Resumen	El encargado introduce sus credenciales y entra en el sistema.

Curso normal			
1.	Encargado: Introduce su nombre de usuario y su contraseña.	2.	Verifica que las credenciales sean correctas
3.	Encargado: Entra en el sistema		

Cursos alternos	
2a	Las credenciales no son correctas.

Cuadro 5.34: Caso de uso - Inicio de sesión

Caso de Uso	cierreSesión	CU2
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF2	
Precondición	El encargado debe estar dentro del sistema.	
Postcondición	El encargado sale del sistema.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de cerrar sesión.
Resumen	El encargado pulsa la opción de cierre de sesión y sale del sistema.

Curso normal			
1.	Encargado: Solicita cierre de sesión.	2.	Cerrar sesión.
3.	Encargado: Sale del sistema.		

Cursos alternos	

Cuadro 5.35: Caso de uso - Cierre de sesión

Para seguir viendo la explicación de los casos de uso, consultar el Apéndice A.

5.3. Modelo conceptual

En esta sección se muestra el modelo conceptual del sistema, donde podemos ver cómo se relacionan las clases y que información almacenan.

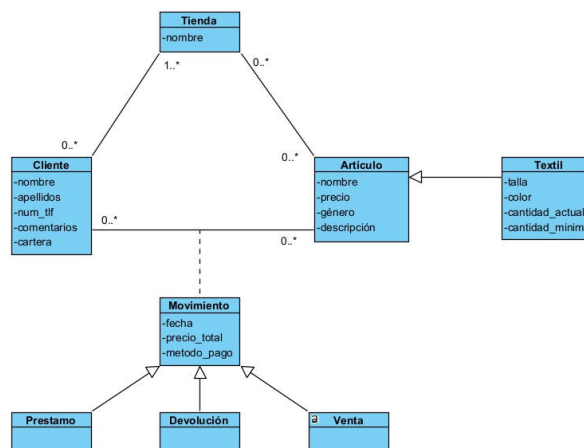


Figura 5.7: Modelo conceptual

En este diagrama podemos ver los principales conceptos que existirán en nuestra aplicación:

- **Tienda:** Se trata de la entidad padre de la aplicación. La tienda es la que tiene clientes, artículos, movimientos, etc.
- **Cliente:** Representa las personas que habitúan a ir a la tienda para comprar artículos, generando movimientos. De cada cliente almacenamos el nombre, el número de teléfono, un apartado de comentarios y una cartera, la cual permite tener dinero en la tienda.
- **Artículo:** Representa cada producto disponible en la tienda. Cada movimiento tiene una lista de artículos. Se almacena información relevante para la diferenciación de todos los artículos de la tienda.
- **Textil:** Representa la talla concreta de cada artículo y la cantidad que se tiene de ella.
- **Movimiento:** Representa cualquier operación que se realiza en la tienda, ya sea un préstamo, una devolución o una venta. Para cada operación, habrá un cliente asignado, una lista de artículos y otra información relevante que puede verse en el diagrama.

5.4. Modelos de comportamiento

En esta sección vamos a especificar cómo opera el sistema ante las acciones del actor mediante diagramas de secuencia del sistema.

5.4.1. Apertura y cierra de sesión

5.4.1.1. Inicio de sesión

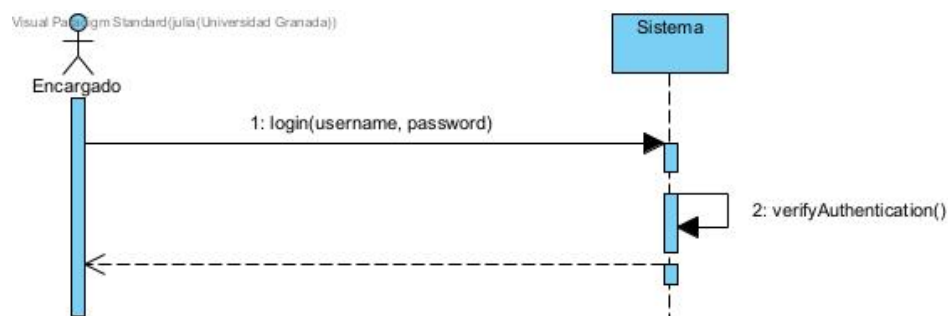


Figura 5.8: Diagrama de secuencia de inicio de sesión

5.4.1.2. Cierre de sesión

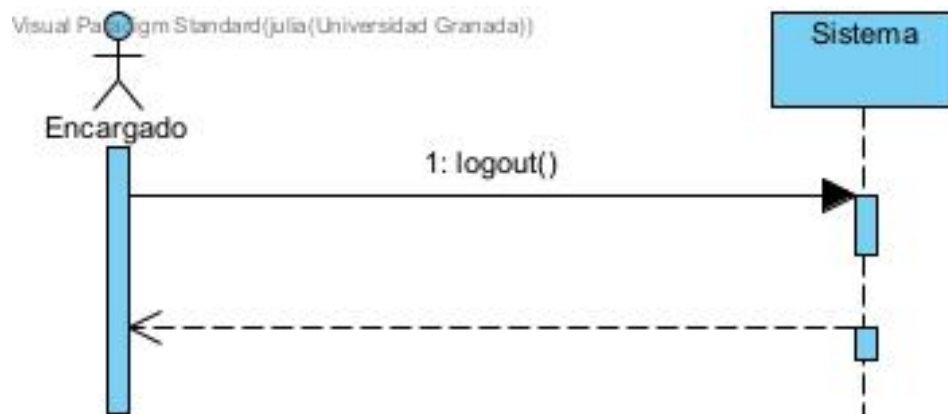


Figura 5.9: Diagrama de secuencia de cierre de sesión

Para seguir viendo la explicación de los diagramas de secuencia del sistema, consultar el Apéndice B.

Capítulo 6

Diseño

6.1. Diagrama de la arquitectura del sistema

Como se puede observar en la imagen, la arquitectura del sistema es una arquitectura sencilla. El usuario interactúa únicamente con la interfaz gráfica de la aplicación y la aplicación solicita a la base de datos la información que el usuario requiere. De la misma forma, la base de datos envía la información a la aplicación y la aplicación se la muestra al usuario. El usuario y la base de datos no se relacionan directamente, siempre estará la aplicación como intermediario.

No es necesaria la existencia de un servidor gracias a la herramienta que se está utilizando como base de datos, Supabase. Esta herramienta da servicio para construir aplicaciones software.

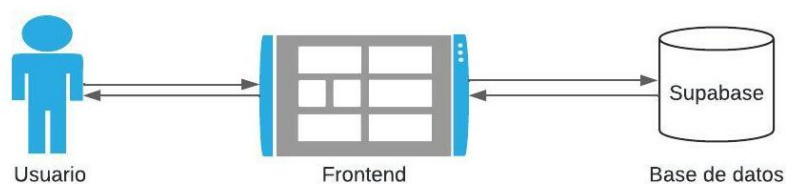


Figura 6.1: Diagrama de la arquitectura del sistema.

6.2. Diseño de la interfaz gráfica

Debemos buscar la usabilidad y accesibilidad en una aplicación de estas características ya que el usuario final puede no tener grandes destrezas tecnológicas. Por tanto, hay que conseguir un diseño sencillo, fácil de usar e intuitivo.

Para ello, habrá que tener en cuenta los siguientes criterios:

- **Botones grandes:** Los botones de la aplicación deben de tener un tamaño adecuado para facilitar la realización de acciones.
- **Tamaño y contraste del texto:** Nos debemos de asegurar que el tamaño de la letra es lo suficientemente grande y que el contraste con el fondo es significativo para facilitar con ello la lectura. El fondo de la aplicación será blanco para conseguir un buen contraste y evitar las distracciones.
- **Claridad en la navegación:** Proporcionar una estructura lógica de navegación y unas flechas visibles que permitan cambiar de pantallas de manera intuitiva.
- **Simplicidad:** El diseño debe ser sencillo. Evitar la sobrecarga de información y priorizar un diseño ordenado.

6.2.1. Pantalla de inicio de sesión

En esta sección se muestra el boceto de la interfaz gráfica del inicio de sesión. Se corresponde con el requisito RF1.

Pantalla de inicio de sesión

Diagrama de la pantalla de inicio de sesión (Login). El diseño se muestra dentro de un recuadro rectangular. En el centro superior del recuadro, el texto "Login" está centrado. Debajo de "Login", hay dos campos de entrada de texto, uno encima del otro, ambos centrados horizontalmente. El primer campo de entrada contiene el texto "Nombre de usuario". El segundo campo de entrada contiene el texto "Contraseña". Debajo de estos dos campos, hay un botón rectangular centrado horizontalmente que contiene el texto "Entrar".

Figura 6.2: Diseño de la pantalla de inicio de sesión.

6.2.2. Pantalla principal

En esta sección se muestran los bocetos de la interfaz gráfica de la pantalla principal y las pantallas correspondientes con una nueva venta y un nuevo préstamo. Estas tres pantallas representan los requisitos RF19, RF20 y RF28. Por tanto, es la pantalla que muestra la caja diaria y permite registrar una nueva venta o un nuevo préstamo.

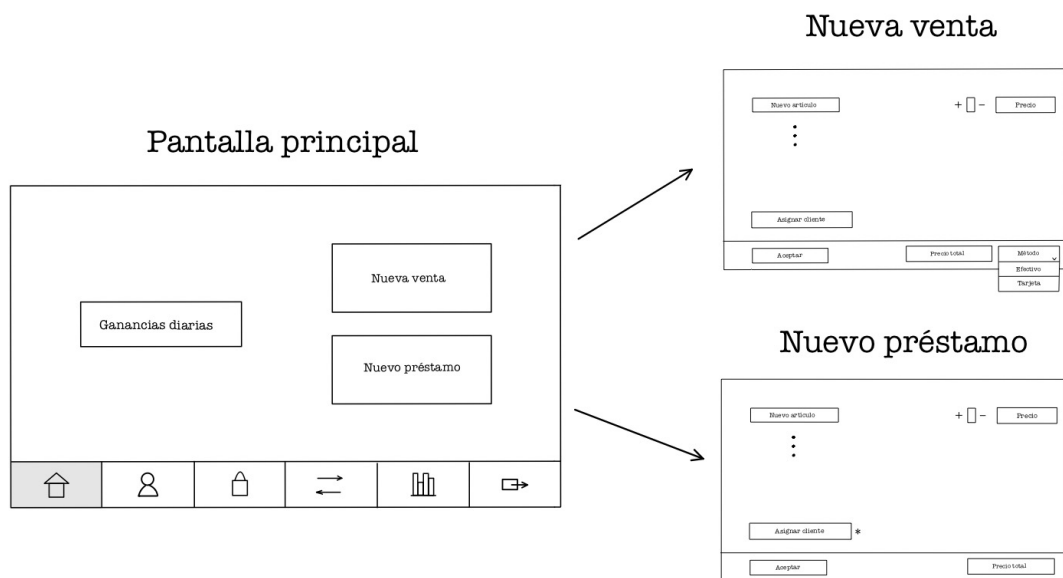


Figura 6.3: Diseño de la pantalla principal.

6.2.3. Pantalla de clientes

En esta sección se muestran los bocetos de la interfaz gráfica de la pantalla clientes. Estas cuatro pantallas representan los requisitos RF12, RF13, RF14, RF15, RF16, RF17 y RF18. En la pantalla clientes podemos ver la lista de clientes existentes, buscar clientes por nombre y filtrar aquellos clientes que tengan préstamos. Si pulsamos encima del nombre del cliente, visualizaremos todos los datos relacionados con este. Además, podemos editar, eliminar y añadir un nuevo cliente.

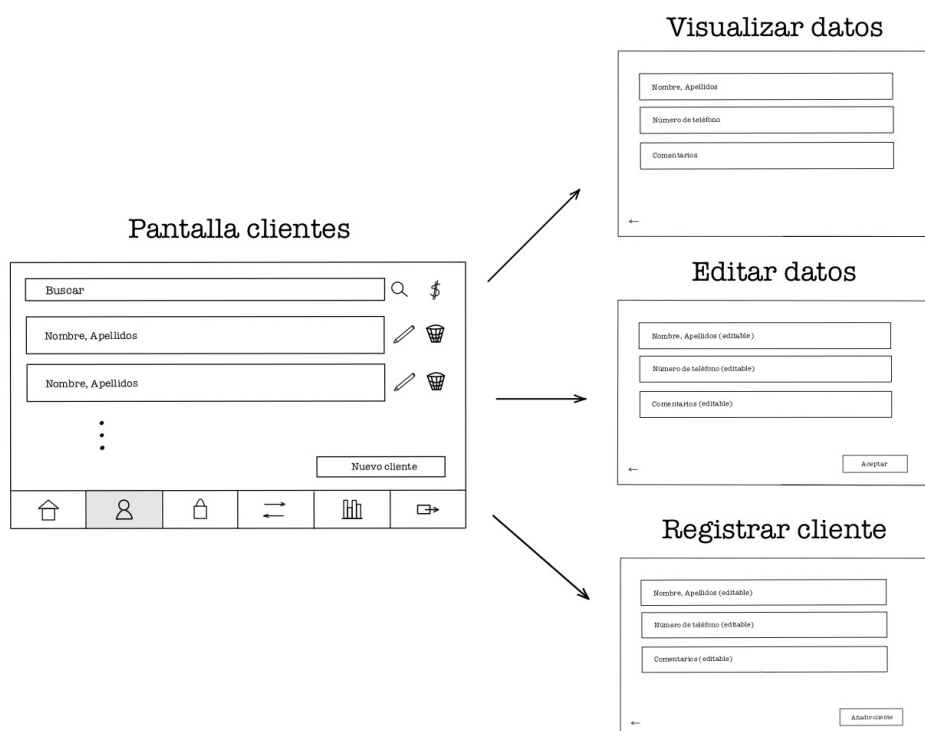


Figura 6.4: Diseño de la pantalla clientes.

6.2.4. Pantalla de artículos

En esta sección se muestran los bocetos de la interfaz gráfica de la pantalla artículos. Estas cinco pantallas representan los requisitos RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10 y RF11. En la pantalla artículos podemos ver la lista de artículos existentes, buscar los artículos por nombre y filtrarlos por categoría. Si pulsamos encima del nombre del artículo, visualizaremos todos los datos relacionados con este. Además, podemos editar, eliminar y añadir un nuevo artículo. Por último, tenemos la lista de renovación de stock donde se introducirán todos los artículos que deba comprar el comerciante.

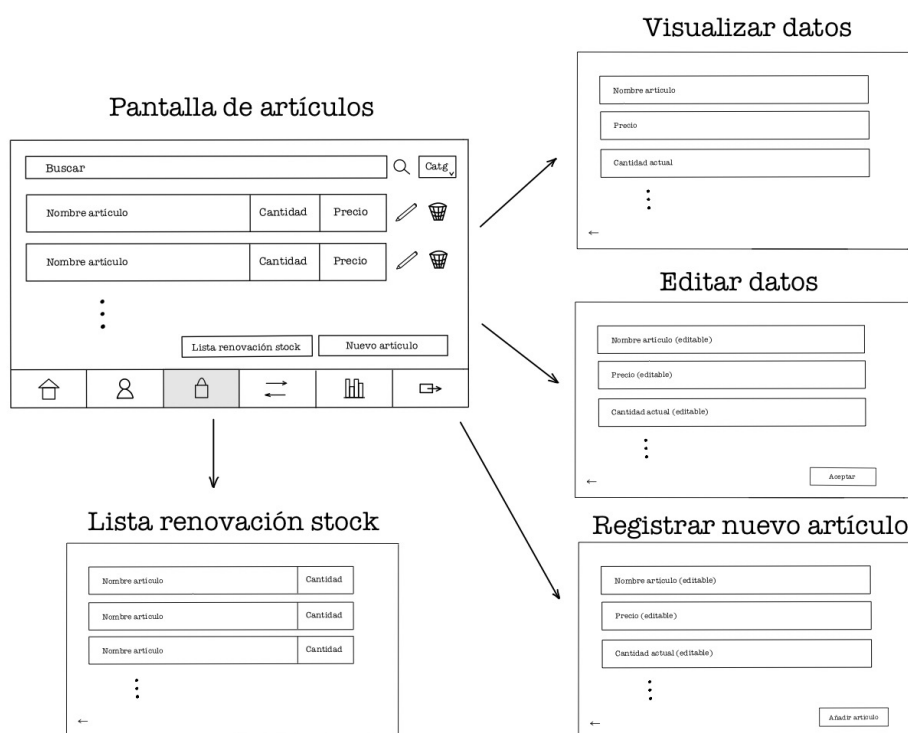


Figura 6.5: Diseño de la pantalla artículos.

6.2.5. Pantalla de movimientos

En esta sección se muestran los bocetos de la interfaz gráfica de la pantalla movimientos. Estas cuatro pantallas representan los requisitos RF21, RF22, RF23, RF24, RF25, RF26 y RF27. En la pantalla movimientos podemos ver la lista de movimientos existentes, buscar los movimientos por fecha o cliente asignado y filtrarlos por tipo de movimiento (venta, préstamo o devolución). Si pulsamos encima del movimiento, visualizaremos todos los datos relacionados con este. Si el movimiento es de tipo "préstamo", además de visualizarlo podremos convertirlo en venta seleccionando aquellos artículos que el cliente desea comprar. Únicamente se podrán devolver las ventas, ya que son los movimientos que generan una subida económica en la caja diaria. Las devoluciones generan una bajada correspondiente con la cantidad devuelta. Si se devuelve un préstamo sin comprar nada, se elimina el movimiento. Podemos eliminar cualquier movimiento, sea del tipo que sea.

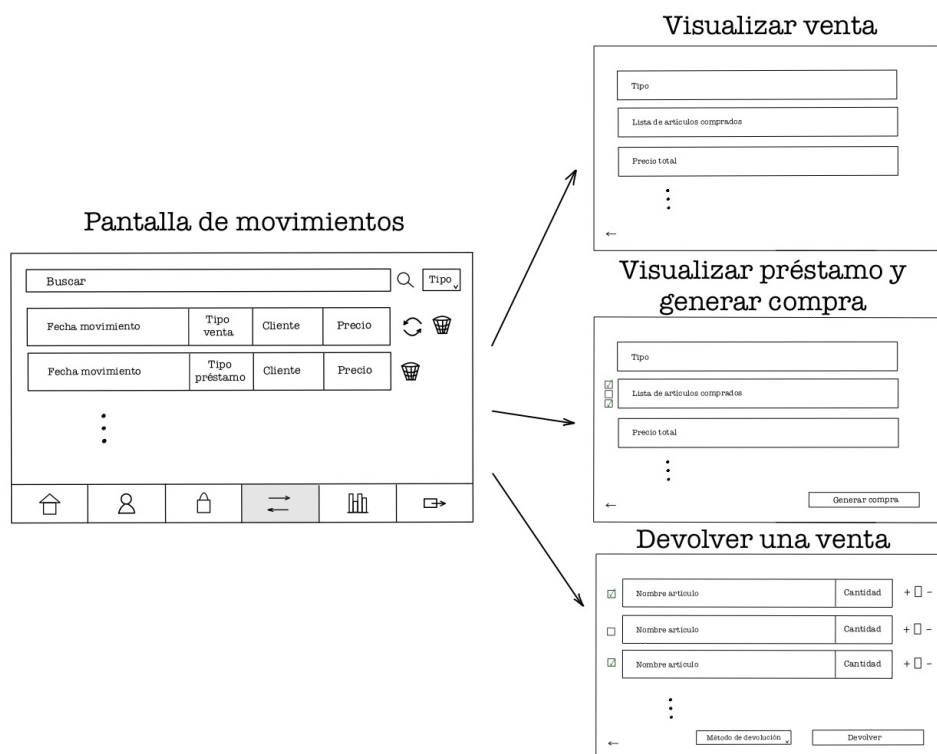


Figura 6.6: Diseño de la pantalla movimientos.

6.2.6. Pantalla de gráficos y cierre de sesión

En esta sección se muestra el boceto de la interfaz gráfica de la pantalla gráficos. Esta pantalla representa el requisito RF29. Aquí podemos observar el progreso económico del negocio en forma de gráfica. Podremos verlo de forma mensual o anual.

Para finalizar, el cierre de sesión podrá hacerse desde cualquier lugar simplemente pulsando en su botón. Esto corresponde con el requisito RF2.

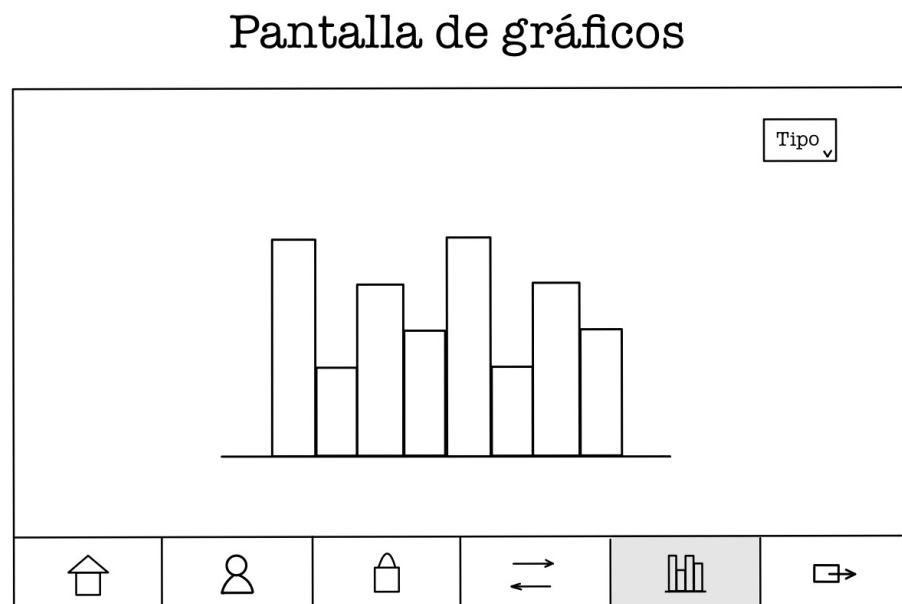


Figura 6.7: Diseño de la pantalla gráficos.

Capítulo 7

División en iteraciones

7.1. Primera iteración

7.1.1. Alcance de la iteración

En esta primera iteración se van a desarrollar los siguientes requisitos: RF1, RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF12, RF13, RF14, RF15, RF16, RF17, RF18.

Además, se deberá de crear la base de datos en Supabase, el proyecto en Flutter y conectarlos entre sí. Al final de la explicación de las iteraciones, se encuentra el diagrama de la base de datos junto a la explicación del proceso de creación de esta.

Creando la aplicación en Flutter, se está asegurando el RNF2 - Compatibilidad. Flutter es una herramienta que permite programar en un lenguaje, Dart, y generar una aplicación compatible para cada sistema operativo (Android e IOS).

Para la autenticación (login / logout), se va a utilizar el sistema de autenticación de Supabase de correo electrónico y contraseña. Para evitar que cualquier persona pueda crearse una cuenta, las credenciales las proporcionará el desarrollador al usuario. Esto permite controlar el acceso al sistema y, con ello, mejorar la seguridad de la información.

Además, Supabase dispone de ".Auth policies" personalizables, lo que permite poner restricciones de acceso a la información de la base. Se pueden restringir ciertas operaciones y especificar aquellos roles que podrán hacer modificaciones en la base de datos. Esto permite proteger los datos almacenados en la base de datos, cumpliendo con el RNF1 - Seguridad.

Para la gestión de clientes y artículos, se crearán las tablas convenientes.

7.1.2. Implementación

La implementación se ha llevado a cabo en relación a los diseños de los bocetos iniciales del capítulo anterior. Además, se han añadido pantallas de categorías de artículos para permitir una mejor organización de los artículos de la tienda. Tras completar la implementación, hemos obtenido las siguientes pantallas:

7.1.2.1. Pantalla de login

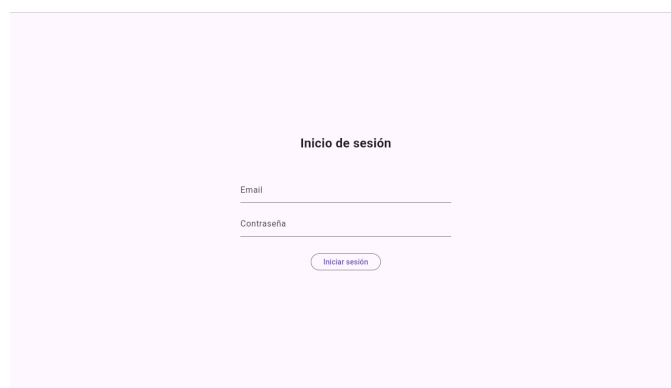


Figura 7.1: Interfaz de usuario de inicio de sesión.

7.1.2.2. Pantalla de principal

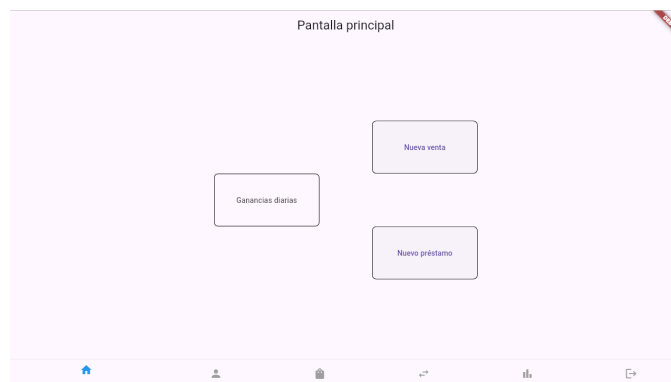


Figura 7.2: Interfaz de usuario de la pantalla principal.

7.1.2.3. Pantalla de visualización de clientes

En esta pantalla se verán todos los clientes registrados en la tienda. Se pueden buscar por nombre o filtrar aquellos que deban dinero a la tienda.

Los clientes que están en positivo en la tienda, es decir, tienen dinero a favor, se muestran con un fondo verde. Los clientes que están en negativo, se muestran con un fondo rojo. Si no tiene dinero a favor ni a deber, se muestran en blanco.

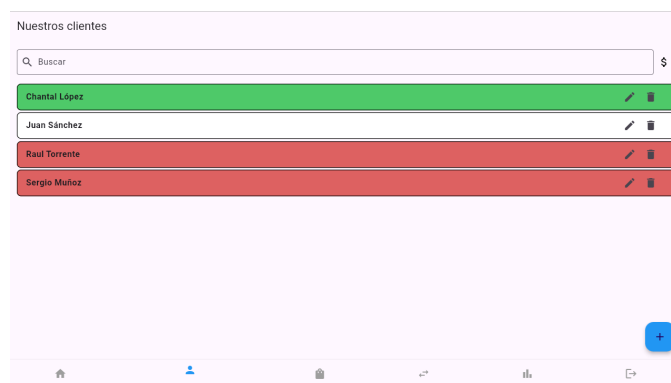


Figura 7.3: Interfaz de usuario de la pantalla de visualización de clientes.

7.1.2.4. Pantalla de añadir nuevo cliente

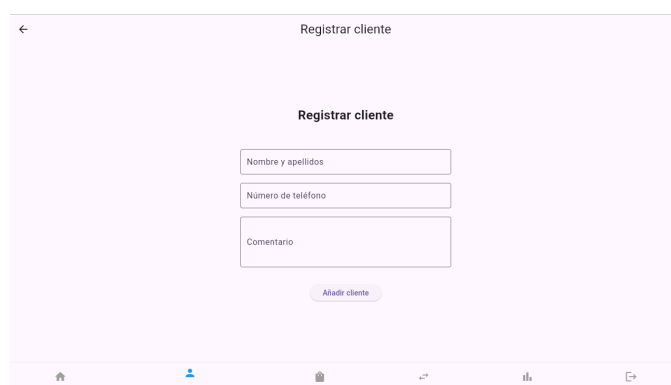


Figura 7.4: Interfaz de usuario de la pantalla de nuevo cliente.

7.1.2.5. Pantalla de editar un cliente existente

← Editar datos

Editar cliente

Nombre
Chantal López

Número de teléfono
098765432

Comentario
La dependiente del bar

Monedero (€)
90.00

Editar cliente

Home User Shop Cart Analytics Share

Figura 7.5: Interfaz de usuario de la pantalla de editar cliente.

7.1.2.6. Pantalla de visualizar los datos de un cliente

← Visualizar datos

Visualizar cliente

Nombre
Sergio Muñoz

Número de teléfono
+34 456 874 635

Comentario

Monedero (€)
-9.50

Home User Shop Cart Analytics Share

Figura 7.6: Interfaz de usuario de la pantalla de visualizar datos de un cliente.

7.1.2.7. Pantalla de visualizar las categorías de artículos

Se pueden añadir tantas categorías como el usuario crea conveniente para poder organizar los productos de su tienda a su gusto.

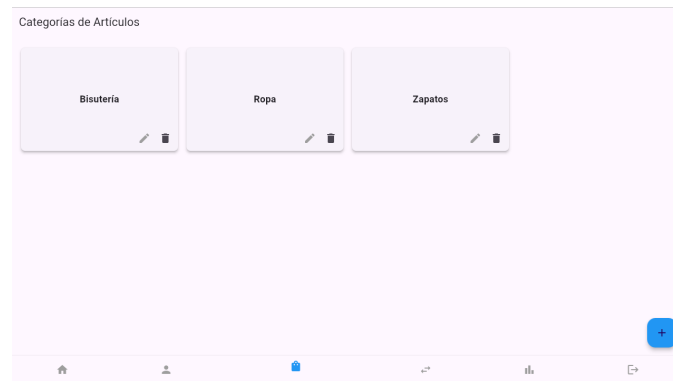


Figura 7.7: Interfaz de usuario de la pantalla de visualizar las categorías.

7.1.2.8. Pantalla de añadir una nueva categoría de artículos

Cuando se añade una nueva categoría, se deberá de especificar el nombre de dicha categoría y se podrán desmarcar los campos que no sean necesarios para esa categoría. Al marcar o desmarcar un campo, se cambia la visibilidad de este en todos los artículos que estén contenidos en esa categoría. Así se evita incluir campos que no tengan relevancia en ciertas categorías.

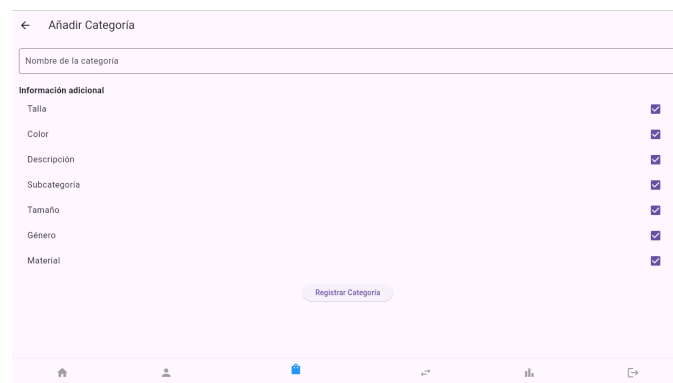


Figura 7.8: Interfaz de usuario de la pantalla de añadir nueva categoría.

7.1.2.9. Pantalla de editar una categoría existente

← Editar Categoría

Editar categoría

Nombre de la categoría
Bisutería

Talla ☐

Color ☐

Descripción ☐

Subcategoría ☐

Tamaño ☐

Género ☐

Material ☐

Editar categoría

Figura 7.9: Interfaz de usuario de la pantalla de editar una categoría.

7.1.2.10. Pantalla de visualizar los artículos de una categoría

En esta pantalla se ven todos los artículos de una determinada categoría. Se puede buscar por nombre o filtrar por subcategoría.

← Nuestros Artículos

Buscar Subcategoría

Anillo
Precio: 6.00 €
Cantidad: 8

Collar
Precio: 22.56 €
Cantidad: 1

Pendientes
Precio: 15.00 €
Cantidad: 2

Reloj
Precio: 6.50 €
Cantidad: 0

+

Figura 7.10: Interfaz de usuario de la pantalla de visualizar artículos.

7.1.2.11. Pantalla de añadir un nuevo artículo a una categoría

← Añadir Nuevo Artículo

Añadir Nuevo Artículo

Nombre

Precio

Cantidad Actual

Cantidad Mínima

Subcategoría Nueva Subcategoría

Género

Color

Material

Figura 7.11: Interfaz de usuario de la pantalla de añadir un nuevo artículo.

7.1.2.12. Pantalla de editar un artículo existente de una categoría

← Editar Artículo

Editar Artículo

Nombre Anillo

Precio 6,00

Cantidad Actual 8

Cantidad Mínima 1

Subcategoría Nueva Subcategoría

Género Unisex

Color

Material Metal

Tamaño

Figura 7.12: Interfaz de usuario de la pantalla de editar un artículo existente.

7.1.3. Pruebas de funcionalidad

Para testear la implementación que se ha llevado a cabo, se han realizado una serie de pruebas de funcionalidad:

- Comprobación de acceso a la aplicación únicamente con las credenciales válidas.
- Comprobación de visualización de todos los clientes almacenados en la base de datos.
- Comprobación del funcionamiento de búsqueda por nombre del cliente.

- Comprobación del filtrado por deudas.
- Comprobación de visualización de los detalles del cliente.
- Comprobación de la edición de los datos del cliente.
- Comprobación de la creación de un nuevo cliente.
- Comprobación de la eliminación de un cliente existente.
- Comprobación de la visualización de las categorías existentes en la tienda.
- Comprobación de la creación de una nueva categoría en la tienda.
- Comprobación de la edición de una categoría existente.
- Comprobación de la visualización de los artículos vinculados con una categoría.
- Comprobación de la búsqueda de un artículo por nombre.
- Comprobación del filtrado de artículos según la subcategoría.
- Comprobación de la visualización de las características un artículo.
- Comprobación de la edición de las características de un artículo.
- Comprobación del registro de un nuevo artículo.

7.1.4. Revisión de la iteración

7.1.4.1. Autenticación

El cliente probó el sistema de autenticación y le pareció correcto.

7.1.4.2. Gestión de clientes

El cliente probó este conjunto de pantallas que gestionan los clientes de la tienda. Pidió una mejora en la accesibilidad en la pantalla de visualizado de clientes. No diferenciar únicamente cuando un cliente debe con un sistema de colores (verde/rojo), añadir también un icono que lo represente. El resto de funcionalidades estaban correctas.

7.1.4.3. Gestión de artículos

El cliente probó este conjunto de pantallas que gestionan los artículos de la tienda y decidió que no se estaban organizando de forma correcta. Las categorías estaban bien, pero a la hora de añadir nuevos artículos, el formulario debería de permitir añadir varias tallas, no únicamente una. Además, por cada talla, se debería de gestionar la cantidad actual, cantidad mínima y color.

El resto de funcionalidades estaban bien.

7.1.5. Plan para la próxima iteración

Para la próxima iteración se propone:

- Implementar las mejoras señaladas por el cliente.
- Implementar la pantalla de la lista de renovación de artículos.
- Implementar las pantallas de gestión de movimientos.

7.2. Segunda iteración

7.2.1. Alcance de la iteración

En esta segunda iteración se van a desarrollar los siguientes requisitos: RF10, RF19, RF20, RF21, RF22, RF23, RF24, RF25, RF26, RF27, RF28.

Además se han mejorado los siguientes requisitos como consecuencia de las mejoras señaladas por el cliente: RF3, RF4, RF6.

7.2.2. Implementación

La implementación se ha llevado a cabo en relación a los diseños de los bocetos iniciales del capítulo anterior. Estos bocetos eran orientativos y el diseño se ha mejorado en esta etapa para obtener una mejor usabilidad. Tras completar la implementación, hemos obtenido las siguientes pantallas:

7.2.2.1. Pantalla principal

En esta pantalla se pueden consultar las ganancias diarias y se muestran las opciones de generar una nueva venta o un nuevo préstamo. Esto se ha puesto en la pantalla principal ya que son las funcionalidades que se usarán con más frecuencia.

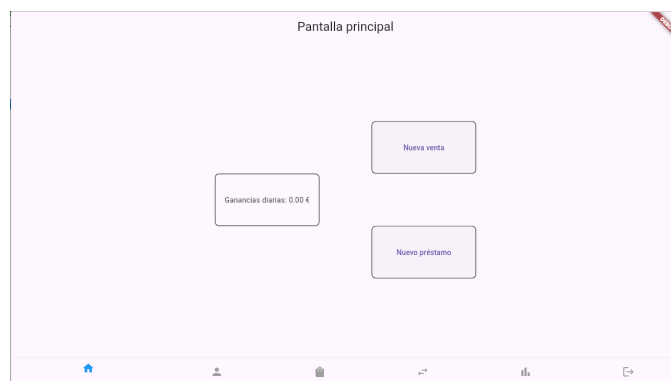


Figura 7.13: Interfaz de usuario de la pantalla principal.

7.2.2.2. Pantalla de nueva venta

En esta pantalla se realiza una nueva venta. En el buscador de la parte superior, se van proponiendo artículos que coincidan con los caracteres buscados. Una vez seleccionado el artículo, en los desplegables de la talla y el color, se proponen únicamente las tallas o colores que existan para dicho

artículo. La cantidad no puede ser inferior a 1. Si la cantidad es mayor a la que hay disponible en la tienda, salta un error avisando de que la cantidad es errónea. Si todo es correcto, pulsando al + se añade el artículo a la lista. Como es una venta, asignar el cliente es opcional. También es una búsqueda dinámica como el buscador de los artículos. El método de pago es obligatorio y se podrá elegir entre Efectivo y Tarjeta. Una vez todos los campos estén completos y se pulse al botón de Aceptar, se generará un nuevo movimiento tipo Venta.

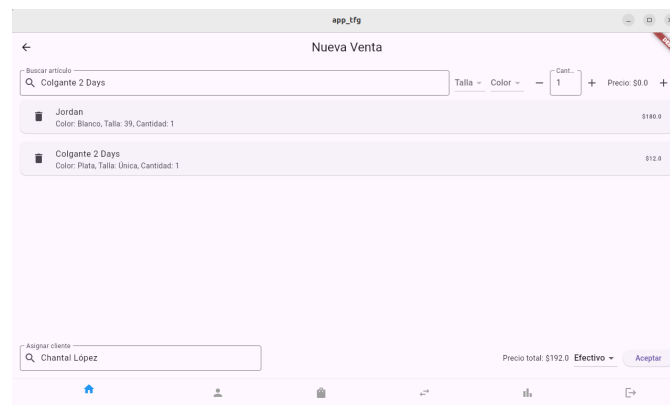


Figura 7.14: Interfaz de usuario de la pantalla de nueva venta.

7.2.2.3. Pantalla de nuevo préstamo

Esta pantalla funciona de la misma forma que la nueva venta. Sin embargo, asignar el cliente será obligatorio y el método de pago no existe ya que los préstamos no se pagan. El dinero de dicho préstamo se verá reflejado de forma negativa en el monedero del cliente. Además, se generará un movimiento de tipo Préstamo.

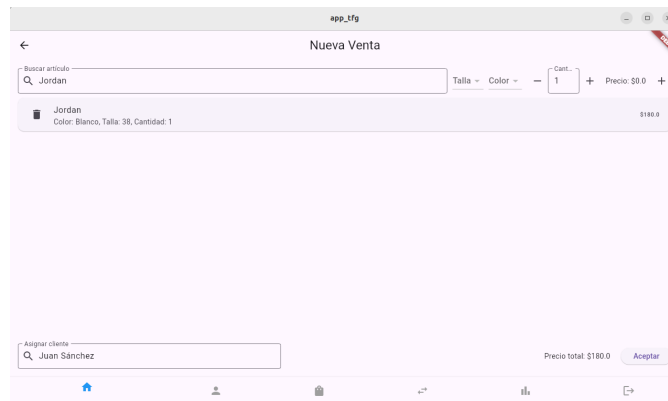


Figura 7.15: Interfaz de usuario de la pantalla de nuevo préstamo.

7.2.2.4. Pantalla de visualización de movimientos

En esta pantalla podemos ver todos los movimientos efectuados en la tienda. Están ordenados de forma que los más recientes se sitúen en la parte superior. Se puede eliminar un movimiento, pero se avisará de que el inventario no se actualizará. Esta funcionalidad es únicamente para permitir borrar movimientos que se generen por error. En la parte superior se puede buscar por cliente. Una vez seleccionado el cliente, solo aparecerán los movimientos que estén asociados a dicho cliente. Además, se puede filtrar por tipo de movimiento: Venta, Préstamo o Devolución.

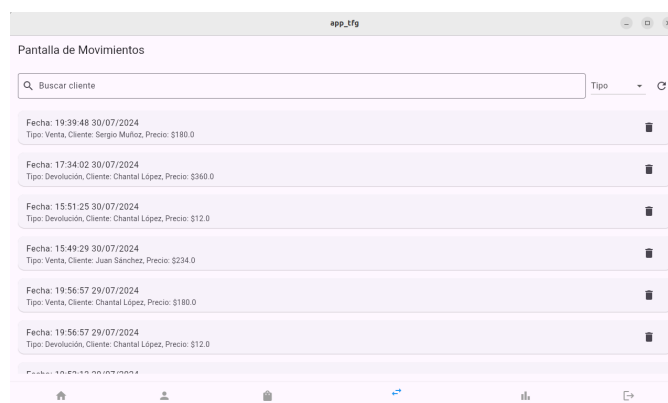


Figura 7.16: Interfaz de usuario de la pantalla de visualización de movimientos.

7.2.2.5. Pantalla de detalles del movimiento

En esta pantalla podemos ver los detalles de un movimiento en específico. Esta pantalla muestra distintas funcionalidades dependiendo del tipo y del estado del movimiento.

Venta: Cuando se trata de una venta, en la esquina superior derecha aparecen dos botones que te proponen una devolución parcial (solo quieres devolver algunos artículos) o una devolución total (se quieren devolver todos los artículos). Además, se pueden observar los datos relevantes de la venta, como los artículos involucrados, el cliente, el método de pago, la fecha y el precio total.

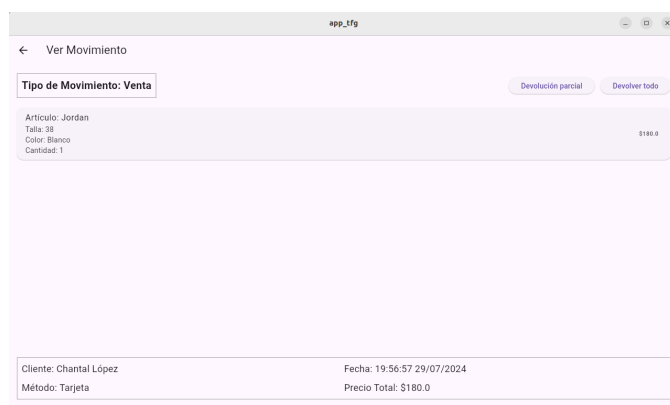


Figura 7.17: Interfaz de usuario de la pantalla de detalles de una venta.

Si se realiza una devolución total, no se muestra ninguna pantalla adicional, directamente se crea un nuevo movimiento tipo Devolución con todos los artículos de la venta. Al realizar una devolución parcial, la pantalla cambia su modo Visualización a modo Devolución. Se permite escoger los artículos que se desean devolver y el método de pago en el que se prefiere realizar la devolución.

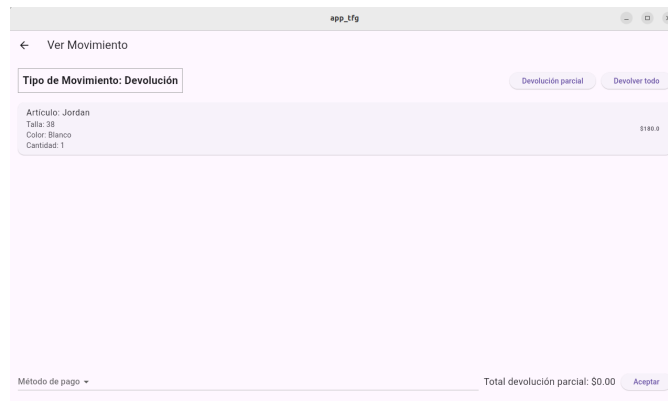


Figura 7.18: Interfaz de usuario de la pantalla de devolución de una venta.

Tras aceptar la devolución, se genera un nuevo movimiento tipo Devolución. En la venta, se verá reflejada esta devolución. Desaparecen los botones de Devolución parcial y Devolución Total y aparece un botón con una referencia a la devolución realizada. De esta forma, se podrá visualizar de forma directa la devolución que se ha hecho de cada venta.



Figura 7.19: Interfaz de usuario de la pantalla de una venta con devolución hecha.

Préstamo: La interfaz es similar a la pantalla de detalles de una venta, muestra la misma información. Sin embargo, los botones de la esquina superior son distintos. Una vez más, se ofrecen dos opciones: Compra o Devolver todo. El botón de Devolver todo funciona de la misma forma que en la Venta.



Figura 7.20: Interfaz de usuario de la pantalla los detalles de un préstamo.

La funcionalidad del botón de Compra es distinta. Este botón permite generar una compra a partir de un préstamo. Al pulsar este botón, se cambia la funcionalidad de la pantalla y permite seleccionar los artículos que se desean comprar. Seleccionar el método de pago es obligatorio. Al seleccionar Aceptar, se generan dos movimientos: un movimiento de venta con los artículos que se han seleccionado para comprar y un movimiento de devolución con aquellos artículos que no se hayan comprado (en el caso de que queden artículos sin seleccionar). Así se verá reflejado en el historial de la tienda aquellos artículos que se compran y se devuelven. A su vez, el inventario se actualiza con estos movimientos.



Figura 7.21: Interfaz de usuario de la pantalla de compra a partir de un préstamo.

Si se ha hecho una devolución, se le asocia al préstamo de la misma forma que vimos en la venta.

Devolución: Esta pantalla también muestra la misma información que los dos movimientos anteriores, los detalles del movimiento. En la esquina superior siempre aparecerá un botón de Movimiento original, que te llevará de forma directa al movimiento del cual proviene dicha devolución.

app_1fg

← Ver Movimiento

Tipo de Movimiento: Devolución Movimiento original

Artículo: Jordan
Talla: 39
Color: Blanco
Cantidad: 2 \$360.0

Cliente: Chantal López
Método: N/A

Fecha: 17:34:02 30/07/2024
Precio Total: \$360.0

Figura 7.22: Interfaz de usuario de la pantalla de detalles de una devolución.

7.2.2.6. Mejora de las tallas en artículos

En la iteración anterior, el cliente pidió que la gestión de las tallas de un artículo se hiciera de forma distinta. Se pidió que para cada artículo se pudieran añadir varias tallas. Para conseguir esto, se modificó la forma de crear un artículo, editarlo y visualizarlo. En la siguiente imagen podemos ver cómo la modificación ha sido correctamente efectuada.

app_1fg

← Visualizar Artículo

Artículo: Jordan

Precio: 180.0

Subcategoría: Nike

Género: Femenino

Material: Piel

Descripción

Talla	Color	Cantidad Actual	Cantidad Mínima
37	Azul	5	1
38	Blanco	5	2
39	Blanco	8	2

Figura 7.23: Interfaz de usuario que muestra el cambio de la gestión de las tallas.

7.2.3. Pruebas de funcionalidad

Para testear la implementación que se ha llevado a cabo, se han realizado una serie de pruebas de funcionalidad:

- Comprobación de la creación de una nueva venta. Esto implica que se almacenen los parámetros correctos y los artículos de dicha venta se almacenen en su tabla correspondiente.
- Comprobación de la creación de un nuevo préstamo, prestando atención a que se cumplan sus características.
- Comprobación de la visualización de todos los movimientos almacenados en la base de datos.
- Comprobación de la eliminación de un movimiento.
- Comprobación de la búsqueda de un movimiento por cliente.
- Comprobación del filtrado de movimientos según su tipo.
- Comprobación de visualización de los detalles de un movimiento.
- Comprobación de la devolución total de una venta o préstamo.
- Comprobación de la modificación de la cartera de un cliente tras la creación, eliminación o devolución de un préstamo.
- Comprobación de la devolución parcial de una venta.
- Comprobación de la realización de una compra a partir de un préstamo.
- Comprobación de la navegación entre dos movimientos relacionados.
- Comprobación de la correcta implementación de las tallas de un artículo.

7.2.4. Revisión de la iteración

El cliente probó el sistema ideado para los movimientos y expresó que estaba conforme con la lógica de la implementación. Además, mostró gratitud por el cambio de las tallas que solicitó en la iteración anterior.

Por tanto, el cliente no puso ninguna objeción y animó a continuar con el planning establecido.

7.2.5. Plan para la próxima iteración

Para la próxima iteración se propone:

- Implementar la pantalla de gráficos.
- Mejorar la interfaz gráfica, realizando a su vez, mejoras en la usabilidad y accesibilidad.
- Testeo de posibles fallos de funcionamiento de la aplicación.

7.3. Tercera iteración

7.3.1. Alcance de la iteración

En esta iteración se llevarán a cabo los siguientes requisitos: RF29, RNF3 y RNF4.

El primero de ellos corresponde con la creación de la pantalla de gráficos. Los últimos son requisitos no funcionales de usabilidad y accesibilidad. Debido a que en esta iteración se mejorará la interfaz gráfica de usuario, se tendrá en cuenta la mejora de la usabilidad y la accesibilidad.

7.3.2. Pantalla de gráficos

Esta es la última pantalla que quedaba por implementar para completar el funcionamiento total de la aplicación. Como podemos observar, en la esquina superior de la pantalla hay un desplegable que permite adaptar la gráfica generada en función de si queremos que sea una gráfica mensual o anual.

La gráfica mensual hace un resumen de las ganancias o pérdidas del mes actual. Además, se adapta al número de días que tenga el mes.

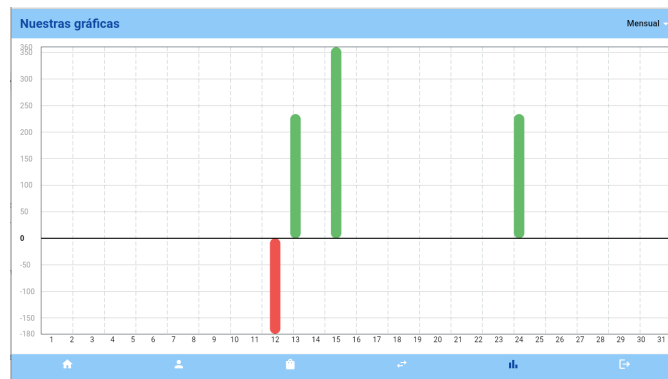


Figura 7.24: Pantalla de gráficos en modo Mensual.

La gráfica anual hace un resumen de las ganancias o pérdidas de año actual. Estas dos gráficas permiten obtener una visión general de progreso del negocio.

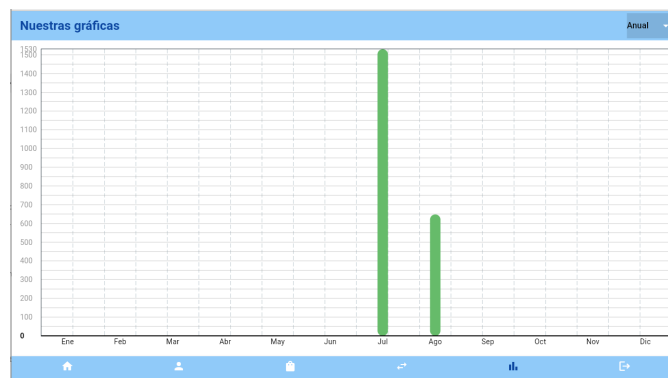


Figura 7.25: Pantalla de gráficos en modo Anual.

Para hacer las cuentas y representarlas en las gráficas, únicamente se tienen en cuenta las ventas y las devoluciones. Los préstamos no se tienen en cuenta ya que es dinero que no ha sido recibido aún.

7.3.3. Mejora de la interfaz gráfica y la accesibilidad

A lo largo de toda la aplicación, se mantiene:

- Un diseño sencillo e intuitivo para que sea usable.
- Fondos lisos y claros para facilitar la lectura del texto.
- Fuente de letra estándar y colores que contrasten con el fondo.

- Etiquetado de las pantallas para que el usuario pueda saber dónde está situado en cada momento.

7.3.4. Colores de la aplicación

Para la interfaz de la aplicación se ha escogido como color primario el azul y como color secundario el amarillo. Estos dos colores son complementarios y hace que sea fácil llamar la atención del usuario realizando una buena combinación de ambos.

Dentro del color primario, se ha trabajado con una gama de azules:

- **Azul claro:** Se ha seleccionado Colors.blue[50].
- **Azul medio:** Se ha seleccionado Colors.blue[200].
- **Azul oscuro:** Se ha seleccionado Colors.blue[900].

Colors.blue[50] Colors.blue.shade50	0xFFE3F2FD
Colors.blue[100] Colors.blue.shade100	0xFFBBDEFB
Colors.blue[200] Colors.blue.shade200	0xFF90CAF9
Colors.blue[300] Colors.blue.shade300	0xFF64B5F6
Colors.blue[400] Colors.blue.shade400	0xFF42A5F5
Colors.blue	0xFF2196F3
Colors.blue[600] Colors.blue.shade600	0xFF1E88E5
Colors.blue[700] Colors.blue.shade700	0xFF1976D2
Colors.blue[800] Colors.blue.shade800	0xFF1565C0
Colors.blue[900] Colors.blue.shade900	0xFF0D47A1

Figura 7.26: Gama de colores azules de Flutter.

Como color secundario, se ha seleccionado Colors.yellow[600].

Colors.yellow[400] Colors.yellow.shade400	0xFFFFEE58
Colors.yellow	0xFFFFEB3B
Colors.yellow[600] Colors.yellow.shade600	0xFFDD835

Figura 7.27: Gama de colores amarillos de Flutter.

Para el fondo de la aplicación y algunos textos, se ha utilizado el blanco, Colors.white.

7.3.4.1. Inicio y cierre de sesión

En esta pantalla de inicio de sesión se utilizan campos de texto que permiten que el usuario proporcione las credenciales a la aplicación. Se hace uso de *validator* para comprobar que los campos no se dejan vacíos. Si las credenciales son incorrectas al presionar *Iniciar sesión*, aparece un diálogo similar al de cerrar sesión informando al usuario del problema.

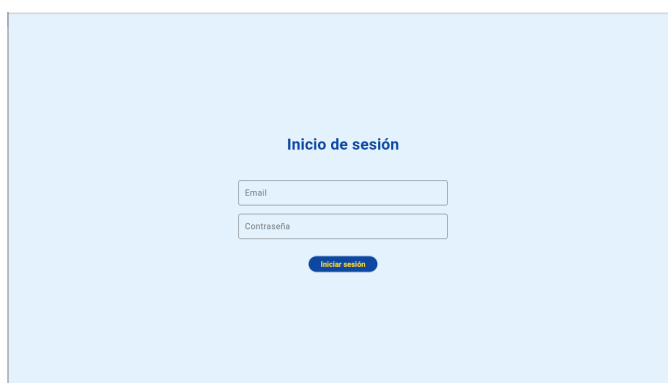
La imagen muestra una pantalla de inicio de sesión con un fondo azul claro. En el centro, el título "Inicio de sesión" está en azul. Debajo, hay dos campos de texto blancos con bordes azules: "Email" y "Contraseña". Debajo de estos campos, hay un botón azul con el texto "Iniciar sesión" en blanco.

Figura 7.28: Pantalla de inicio de sesión.

Al presionar el botón de *Confirmar* del diálogo, se cierra la sesión activa del usuario.

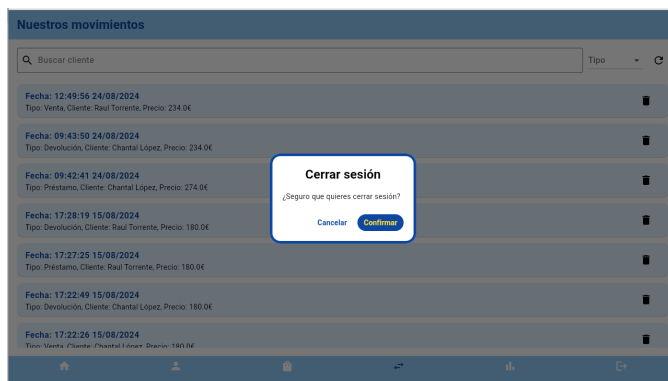
La imagen muestra una pantalla de "Nuestros movimientos" con un fondo gris oscuro. En la parte superior, hay una barra de búsqueda con el texto "Buscar cliente" y un botón "Tipo". Debajo, hay una lista de movimientos con columnas para "Fecha", "Tipo", "Cliente", "Precio" y "Devolución". En el centro de la pantalla, hay un diálogo de "Cerrar sesión" con el texto "¿Seguro que quieres cerrar sesión?" y dos botones: "Cancelar" y "Confirmar".

Figura 7.29: Cierre de sesión.

7.3.4.2. Pantalla principal

Esta pantalla está compuesta por un campo de texto no editable, para mostrar las ganancias diarias, y dos *ElevatedButton* que permiten crear una nueva venta o préstamo.

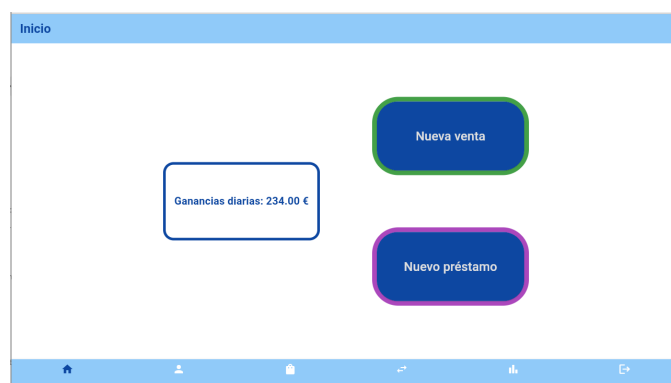


Figura 7.30: Pantalla principal.

La pantalla de *Nueva venta* y *Nuevo préstamo* parten de la misma implementación y se diferencian con un booleano que especifica el movimiento que se está creando. Para la búsqueda se utiliza un campo de texto que escucha cada cambio y encuentra los artículos dinámicamente. Una vez encontrado el artículo, los desplegables de *Talla* y *Color* realizan una consulta a la base de datos y ofrecen las opciones que tiene ese artículo para esos valores. La cantidad mínima es 1 y la cantidad máxima la determina la cantidad de stock que haya en la tienda. Si la cantidad seleccionada sobrepasa la cantidad existente en la tienda, se muestra un diálogo avisando de que la cantidad seleccionada es incorrecta y se muestra la cantidad posible. El precio, se va actualizando dinámicamente, multiplicando el precio del producto por la cantidad seleccionada de este. Al pulsar el símbolo $+$, se genera una *Card* con la información especificada y, a continuación, se puede añadir otro artículo a la factura.

Si es una venta, el campo de cliente, puede dejarse vacío. Si es un préstamo, es obligatorio que se asigne un cliente ya que es un movimiento que no se paga en el momento. El campo de búsqueda del cliente funciona de la misma forma que la búsqueda del artículo. Si es una venta, aparece el desplegable para seleccionar el método de pago y es obligatorio rellenarlo, ya que deberá ser pagado en el momento. Ambos movimientos finalizan al presionar el botón de *Aceptar*.

Cada vez que se añade un nuevo movimiento, se manda una notificación mediante el uso de un *Provider ChangeNotifier* para avisar de que el resto

de pantallas deben actualizarse con la nueva información que se acaba de añadir. El resto de pantallas escuchan con un *Consumer*.

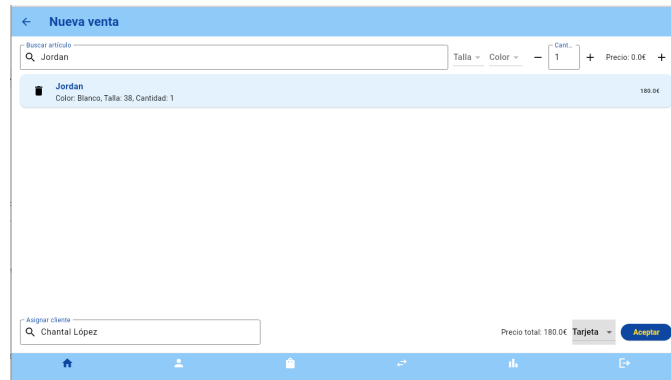


Figura 7.31: Pantalla de nueva venta.

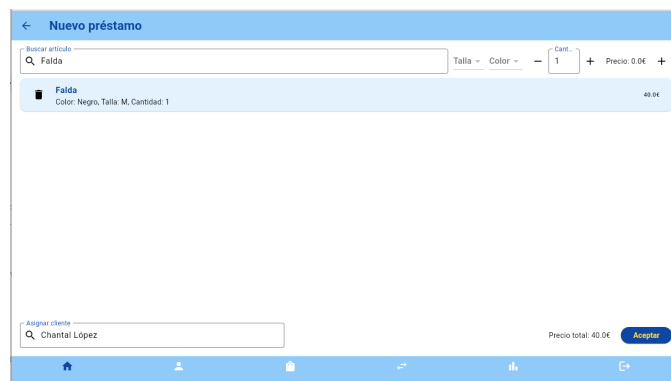


Figura 7.32: Pantalla de nuevo préstamo.

7.3.4.3. Pantalla de clientes

En esta pantalla, el buscador funciona de la misma forma que explicamos con anterioridad. A la derecha, podemos ver un icono de dinero que activa y desactiva un booleano para filtrar aquellos clientes que se encuentran en dinero negativo en la tienda. Además, en cada *Card* de clientes con dinero positivo o negativo podemos ver una flecha. Esto se ha llevado a cabo como requisito que se añadió en la primera iteración. El cliente expresó que quería poder diferenciar el estado positivo o negativo del cliente no solamente con el color, sino también con algún símbolo, para aquellas personas que no diferencien los colores de forma correcta. Por tanto, se añadió una flecha hacia arriba o hacia abajo para ser más accesibles.

La parte central de la pantalla es un *StreamBuilder* que reconstruye su contenido cada vez que hay algún *insert*, *update* o *delete*. De esta forma, el contenido que se muestra en la pantalla de visualización de los clientes, está siempre actualizado. Además, escucha los *ChangeNotifier* con el *Consumer* para actualizar los datos monetarios del cliente que se vean afectados con los movimientos de la tienda.

Para eliminar un cliente, no debe de estar vinculado a ningún movimiento. Si está vinculado a un movimiento, debe de existir, para mantener el histórico de la tienda.

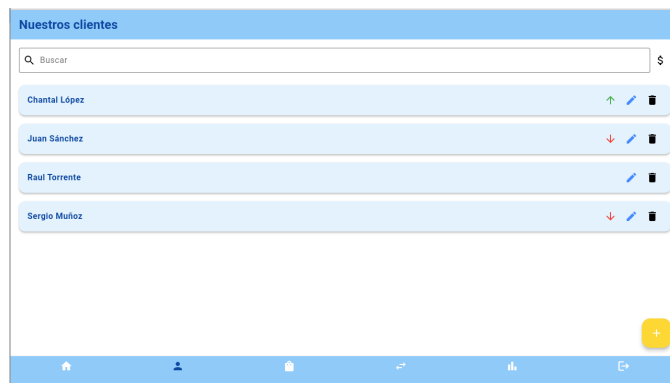


Figura 7.33: Pantalla de visualización de clientes.

Las tres pantallas siguientes son muy similares. La primera de ellas, la que muestra los detalles, son unos campos de texto inicializados a los valores almacenados en la base de datos y con la propiedad *readonly*. La segunda pantalla, la pantalla de edición, tiene los mismos campos de texto inicializados a dichos valores, pero sin la propiedad *readonly* para permitir modificar la información. Además, se comprueba que el número de teléfono sea correcto mediante una expresión regular. En la tercera pantalla, la pantalla de registro, los campos no están inicializados y se comprueba que el nombre

esté relleno mediante un *validator*.

The screenshot shows a mobile application interface with a blue header bar containing a back arrow and the text 'Detalles del cliente'. Below the header, the title 'Datos del cliente' is displayed. The form contains four input fields with pre-filled data: 'Nombre' with 'Chantal López', 'Número de teléfono' with '098765431', 'Comentario' with 'La dependienta del bar', and 'Monedero (€)' with '1260.00'. A bottom navigation bar with icons for home, profile, shop, chat, analytics, and share is visible at the bottom.

Figura 7.34: Pantalla de visualización de los datos de un cliente.

The screenshot shows a mobile application interface with a blue header bar containing a back arrow and the text 'Editar datos del cliente'. Below the header, the title 'Editar cliente' is displayed. The form contains four input fields with pre-filled data: 'Nombre' with 'Chantal López', 'Número de teléfono' with '098765431', 'Comentario' with 'La dependienta del bar', and 'Monedero (€)' with '1260.00'. A blue button labeled 'Editar cliente' is positioned below the form. A bottom navigation bar with icons for home, profile, shop, chat, analytics, and share is visible at the bottom.

Figura 7.35: Pantalla de edición de los datos de un cliente.

The screenshot shows a mobile application interface with a blue header bar containing a back arrow and the text 'Añadir un cliente'. Below the header, the title 'Nuevo cliente' is displayed. The form contains three input fields: 'Nombre y apellidos', 'Número de teléfono', and 'Comentario'. A blue button labeled 'Añadir cliente' is positioned below the form. A bottom navigation bar with icons for home, profile, shop, chat, analytics, and share is visible at the bottom.

Figura 7.36: Pantalla de registro de un nuevo cliente.

7.3.4.4. Pantalla de artículos

Esta pantalla muestra las categorías en forma de *Card* mediante un *StreamBuilder*, de la misma forma que se explicó en la pantalla de los clientes. Se puede añadir o editar una categoría, de la forma que se muestra en las siguientes imágenes.

Para eliminar una categoría, no debe de tener artículos dentro de ella.

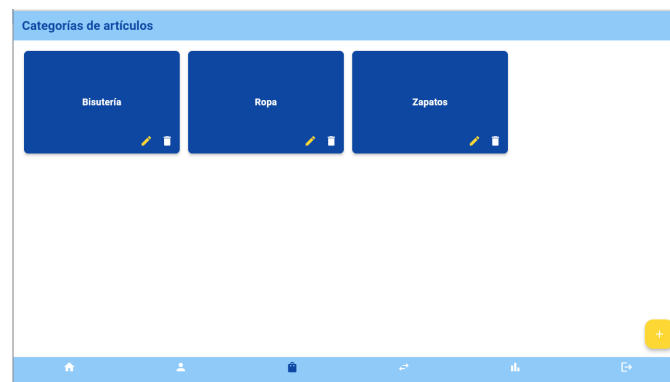


Figura 7.37: Pantalla de visualización de las categorías.



Figura 7.38: Pantalla de edición de las propiedades de una categoría.

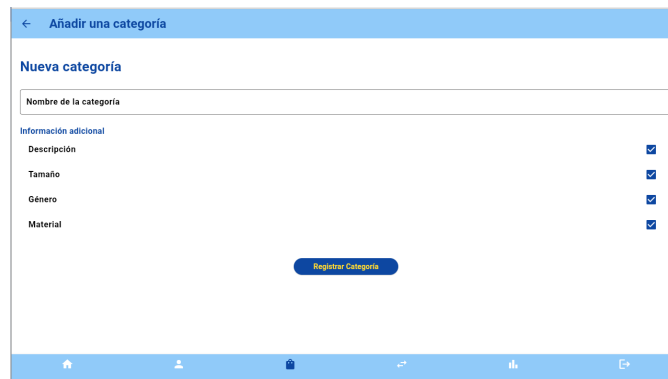


Figura 7.39: Pantalla de registro de una nueva categoría.

La pantalla de visualización de artículos es muy similar a la pantalla de visualización de clientes. Un aspecto a destacar es el desplegable de *Subcategoría*. Este desplegable realiza una consulta en la base de datos y trae las posibles subcategorías de artículos dentro de esa categoría. Permite seleccionar una subcategoría y filtrar por dicha subcategoría.

Para eliminar un artículo, no debe de estar vinculado a ningún movimiento. Si está vinculado a un movimiento, debe de mantenerse para mantener el histórico de la tienda.

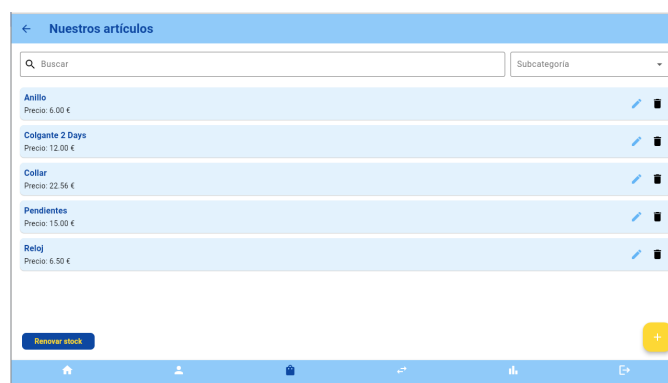


Figura 7.40: Pantalla de visualización de los artículos de una categoría.

Las pantallas de visualización de los detalles de un artículo, editar o añadir son similares a las pantallas comentadas en el apartado de clientes. Al añadir un artículo, es obligatorio añadir mínimo una talla, se comprueba mediante un *validator*.

← Detalles del artículo

Datos del artículo

Nombre
Anillo

Precio (€)
6.0

Subcategoría

Género
Unisex

Material
Metal

Dimensiones
14

Descripción
Muy bonito.

No hay tallas registradas

Figura 7.41: Pantalla de visualización de los datos de un artículo.

← Editar datos del artículo

Editar artículo

Nombre
Anillo

Precio (€)
6.0

Subcategoría Nueva Subcategoría

Género
Unisex

Material
Metal

Dimensiones
14

Descripción
Muy bonito.

Añadir otra talla

Figura 7.42: Pantalla de edición de los datos de un artículo.

← Añadir un artículo

Nuevo artículo

Nombre

Precio (€)

Subcategoría Nueva Subcategoría

Género

Material

Dimensiones

Descripción

Talla Color Cantidad Actual Cantidad Mínima

Figura 7.43: Pantalla de registro de un nuevo artículo.

La pantalla de stock muestra los artículos cuya cantidad actual es crítica y necesita ser renovada. Se muestra mediante campo de texto *readonly*.

The screenshot shows a mobile application interface titled 'Lista de renovación de stock'. It displays two items in a list, each with a 'Nombre artículo', 'Talla', 'Color', 'Cantidad actual', and 'Cantidad mínima' field. The first item is 'Falda' (S, Azul) with 'Cantidad actual' 1 and 'Cantidad mínima' 4. The second item is 'Falda' (M, Negro) with 'Cantidad actual' 2 and 'Cantidad mínima' 3. The 'Cantidad actual' fields are highlighted in red, indicating critical stock levels. The interface has a blue header and a blue bottom navigation bar with icons for home, user, shopping cart, search, and settings.

Nombre artículo	Talla	Color	Cantidad actual	Cantidad mínima
Falda	S	Azul	1	4
Falda	M	Negro	2	3

Figura 7.44: Pantalla de visualización la lista de renovación del stock.

7.3.4.5. Pantalla de movimientos

La pantalla de visualización de movimientos es similar a las anteriores que muestran todas las instancias almacenadas. La búsqueda permite buscar movimientos de un cliente en específico. El desplegable filtra por tipo de movimiento, por lo que permite seleccionar *Venta*, *Préstamo* o *Devolución*. La rueda situada en la derecha sirve para quitar la selección del desplegable.

Para la parte central, donde se muestran los movimientos en forma de *Card*, se utiliza un *StreamBuilder* que reconstruye su contenido cada vez que hay algún *insert*, *update* o *delete*. De esta forma, el contenido que se muestra en la pantalla de movimientos, está siempre actualizado. El icono de la papelera permite eliminar un movimiento solo si no está vinculado a otro. Si no se puede eliminar, se muestra un diálogo avisando del motivo.



Figura 7.45: Pantalla de visualización los movimientos de la tienda.

Esta pantalla recoge muchas funcionalidades y muestra unas u otras en función del tipo de movimiento y su estado. Para *Ventas*, si no se ha hecho una devolución anteriormente, se muestran dos botones en la parte superior que permiten realizar una devolución parcial o total. Si se decide una devolución total de la venta, se muestra un diálogo de confirmación. Al confirmar, se crea un movimiento de devolución con los datos de la venta. Si se elige una devolución parcial, la interfaz de la visualización de los detalles del movimiento cambia ligeramente y se asemeja a la creación de una *Nueva venta*. Permite seleccionar los artículos que se desean devolver y el método de pago con el que se desea recibir la devolución. Esto se consigue con un *booleano* que cambia de estado al pulsar el botón de *Devolución parcial*. Si está activo, desaparecen algunos campos de detalle del movimiento para mostrar los elementos que permiten la creación de la devolución parcial. Cuando una venta sí que tiene una devolución hecha, ambos movimientos se vinculan. En la parte superior derecha desaparecen los dos botones que se comentaban antes y aparece un botón que te lleva directamente a los

detalles de la devolución. En la devolución también estará ese mismo botón que te redirige a la venta de la que proviene dicha devolución. De esta forma, ambos movimientos están relacionados entre ellos.

Si es un *Préstamo*, funciona de la misma forma que la venta, pero en lugar del botón de *Devolución parcial*, estará el botón de *Compra*. Este botón te redirige a la misma pantalla de selección de artículos y creación de movimientos de la que hablábamos anteriormente. Sin embargo, se llama a otro método distinto, ya que los artículos seleccionados serán comprados en vez de devueltos. Además, los artículos que no se seleccionen, se mostrarán reflejados en un movimiento de devolución. Por tanto, al pulsar este botón, se podrán comprar todos los artículos, generando un único movimiento de *Venta* o se podrá hacer una compra parcial, generando dos movimientos: una *Venta* con los artículos seleccionados y una *Devolución* con los artículos que restan.

Si es una *Devolución*, únicamente se podrán ver los detalles de la devolución y se podrá utilizar el botón de la parte superior derecha para ir al movimiento del que proviene.

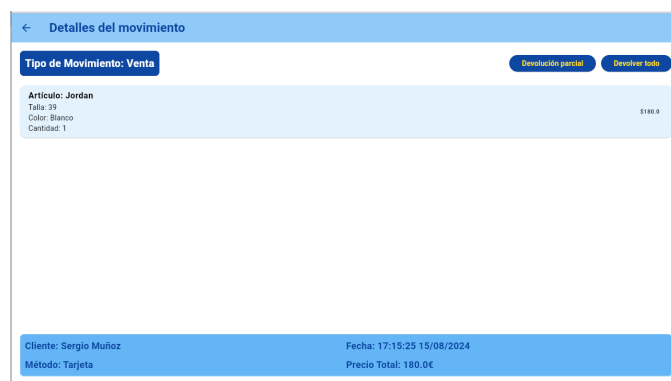


Figura 7.46: Pantalla de visualización de los detalles de un movimiento.

7.3.4.6. Pantalla de gráficos

Esta pantalla muestra un resumen de las ganancias de forma mensual o anual, según se elija en el desplegable de la parte superior derecha. Para conseguir ser más accesibles, las estadísticas positivas no solo se muestran con el color verde, sino que también se reflejan positivas en el eje. Las estadísticas negativas, se reflejan en la parte inferior del eje, además del color rojo. Esto se realiza con el pensamiento de las personas daltónicas que no puedan diferenciar los colores. De esta forma, quedan claras cuáles son las ganancias y cuáles son las pérdidas prestando atención al eje de la izquierda.

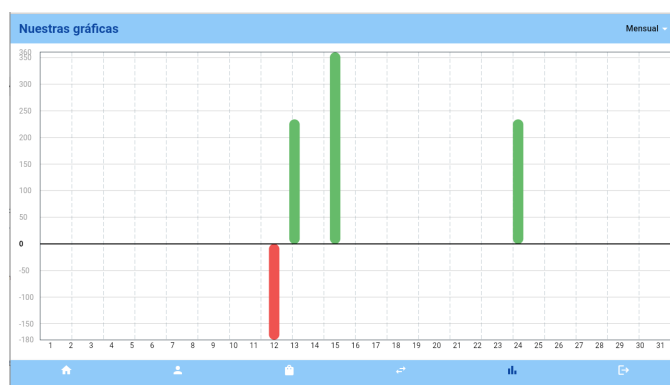


Figura 7.47: Pantalla de gráficos en modo mensual.

7.3.5. Pruebas de funcionalidad

Para testear la implementación que se ha llevado a cabo, se han realizado una serie de pruebas de funcionalidad:

- Comprobación de la representación de las ganancias del negocio en modo mensual.
- Comprobación de la representación de las ganancias del negocio en modo anual.

7.3.6. Revisión de la iteración

Tras terminar la iteración, se le mostró al usuario el resultado y estuvo contento con el cambio de interfaz y la representación de las gráficas. Le resultó sencillo navegar por la aplicación gracias al etiquetado superior y le resultó intuitiva para operar. Además, también agradeció que se hubiera mejorado la accesibilidad de la aplicación.

Capítulo 8

Base de datos e implementación

8.1. Creación y modificaciones de la base de datos

8.1.1. Creación de la base de datos

Para crear la base de datos se ha utilizado la herramienta de Supabase. Esta herramienta permite crear una base de datos y sus tablas correspondientes de forma sencilla. Además, tiene una fácil integración con Flutter, por lo que conectar la aplicación a la base de datos no fue complicado. Fue suficiente con añadir librerías y las claves de configuración.

Una vez configurada la base de datos dentro del proyecto, se genera una instancia que se utilizará cada vez que sea necesario hacer operaciones en la base de datos:

```
app_tfg > lib > utils > common.dart > ...  
1  import 'package:supabase_flutter/supabase_flutter.dart';  
2  
3  final client = Supabase.instance.client;  
4  |
```

Figura 8.1: Instancia de la base de datos.

8.1.2. Tablas de la base de datos

Las tablas que están dentro de la base de datos son:

- **clientes:** En esta tabla se almacena toda la información relativa a los clientes. Podemos encontrar campos como el nombre, el teléfono, un comentario para añadir más información y una cartera, que permitirá que el cliente pueda tener dinero negativo o positivo en la tienda.
- **categorias:** En esta tabla se crean categorías para organizar los artículos dentro de la tienda. Sus atributos `tieneTamano`, `tieneDescripcion`, `tieneMaterial`, `tieneGenero`, son booleanos que permiten personalizar los atributos que tendrá un artículo. Si no se seleccionan alguno de estos atributos, todos los artículos que se creen bajo esa categoría no tendrán ese atributo. El atributo `nombre` indica el nombre de la categoría.
- **articulos:** En esta tabla se almacena la información de cada uno de los artículos de la tienda. Sus atributos fijos son el nombre, el precio y la subcategoría (que permite categorizar más fino dentro de una categoría y posteriormente filtrar por dichas subcategorías). El resto de atributos son opcionales y se elegirán en la creación de la categoría de dicho artículo. Un artículo representa un modelo, dentro de una prenda podrán haber varias tallas o varios colores, lo que se contempla en la siguiente tabla.
- **tallas:** En esta tabla se registran las tallas y los colores disponibles de cada uno de los artículos. Para cada talla se especifica un color, la cantidad actual y la cantidad mínima. La cantidad mínima sirve para detectar cuando un artículo necesita ser renovado. Cuando se tienen igual o menos artículos de los especificados en la cantidad mínima, se mete en una lista de renovación de stock para indicar que deben ser renovados.
- **movimientos:** En esta tabla se registra toda la información relacionada con los movimientos de la tienda. Se registra el precio total del movimiento, el cliente asignado, el método de pago, la fecha, el tipo de movimiento y el movimiento anterior, este atributo sirve para relacionar unos tickets con otros. Cuando se hace una devolución, se le asigna el movimiento anterior del que procede para poder ver cuál es su procedencia. Además, a las ventas / préstamos también se le asignan las devoluciones para una mejor sincronización.
- **articulosMov:** En esta tabla se registran todos los artículos que pertenecen a cada movimiento. Es la tabla resultante de una relación

muchos a muchos. Además, también se almacena como información adicional, la cantidad comprada de esa talla y el precio parcial.

8.1.3. Modificaciones intermedias de la base de datos

Debido al cambio que solicitó el cliente con las tallas, se tuvo que crear una nueva tabla, la tabla tallas. Esto permitió que un artículo pudiera tener distintas tallas o distintos colores.

8.1.4. Diagrama final del diseño de la base de datos

En este diagrama podemos ver los atributos que posee cada una de las tablas y como se relacionan entre ellas. Las líneas discontinuas indican las claves externas y los atributos con la llave a la izquierda representan las claves primarias de cada tabla.

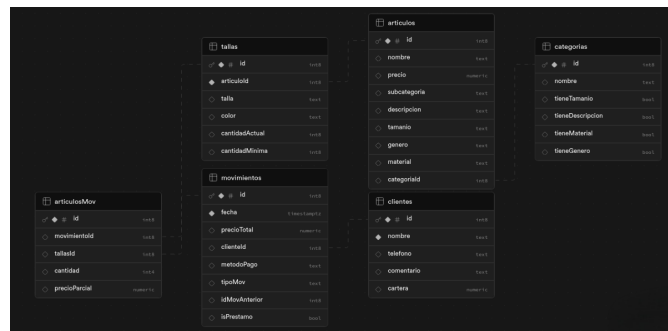


Figura 8.2: Diagrama de la base de datos.

8.2. Diagrama de clases

Tras finalizar la implementación de la aplicación, se ha construido un diagrama de clases para poder observar las clases que componen el proyecto y las relaciones entre estas. También se pueden ver los atributos y los métodos que tiene cada una de las clases.

8.2.0.1. Common

No es una clase, es un documento donde se declara la instancia de la base de datos. Este documento se importa en cada una de las clases que necesitan la base de datos. De esta forma, hay una única instancia de la base de datos en vez de múltiples instancias.

8.2.0.2. ChangeNotifier

Es una clase que extiende *ChangeNotifier*, esto permite notificar cambios dentro de la aplicación para poder actualizar las pantallas en los momentos precisos.

8.2.0.3. Main

En esta clase se define el tema de la aplicación, los colores que se utilizarán en cada widget por defecto. Además, mediante el método *getAuth()* se comprueba si el usuario se ha autenticado. Si no está autenticado, lo redirige a la pantalla de *Login*, mientras que si lo está, lo redirige a la *BottomNav*.

8.2.0.4. Login

Esta clase, recopila la información mediante los campos de texto de la interfaz gráfica. Cuando tiene el email y la contraseña, los pasa como parámetro al método *signInWithPassword(email, password)* que proporciona la base de datos. Si las credenciales son correctas, el *Main* redirigirá al usuario a la *BottomNav*, mientras que si son incorrectas, mostrará un diálogo informando del error.

8.2.0.5. BottomNav

Esta es la clase de la barra de navegación. Por defecto, cada vez que se inicie sesión, la barra de navegación redirigirá al usuario a la pantalla principal. La barra de navegación tiene 6 iconos con 5 pantallas distintas: la *MainPage*, la *ClientView*, la *CategoryView*, la *MovimientosView* y la *ChartScreen*. El sexto icono se corresponde con el cerrado de sesión. Al pulsar esa opción, aparece un diálogo que pregunta si se quiere cerrar la sesión. Al confirmar el diálogo, se utiliza el método *signOut()* que proporciona la base de datos para cerrar la sesión.

8.2.0.6. Mainpage

Esta pantalla muestra las ganancias diarias y dos botones que redirigen a la pantalla de *newMovement*, uno para *Nueva venta* y otro para *Nuevo préstamo*. La clase *newMovement* necesita un booleano para saber si es venta o préstamo, *isVenta*, que se le proporciona en base al botón que se seleccione.

Para calcular las ganancias diarias, se utiliza el método *fetchDailyEarnings()*. Este método recorre todos los movimientos que se han hecho en el día actual. Suma en una variable los precios totales de las ventas y resta los precios totales de las devoluciones. No se tienen en cuenta los precios totales de los préstamos porque no es dinero que haya entrado en la tienda. Finalmente, asina esa variable a la variable que se muestra por pantalla, *dailyEarnings*.

8.2.0.7. NewMovement

En esta pantalla, se necesitan varios métodos para cumplir las especificaciones anteriormente comentadas:

- **fetchArticles(String query)**: Busca artículos dentro de la tabla cuyo nombre contenga la cadena de caracteres proporcionada por parámetro, *query*.
- **fetchClients(String query)**: Busca clientes dentro de la tabla cuyo nombre contenga la cadena de caracteres proporcionada por parámetro, *query*.
- **updateQuantity(int articleId, String size, String color, int newQuantity, int currentQuantity)**: Actualiza la cantidad actual de una artículo para una talla y color específico.
- **fetchSizesAndColors(int articleId)**: Devuelve las tallas y los colores disponibles para un determinado artículo.
- **fetchCurrentQuantity(int articleId, String size, String color)**: Devuelve la cantidad actual de un artículo para una talla y un color específico.
- **insertArticulosMov(int movimientoId)**: Se hace un *insert* para cada artículo de un movimiento.
- **insertVenta()**: Añade a la tabla *Movimientos* una venta con los datos recolectados de los campos de la interfaz gráfica.

- **updateClientCartera(int clientId, double amount):** Actualiza la cartera de un cliente, restándole la cantidad especificada. Esta función se utiliza cuando se crea un nuevo préstamo.
- **insertPrestamo():** Se añade a la tabla (Movimientos) un préstamo con los datos recolectados de los campos de la interfaz gráfica.
- **addItem():** Comprueba que se hayan completado todos los campos del artículo de forma correcta y lo añade a la lista de artículos del movimiento que se añadirán al finalizar la venta o préstamo.
- **removeItem():** Elimina un artículo de la lista de compra.
- **calculateTotal():** Calcula el precio total del ticket de compra.
- **calculatePartialPrice():** Calcula el precio parcial de cada uno de los artículos de la lista.
- **incrementQuantity():** Incrementa la cantidad seleccionada del artículo especificado en una unidad.
- **decrementQuantity():** Decrementa la cantidad seleccionada del artículo especificado en una unidad.
- **onWillPop():** Este método se llama cuando se desea interrumpir el proceso de creación de un nuevo movimiento. Antes de salir, deshace todos los cambios que se hayan hecho hasta el momento.

8.2.0.8. ClientView

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState():** Método que se llama al cargar por primera vez la pantalla, hace una llamada a *loadStreams()*.
- **loadStream():** Hace una consulta a la base de datos para rellenar el *Stream* de clientes.
- **toggleDebts():** Cambia de estado la variable de filtrado de clientes que tienen deudas en la tienda.
- **deleteClient():** Elimina un cliente si no tiene movimientos vinculados.

8.2.0.9. ClientRegister

El método necesario para cumplir con las especificaciones de esta pantalla es:

- **addClientToDatabase(String name, String phone, String comments)**: Añade el cliente a la base de datos con los valores recogidos a través de los campos de la interfaz gráfica.

8.2.0.10. ClientEdit

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla, hace una llamada a *loadClientData(String clientId)*.
- **loadClientData(String clientId)**: Obtiene los datos del cliente seleccionado y asigna esos valores a los campos de texto de la interfaz gráfica.
- **editClient(String clientId, String name, String phone, String comments, double money)**: Edita los datos del cliente con la información aportada.

8.2.0.11. ClientDetail

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla, hace una llamada a *loadClientData(String clientId)*.
- **loadClientData(String clientId)**: Obtiene los datos del cliente seleccionado y asigna esos valores a los campos de texto de la interfaz gráfica.

8.2.0.12. CategoryView

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState():** Método que se llama al cargar por primera vez la pantalla. Inicializa el *Stream* de categorías mediante una llamada a la base de datos.
- **deleteCategory(String categoryId):** Elimina la categoría seleccionada si no tiene ningún artículo vinculado.

8.2.0.13. NewCategory

El método necesario para cumplir con las especificaciones de esta pantalla es:

- **addCategory():** Añade la categoría a la base de datos con los valores recogidos a través de los campos de la interfaz gráfica.

8.2.0.14. EditCategory

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState():** Método que se llama al cargar por primera vez la pantalla. Llama al método *fetchCategoryDetails()*.
- **dispose():** Hace el *dispose* del controlador del nombre de la categoría.
- **fetchCategoryDetails():** Obtiene los datos de la categoría seleccionada y asigna esos valores a los campos de la interfaz gráfica.
- **updateCategory():** Edita los datos de la categoría con la información aportada.

8.2.0.15. ArticleView

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState():** Método que se llama al cargar por primera vez la pantalla. Llama al método *fetchArticles()*.
- **fetchArticles():** Obtiene todos los artículos de la categoría seleccionada mediante una consulta a la base de datos. Además, actualiza el *Stream* de artículos con esta información.
- **deleteArticle(String articleId):** Elimina el artículo seleccionado si no está vinculado a ningún movimiento.

8.2.0.16. NewArticle

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla. Llama a los métodos *fetchCategoryDetails()* y *addSizeRow()*. El último método obliga a que haya mínimo una talla por artículo.
- **fetchCategoryDetails()**: Obtiene los datos de la categoría y adapta el formulario de nuevo artículo con las flags activas de dicha categoría.
- **addSizeRow()**: Añade nuevos controladores de texto cuando se invoca.
- **removeSizeRow(int index)**: Elimina los controladores de texto del índice especificado.
- **addTallasToDatabase()**: Añade las tallas que se han completado en el formulario a la base de datos. Hace un *insert* por cada talla, con sus detalles correspondientes.
- **addArticleToDatabase()**: Añade el artículo con los datos recogidos de los campos de la interfaz gráfica a la base de datos.
- **dispose()**: Hace el *dispose* de todos los controladores utilizados.

8.2.0.17. EditArticle

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla. Llama a los métodos *fetchCategoryDetails()* y *fetchArticleDetails()*.
- **fetchCategoryDetails()**: Obtiene los datos de la categoría y adapta los campos que se muestran según los flags activos.
- **fetchArticleDetails()**: Obtiene los datos del artículo e inicializa los campos de la interfaz gráfica con dichos valores.
- **fetchTallas()**: Obtiene los datos de todas las tallas vinculadas a ese artículo y lo almacena en una variable.
- **initializeSizeControllers()**: Inicializa todos los controladores de texto de las tallas con la información previamente obtenida.

- **addSizeRow()**: Añade nuevos controladores de texto cuando se invoca.
- **removeSizeRow(int index)**: Elimina los controladores de texto del índice especificado.
- **updateArticle()**: Actualiza los datos del artículo con la información modificada a través de la interfaz gráfica.

8.2.0.18. ArticleDetails

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla. Llama a los métodos *fetchCategoryDetails()* y *fetchArticleDetails()*.
- **fetchCategoryDetails()**: Obtiene los datos de la categoría y adapta los campos que se muestran según los flags activos.
- **fetchArticleDetails()**: Obtiene los datos del artículo e inicializa los campos de la interfaz gráfica con dichos valores.
- **fetchTallas()**: Obtiene los datos de todas las tallas vinculadas a ese artículo y lo almacena en una variable.
- **initializeSizeControllers()**: Inicializa todos los controladores de texto de las tallas con la información previamente obtenida.

8.2.0.19. Stock

El método necesario para cumplir con las especificaciones de esta pantalla es:

- **getLowStockArticles()**: Obtiene una lista de las tallas de los artículos cuya cantidad actual sea igual o inferior a la cantidad mínima establecida.

8.2.0.20. MovimientosView

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla, hace una llamada a *loadStreams()*.
- **loadStreams()**: Hace una consulta a la base de datos para rellenar el *Stream* de movimiento ordenados por fecha y el *Stream* de los clientes ordenados por nombre.
- **filterMovimientos()**: Vuelve a cargar la página, para que se actualice con los movimientos filtrados.
- **actualizarCarteraCliente(int clienteId, double monto)**: Actualiza la cartera del cliente, añadiendo el monto a la cantidad actual. Se llama cuando se elimina un movimiento de tipo préstamo.
- **deleteMovimiento(int movId)**: Se elimina el movimiento especificado si no está vinculado a ningún otro. Además, se actualiza el stock de los artículos que estuvieran dentro del movimiento, se borran las entradas en la tabla *ArticulosMov* que estuvieran relacionados con ese movId y, finalmente, se elimina el movimiento.
- **getClientNombre(int clienteId)**: Obtiene el nombre del cliente con ese ID.
- **fetchClients(String query)**: Obtiene los clientes cuyo nombre contengan la cadena de caracteres pasada por parámetro, *query*.
- **formatDate()**: Le da formato a la fecha actual.

8.2.0.21. MovementDetail

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **fetchMovimiento()**: Obtiene la información del movimiento que se va a mostrar en la pantalla mediante una consulta a la base de datos.
- **fetchItems()**: Obtiene todos los artículos vinculados a ese movimiento.
- **getClientNombre(int clienteId)**: Obtiene el nombre de un cliente a partir de su ID.
- **getArticuloNombre(int articuloId)**: Obtiene el nombre de un artículo a partir de su ID.
- **actualizarCarteraCliente(int clienteId, double monto)**: Actualiza la cartera del cliente, añadiendo el monto a la cantidad actual. Se llama cuando se devuelve un movimiento de tipo préstamo.

- **devolverTodo()**: Devuelve todos los artículos de una venta o un préstamo y actualiza el inventario. Si es un préstamo, actualiza la cartera del cliente. Si es una venta, el método de pago de la devolución será igual que el método de pago de la venta. Además, relaciona el movimiento original con el movimiento de devolución para una navegación rápida.
- **devolverParcial()**: Similar al método *devolverTodo()* pero permite seleccionar los artículos que se desean devolver. Permite seleccionar el método de pago de la devolución.
- **comprarParcial()**: Este método solo aplica para préstamos. Permite convertir un préstamo en venta, comprando los artículos seleccionados. Los artículos que no se seleccionan, formarán un movimiento de devolución y se actualizará la cartera del cliente con ese precio parcial devuelto. El movimiento original se vinculará con la devolución. Si no hay ninguna devolución porque se han comprado todos los artículos, no se vinculará a ningún movimiento.
- **onItemSelect(int tallasId, double precioParcial)**: Filtra la lista de artículos y solo se queda con aquellos que estén seleccionados.

8.2.0.22. ChartScreen

Los métodos necesarios para cumplir con las especificaciones de esta pantalla son:

- **initState()**: Método que se llama al cargar por primera vez la pantalla, hace una llamada a *fetchEarnings()*.
- **fetchEarnings()**: Dependiendo del valor seleccionado en el desplegable, mensual o anual, hace una llamada a *fetchMonthlyEarnings()* o *fetchAnnualEarnings()*.
- **fetchMonthlyEarnings()**: Determina los días que hay en el mes y, para cada uno de los días de ese mes, hace un recuento del dinero ganado, sumando el precio total de las ventas y restando el precio total de las devoluciones procedentes de ventas. Las devoluciones procedentes de préstamos no se restan porque es dinero que nunca entró en la tienda. Finalmente vuelve a cargar la pantalla para que se apliquen los cálculos.
- **fetchAnnualEarnings()**: Realiza un cálculo similar a el mensual, pero en vez de mostrar las ganancias día a día, las muestra mes a mes. Finalmente vuelve a cargar la pantalla para que se apliquen los cálculos.

- **onOptionChanged(String newValue):** Cambia el valor de la variable que controla el desplegable de la interfaz gráfica.

Capítulo 9

Conclusiones y trabajos futuros

9.1. Conclusiones

Al finalizar el proyecto de diseño e implementación de una aplicación para la gestión de una tienda minorista, se han alcanzado diversos objetivos importantes que nos permiten evaluar el éxito del trabajo realizado. A continuación se expondrán los objetivos que se han completado y las acciones que se han llevado a cabo para completarlos:

- **Objetivo 1 - Análisis de los comercios minoristas:** Este objetivo se ha llevado a cabo en el capítulo 3 de este documento. Gracias a este análisis, se ha podido comprender mejor los requisitos necesarios para desarrollar la aplicación.
- **Objetivo 2 - Estudio de aplicaciones similares:** Este objetivo se ha llevado a cabo en el apartado 3.6 de este documento. Mediante este estudio, se ha podido observar las funcionalidades necesarias que debía de tener la aplicación. Además, hemos podido encontrar funcionalidades innovadoras que no existieran en el mercado actual.
- **Objetivo 3 - Análisis de usabilidad y accesibilidad:** Este análisis se ha llevado a cabo en el apartado 3.4 de este documento. Gracias a este análisis, hemos podido desarrollar una interfaz gráfica usable y accesible. De esta forma, un mayor número de usuarios podrán utilizar la aplicación desarrollada.
- **Objetivo 4 - Análisis de tecnologías a utilizar:** Este objetivo se ha desarrollado en el capítulo 4 de este documento. Este análisis ha

permitido obtener las mejores tecnologías para el desarrollo de la aplicación. En ese capítulo se encuentra la justificación de dicha elección.

- **Objetivo 5 - Desarrollo e implementación de la aplicación:** Este objetivo se ha llevado a cabo en los capítulos 5, 6, 7 y 8 de este documento. La aplicación se ha desarrollado mediante iteraciones que permitían mejoras incrementales. Tras la última iteración, se consiguió un entregable que cumplía con los requisitos planteados al inicio del proyecto y aquellos resultantes de las valoraciones de las iteraciones.
- **Objetivo 6 - Validación de la aplicación por usuarios reales:** Este objetivo no se ha conseguido cumplir en su totalidad. Tras cada iteración, el resultado era valorado por un usuario real. Sin embargo, no se ha llegado a poner en un entorno real. Por ello, deberá de hacerse como trabajo futuro.

Una vez presentados los objetivos que se han cumplido a lo largo del proyecto, se presentan los principales logros conseguidos:

- **Funcionalidades certeras:** Al realizar una investigación sobre los negocios minoristas, he podido entender mejor cuáles son las necesidades a las que se enfrentan y poder definir unos requisitos más acertados a la hora del diseño de la aplicación. Además, tras cada iteración, se obtenía una validación de usuario que permitía saber si la aplicación se estaba desarrollando adecuadamente.
- **Novedad de la aplicación:** A diferencia de otras soluciones existentes en el mercado, la nueva aplicación desarrollada introduce funcionalidades innovadoras que ofrecen servicios únicos que hasta ahora no estaban disponibles. Una de las funcionalidades que ninguna otra aplicación del mercado contemplaba, es la introducción de un sistema de préstamos.
- **Compatibilidad con Android e IOS:** Al desarrollar la aplicación en Flutter con el lenguaje de programación Dart, la aplicación se puede adaptar a ambos sistemas operativos con facilidad. Esto permite que la aplicación pueda ser utilizada por un mayor número de usuarios.
- **Accesibilidad y usabilidad:** Se han tenido en cuenta los principios de accesibilidad y usabilidad para desarrollar la aplicación, obteniendo como resultado una aplicación sencilla de usar para una gran cantidad de usuarios.

9.2. Trabajos futuros

Durante el desarrollo de la aplicación, se han realizado una serie de pruebas de funcionalidad que aseguran unos resultados válidos. Sin embargo, para continuar con el desarrollo y mejora de la aplicación, se propone como trabajo futuro la introducción de la misma en un entorno real que permita evaluar su funcionamiento en condiciones auténticas. Este proceso no solo busca identificar posibles fallos o áreas de mejora, sino también adaptar la aplicación a las necesidades y expectativas de los usuarios finales. Para ello se deberá:

- **Seleccionar un entorno de prueba:** Se deberá de elegir un entorno de prueba que refleje de forma precisa el contexto en el que se espera que la aplicación sea utilizada. Por tanto, deberá de ser una pequeña tienda minorista sin informatizar con una pequeña clientela.
- **Involucramiento de usuarios reales:** Durante esta fase, se pondrá en marcha la aplicación en el entorno previamente seleccionado y se observará en detalle la interacción de dichos usuarios con la aplicación.
- **Recolección de datos y análisis:** Con los datos obtenidos de la fase anterior, se harán propuestas de mejora para que el funcionamiento de la aplicación se adapte al entorno.
- **Mejora de la implementación:** Esta etapa trata de aplicar las mejoras que se han diseñado en el código de la aplicación. Es decir, se harán físicos los cambios planteados de forma teórica.
- **Lanzamiento final:** Finalmente, tras someter la aplicación a unas pruebas con usuarios reales, se consigue un resultado más ajustado a las necesidades reales.

Tras completar esta valoración con usuarios finales, el resultado de la aplicación quedaría mejorado.

Al finalizar la implementación de la aplicación, se deberá hacer un manual de usuario para explicar todas las funcionalidades de la aplicación. De esta forma, será mas sencillo el entendimiento del funcionamiento de la aplicación.

Apéndice A

Descripción de casos de uso

Caso de Uso	inicioSesión		CU1
Actores	Encargado		
Tipo	Obligatorio		
Referencias	RF1		
Precondición	El encargado debe estar registrado en el sistema.		
Postcondición	El encargado entra en el sistema.		
Autor	Julia Cano		
Fecha	02/03/2024	Versión	1.0

Propósito	El encargado debe ser capaz de acceder al sistema.
Resumen	El encargado introduce sus credenciales y entra en el sistema.

Curso normal			
1.	Encargado: Introduce su nombre de usuario y su contraseña.	2.	Verifica que las credenciales sean correctas
3.	Encargado: Entra en el sistema		

Cursos alternos	
2a	Las credenciales no son correctas.

Cuadro A.1: Caso de uso - Inicio de sesión

Caso de Uso	cierreSesión	CU2
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF2	
Precondición	El encargado debe estar dentro del sistema.	
Postcondición	El encargado sale del sistema.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de cerrar sesión.
Resumen	El encargado pulsa la opción de cierre de sesión y sale del sistema.

Curso normal			
1.	Encargado: Solicita cierre de sesión.	2.	Cerrar sesión.
3.	Encargado: Sale del sistema.		

Cursos alternos	

Cuadro A.2: Caso de uso - Cierre de sesión

Caso de Uso	nuevoArtículo	CU3
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF3	
Precondición		
Postcondición	El artículo es creado.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de introducir nuevos artículos en la tienda.
Resumen	El encargado introduce los datos del artículo y lo registra en el sistema.

Curso normal			
1.	Encargado: Quiere introducir un nuevo artículo.	2.	Solicitar datos del nuevo artículo.
3.	Encargado: Introduce datos necesarios del artículo.	4.	Registrar el artículo.
		5.	Actualizar lista de artículos existentes.

Cursos alternos	
3a	El encargado introduce la información en un formato erróneo.

Cuadro A.3: Caso de uso - Introducción de un nuevo artículo

Caso de Uso	editarArtículo	CU4
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF4	
Precondición	El artículo a editar debe estar registrado en el sistema.	
Postcondición	La información del artículo es modificada.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de editar información de los artículos de la tienda.
Resumen	El encargado introduce los nuevos datos del artículo y lo edita.

Curso normal			
1.	Encargado: Quiere editar un artículo.	2.	Solicitar información a editar del artículo.
3.	Encargado: Introduce nuevos datos del artículo.	4.	Actualizar el artículo.

Cursos alternos	
3a	El encargado introduce la información en un formato erróneo.

Cuadro A.4: Caso de uso - Edición de un artículo existente

Caso de Uso	eliminarArtículo	CU5
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF5	
Precondición	El artículo debe estar registrado en el sistema.	
Postcondición	El artículo es eliminado.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de eliminar los artículos de la tienda.
Resumen	El encargado elimina el artículo.

Curso normal			
1.	Encargado: Quiere eliminar un artículo.	2.	Enviar mensaje de confirmación.
3.	Encargado: Aceptar la eliminación.	4.	Verificar que el artículo no esté vinculado a movimientos existentes.
		5.	Eliminar el artículo.

Cursos alternos	
3a	El encargado rechaza el mensaje de confirmación del sistema.
4a	El sistema encuentra movimientos vinculados a ese artículo.

Cuadro A.5: Caso de uso - Eliminación de un artículo

Caso de Uso	visualizarDatosArtículos	CU6
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF6	
Precondición	Debe de existir el artículo.	
Postcondición	Los datos del artículo son visualizados.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los datos de los artículos de la tienda.
Resumen	El encargado ve los datos.

Curso normal			
1.	Encargado: Quiere visualizar los datos de un artículo.	2.	Mostrar los datos del artículo dado.

Cursos alternos	

Cuadro A.6: Caso de uso - Visualización de los datos de un artículo

Caso de Uso	buscarArtículo	CU7
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF7	
Precondición	Debe existir el artículo.	
Postcondición	Muestra el artículo que coincida con el nombre especificado.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de buscar un artículo por nombre.
Resumen	El encargado introduce el nombre del artículo y el sistema le muestra las coincidencias.

Curso normal			
1.	Encargado: Introduce el nombre de un artículo.	2.	Filtrar la lista de artículos existentes en base al nombre introducido.
		3.	Mostrar el artículo coincidente con el nombre.

Cursos alternos	
3a	No existe ninguna coincidencia y no muestra nada.

Cuadro A.7: Caso de uso - Búsqueda de un artículo por nombre

Caso de Uso	categorizarArtículo	CU8
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF8	
Precondición	Deben existir categorías.	
Postcondición	Muestra los artículos de una determinada categoría.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de categorizar los artículos de la tienda.
Resumen	El encargado selecciona la categoría y se muestran todos los artículos que pertenezcan a esta.

Curso normal			
1.	Encargado: Selecciona la categoría.	2.	Filtrar la lista de artículos existentes en base a la categoría.
		3.	Mostrar los artículos de esa categoría.

Cursos alternos	
3a	No existen artículos de esa categoría y no muestra nada.

Cuadro A.8: Caso de uso - Categorización de un artículo

Caso de Uso	visualizarListaArtículos	CU8
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF9	
Precondición	Debe existir al menos un artículo.	
Postcondición	Muestra los artículos registrados en la tienda.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los artículos de la tienda.
Resumen	El encargado ve los artículos registrados en la tienda.

Curso normal			
1.	Encargado: Quiere ver los artículos registrados.	2.	Muestra todos los artículos existentes en la tienda.

Cursos alternos	

Cuadro A.9: Caso de uso - Visualización de la lista de los artículos existentes

Caso de Uso	visualizarListaRenovación	CU10
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF11	
Precondición		
Postcondición	Muestra los artículos que se deben de renovar para la tienda.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los artículos de la tienda.
Resumen	El encargado ve los artículos registrados en la tienda.

Curso normal			
1.	Encargado: Quiere ver los artículos que necesita comprar.	2.	Muestra los artículos con un stock escaso.

Cursos alternos	

Cuadro A.10: Caso de uso - Visualización de lista de renovación de artículos

Caso de Uso	nuevoCliente	CU11
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF12	
Precondición		
Postcondición	El cliente es creado.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de registrar nuevos clientes en la tienda.
Resumen	El encargado pregunta al cliente e introduce los datos en el sistema

Curso normal			
1.	Encargado: Quiere registrar un nuevo cliente.	2.	Solicita los datos del nuevo cliente.
3.	Encargado: Pregunta datos al cliente.		
4.	Cliente: Proporciona datos al encargado.		
5.	Encargado: Introduce los datos.	6.	Verifica el formato de los datos.
		7.	Registra el cliente
		8.	Actualizar la lista de clientes existentes.

Cursos alternos	
4a	El cliente se niega a dar ciertos dato.
6a	El formato de los datos es erróneo.

Cuadro A.11: Caso de uso - Registro de un nuevo cliente habitual

Caso de Uso	editarCliente	CU12
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF13	
Precondición	Debe existir el cliente previamente	
Postcondición	Los datos del cliente son modificados.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de modificar los datos del cliente.
Resumen	El encargado pregunta al cliente y modifica los datos.

Curso normal			
1.	Encargado: Quiere modificar los datos de un cliente existente.	2.	Solicita los nuevos datos del cliente.
3.	Encargado: Pregunta datos al cliente.		
4.	Cliente: Proporciona datos al encargado.		
5.	Encargado: Introduce los nuevos datos.	6.	Verifica el formato de los datos.
		7.	Actualiza los datos del cliente

Cursos alternos	
4a	El cliente proporciona los datos de forma errónea.
6a	El formato de los datos es erróneo.

Cuadro A.12: Caso de uso - Edición de los datos de un cliente existente

Caso de Uso	eliminarCliente	CU13
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF14	
Precondición	Debe existir el cliente previamente.	
Postcondición	El cliente se elimina.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de eliminar a un cliente.
Resumen	El encargado elimina al cliente.

Curso normal			
1.	Encargado: Quiere eliminar un cliente.	2.	Envía un mensaje de confirmación.
3.	Encargado: Acepta el mensaje de confirmación.	4.	Verifica que el cliente no esté vinculado a ningún movimiento activo.
		5.	Elimina al cliente y actualiza la lista de clientes existentes.

Cursos alternos	
3a	El encargado rechaza el mensaje de confirmación.
4a	El sistema identifica un movimiento vinculado al cliente.

Cuadro A.13: Caso de uso - Eliminación de un cliente

Caso de Uso	visualizarDatosCliente	CU14
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF15	
Precondición	Debe existir el cliente previamente.	
Postcondición	Se visualizan los datos almacenados de dicho cliente.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los datos de un cliente.
Resumen	El encargado selecciona un cliente y visualiza sus datos.

Curso normal			
1.	Encargado: Quiere visualizar los datos de un cliente.	2.	Muestra los datos del cliente.

Cursos alternos	

Cuadro A.14: Caso de uso - Visualización de los datos de un cliente

Caso de Uso	visualizarListaClientes	CU15
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF16	
Precondición	Debe existir al menos un cliente.	
Postcondición	Se visualiza la lista de clientes registrados en la tienda.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los clientes registrados en la tienda.
Resumen	El encargado visualiza la lista de clientes.

Curso normal			
1.	Encargado: Quiere visualizar la lista de clientes existentes.	2.	Muestra la lista de clientes.

Cursos alternos	

Cuadro A.15: Caso de uso - Visualización de la lista de clientes existentes

Caso de Uso	buscarCliente	CU16
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF17	
Precondición	Debe existir al menos un cliente	
Postcondición	Se obtienen los clientes coincidentes con la búsqueda.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de buscar clientes por nombre.
Resumen	El encargado introduce un nombre y obtiene los clientes coincidentes.

Curso normal			
1.	Encargado: Introduce un nombre de un cliente.	2.	Realiza una búsqueda entre los clientes disponibles.
		3.	Muestra los clientes coincidentes con la búsqueda.

Cursos alternos	
3a	No hay clientes coincidentes

Cuadro A.16: Caso de uso - Búsqueda de un cliente por nombre

Caso de Uso	filtrarClientes	CU17
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF18	
Precondición	Debe existir al menos un cliente	
Postcondición	Se obtienen los clientes con préstamos.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de filtrar clientes con préstamos.
Resumen	El encargado filtra y visualiza los clientes que tienen préstamos en la tienda.

Curso normal			
1.	Encargado: Quiere filtrar los clientes con préstamos.	2.	Realiza un filtrado de clientes con préstamos.
		3.	Muestra los clientes coincidentes con el filtrado.

Cursos alternos	
3a	No hay clientes coincidentes

Cuadro A.17: Caso de uso - Filtrado de clientes con préstamos

Caso de Uso	nuevaVenta	CU18
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF19	R10
Precondición	Deben existir los artículos de la venta	
Postcondición	Se registra una nueva venta.	
Autor	Julia Cano	
Fecha	02/03/2024	1.0

Propósito	El encargado debe ser capaz de introducir una venta.
Resumen	El encargado introduce los artículos que va a vender, rellena la información necesaria y registra la nueva venta en el sistema.

Curso normal			
1.	Encargado: Quiere introducir una nueva venta.	2.	Solicita los artículos vendidos.
3.	Encargado: Introduce los artículos y las cantidades.	4.	Solicita el cliente asignado.
5.	Encargado: Introduce el cliente.	6.	Solicita el método de pago.
7.	Encargado: Introduce el método de pago.	8.	Registrar la venta en la base de datos.
		9.	include «actualizaciónInventario»

Cursos alternos	
5a	No introduce el cliente, puesto que la asignación es opcional.

Cuadro A.18: Caso de uso - Introducción de una nueva venta

Caso de Uso	nuevoPréstamo	CU19
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF20	
Precondición	Deben existir los artículos del préstamo.	
Postcondición	Se registra un préstamo.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de introducir un préstamo.
Resumen	El encargado introduce los artículos que va a vender, rellena la información necesaria y registra el préstamo en el sistema.

Curso normal			
1.	Encargado: Quiere introducir un préstamo.	2.	Solicita los artículos a prestar.
3.	Encargado: Introduce los artículos y las cantidades.	4.	Solicita el cliente asignado.
5.	Encargado: Introduce el cliente.	6.	Registrar el préstamo en la base de datos.
		7.	include «actualizaciónInventario»

Cursos alternos	

Cuadro A.19: Caso de uso - Introducción de un nuevo préstamo

Caso de Uso	realizarDevolución	CU20
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF21	R10
Precondición	Deben existir la venta que vamos a devolver.	
Postcondición	Se realiza la devolución de los productos.	
Autor	Julia Cano	
Fecha	02/03/2024	1.0

Propósito	El encargado debe ser capaz de hacer una devolución.
Resumen	El encargado busca la venta y realiza la devolución de los productos correspondientes.

Curso normal			
1.	Encargado: Quiere realizar una devolución.		
2.	Encargado: Busca la venta que va a devolver.	3.	Solicita los artículos que se van a devolver.
4.	Encargado: Introduce los artículos y las cantidades.	5.	Solicita el método de devolución de dinero.
6.	Encargado: Introduce método de devolución de dinero.	7.	Registrar la devolución en la base de datos.
		8.	include «actualizaciónInventario»

Cursos alternos	
2a	No se encuentra la venta

Cuadro A.20: Caso de uso - Introducción de una devolución

Caso de Uso	eliminarMovimiento	CU21
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF22	R10
Precondición	Debe existir el movimiento previamente.	
Postcondición	El movimiento se elimina.	
Autor	Julia Cano	
Fecha	02/03/2024	1.0

Propósito	El encargado debe ser capaz de eliminar un movimiento.
Resumen	El encargado elimina al movimiento.

Curso normal			
1.	Encargado: Quiere eliminar un movimiento.	2.	Envía un mensaje de confirmación.
3.	Encargado: Acepta el mensaje de confirmación.		
		4.	Elimina al movimiento y actualiza la lista de movimientos existentes.
		5.	include «actualizaciónInventario».

Cursos alternos	
3a	El encargado rechaza el mensaje de confirmación.

Cuadro A.21: Caso de uso - Eliminación de un movimiento

Caso de Uso	visualizarDatosMovimiento	CU22
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF23	
Precondición	Debe existir el movimiento previamente.	
Postcondición	Se visualizan los datos almacenados de dicho movimiento.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los datos de un movimiento.
Resumen	El encargado selecciona un movimiento y visualiza sus datos.

Curso normal			
1.	Encargado: Quiere visualizar los datos de un movimiento.	2.	Muestra los datos del movimiento.

Cursos alternos	

Cuadro A.22: Caso de uso - Visualización de los datos de un movimiento

Caso de Uso	visualizarListaMovimientos	CU23
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF24	
Precondición	Debe existir al menos un movimiento.	
Postcondición	Se visualiza la lista de movimientos registrados en la tienda.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar los movimientos registrados en la tienda.
Resumen	El encargado visualiza la lista de movimientos.

Curso normal			
1.	Encargado: Quiere visualizar la lista de movimientos existentes.	2.	Muestra la lista de movimientos.

Cursos alternos	

Cuadro A.23: Caso de uso - Visualización de la lista de movimientos existentes

Caso de Uso	filtrarMovimientos	CU24
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF25	
Precondición	Debe existir al menos un movimiento	
Postcondición	Se obtienen los movimientos del tipo correspondiente.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de filtrar movimientos según su tipo.
Resumen	El encargado filtra y visualiza los movimientos de un tipo seleccionado.

Curso normal			
1.	Encargado: Introduce el tipo de movimiento por el que quiere filtrar.	2.	Realiza un filtrado de movimientos según ese tipo.
		3.	Muestra los movimientos coincidentes con el filtrado.

Cursos alternos	
3a	No hay movimientos coincidentes

Cuadro A.24: Caso de uso - Filtrado de movimientos según su tipo

Caso de Uso	buscarMovimientos	CU25
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF26	
Precondición	Debe existir al menos un movimiento	
Postcondición	Se obtienen los movimientos coincidentes con la búsqueda.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de buscar movimientos por fecha o vinculados a un cliente.
Resumen	El encargado introduce una fecha o un nombre y obtiene los movimientos coincidentes.

Curso normal			
1.	Encargado: Introduce una fecha o un nombre de un cliente.	2.	Realiza una búsqueda entre los movimientos disponibles.
		3.	Muestra los movimientos coincidentes con la búsqueda.

Cursos alternos	
3a	No hay movimientos coincidentes

Cuadro A.25: Caso de uso - Búsqueda de un movimiento por fecha o cliente

Caso de Uso	compraDesdePréstamo		CU26
Actores	Encargado		
Tipo	Obligatorio		
Referencias	RF27	Include	R10
Precondición	Debe existir el préstamo		
Postcondición	Se elimina el préstamo y se transforma en una venta.		
Autor	Julia Cano		
Fecha	02/03/2024	Versión	1.0

Propósito	El encargado debe ser capaz de generar una compra a partir de un préstamo.
Resumen	El encargado introduce los productos con los que el cliente se queda y genera una compra a partir de un préstamo anterior.

Curso normal			
1.	Encargado: Identifica el préstamo.		
2.	Encargado: Quiere realizar una venta a partir de ese préstamo.	3.	Solicita los productos que van a ser comprados.
4.	Encargado: Introduce los artículos con los que el cliente se queda.	5.	Solicita el método de pago.
6.	Encargado: Introduce el método de pago de preferencia del cliente.	7.	Genera una venta con los datos recopilados.
		8.	Elimina el movimiento préstamo anterior.
		9.	incluye «actualizaciónInventario»

Cursos alternos	
1a	El encargado no encuentra el préstamo

Cuadro A.26: Caso de uso - Generación de una compra a partir de un préstamo

Caso de Uso	visualizarCajaDiaria	CU27
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF28	
Precondición		
Postcondición	Se muestra el cómputo de dinero recopilado en el día.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar la caja diaria.
Resumen	El encargado ve el valor de la caja diaria.

Curso normal			
1.	Encargado: Quiere ver el valor de la caja diaria.	2.	El sistema calcula y muestra el valor de las ganancias diarias.

Cursos alternos	

Cuadro A.27: Caso de uso - Visualización de la caja diaria

Caso de Uso	visualizarGráficos	CU28
Actores	Encargado	
Tipo	Obligatorio	
Referencias	RF29	
Precondición	Debe existir al menos un movimiento.	
Postcondición	Se muestra una gráfica resumen de las ganancias mensuales o anuales.	
Autor	Julia Cano	
Fecha	02/03/2024	Versión 1.0

Propósito	El encargado debe ser capaz de visualizar gráficas que reflejen el progreso económico del negocio.
Resumen	El encargado ve las gráficas generadas.

Curso normal			
1.	Encargado: Quiere ver el valor de las gráficas.		
2.	Encargado: Elige el tipo de gráfica que desea ver, mensual o anual.	3.	El sistema calcula y muestra una gráfica que represente el progreso de la tienda durante el periodo de tiempo establecido.

Cursos alternos	

Cuadro A.28: Caso de uso - Visualización de gráficas

Apéndice B

Diagramas de secuencia

En esta sección vamos a especificar cómo opera el sistema ante las acciones del actor mediante diagramas de secuencia del sistema.

B.0.1. Apertura y cierre de sesión

B.0.1.1. Inicio de sesión

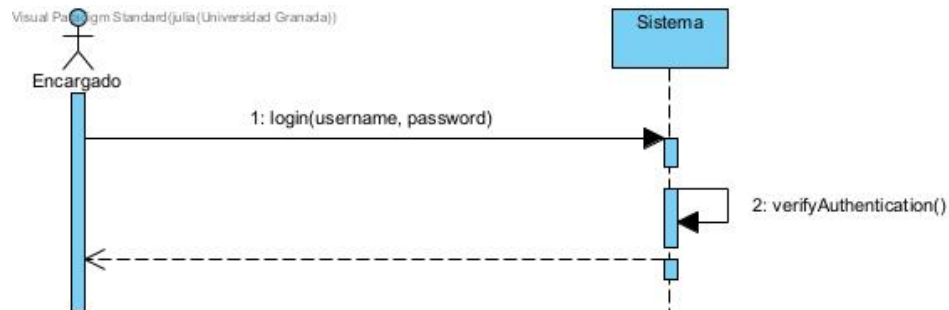


Figura B.1: Diagrama de secuencia de inicio de sesión

B.0.1.2. Cierre de sesión

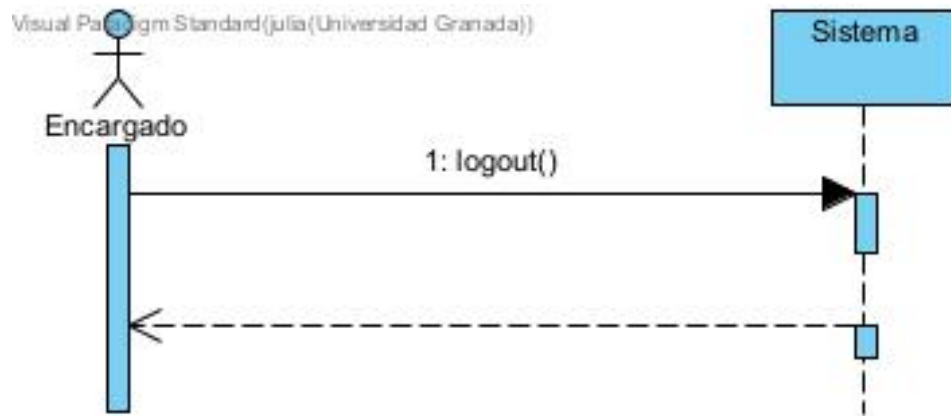


Figura B.2: Diagrama de secuencia de cierre de sesión

B.0.2. Gestión de artículos

B.0.2.1. Introducción de un nuevo artículo

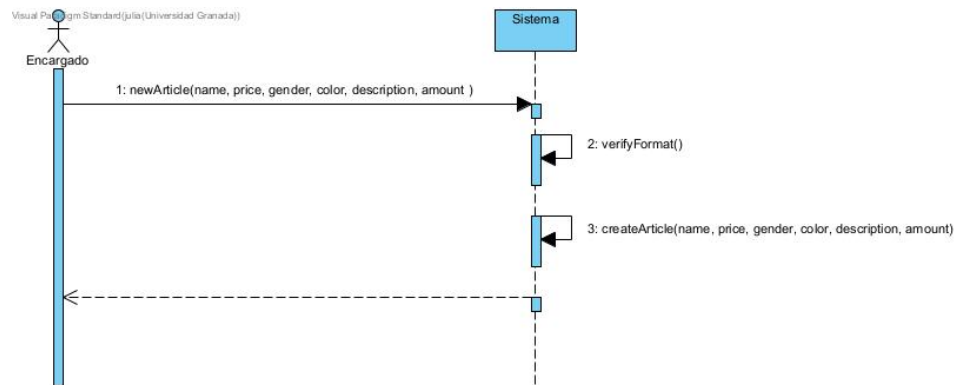


Figura B.3: Diagrama de secuencia de un nuevo artículo

B.0.2.2. Edición de un artículo existente

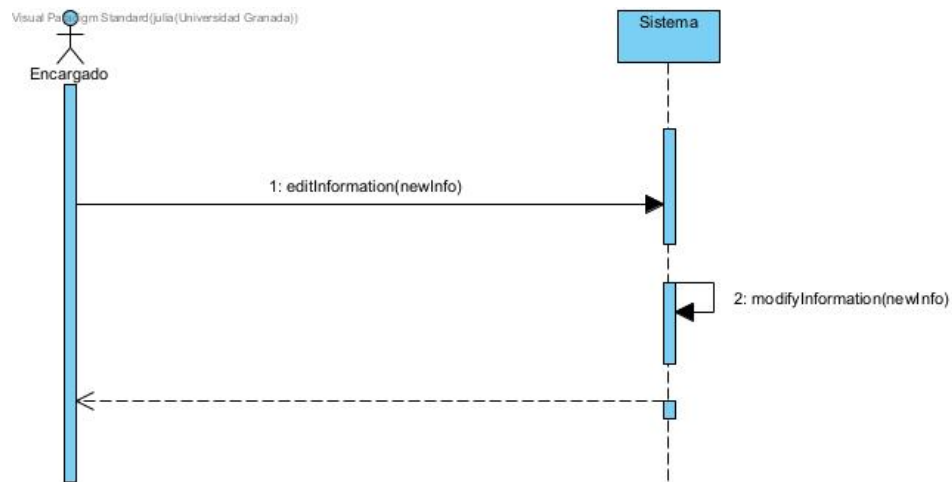


Figura B.4: Diagrama de secuencia de edición de un artículo

B.0.2.3. Eliminación de un artículo

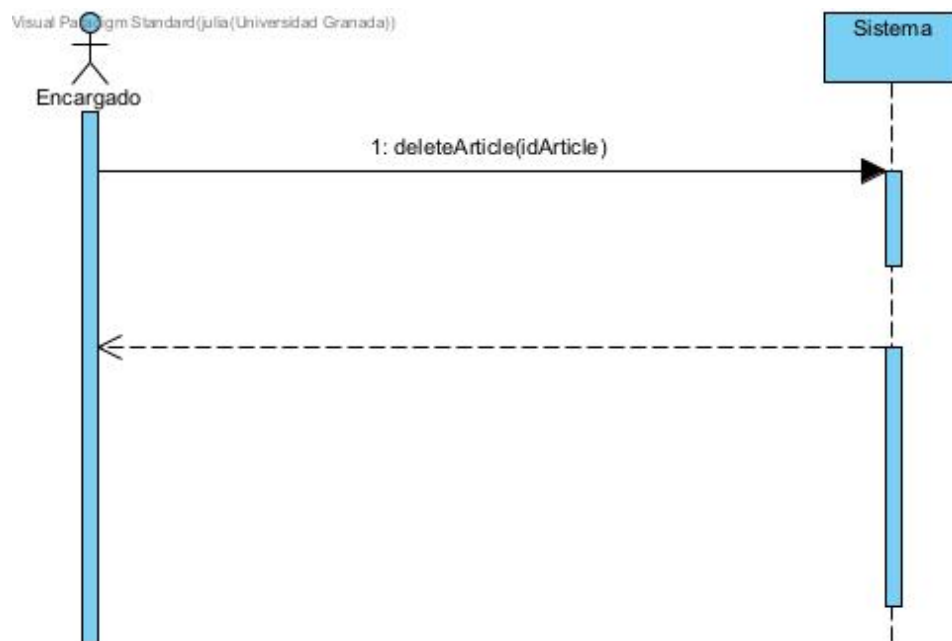


Figura B.5: Diagrama de secuencia de eliminación de un artículo

B.0.2.4. Visualización de los datos de un artículo

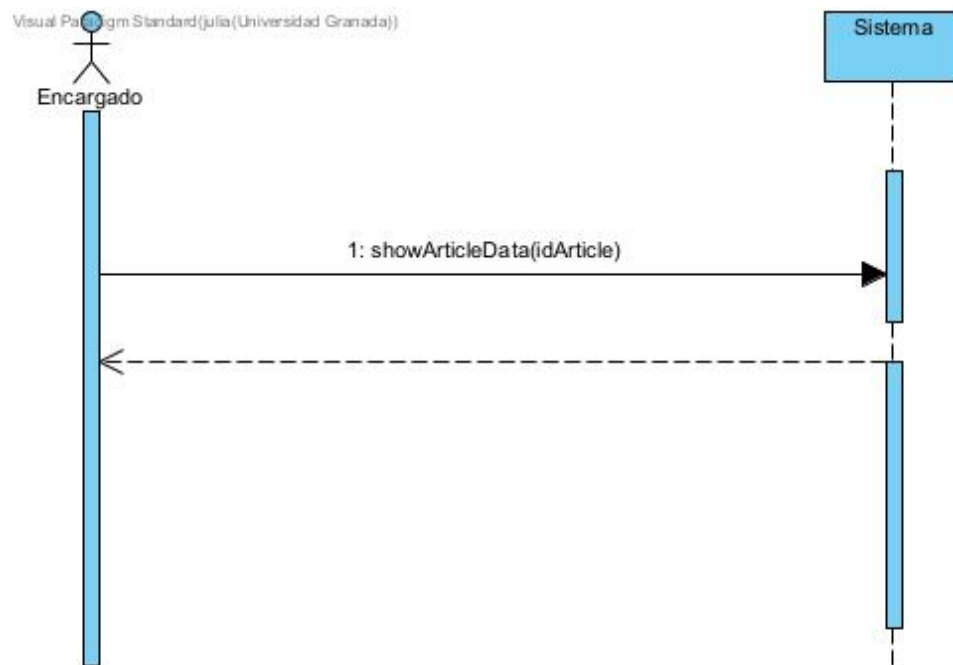


Figura B.6: Diagrama de secuencia de visualización de datos de un artículo

B.0.2.5. Búsqueda de un artículo por nombre

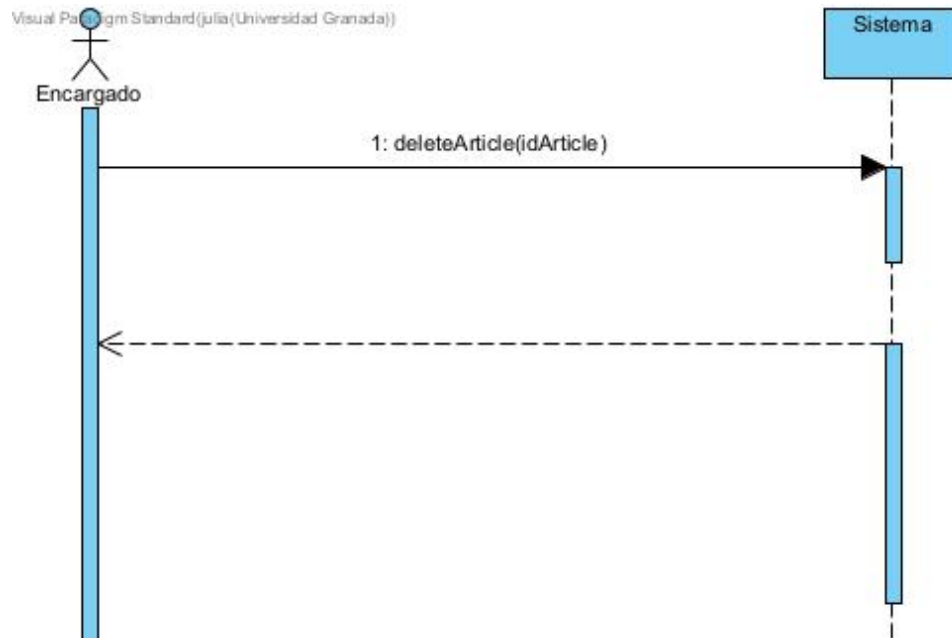


Figura B.7: Diagrama de secuencia de búsqueda de un artículo

B.0.2.6. Categorización de un artículo

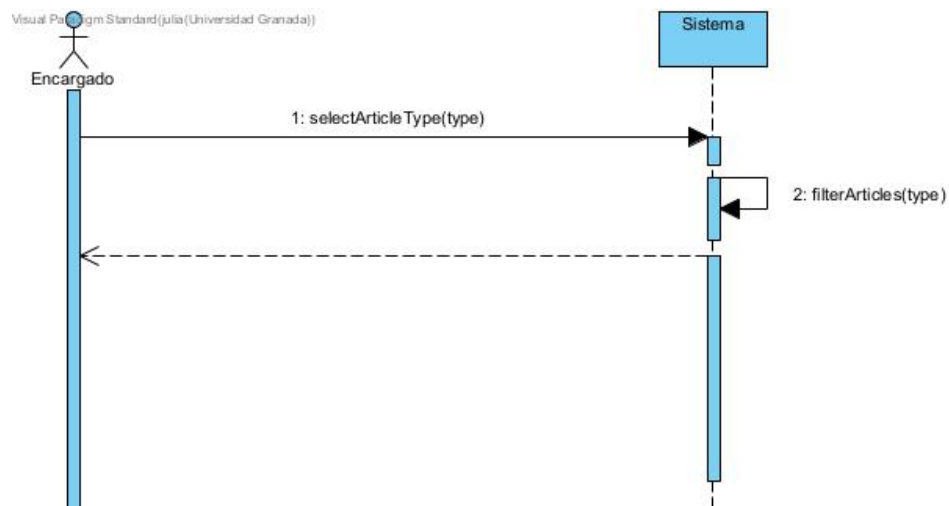


Figura B.8: Diagrama de secuencia de categorización de un artículo

B.0.2.7. Visualización de la lista de artículos



Figura B.9: Diagrama de secuencia de visualización de la lista de artículos

B.0.3. Gestión de inventario

B.0.3.1. Actualización del inventario de forma automática

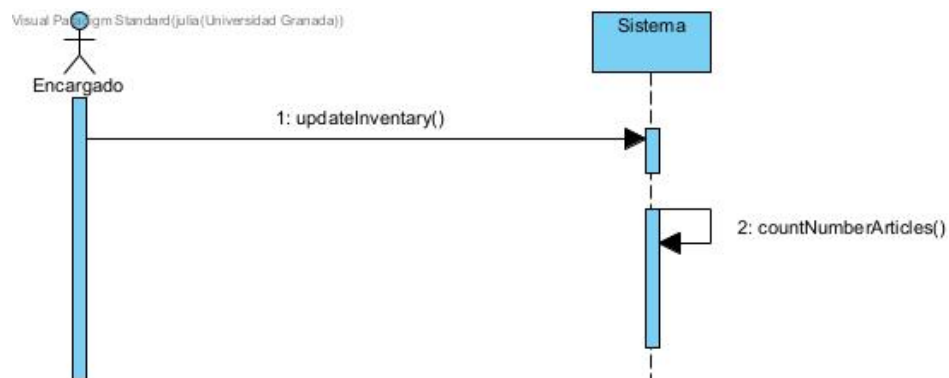


Figura B.10: Diagrama de secuencia de actualización de inventario

B.0.3.2. Visualización de la lista de renovación de artículos



Figura B.11: Diagrama de secuencia de visualización de la lista de renovación de artículos

B.0.4. Gestión de clientes

B.0.4.1. Registro de un nuevo cliente habitual

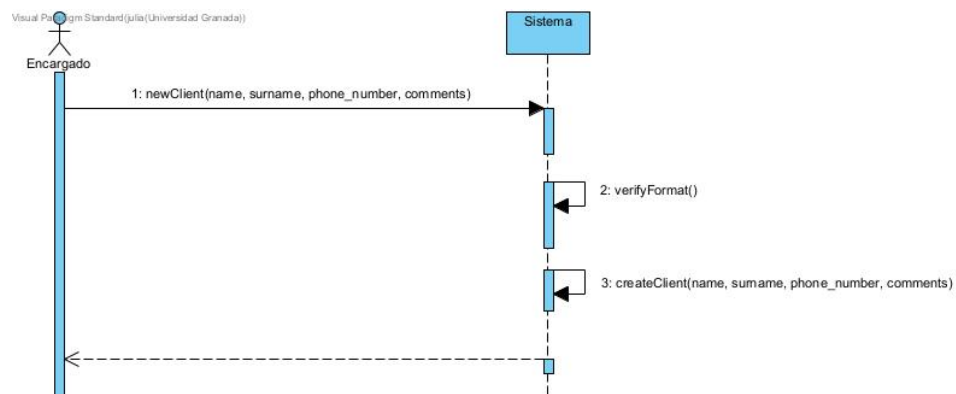


Figura B.12: Diagrama de secuencia de registro de un cliente

B.0.4.2. Edición de los datos de un cliente existente

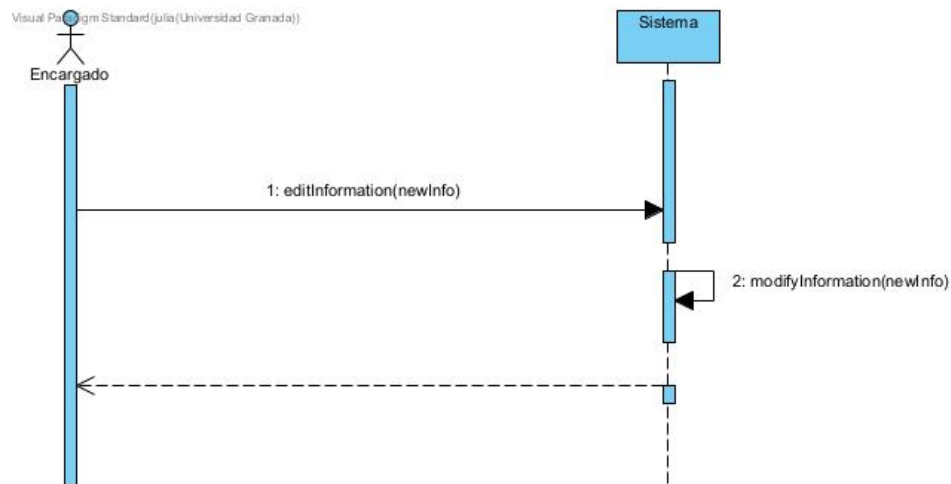


Figura B.13: Diagrama de secuencia de edición de los datos de un cliente

B.0.4.3. Eliminación de un cliente existente

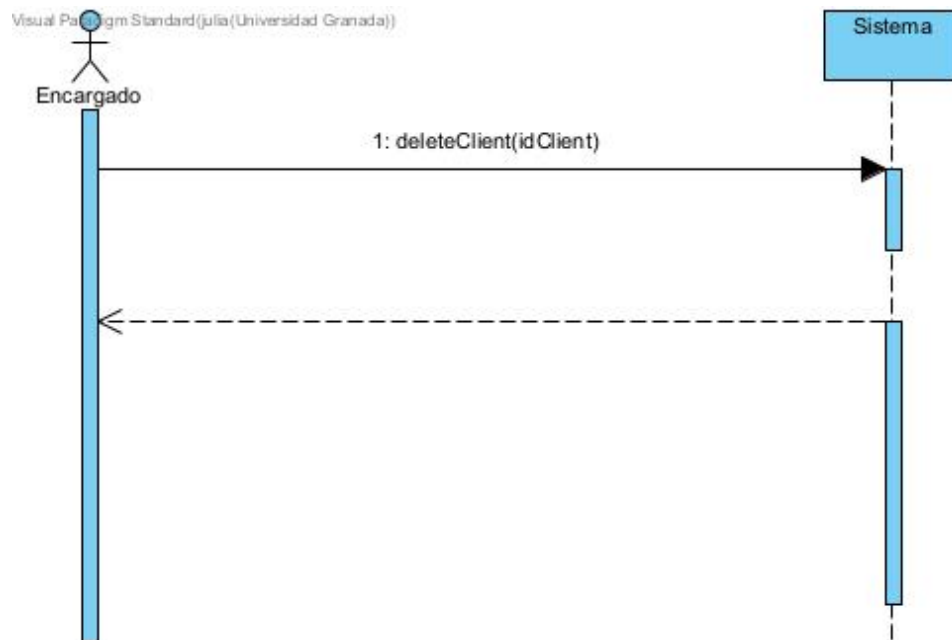


Figura B.14: Diagrama de secuencia de eliminación de un cliente

B.0.4.4. Visualización de los datos de un cliente

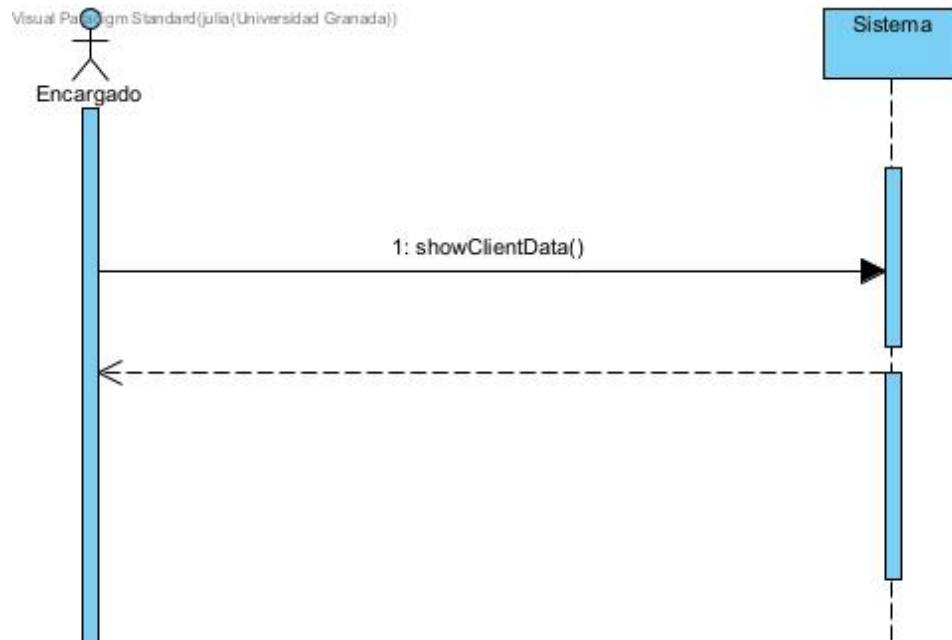


Figura B.15: Diagrama de secuencia de visualización de los datos de un cliente

B.0.4.5. Visualización la lista de clientes existentes



Figura B.16: Diagrama de secuencia de visualización de la lista de clientes

B.0.4.6. Búsqueda de un cliente por nombre

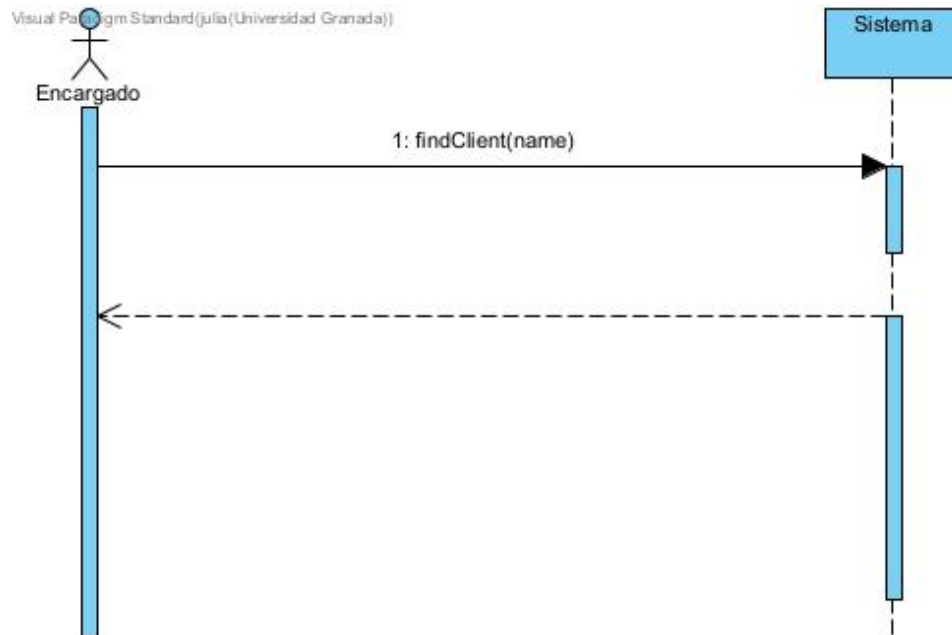


Figura B.17: Diagrama de secuencia de búsqueda de un cliente

B.0.4.7. Filtrado de clientes con préstamos

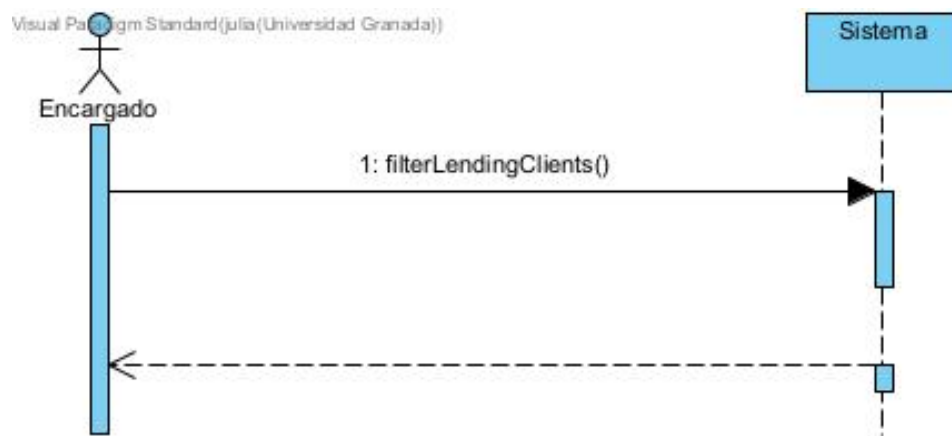


Figura B.18: Diagrama de secuencia de filtrado de clientes

B.0.5. Gestión de movimientos

B.0.5.1. Introducción de una nueva venta

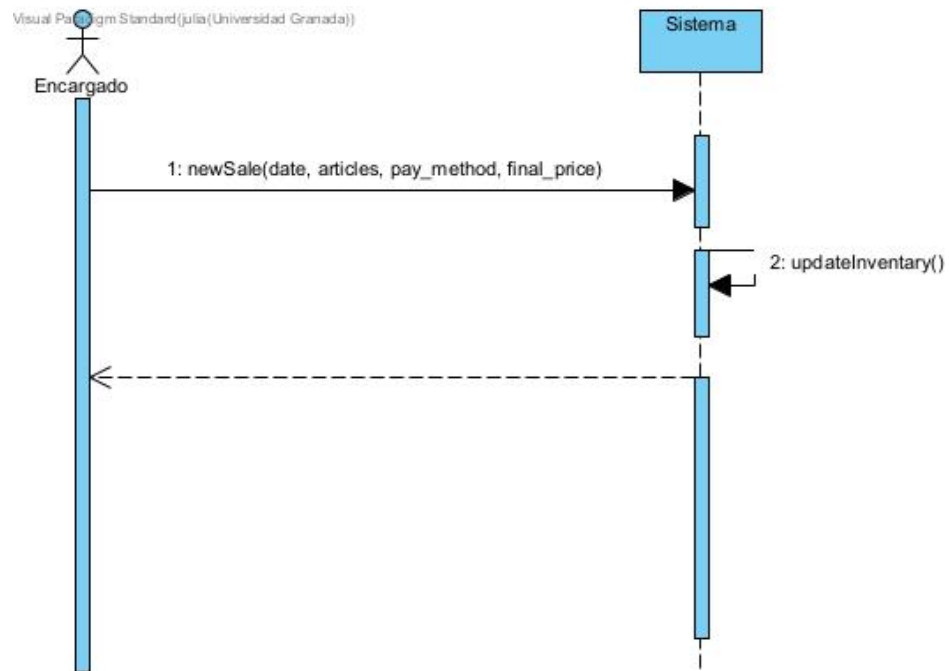


Figura B.19: Diagrama de secuencia de nueva venta

B.0.5.2. Introducción de un nuevo préstamo

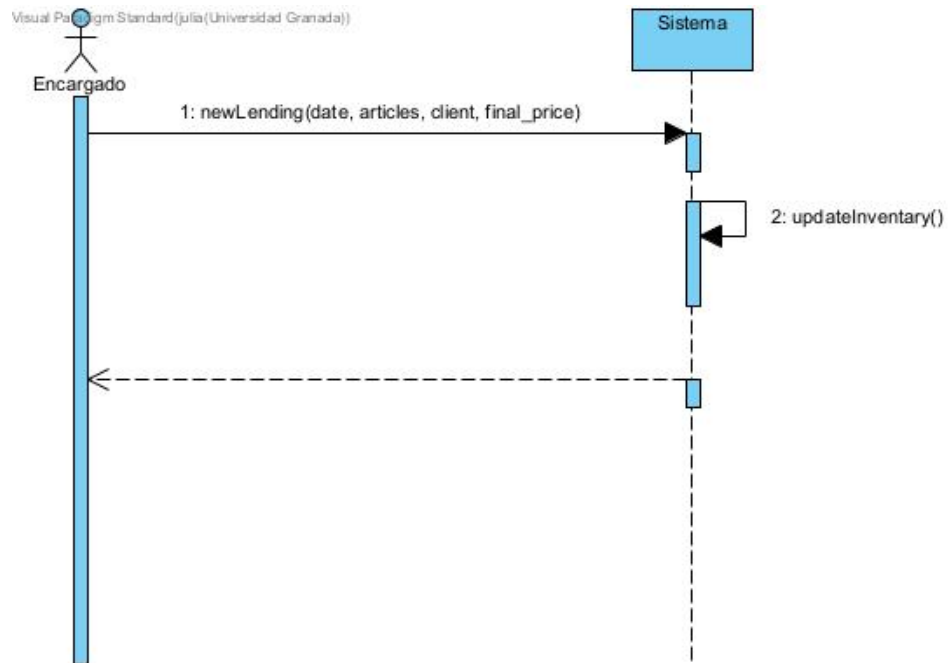


Figura B.20: Diagrama de secuencia de nuevo préstamo

B.0.5.3. Introducción de una nueva devolución

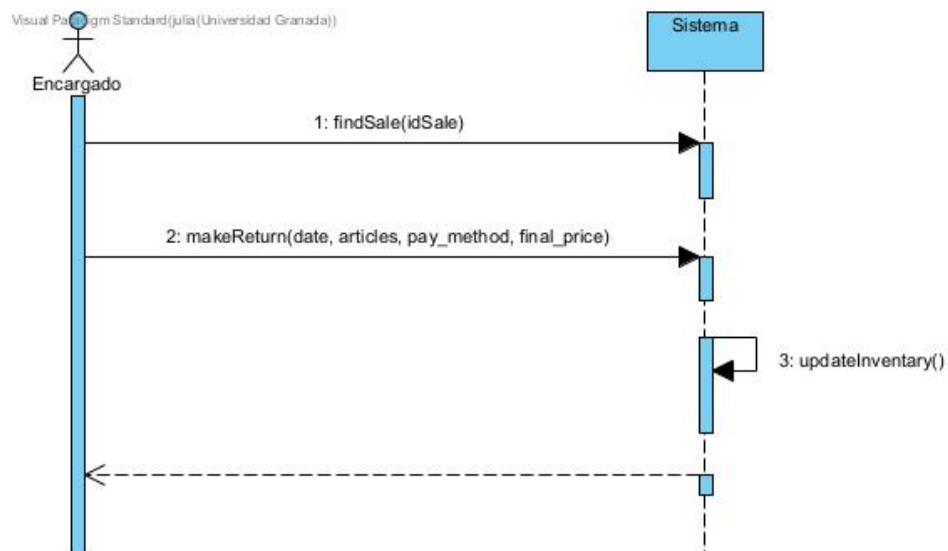


Figura B.21: Diagrama de secuencia de nueva devolución

B.0.5.4. Eliminación de un movimiento

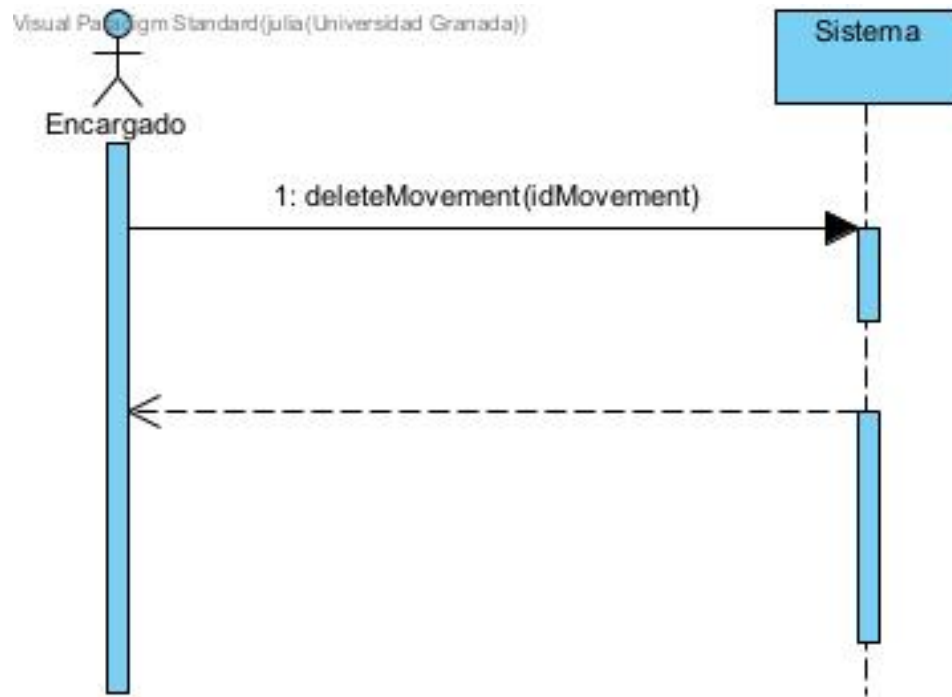


Figura B.22: Diagrama de secuencia de eliminación de un movimiento

B.0.5.5. Visualización de los datos de un movimiento

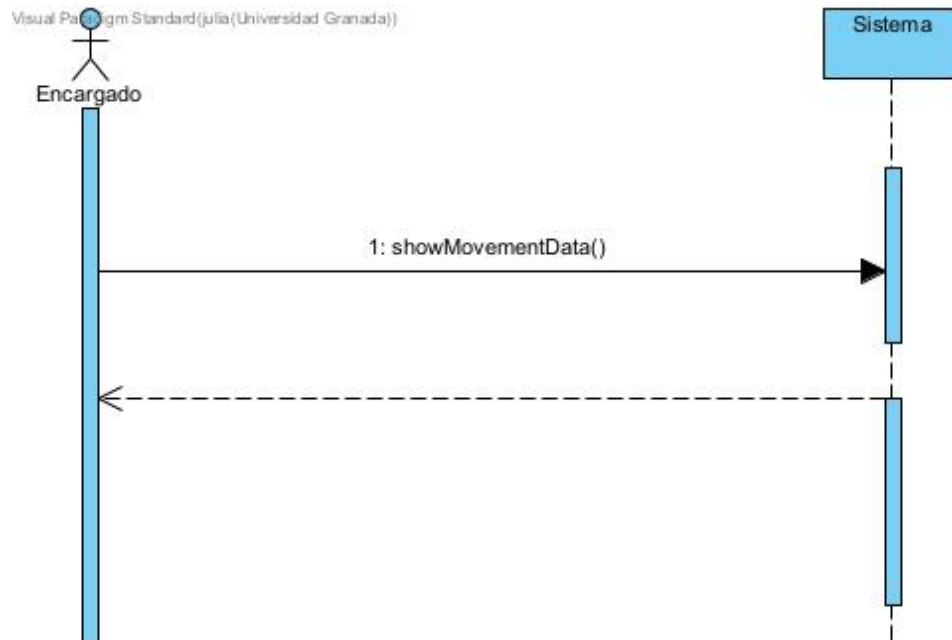


Figura B.23: Diagrama de secuencia de visualización de datos de un movimiento

B.0.5.6. Visualización de la lista de movimientos existentes

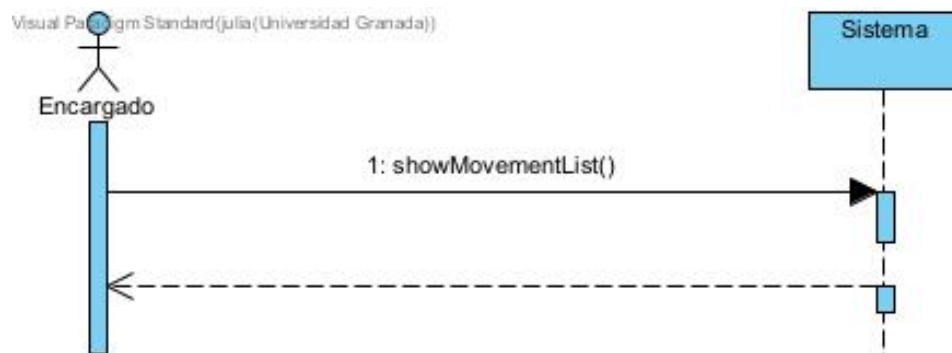


Figura B.24: Diagrama de secuencia de visualización la lista de movimientos

B.0.5.7. Filtrado de los movimientos según su tipo

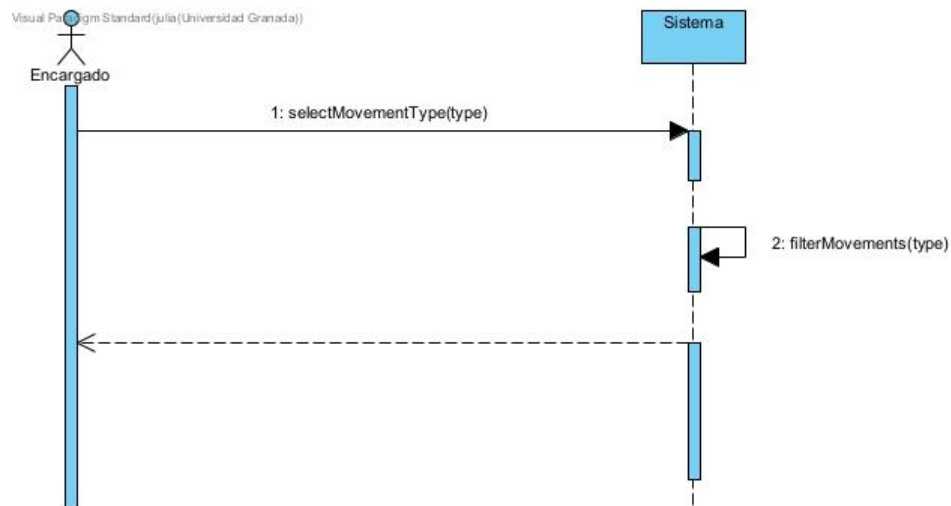


Figura B.25: Diagrama de secuencia del filtrado de movimientos

B.0.5.8. Búsqueda de movimientos por fecha o cliente

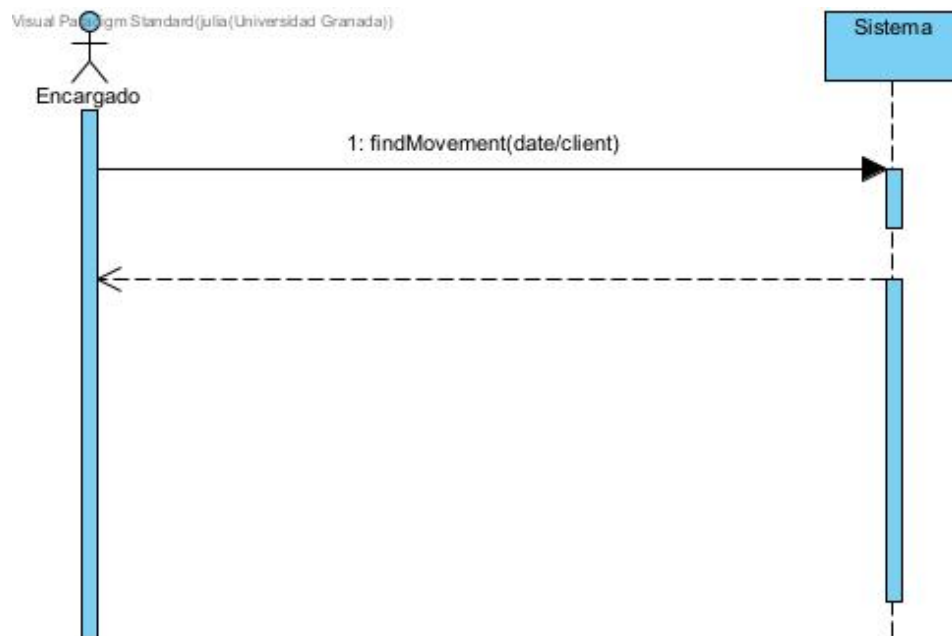


Figura B.26: Diagrama de secuencia de búsqueda de movimientos

B.0.5.9. Generación de una compra a partir de un préstamo

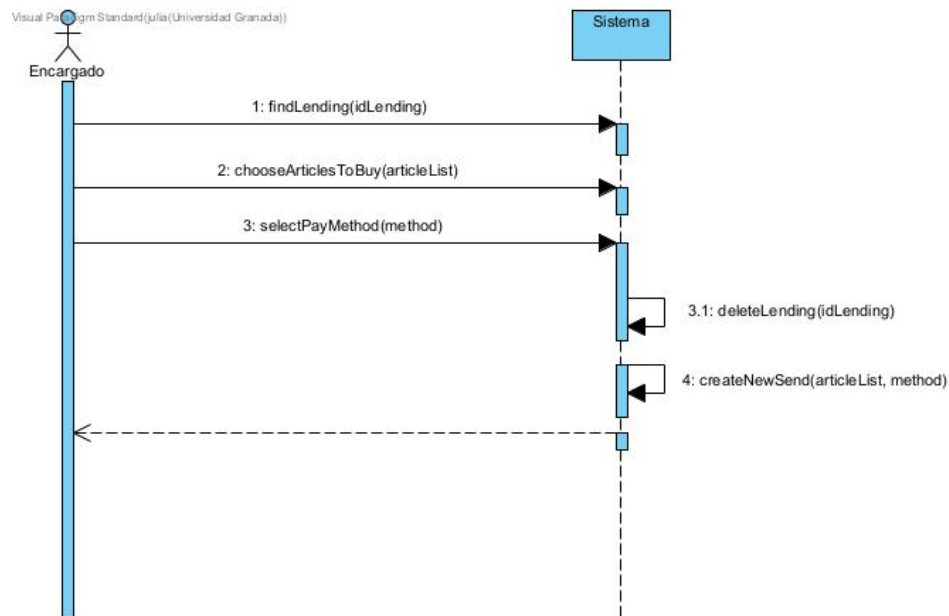


Figura B.27: Diagrama de secuencia de generación de una compra desde un préstamo

B.0.6. Gestión de resúmenes y gráficas

B.0.6.1. Visualización de la caja diaria

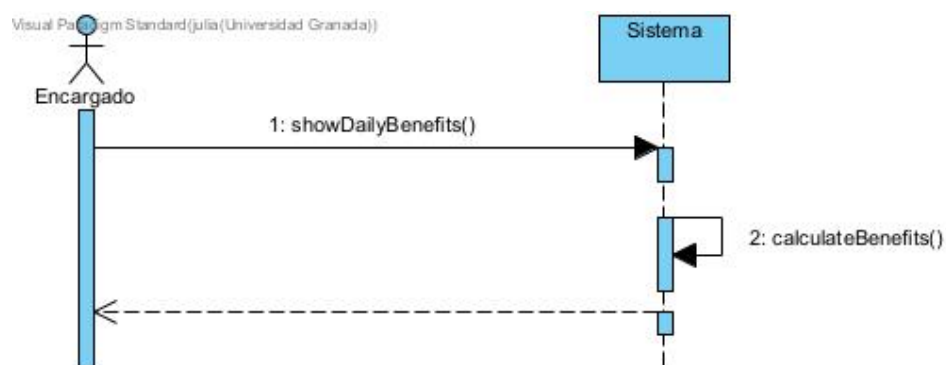


Figura B.28: Diagrama de secuencia de visualización de la caja diaria

B.0.6.2. Visualización de gráficos

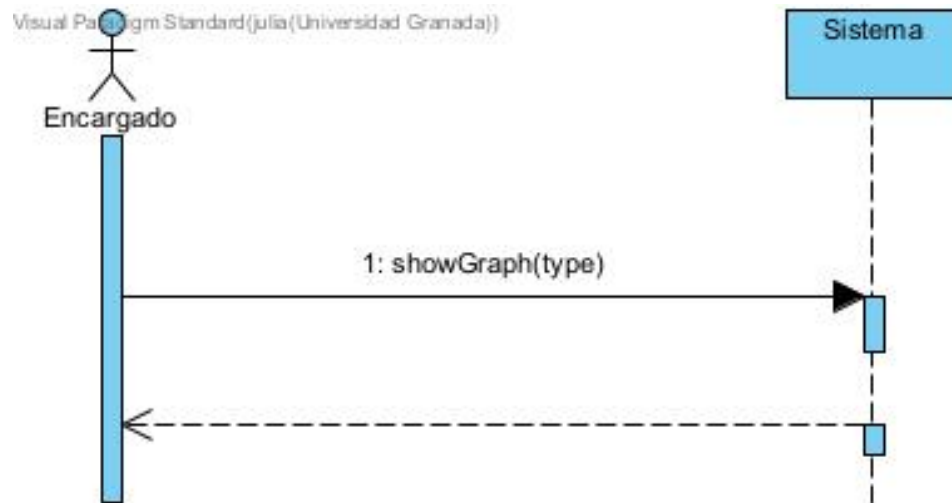


Figura B.29: Diagrama de secuencia de visualización de gráficos

Referencias

- Autonomosyemprendedores. (2024, 14 de Feb). *Máster empresariales: una opción para autónomos que tienen negocios*. <https://www.autonomosyemprendedor.es/articulo/actualidad/master-empresariales-opcion-que-autonomos-que-tienen-negocios/20240214104549034620.html>. (Accedido el [fecha de acceso aquí])
- BBVA. (2024). *Superando la brecha digital: soluciones para conectar a las personas mayores con la tecnología*. Descargado de <https://www.bbva.com/es/innovacion/superando-la-brecha-digital-soluciones-para-conectar-a-las-personas-mayores-con-la-tecnologia/> (Accedido el [fecha de acceso aquí])
- CeroIdeas. (2024). *Flutter*. Descargado de <https://ceroideas.es/que-es-flutter-caracteristicas-y-funcionalidades/>
- Clover. (2024). *Clover*. Descargado de <https://www.la.clover.com/>
- del Estado, J. (2018). *Boe-a-2018-16673*. Descargado de <https://www.boe.es/buscar/act.php?id=B0E-A-2018-16673>
- Dueñas Nogueras, J. (2014). *Calidad y servicios de proximidad en el pequeño comercio* (1.ª ed.). IC Editorial.
- Fernández Casado, P. E. (2018). *Usabilidad web: teoría y uso*. RA-MA Editorial.
- Fernández Casado, P. E. (2021). *Accesibilidad web: diseño de aplicaciones*. Ra-Ma.
- Gago Muñoz, M. (2023). *Dinamización del punto de venta en el pequeño comercio* (1.ª ed.). Bookwire GmbH.
- García, J. C. B. (2003). *La gestión moderna del comercio minorista: el enfoque práctico de las tiendas de éxito*. Esic Editorial.
- GoodNotes. (2024). *Goodnotes*. Descargado de <https://www.goodnotes.com/features>
- Highsmith, J. A. (2004). *Agile project management: Creating innovative products* (1st ed.). Addison-Wesley.
- Iberdrola. (Año de Publicación). *Qué es la brecha digital*. Descargado de <https://www.iberdrola.com/compromiso-social/que-es-brecha-digital>
- ISO. (2008). *Iso 9241-171: "guidance on software accessibility"*. Descargado de <https://www.iso.org/es/contents/data/standard/03/90/39080.html?browse=ics>
- ISO. (2019). *Iso 9241-11: usability: Definitions and concepts*. Descargado de <https://www.iso.org/standard/63500.html>
- LaTeX. (2024). *Latex*. Descargado de <https://es.wikipedia.org/wiki/LaTeX>

myGestión. (2024). *mygestión*. Descargado de <https://www.mygestion.com/caracteristicas-erp>

Paradigm, V. (2024). *Visual paradigm*. Descargado de <https://www.visual-paradigm.com/support/documents/vpuserguide.jsp>

Supabase. (2024). *Supabase*. Descargado de <https://supabase.com/>

Vargas Belmonte, A. (2023). *Gestión económica básica del pequeño comercio* (1.^a ed.). Bookwire GmbH.

W3C. (2023). *Accessibility principles*. Descargado de <https://www.w3.org/WAI/fundamentals/accessibility-principles/>

Wikipedia. (2024). *Minorista*. Descargado 2024-02-22, de <https://es.wikipedia.org/wiki/Minorista>