



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍA INFORMÁTICA

Grado en Ingeniería Informática

Mención en Ingeniería de Software

Optimización B2B de precios: Aplicación web de monitorización
de precios e inventario aplicando Análisis predictivo

AUTOR:

Martín Calvo, Enrique

Tutora:

Gonzalo Tasis, Margarita

Valladolid, 18 Junio

Agradecimientos

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

También quiero mencionar especialmente a mi madre, sin la cuál no podría haber realizado estos estudios. Le agradezco haber estado apoyándome en todo momento durante estos cinco años y aguantarme durante los exámenes haciéndome ver que todo era posible si trabajaba por ello.

Tampoco quiero olvidarme de todos los amigos que he hecho en la carrera por hacer más amenos estos años y ayudarme cuando ha sido necesario pero en especial a aquellos con los que comencé.

Por último, quiero agradecer a la persona que me ha servido como referente durante esta etapa y que me ha ayudado con sus consejos a llegar hasta aquí, Ignacio.

Resumen

En este documento se describe el desarrollo de un sistema para la gestión de inventario y proveedores, que incluye alertas sobre tendencias de precios y predicciones de precios para el siguiente periodo de tiempo, diseñado específicamente para pequeñas y medianas empresas. Siguiendo la metodología SCRUM, esta aplicación ha sido desarrollada para dispositivos Linux y se ejecuta localmente. Con esta aplicación, los clientes podrán realizar un seguimiento integral de los productos y precios de sus proveedores, lo que les permitirá tomar decisiones informadas basadas en dicha información.

Palabras Clave : Precios, proveedores, tendencias, visualización, modelo predictivo

Abstract

This document describes the development of a system for inventory and supplier management, which includes alerts on price trends and price predictions for the next time period, specifically designed for small and medium-sized enterprises. Following the SCRUM methodology, this application has been developed for Linux devices and runs locally. With this application, customers will be able to comprehensively track the products and prices of their suppliers, enabling them to make informed decisions based on this information.

Keywords : Price trends, suppliers, visualization, predictive model

Índice general

1. Introducción	1
1.1. Consideraciones iniciales	1
1.1.1. Contexto	1
1.1.2. Motivación	1
1.1.3. Objetivos	2
1.2. Análisis de aplicaciones similares	2
1.2.1. QUICKBOOKS	2
1.2.2. ODOO	3
1.2.3. SAP BUSINESS ONE	4
1.2.4. MINDEREST	5
1.2.5. OPTIMUS PRICE	5
1.2.6. Conclusiones extraídas del análisis	6
1.3. Estructura de la memoria	7
2. Planificación del Proyecto	9
2.1. Introducción	9
2.2. Descripción de la metodología utilizada	9
2.2.1. Características principales de la metodología Agile	9
2.2.2. Características principales de SCRUM	10
2.3. Planificación inicial del proyecto	10
2.3.1. Visión general de la planificación	10
2.3.2. Visión detallada de la planificación	11
2.4. Plan de gestión de riesgos	12
2.4.1. Gestión de riesgos	13
2.5. Presupuesto del Proyecto	18
2.5.1. Recursos de personal	18
2.5.2. Recursos materiales	19
2.5.3. Recursos tecnológicos	19
2.5.4. Recursos de localización	19
2.5.5. Recursos de infraestructura	19
2.5.6. Presupuesto inicial	20
2.5.7. Presupuesto real	20
3. Análisis	21
3.1. Introducción	21
3.2. Actores y Roles del sistema	21
3.2.1. Actores Principales	21
3.3. Requisitos	21
3.3.1. Requisitos funcionales	22
3.3.2. Requisitos funcionales de información	22
3.3.3. Requisitos no funcionales	24

3.4.	Modelo de Dominio	24
3.5.	Casos de Uso	25
3.5.1.	Presentación Casos de Uso	25
3.5.2.	Diagrama de Casos de Uso	26
3.5.3.	Descripción textual de los casos de uso	28
3.5.4.	Matriz de correspondencia de requisitos y casos de uso	45
3.5.5.	Realización en Análisis de los casos de uso	46
4.	Diseño	57
4.1.	Arquitectura del sistema	57
4.1.1.	Patrón arquitectónico Cliente-Servidor	57
4.1.2.	Arquitectura del cliente	58
4.1.3.	Arquitectura del servidor	61
4.2.	Otros patrones empleados	63
4.2.1.	Patrón Inyección de Dependencias (Dependency Injection)	63
4.2.2.	Patrón Repositorio (Repository Pattern)	64
4.2.3.	Patrón Servicio (Service Pattern)	64
4.2.4.	Patrón DAO/DTO	64
4.2.5.	Patrón Singleton	65
4.3.	Diagrama detallado de clases	65
4.4.	Modelo de datos	66
4.4.1.	Diccionario de datos	66
4.5.	Privacidad desde el diseño y por defecto	68
4.5.1.	Spring Security con JWT	69
4.6.	Diseño de la interfaz de usuario	69
4.6.1.	Usabilidad	69
4.6.2.	Prototipos de la interfaz	70
4.7.	Despliegue de la aplicación	72
4.8.	Decisiones de Diseño	72
5.	Implementación	73
5.1.	Tecnologías utilizadas	73
5.1.1.	Generación de diagramas	73
5.1.1.1.	Creación de la memoria	73
5.1.1.2.	Entorno de desarrollo: IntelliJ IDE	74
5.1.2.	Spring Framework	74
5.1.2.1.	SpringBoot	75
5.1.3.	Angular Framework	75
5.1.4.	Node.Js	76
5.1.5.	Tecnología de pruebas	76
5.1.5.1.	Postman	76
5.1.6.	Control de versiones	77
5.1.6.1.	Git y Gitlab	77
5.1.7.	Sistema Gestor de Base de Datos	77
5.1.7.1.	DBeaver	77
5.2.	Modelo predictivo de los datos	78
5.2.1.	Proceso de Implementación	78

6. Pruebas	81
6.1. Ejecución de las pruebas	81
6.1.1. Pruebas Funcionales	81
6.1.2. Pruebas No Funcionales	81
6.2. Pruebas de aceptación por cada caso de uso	82
6.3. Resultados de las pruebas	87
6.3.1. Pruebas de usabilidad	88
7. Conclusiones	89
7.1. Objetivos conseguidos	89
7.2. Análisis personal	89

Capítulo 1

Introducción

1.1. Consideraciones iniciales

1.1.1. Contexto

Las pequeñas y medianas empresas están sufriendo un recorte de sus márgenes de beneficio. El mercado cambia rápidamente, le afectan múltiples factores y los precios fluctúan. Un conocimiento detallado de las tendencias en los precios de los proveedores proporciona varias ventajas competitivas a las pequeñas y medianas empresas:

- **Toma de Decisiones Informadas:** Al entender las tendencias de precios, las empresas pueden tomar decisiones de compra más informadas. Saben cuándo es el mejor momento para comprar grandes volúmenes y cuándo es mejor esperar, lo que puede resultar en ahorros significativos.
- **Optimización de Inventarios:** Con información precisa sobre las fluctuaciones de precios, las empresas pueden gestionar sus inventarios de manera más eficiente. Esto evita tanto el exceso de inventario, que puede generar costos adicionales, como la escasez, que puede llevar a perder ventas.
- **Negociación con Proveedores:** Conocer las tendencias de precios permite a las empresas negociar mejores condiciones con sus proveedores. Pueden argumentar a favor de descuentos o términos más favorables basándose en datos históricos y tendencias futuras.
- **Estrategias de Precios de Venta:** Las empresas pueden ajustar sus precios de venta en función de los costos de adquisición previstos. Esto les permite mantener márgenes de beneficio estables y ofrecer precios competitivos a sus clientes.
- **Planificación Financiera:** Al prever cómo evolucionarán los costos de las materias primas, las empresas pueden planificar mejor sus finanzas. Esto incluye la elaboración de presupuestos más precisos y la anticipación de necesidades de financiamiento.
- **Mitigación de Riesgos:** Con un conocimiento profundo de las tendencias de precios, las empresas pueden identificar y mitigar riesgos asociados con la volatilidad del mercado. Pueden establecer estrategias de cobertura o contratos a largo plazo que les protejan contra aumentos repentinos de precios.

1.1.2. Motivación

En la pequeña y mediana empresa, los márgenes de beneficio son uno de los indicadores más importantes de su salud financiera. Miden qué parte de los ingresos generados por una empresa se retiene como ganancia después de deducir los costos de producción, operación e impuestos. Para

los empresarios de estas pymes, comprender y mejorar los márgenes de beneficio puede ayudarles a aumentar la rentabilidad, hacer crecer su negocio obteniendo una ventaja competitiva en el mercado.

Los márgenes de beneficio están afectados (entre otros factores) por las fluctuaciones de los precios de los proveedores. Ajustar la estrategia de precios y detectar tendencias en ellos, mejoran la estrategia financiera y proporcionan una ventaja competitiva frente a otras empresas del sector.

Para gestionar eficazmente estos desafíos, las pymes necesitan herramientas que les permitan monitorizar y analizar los precios de los proveedores, facilitando la toma de decisiones informadas y estratégicas.

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

1.1.3. Objetivos

Se busca desarrollar una aplicación web capaz de permitir a los usuarios llevar un registro del inventario de los productos, de los proveedores y un análisis sobre la tendencia de los precios, asegurando siempre la seguridad de los datos de los usuarios. Al mismo tiempo en su uso debe ser fácil de recordar.

1.2. Análisis de aplicaciones similares

Se ha investigado y apenas existen aplicaciones que permitan la gestión de inventario y la gestión de cuentas y precios al mismo tiempo destinadas para las Pymes y con un alto grado de personalización. Es por ello por lo que se ha optado por un análisis de las mejores aplicaciones de gestión de inventario y de cuentas del mercado por separado. Se comenzará por las aplicaciones de gestión de inventario:

1.2.1. QUICKBOOKS

Descripción: Este sistema compuesto tanto de una aplicación web como de una versión de escritorio es una solución versátil, escalable y de fácil acceso que puede adaptarse a diversas necesidades empresariales, destaca por su integración con otras aplicaciones y servicios, ofreciendo una solución completa para la gestión financiera empresarial.

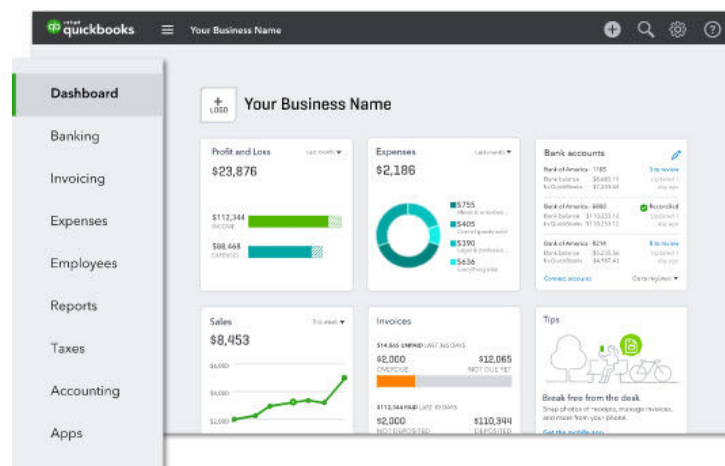


Figura 1.1: Interfaz QUICKBOOKS

Fortalezas detectadas:

- Abarca desde contabilidad básica hasta gestión de inventario y facturación.
- El sistema es fácil de usar y presenta una pequeña curva de aprendizaje.
- Facilita la conciliación bancaria y proporciona seguimiento en tiempo real de las transacciones financieras.
- Capacidad para generar facturas para clientes.
- Gestión específica de clientes, proveedores y empleados.
- Generación y consulta de los libros de contabilidad.
- Permite importar datos sobre ventas, facturas, etc en distintos formatos como CSV.
- Empleo de nube para el almacenamiento de los datos lo que permite la escalabilidad.

Debilidades detectadas:

- A pesar de que tiene distintos planes, pueden verse como caros para algunas empresas.
- Las funcionalidades que ofrece la aplicación están muy determinadas y para algunas empresas puede que no se ajuste al modelo que tienen de organización o que no les satisfaga la necesidad que tienen.
- Dependencia de internet al estar implementada en la nube.

1.2.2. ODOO

Descripción: Es un SaaS dividido en un conjunto de aplicaciones modulares. Estas aplicaciones abarcan áreas como contabilidad, gestión de inventarios, ventas, compras, recursos humanos, fabricación y más. La integración entre estas aplicaciones permite a las empresas tener una visión completa de sus operaciones y facilita la automatización de procesos.

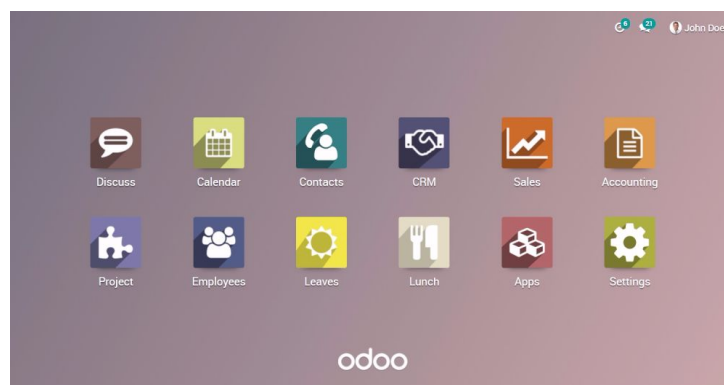


Figura 1.2: Interfaz ODOO[12]

Fortalezas detectadas:

- Posibilidad de modificar la funcionalidad según las necesidades de la empresa a ser de código abierto.
- Debido a la modularidad que tiene se puede elegir que funcionalidades se quieren emplear.

- Presenta una interfaz de fácil empleo y muy amigable para el usuario.
- Presenta una comunidad de usuarios activa, con actualizaciones, etc.

Debilidades detectadas:

- La implementación y la configuración pueden presentar una gran curva de aprendizaje.
- La personalización de algunas funciones necesitara ciertos conocimientos sobre desarrollo.
- Para un soporte técnico más especializado será necesario tener una suscripción.

1.2.3. SAP BUSINESS ONE

Descripción: Sap Busines One ofrece funciones clave, como contabilidad, gestión de inventario, gestión de relaciones con clientes (CRM), compras y ventas. La plataforma se integra con diversas aplicaciones y módulos, permitiendo a las empresas personalizar su implementación según sus necesidades específicas.

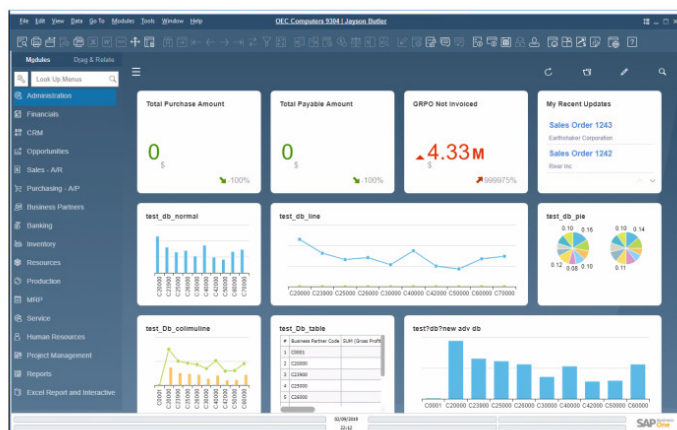


Figura 1.3: Interfaz SAP BUSINESS ONE[13]

Fortalezas detectadas:

- Es ideal para sus operaciones internacionales, debido a todos los idiomas, divisas y legislaciones fiscales que soporta.
- Ofrece la posibilidad de generar informes y análisis de los datos.

Debilidades detectadas:

- El coste de implementación y de licenciamiento junto con los servicios asociados puede ser alto.
- La complejidad de la plataforma genera una curva de aprendizaje media.
- La personalización de la aplicación puede resultar limitada para algunas empresas.
- La implementación depende de la infraestructura de hardware y el software.

Una vez analizadas las de gestión de inventario, se analizará las de gestión de cuentas y precios con la capacidad de mostrar los datos:

1.2.4. MINDEREST

Descripción: está diseñada para adaptarse a las necesidades específicas de los clientes y proporcionar información valiosa para optimizar la estrategia de precios de una empresa en el entorno digital.

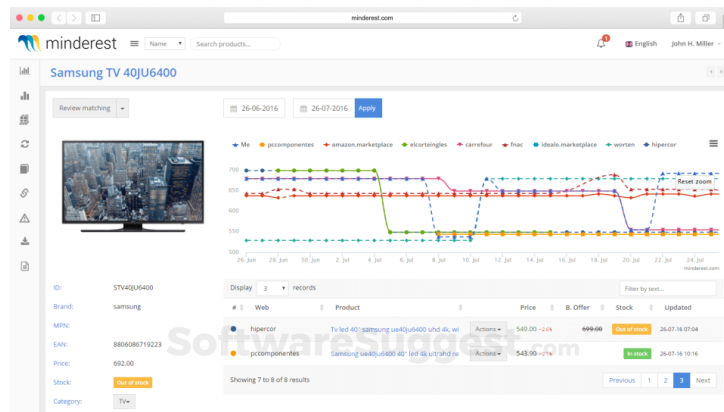


Figura 1.4: Interfaz MINDEREST[10]

Fortalezas detectadas:

- Monitorización de precios en tiempo real.
- Personalización según las necesidades específicas del cliente.
- Proporciona análisis detallados de la competencia.

Debilidades detectadas:

- La implementación y utilización de las soluciones pueden requerir cierto nivel de conocimientos técnicos.
- Puede tener un coste elevado para una Pyme.
- Es necesario tener una capacitación para alguna de las soluciones empleadas.

1.2.5. OPTIMUS PRICE

Descripción: Esta aplicación utiliza algoritmos avanzados de aprendizaje automático para analizar datos de precios en tiempo real, comportamiento del consumidor y condiciones del mercado.

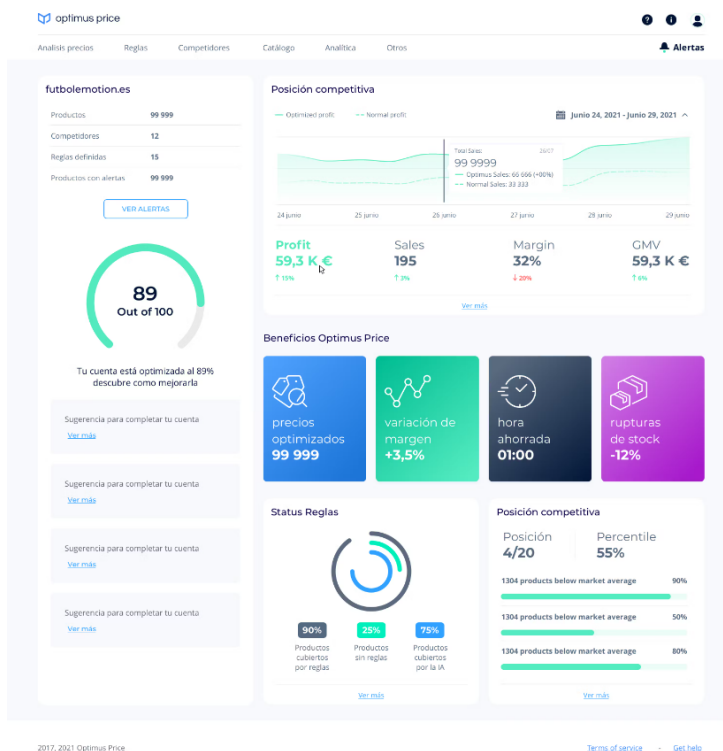


Figura 1.5: Interfaz Optimus Price[3]

Fortalezas detectadas:

- Monitoriza y analiza los precios de la competencia.
- Ofrece la capacidad de ajustar los precios en tiempo real.
- Empleo de machine learning avanzado.
- Amplia gama de opciones de visualización.
- Es personalizable.

Debilidades detectadas:

- Puede tener un coste elevado para una Pyme.
- La curva de aprendizaje de las funcionalidades relacionadas con la IA es grande.
- Necesidad de los mejores datos posibles para obtener la mayor eficiencia posible.
- Se enfoca principalmente para grandes y medianas empresas.

1.2.6. Conclusiones extraídas del análisis

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

- **Alto coste:** la mayoría de estas aplicaciones son gratuitas en su versión básica, la cuál puede no tener todas las funcionalidades necesarias para una Pyme y la versión que ofrece todas las funcionalidades suele tener un precio demasiado elevado.

- **No información sobre tendencias de precios:** casi todas las aplicaciones analizadas no ofrecen información o avisos a los usuarios sobre las tendencias de los precios y si se produce una desviación en las mismas.
- **Machine Learning:** solo una de las aplicaciones estudiadas ofrece una funcionalidad relacionada con el machine learning, sin embargo no es empleado para el cálculo de precios futuros que permitan al usuario anticiparse al mercado.
- **Dificultad para la configuración:** en las aplicaciones analizadas es necesario tener un cierto nivel de conocimientos para la configuración de las funcionalidades.

1.3. Estructura de la memoria

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

- **Capítulo 1: Introducción** explica las consideraciones previas, motivos, situación y otras informaciones interesantes para conocer el alcance y objetivos del trabajo.
- **Capítulo 2: Planificación del Proyecto** trata todo lo referente a la metodología que se va a utilizar y como se va a estructurar el trabajo durante la duración del proyecto.
- **Capítulo 3: Análisis** explicación de la etapa y los artefactos generados.
- **Capítulo 4: Diseño** explicación de la etapa y los artefactos generados.
- **Capítulo 5: Implementación** explicación de la etapa y los artefactos generados.
- **Capítulo 6: Prueba** análisis de los procedimientos llevados adelante para detectar y corregir errores en el sistema.
- **Capítulo 7: Conclusiones** experiencias y aprendizajes adquiridos.
- **Bibliografía y Webgrafía.**
- **Anexos.**

Capítulo 2

Planificación del Proyecto

2.1. Introducción

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

Otros objetivos son ensayar la planificación de proyectos y poner en práctica los métodos de control, prevención y mitigación (aprendidos durante el grado) para corregir los posibles errores.

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

2.2. Descripción de la metodología utilizada

En este proyecto se empleará metodología agile, concretamente hemos elegido SCRUM . En esta sección primero se comentarán las características principales de la metodología agile y después se especificarán las de la metodología SCRUM.

2.2.1. Características principales de la metodología Agile

Las metodología agile destaca por tener un enfoque iterativo, lo que implica la realización de entregas de manera cíclica. En cada entrega, se abordan todas las fases del ciclo de desarrollo, desde la toma de requisitos hasta el diseño, la verificación y la entrega final. La principal diferencia respecto a los antiguos modelos en cascada radica en que, en los procesos ágiles, se entrega valor de forma continua y se recibe retroalimentación a lo largo de todo el proyecto.

Esta metodología utiliza estructuras de trabajo para la gestión de proyectos, divididos en fases dinámicas conocidas como "sprints". Tras cada sprint, los equipos reflexionan sobre lo sucedido, evaluando posibles mejoras para ajustar la estrategia en los siguientes sprints.[4]

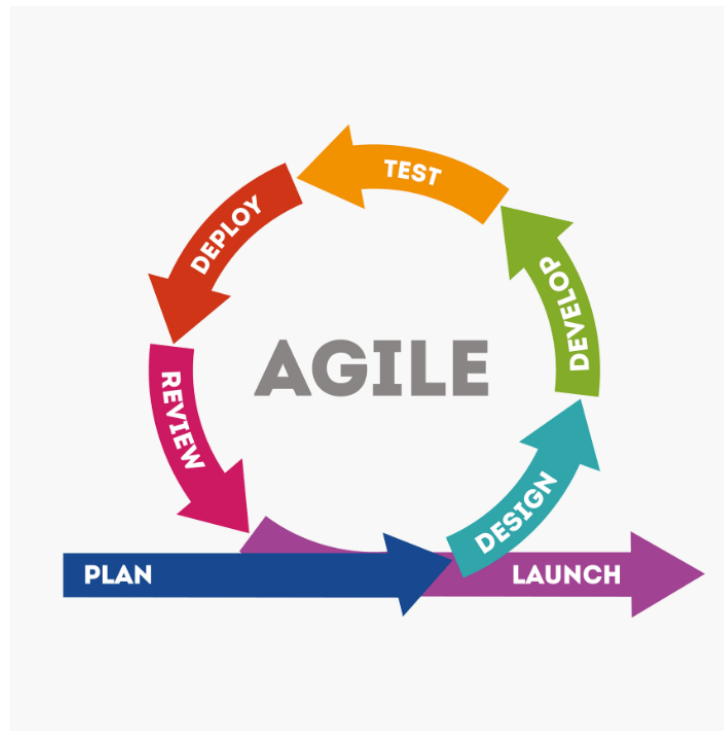


Figura 2.1: Ciclo Metodología Agile

2.2.2. Características principales de SCRUM

Como se ha comentado en la sección anterior, la metodología ágil es una de las más utilizadas en la realización de proyectos software.

SCRUM es una metodología ágil de gestión de proyectos, especialmente eficaz para el desarrollo de software. Se basa en un enfoque iterativo e incremental, dividido en sprints de corta duración. SCRUM utiliza roles claramente definidos, como el Scrum Master, el Product Owner y el Equipo de Desarrollo, para fomentar la colaboración y la responsabilidad. Las reuniones regulares, como la planificación de sprint, la revisión y la retrospectiva, promueven la transparencia y la mejora continua. SCRUM destaca por su capacidad para adaptarse a cambios en los requisitos del proyecto, entregando de manera constante productos funcionales y valiosos.[5]

2.3. Planificación inicial del proyecto

En este apartado se expondrá la planificación que se ha empleado durante la realización del trabajo. Como no existe una experiencia previa por parte del alumno en desarrollos de este alcance y ha sido necesario compaginarlo con otros requerimientos laborales se ha optado por una planificación semanal en lugar de diaria. El alcance inicial del proyecto se ha fijado en 300 horas porque es la cantidad de horas propuesta para la finalización de un Trabajo de Fin de Grado.

2.3.1. Visión general de la planificación

Se asignará a cada una de las semanas de trabajo del proyecto aproximadamente 16h de trabajo. Durante ese período se llevarán a cabo las tareas asignadas en cada sprint backlog.

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

- Al asignar un número de horas fijas por semana, favorece que se establezca una rutina permitiendo una flexibilidad diaria dentro de la planificación semanal.

- Se ha decidido realizar 4h el sábado y el domingo y el resto de las 8h restantes realizarlas entre el lunes y el viernes.
- Las 4 últimas semanas se emplearán en el refinamiento, primero 2 semanas para la implementación y después otras 2 para la memoria.

Siguiendo estas directrices el trabajo se realizará en 17 semanas.

2.3.2. Visión detallada de la planificación

En esta sección, se va a detallar la planificación que se ha llevado a cabo para el proyecto de software. SCRUM ha sido la elección clave debido a su enfoque ágil y estructurado, proporcionando un marco robusto para el desarrollo iterativo. A través de la organización de sprints, y las reuniones regulares, se ha establecido una hoja de ruta clara para garantizar la eficiencia en el proceso de desarrollo.

Siguiendo uno de los principios de SCRUM se realizarán unas reuniones periódicas semanalmente, concretamente los viernes, donde se revisará el trabajo realizado a lo largo de la semana y se acordará el trabajo a realizar en la siguiente.

Asimismo se ha dividido casi toda la carga de trabajo de este TFG en sprints. Estos se han agrupado dentro de la conocida fase de desarrollo del TFG.

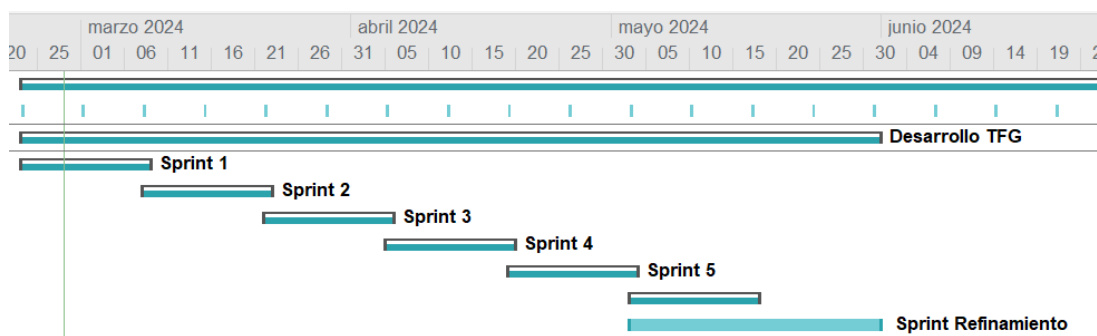


Figura 2.2: Fase Desarrollo Gantt

↳ Desarrollo TFG	71 días	vie 23/02/24	vie 31/05/24	261 hrs
↳ Sprint 1	11 días	vie 23/02/24	vie 08/03/24	37 hrs
↳ Sprint 2	11 días	vie 08/03/24	vie 22/03/24	37 hrs
↳ Sprint 3	11 días	vie 22/03/24	vie 05/04/24	37 hrs
↳ Sprint 4	11 días	vie 05/04/24	vie 19/04/24	37 hrs
↳ Sprint 5	11 días	vie 19/04/24	vie 03/05/24	37 hrs
↳ Sprint 6	11 días	vie 03/05/24	vie 17/05/24	37 hrs
Sprint Refinamiento	11 días	vie 17/05/24	vie 31/05/24	39 hrs

Figura 2.3: Fase Desarrollo Tareas

En las figuras 2.2 y 2.3 se puede apreciar cómo se ha dividido el desarrollo en 7 sprints, 6 regulares y uno de refinamiento, todos ellos con una duración de 2 semanas. Cada sprint regular se divide en Análisis, Diseño, Implementación, Pruebas y Memoria, correspondiéndose cada una con las fases de desarrollo de un proyecto software, salvo Memoria que simplemente se corresponde a ir documentando el proceso.

Una vez acabado el desarrollo se realizará otro sprint de Refinamiento pero este orientado a la memoria, también durará 2 semanas justo hasta la fecha en la preferiblemente será entregado el TFG.

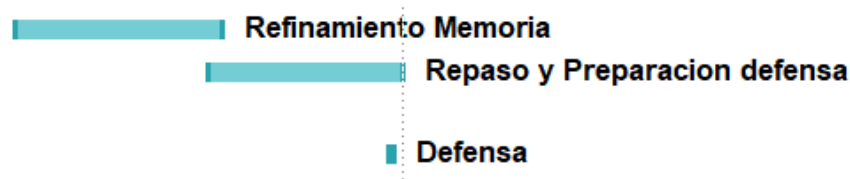


Figura 2.4: Sprint Refinamiento Gantt

Refinamiento Memoria	11 días	vie 31/05/24	vie 14/06/24	39 hrs
Repaso y Preparacion defensa	10 días	vie 14/06/24	jue 27/06/24	0 hrs
Defensa	2 hrs	jue 27/06/24	jue 27/06/24	0 hrs

Figura 2.5: Sprint Refinamiento Tareas

Como se ve en las imágenes anteriores, existen 2 tareas extra relacionadas con la defensa. El trabajo realizado en ellas no se considera que deba entrar dentro de las 300 horas del TFG debido a que en ese momento ya ha sido entregado.

2.4. Plan de gestión de riesgos

En cualquier proyecto software que se lleve a cabo existen una serie de contratiempos asociados a distintos factores, los cuales pueden generar retrasos dentro de la planificación inicial o incluso hasta oportunidades para aprender.

El Plan de Gestión de Riesgos pretende definir todos esos posibles contratiempos que puedan ocurrir, definiendo para ello tanto los detalles, como los de la solución con la que se tiene la intención de subsanarlo.

Antes de comenzar es necesario comenzar por definir qué entendemos por riesgo. Para ello se ha utilizado la comparativa planteada en el libro *Software Project Management* de B. Hughes[8]. En este documento se presentan los riesgos como la combinación de dos definiciones diferentes. La primera, extraída de la guía “PMBOK”[9] considera los riesgos como “una situación o evento desconocido que, en el caso de ocurrir, podría tener un impacto positivo o negativo en los objetivos del proyecto”. La segunda, extraída de la metodología “PRINCE2”[17], se define como “la posibilidad de exposición a las consecuencias adversas de un evento futuro”. Ambas combinadas, nos otorgan una definición concisa de que podemos entender por riesgo. En este documento solo se considerará dichos contratiempos que puedan suponer un problema a la hora de alcanzar el objetivo deseado.

Una vez definido lo que es el riesgo, se debe definir como se clasificarán. La clasificación se realizará según la probabilidad e impacto de ocurrencia:

- **Probabilidad:** es la posibilidad de que ocurra un evento no deseado.
- **Impacto:** es la magnitud de las consecuencias asociadas a dicho evento.

Para la categorización de los riesgos según la probabilidad se ha empleado esta tabla:

Nivel de probabilidad	Rango
Alta	>50 % de probabilidad de ocurrencia
Media	10-50 % de probabilidad de ocurrencia
Baja	<10 % de probabilidad de ocurrencia

Tabla 2.1: Probabilidad de ocurrencia

Para la categorización de los riesgos según el impacto se ha empleado esta tabla:

Nivel de impacto	Rango
Alta	>30 % de probabilidad de ocurrencia
Media	10-30 % de probabilidad de ocurrencia
Baja	<10 % de probabilidad de ocurrencia

Tabla 2.2: Impacto de ocurrencia

2.4.1. Gestión de riesgos

A continuación, se enumerarán los riesgos que se han tenido en cuenta en la Fase de inicio del proyecto. Para cada uno de ellos se proporciona una descripción que introduce cuál es su categoría, una pequeña descripción, la probabilidad de ocurrencia y el impacto. También se explicarán las acciones que se llevarán a cabo durante el proyecto para minimizar la exposición a dichos riesgos. Finalmente se enumerarán una serie de acciones que se llevaran a cabo para que el riesgo tenga el menor impacto posible sobre la realización del proyecto.

Nombre del Riesgo	Indisponibilidad
Categoría	Planificación
Descripción	Indisponibilidad de algún miembro del equipo de desarrollo debido a distintas causas.
Probabilidad	Media
Impacto	Alto
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Identificar los puntos críticos del proyecto que podrían ser más vulnerables a interrupciones. ▪ Permitir atención especial y medidas preventivas en áreas de alto riesgo.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Incluir márgenes de tiempo adicionales en la planificación para acomodar posibles interrupciones. ▪ Dividir el proyecto en hitos intermedios y establecer revisiones regulares.

Tabla 2.3: Riesgo 1:Indisponibilidad

Nombre del Riesgo	Acceso limitado a recursos
Categoría	Tecnología
Descripción	Existencia de un acceso limitado a los recursos software o físicos que se desee emplear para el desarrollo del trabajo.
Probabilidad	Baja
Impacto	Medio
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Identificar y evaluar los recursos de software necesarios para el proyecto desde el principio. ▪ Buscar alternativas si algunos recursos son de difícil acceso.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Contactar con instituciones académicas, bibliotecas o comunidades que puedan proporcionar acceso a los recursos necesarios. ▪ Explorar oportunidades para obtener acceso a través de colaboraciones externas.

Tabla 2.4: Riesgo 2: Acceso limitado a recursos

Nombre del Riesgo	Cambio Requisitos
Categoría	Requisitos
Descripción	Los requisitos del cliente pueden cambiar a lo largo del desarrollo del TFG, lo que podría impactar en el diseño y la implementación del software.
Probabilidad	Alta
Impacto	Alto
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Asegurarse de tener un entendimiento claro y documentado de los requisitos iniciales. ▪ Permitir cambios y establecer una gestión pro activa a la hora de realizarlos. ▪ Establecer una fecha máxima de aceptación de cambio de requisitos.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Proporcionar un marco estructurado para gestionar cambios y evaluar su impacto. ▪ Implementar un proceso formal de gestión de cambios.

Tabla 2.5: Riesgo 3:Cambio Requisitos

Nombre del Riesgo	Falta de experiencia
Categoría	Tecnología
Descripción	Falta de experiencia con la tecnología elegida.
Probabilidad	Media
Impacto	Medio
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Realizar una investigación exhaustiva para desarrollar habilidades y aprender la tecnología.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Solicitar ayuda de personas más experimentadas. ▪ Aprovechar la experiencia de profesionales para evitar problemas comunes y mejorar la calidad del trabajo.

Tabla 2.6: Riesgo 4:Falta de Experiencia

Nombre del Riesgo	Insatisfacción de requisitos
Categoría	Tecnología
Descripción	La arquitectura y tecnología elegidas puede ser que no lleguen a satisfacer los requisitos.
Probabilidad	Baja
Impacto	Alto
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Realizar un buen estudio para elegir la arquitectura y tecnología para elegir una adecuada.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Reajustar el proyecto. ▪ Incluir nuevas tecnologías que si que nos permitan cumplir con los requisitos.

Tabla 2.7: Riesgo 5: Insatisfacción de requisitos

Nombre del Riesgo	Retrasos respecto a la planificación
Categoría	Planificación
Descripción	Debido a la inexperiencia o a distintas causas el producto se encuentra en retraso según la planificación.
Probabilidad	Media
Impacto	Medio
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Intentar seguir la planificación lo máximo posible y no perder demasiado tiempo.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Aumentar numero de horas trabajadas por semana para subsanar el retraso.

Tabla 2.8: Riesgo 6: Retraso

Nombre del Riesgo	Desarrollo erróneo del producto backlog
Categoría	Producto
Descripción	Desarrollo erróneo del producto backlog.
Probabilidad	Media
Impacto	Alto
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Intentar que los requisitos queden lo más claro posible antes de empezar el desarrollo. ▪ Fomentar una comunicación activa entre el cliente y el desarrollador.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Establecer sesiones regulares de refinamiento del Product Backlog para adaptarse a cambios en los requisitos y mejorar la calidad. ▪ Implementar revisiones periódicas del producto backlog para identificar y corregir posibles errores. ▪ Incorporar retroalimentación del cliente de manera regular.

Tabla 2.9: Riesgo 7: Desarrollo erróneo

Nombre del Riesgo	Planificación errónea
Categoría	Planificación
Descripción	Debido a la inexperiencia en la gestión de proyectos y en realizar planificaciones se genera una planificación demasiado optimista y muy irreal.
Probabilidad	Alta
Impacto	Alto
Acciones Mitigantes	<ul style="list-style-type: none"> ▪ Proporcionar al estudiante recursos y entrenamiento en las mejores prácticas de gestión de proyectos. ▪ EL tutor del TFG que este experimentado para que pueda guiar al estudiante en la planificación y ejecución del proyecto.
Acciones Correctivas	<ul style="list-style-type: none"> ▪ Establecer puntos de revisión periódicos durante el desarrollo del proyecto para ajustar la planificación según sea necesario. ▪ Fomenta un ambiente donde el estudiante pueda aprender de los errores y experiencias durante el proyecto. ▪ Establece canales de comunicación abiertos para que el estudiante pueda expresar cualquier preocupación o desafío en la planificación.

Tabla 2.10: Riesgo 8: Desarrollo erróneo

2.5. Presupuesto del Proyecto

A continuación se describe el desglose del presupuesto del proyecto, centrándonos específicamente en los recursos humanos, equipos y tiempo asignados. La clave para el éxito radica en comprender cómo estos elementos se combinan para impulsar el progreso del proyecto. Analizaremos la planificación estratégica detrás de la asignación de personal, la elección de equipos y la gestión del tiempo. Este enfoque nos permitirá obtener una visión precisa de la inversión necesaria en términos de capital humano y equipos, asegurando una ejecución eficiente y exitosa del proyecto.

2.5.1. Recursos de personal

Tradicionalmente, los equipos que gestionan desarrollan proyectos software están configurados por diferentes miembros que desempeñan distintos roles en la empresa. Sin embargo, en este caso todos los roles del equipo de desarrollo están divididos entre el profesor y el alumno que lo realizan. Donde el tutor se encarga de las tareas relacionadas con tratar con el cliente y el alumno de las labores de desarrollo, diseño, análisis, etc. Para conocer los gastos asociados a cada rol y el salario medio en España se ha investigado en el sitio web *talent* los datos para el año 2024.

- Líder del Proyecto: 18,46€/hora
- Desarrollador de Software(FullStack): 18€/hora
- QA Tester: 14,62€/hora
- Diseñador de Interfaz de Usuario(UX/UI): 15,38€/hora

2.5.2. Recursos materiales

Existen diferentes gamas para cada uno de los productos necesarios así que se tomará el precio del equipo empleado para el desarrollo del proyecto. El coste proporcional viene determinado por el tiempo de vida de los elementos (700 días) y el tiempo de desarrollo del proyecto (70 días).

Recurso	Precio(€)/unidad	Coste proporcional
Ordenador	800	80
Monitor	122	12
Teclado	20	2
Ratón	60	6
Mesa de Oficina	95	10
Total(€)		110

Tabla 2.11: Tabla de recursos materiales

2.5.3. Recursos tecnológicos

- IntelliJ Idea: 16,90€/mes
- Gitlab: 0€
- Astah(individual): 9,99€/mes
- Microsoft Teams: 120€/mes por usuario

2.5.4. Recursos de localización

Según el análisis de mercado llevado a cabo, el precio mostrado es la media de los analizados:

- Oficina 500€/mes

2.5.5. Recursos de infraestructura

Existen numerosas ofertas que dependen de la situación del mercado.

- Calefacción: 75€/mes
- Conexión de red: 50€/mes
- Agua: 20€/mes
- Luz: 50€/mes

2.5.6. Presupuesto inicial

Con los precios mostrados en la sección anterior se va a calcular el presupuesto estimado del proyecto. Para la estimación, se empleará la duración del proyecto planificada de 300 horas, divididas en 10 semanas y considerando a los trabajadores como autónomos.

Recurso	Precio(€)/UT	Coste proporcional(€)
Líder de Proyecto	18,46	5.538
Fullstack	18	5.400
QA Tester	14,62	4.386
UX/UI	15,38	4.614
Recursos materiales	110	110
Intelij Idea	16,90	51
Gitlab	0	0
Astah(individual)	10	30
Teams	120	360
Oficina	500	1500
Calefacción	75	225
Red	50	150
Agua	20	60
Luz	50	150
Total(€)		22.464

Tabla 2.12: Tabla de Presupuesto inicial

2.5.7. Presupuesto real

En el presupuesto real los costes de personal, de recursos tecnológicos, de infraestructura y de localización son 0 debido a que el desarrollador es el alumno y los demás recursos son suministrados por la universidad.

Recurso	Precio(€)/UT	Coste proporcional(€)
Líder de Proyecto	0	0
Fullstack	0	0
QA Tester	0	0
UX/UI	0	0
Recursos materiales	110	110
IntelliJ Idea	0	0
Gitlab	0	0
Astah(individual)	0	0
MS Teams	0	0
Oficina	0	0
Calefacción	0	0
Red	0	0
Agua	0	0
Luz	0	0
Total(€)		22.464

Tabla 2.13: Tabla de Presupuesto inicial

Capítulo 3

Análisis

3.1. Introducción

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

3.2. Actores y Roles del sistema

3.2.1. Actores Principales

El único actor de nuestro sistema, y por lo tanto el principal, es el usuario. Podemos definir al usuario como el gestor de la compañía, que se encargará de comprobar los precios y de gestionar el inventario.

Hay roles dentro del sistema pero en la práctica resulta ser uno solo predeterminado que engloba todas las funciones y responsabilidades, simplificando así la gestión de permisos y accesos de los usuarios de manera eficiente.

3.3. Requisitos

Debido a que esto es un proyecto software real, a continuación se reflejarán los requisitos del proyecto para posibles consultas futuras con el cliente.

Para la clasificación de los requisitos del sistema se empleará la clasificación del libro de Sommerville [15] :

- **Requisitos funcionales:** Son declaraciones de los servicios que el sistema debería proporcionar, cómo el sistema debería reaccionar ante entradas específicas y cómo debería comportarse en situaciones particulares.
- **Requisitos no funcionales:** Son restricciones sobre los servicios o funciones ofrecidos por el sistema. Incluyen restricciones temporales, restricciones en el proceso de desarrollo y restricciones impuestas por estándares. Dentro de los requisitos no funcionales existen otro tipo de requisitos como son los de producto, de organización y externos que a su vez se dividen en otros.

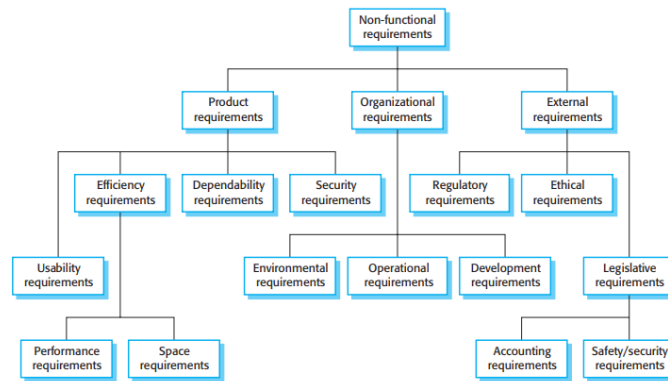


Figura 3.1: División Requisitos no Funcionales

3.3.1. Requisitos funcionales

RF-1: Registro de usuarios: el sistema debe permitir a los nuevos usuarios registrarse otorgando toda la información detallada en el requisito de información correspondiente. [RFI-1]

RF-2: Histórico de precios: el sistema debe realizar una representación gráfica del registro de los precios por semana y año.

RF-3: Tendencias de precios: el sistema debe realizar un análisis de tendencias de precios de proveedores y emplear gráficos que muestren su evolución.

RF-4: Alertas de Precios: el sistema deberá notificar al usuario cuando se produzca un cambio repentino en las tendencias de precios, empleando umbrales para lograrlo.

RF-5: Análisis predictivo: el sistema deberá emplear modelos de análisis predictivo de precios para el pronóstico del precio del mes siguiente y permitir su visualización.

RF-6: Importación de Informes: el sistema deberá permitir la importación de informes de datos en formato CSV y PDF, y realizar la validación de sus datos.

RF-7: Inicio de sesión: el sistema deberá permitir al usuario con cuenta registrada iniciar sesión

RF-8: Proveedores: el sistema deberá permitir el registro manual de proveedores según los datos del [RFI-4], editar y consultar dicha información, al mismo tiempo de eliminarlos.

RF-9: Productos: el sistema deberá permitir el registro manual de productos según los datos del [RFI-3], editar y consultar dicha información, al mismo tiempo de eliminarlos.

RF-10: Búsqueda de Productos: el sistema deberá realizar la búsqueda de un producto empleando su id interno o su nombre.

RF-11: Búsqueda de Proveedores: el sistema deberá realizar la búsqueda de un proveedor empleando su cif o su nombre.

3.3.2. Requisitos funcionales de información

RFI-1: El sistema deberá almacenar los siguientes datos sobre el Usuario:

- Email

- Contraseña
- Teléfono

RFI-2: El sistema deberá almacenar los siguientes datos sobre los Precio:

- Producto
- Cantidad
- Precio
- Fecha venta

RFI-3: El sistema deberá almacenar los siguientes datos sobre el Producto:

- Nombre
- Id que tiene el proveedor sobre él
- Id que tiene el cliente sobre él
- Temperatura mínima de conservación
- Temperatura máxima de conservación
- Familia a la que pertenece el producto

RFI-4: El sistema deberá almacenar los siguientes datos sobre el Proveedor:

- Nombre
- CIF(Código de Identificación Fiscal)
- Dirección
- Provincia
- Teléfono

RFI-5: El sistema deberá almacenar los siguientes datos sobre el Presupuesto:

- Producto
- Cantidad
- Precio
- Fecha venta
- Proveedor

3.3.3. Requisitos no funcionales

RNF-1: El sistema deberá estar configurado para que funcione en local en un PC.

RNF-2: La base de datos del sistema deberá ser MariaDB.

RNF-3: El sistema deberá dar respuesta rápida a las consultas y visualizaciones.

RNF-4: El sistema deberá ser escalable para poder manejar grandes conjuntos de datos.

RNF-5: El funcionamiento del sistema deberá ser fácil de recordar.

RNF-6: El sistema deberá implementar un enfoque eficiente para el manejo de errores de usabilidad, garantizando una presentación clara y comprensible de mensajes de error.

RNF-7: El proceso de importación de informes del sistema deberá ser eficiente y sin pérdida de datos y manejar los posibles errores durante la importación.

Debido a que la metodología empleada en la realización del proyecto es SCRUM del tipo Agile, el cliente ha solicitado una nueva funcionalidad lo que ha supuesto un nuevo caso de uso y añadir el siguiente RF que se corresponderá con el CU-20:

RF-12: Presupuestos: el sistema deberá permitir el registro manual de presupuestos según los datos del [RFI-5].

3.4. Modelo de Dominio

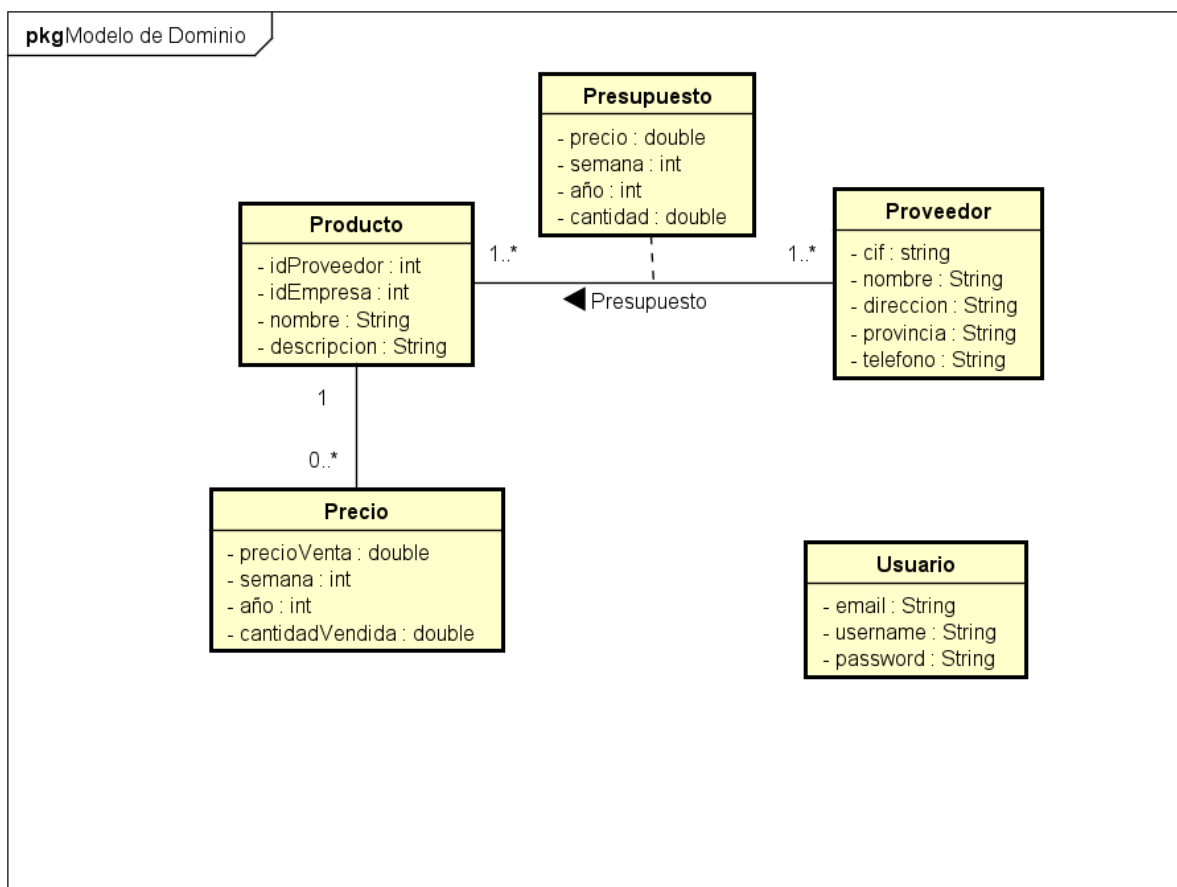


Figura 3.2: Modelo de Dominio

3.5. Casos de Uso

3.5.1. Presentación Casos de Uso

- CU-1: Registrarse
- CU-2: Iniciar sesión
- CU-3: Cerrar sesión
- CU-4: Modificar perfil del usuario
- CU-5: Recuperar contraseña del usuario
- CU-6: Mostrar proveedores
- CU-7: Añadir proveedor
- CU-8: Editar proveedor
- CU-9: Buscar proveedor
- CU-10: Ver detalles proveedor
- CU-11: Eliminar proveedor
- CU-12: Mostrar productos
- CU-13: Añadir producto
- CU-14: Editar producto
- CU-15: Buscar producto
- CU-16: Ver detalles producto
- CU-17: Eliminar producto
- CU-18: Subir datos
- CU-19: Consultar analíticas sobre los datos
- CU-20: Añadir presupuesto

3.5.2. Diagrama de Casos de Uso

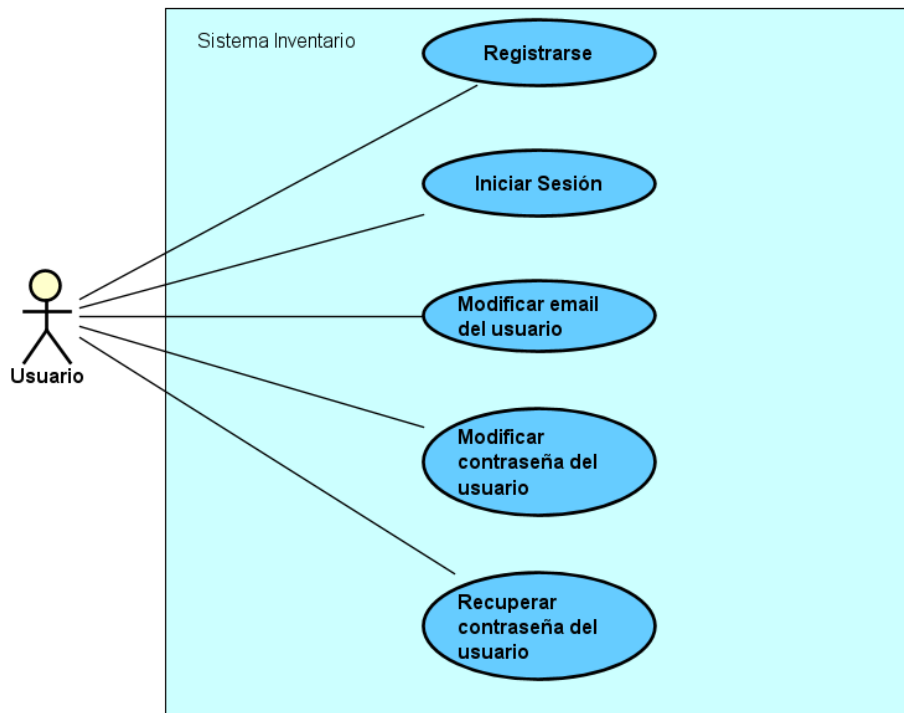


Figura 3.3: Diagrama de casos de uso de Usuario

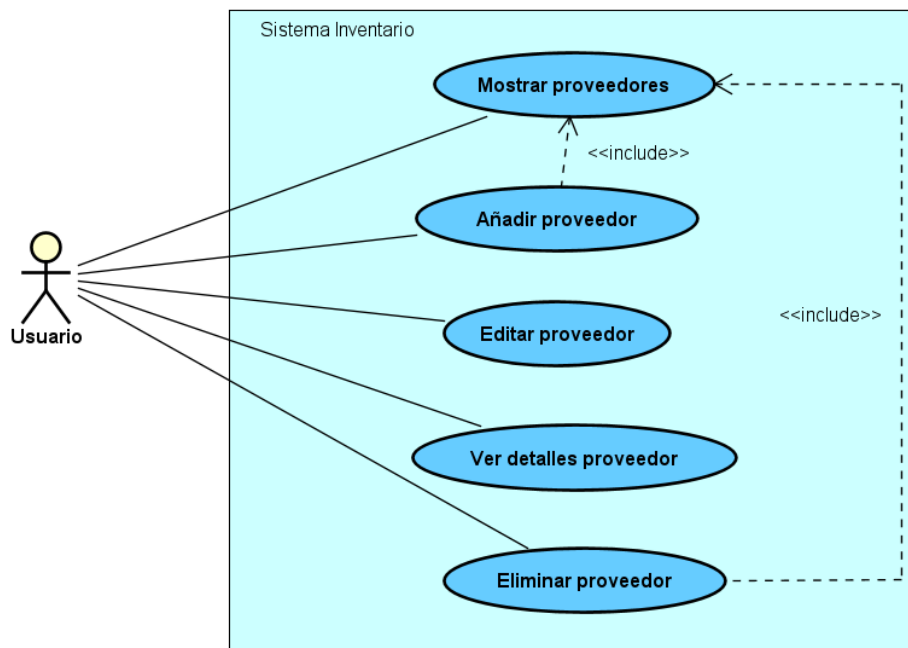


Figura 3.4: Diagrama de casos de uso de Proveedores

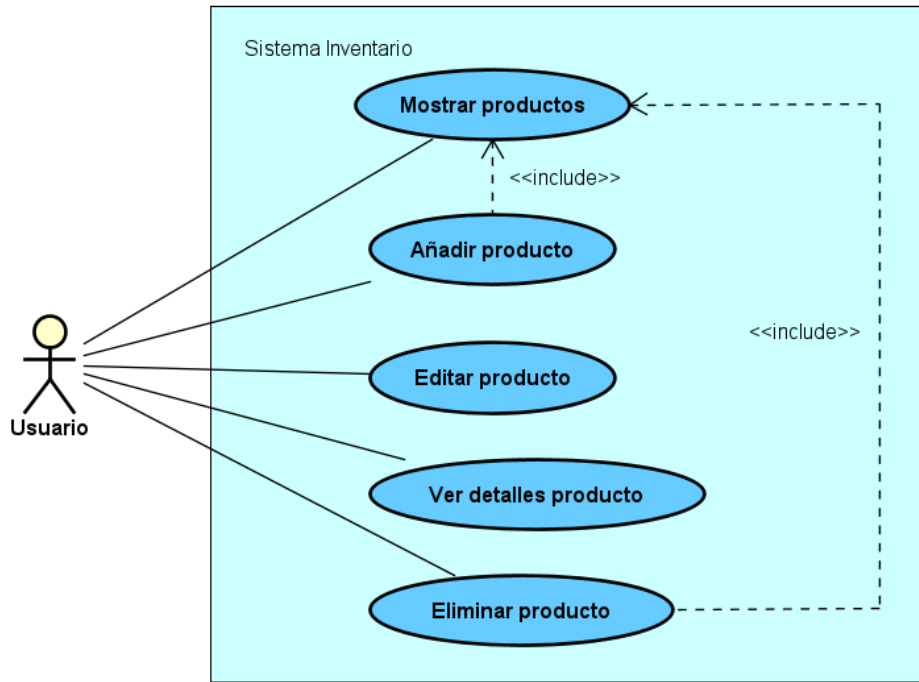


Figura 3.5: Diagrama de casos de uso de Productos

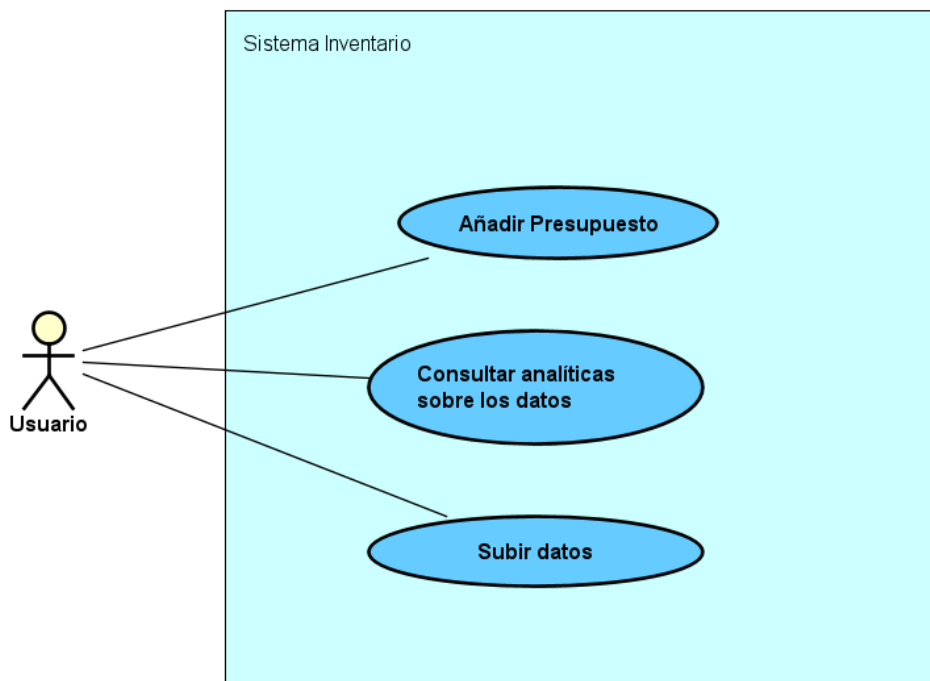


Figura 3.6: Diagrama de casos de uso de Usuario

3.5.3. Descripción textual de los casos de uso

CU-1: Registrarse	
Actor	Usuario
Resumen	Un nuevo usuario quiere crear una cuenta en el sistema que le permita identificarse y acceder a la funcionalidad
Precondición	El actor ha accedido al sistema mediante la aplicación web
Postcondición	El actor queda registrado en el sistema, cuenta con acceso a toda la funcionalidad y su información de sesión está cargada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor comienza el proceso de registro 2. El sistema solicita nombre de usuario, correo electrónico y contraseña del actor 3. El actor introduce los datos solicitados 4. El sistema valida el formato de los datos solicitados 5. El actor confirma el registro 6. El sistema comprueba que el correo electrónico no ha sido usado en otro registro y registra al usuario 7. El sistema informa al usuario del éxito en el registro y le introduce en la aplicación y el caso de uso finaliza 	
Secuencias Alternativas	
1A,3A,5A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos introducidos es incorrecto 2.El sistema informa al actor 3.El caso de uso continua en el paso 3
6A	<ol style="list-style-type: none"> 1.El sistema comprueba que el correo electrónico ya ha sido utilizado por otro usuario 2.El sistema informa al actor 3.El caso de uso continua en el paso 3

Tabla 3.1: Descripción del CU Registrarse"

CU-2: Iniciar sesión	
Actor	Usuario
Resumen	Un usuario pretende utilizar la aplicación y lo primero que debe hacer es identificarse ante el sistema para poder acceder a su funcionalidad
Precondición	El actor está previamente registrado en el sistema
Postcondición	El actor tiene acceso a la funcionalidad del sistema y su información de sesión está cargada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor comienza el proceso de inicio de sesión 2. El sistema solicita el correo electrónico o nombre de usuario y la contraseña 3. El actor introduce los datos solicitados 4. El sistema valida el formato de los datos solicitados 5. El actor confirma el inicio de sesión 6. El sistema comprueba que las credenciales se corresponden con un usuario registrado 7. El sistema carga la información de sesión del usuario y el caso de uso finaliza 	
Secuencias Alternativas	
3A,6A	Si el actor cancela
Excepciones	
6A	<ol style="list-style-type: none"> 1.El sistema comprueba que las credenciales no se corresponden con un usuario registrado 2.El sistema informa al actor 3.El caso de uso continua en el paso 3

Tabla 3.2: Descripción del CU Iniciar-sesión"

CU-3: Cerrar sesión	
Actor	Usuario
Resumen	Un usuario que previamente ha iniciado sesión quiere cerrar la misma después del uso de la aplicación
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La sesión del actor ha sido cerrada y no podrá accederse a la funcionalidad hasta el siguiente inicio de sesión
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor solicita cerrar sesión 2. El sistema cierra la sesión y elimina la información sobre el usuario que se haya cargado para la misma y el caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	

Tabla 3.3: Descripción del CU 'Cerrar-sesión'

CU-4: Modificar perfil del usuario	
Actor	Usuario
Resumen	El usuario quiere cambiar el correo electrónico con el que le reconoce el sistema o la contraseña de acceso
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La cuenta asociada al actor tendrá en su registro en el sistema la información con el dato modificado y esa nueva información de sesión está cargada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor selecciona cambiar su información de perfil y selecciona el dato que pretende cambiar 2. El sistema solicita el nuevo valor modificado para el dato 3. El actor introduce el nuevo valor 4. El sistema valida el formato del nuevo valor introducido y solicita confirmación 5. El actor confirma el cambio 6. El sistema guarda la nueva información y carga la nueva información del usuario y el caso de uso finaliza 	
Secuencias Alternativas	
2A, 4A, 7A	Si el actor cancela
Excepciones	
5A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos introducidos no es correcto 2.El sistema informa al actor 3.El caso de uso continua en el paso 3

Tabla 3.4: Descripción del CU 'Modificar perfil del usuario'

CU-5: Recuperar contraseña del usuario	
Actor	Usuario
Resumen	El usuario quiere recuperar la contraseña de acceso
Precondición	El actor está registrado en el sistema
Postcondición	La cuenta asociada al actor tendrá en su registro en el sistema la información con el dato modificado
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de recuperar contraseña 2. El sistema solicita el email o el nombre de usuario 3. El actor introduce el valor 4. El sistema valida el formato del valor introducido 5. El sistema solicita la nueva contraseña 2 veces 6. El actor introduce las 2 contraseñas 7. El sistema comprueba que son las mismas y solicita confirmación 8. El actor confirma 9. El sistema guarda la nueva información y la carga en la sesión 10. El caso de uso finaliza 	
Secuencias Alternativas	
3A, 6A	Si el actor cancela
Excepciones	
7A	<ol style="list-style-type: none"> 1.El sistema comprueba que las contraseñas no coinciden 2.El sistema informa al actor 3.El caso de uso vuelve al paso 5

Tabla 3.5: Descripción del CU 'Recuperar contraseña del usuario'

CU-6: Mostrar proveedores	
Actor	Usuario
Resumen	El usuario quiere consultar la lista de proveedores almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de mostrar proveedores 2. El sistema muestra los proveedores y el caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	
2A	<ol style="list-style-type: none"> 1.El sistema comprueba que no hay proveedores almacenados en el sistema 2.El sistema informa al actor 3.El caso de uso finaliza

Tabla 3.6: Descripción del CU 'Mostrar proveedores'

CU-7: Añadir proveedor	
Actor	Usuario
Resumen	El usuario quiere registrar un nuevo proveedor en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La información del nuevo proveedor quedará registrada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de añadir proveedor 2. El sistema solicita los datos 3. El actor introduce los datos 4. El sistema comprueba que los datos seann correctos 5. El sistema solicita confirmación 6. El actor confirma 7. El sistema comprueba que no existe un proveedor con esos datos 8. El sistema almacena la información del proveedor y a continuación muestra todos los proveedores 9. El caso de uso finaliza 	
Secuencias Alternativas	
3A, 6A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3
7A	<ol style="list-style-type: none"> 1.El sistema comprueba que ya existe un proveedor con esos datos. 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3

Tabla 3.7: Descripción del CU 'Añadir proveedor'

CU-8: Editar proveedor	
Actor	Usuario
Resumen	El usuario quiere editar uno de los proveedores almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La nueva información del proveedor quedará registrada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de editar proveedor 2. El sistema solicita los datos 3. El actor introduce los datos 4. El sistema comprueba el formato de los datos 5. El sistema solicita confirmación 6. El actor confirma 7. El sistema comprueba que existe un proveedor con esos datos 8. El sistema almacena la información del proveedor y a continuación muestra todos los proveedores 9. El caso de uso finaliza 	
Secuencias Alternativas	
3A, 6A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3
7A	<ol style="list-style-type: none"> 1.El sistema comprueba que no existe un proveedor con esos datos. 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3

Tabla 3.8: Descripción del CU 'Editar proveedor'

CU-9: Buscar proveedor	
Actor	Usuario
Resumen	El usuario quiere encontrar uno de los proveedores almacenados en el sistema de manera rápida
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de buscar proveedor introduciendo el cif del proveedor o el nombre en el buscador 2. El sistema comprueba el formato de los datos 3. El sistema comprueba que existe un proveedor con esos datos 4. El sistema muestra el proveedor buscado 5. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	
2A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 1
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que no existe un proveedor con esos datos. 2.El sistema informa al actor 3.El caso de uso finaliza

Tabla 3.9: Descripción del CU 'Buscar proveedor'

CU-10: Ver detalles proveedor	
Actor	Usuario
Resumen	El usuario quiere ver los datos de uno de los proveedores almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de ver detalles del proveedor 2. El sistema recupera los datos del proveedor 3. El sistema muestra los datos 4. El actor cierra la ventana donde se muestran los datos 5. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	

Tabla 3.10: Descripción del CU 'Ver detalles proveedor'

CU-11: Eliminar proveedor	
Actor	Usuario
Resumen	El usuario quiere eliminar uno de los proveedores almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La información del proveedor quedará eliminada del sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de eliminar proveedor 2. El sistema busca el proveedor 3. El sistema elimina la información del proveedor y a continuación muestra todos los proveedores 4. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	

Tabla 3.11: Descripción del CU 'Eliminar proveedor'

CU-12: Mostrar productos	
Actor	Usuario
Resumen	El usuario quiere consultar la lista de productos almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de mostrar productos 2. El sistema recopila los productos almacenados 3. El sistema muestra los productos 4. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	
2A	<ol style="list-style-type: none"> 1.El sistema comprueba que no hay productos almacenados en el sistema 2.El sistema informa al actor 3.El caso de uso queda sin efecto

Tabla 3.12: Descripción del CU 'Mostrar productos'

CU-13: Añadir producto	
Actor	Usuario
Resumen	El usuario quiere registrar un nuevo producto en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La información del nuevo producto quedará registrada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de añadir producto 2. El sistema solicita los datos 3. El actor introduce los datos 4. El sistema comprueba el formato de los datos 5. El sistema solicita confirmación 6. El actor confirma 7. El sistema comprueba que no existe un producto con esos datos 8. El sistema almacena la información del producto y a continuación muestra todos los productos 9. El caso de uso finaliza 	
Secuencias Alternativas	
3A, 6A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3
7A	<ol style="list-style-type: none"> 1.El sistema comprueba que ya existe un producto con esos datos. 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3

Tabla 3.13: Descripción del CU 'Añadir producto'

CU-14: Editar producto	
Actor	Usuario
Resumen	El usuario quiere editar uno de los productos almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La nueva información del producto quedará registrada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de editar producto 2. El sistema solicita los datos 3. El actor introduce los datos 4. El sistema comprueba el formato de los datos 5. El sistema solicita confirmación 6. El actor confirma 7. El sistema comprueba que no existe un producto con esos datos 8. El sistema almacena la información del producto y a continuación muestra todos los productos 9. El caso de uso finaliza 	
Secuencias Alternativas	
3A, 6A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3
7A	<ol style="list-style-type: none"> 1.El sistema comprueba que ya existe un producto con esos datos. 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3

Tabla 3.14: Descripción del CU 'Editar producto'

CU-15: Buscar producto	
Actor	Usuario
Resumen	El usuario quiere encontrar uno de los productos almacenados en el sistema de manera rápida
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de buscar producto introduciendo el id del producto o el nombre en el buscador 2. El sistema comprueba el formato de los datos 3. El sistema comprueba que existe un producto con esos datos 4. El sistema muestra el producto buscado 5. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	
2A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 1
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que no existe un producto con esos datos. 2.El sistema informa al actor 3.El caso de uso finaliza

Tabla 3.15: Descripción del CU 'Buscar producto'

CU-16: Ver detalles producto	
Actor	Usuario
Resumen	El usuario quiere ver los datos de uno de los productos almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de ver detalles del productos 2. El sistema recupera los datos del producto 3. El sistema muestra los datos 4. El actor cierra la ventana donde se muestran los datos 5. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	

Tabla 3.16: Descripción del CU 'Ver detalles producto'

CU-17: Eliminar producto	
Actor	Usuario
Resumen	El usuario quiere eliminar uno de los productos almacenados en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La información del producto quedará eliminada del sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de eliminar producto 2. El sistema busca el producto 3. El sistema elimina la información del producto y a continuación muestra todos los productos 4. El caso de uso finaliza 	
Secuencias Alternativas	
Excepciones	

Tabla 3.17: Descripción del CU 'Eliminar producto'

CU-18: Subir datos	
Actor	Usuario
Resumen	El usuario desea subir los documentos con los distintos datos sobre los precios de los proveedores
Precondición	El usuario debe tener una sesión activa en el sistema
Postcondición	La información sobre los productos y sus precios de la semana quedará registrada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El usuario comienza el proceso de subir datos 2. El sistema solicita el archivo con los datos 3. El usuario introduce el archivo solicitado 4. El sistema valida el formato del archivo 5. El sistema procesa los datos del archivo y almacena los datos e informa al usuario 6. El caso de uso finaliza 	
Secuencias Alternativas	
3A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato del archivo introducido es incorrecto 2.El sistema informa al actor 3.El caso de uso continua en el paso 2

Tabla 3.18: Descripción del CU 'Subir datos'

CU-19: Consultar analíticas sobre los datos	
Actor	Usuario
Resumen	El usuario desea consultar lo datos de los precios en un intervalo de tiempo determinado
Precondición	El usuario debe tener una sesión activa en el sistema
Postcondición	
Secuencia Principal	
<ol style="list-style-type: none"> 1. El usuario comienza el proceso de consultar datos 2. El sistema solicita un intervalo de tiempo entre los que se mostrarán los datos 3. El usuario introduce el intervalo solicitado 4. El sistema valida el formato de los datos 5. El sistema recupera los datos de los precios entre ese intervalo y se los envía al modelo 6. El modelo calcula el valor aproximado del precio en la unidad de tiempo siguiente 7. El sistema muestra los datos al usuario 8. El caso de uso finaliza 	
Secuencias Alternativas	
3A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos introducidos es incorrecto 2.El sistema informa al actor 3.El caso de uso continua en el paso 2

Tabla 3.19: Descripción del CU 'Consultar analíticas sobre los datos'

CU-20: Añadir presupuesto	
Actor	Usuario
Resumen	El usuario quiere registrar un nuevo presupuesto en el sistema
Precondición	El actor tiene una sesión activa en el sistema
Postcondición	La información del nuevo presupuesto quedará registrada en el sistema
Secuencia Principal	
<ol style="list-style-type: none"> 1. El actor inicia el proceso de añadir presupuesto 2. El sistema solicita los datos 3. El actor introduce los datos 4. El sistema comprueba el formato de los datos y solicita confirmación 5. El actor confirma 6. El sistema almacena la información del presupuesto e informa al usuario 7. El caso de uso finaliza 	
Secuencias Alternativas	
3A, 5A	Si el actor cancela
Excepciones	
4A	<ol style="list-style-type: none"> 1.El sistema comprueba que el formato de los datos no es correcto 2.El sistema informa al actor 3.El caso de uso vuelve al paso 3

Tabla 3.20: Descripción del CU 'Añadir presupuesto'

3.5.4. Matriz de correspondencia de requisitos y casos de uso

CU/RF	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	RF11	RF12
CU1	■											
CU2							■					
CU3												
CU4												
CU5												
CU6									■			
CU7									■			
CU8												
CU9											■	
CU10									■			
CU11									■			
CU12								■				
CU13								■				
CU14								■				
CU15										■		
CU16								■				
CU17								■				
CU18				■								
CU19		■	■		■	■						
CU20												■

Figura 3.7: Matriz de correspondencia de CU/RF

3.5.5. Realización en Análisis de los casos de uso

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

Además, en los diagramas de hace referencia a los casos de uso auxiliares que por su sencillez se han excluido de esta sección.

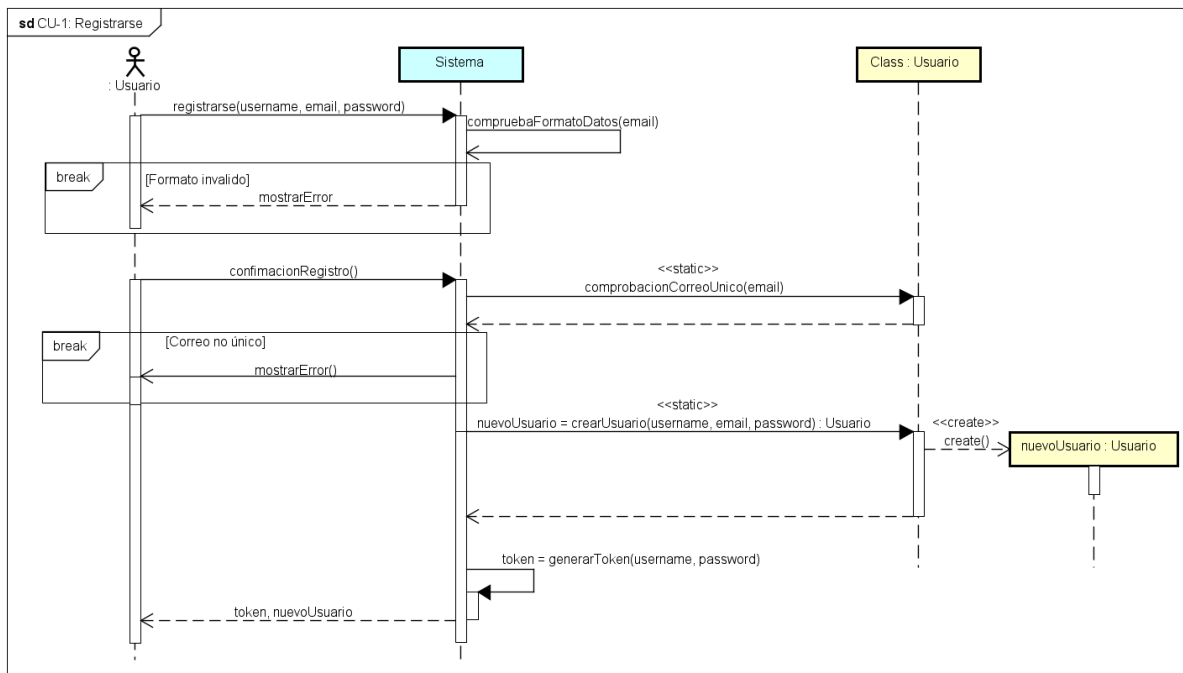


Figura 3.8: Secuencia en análisis del CU-1: Registrarse

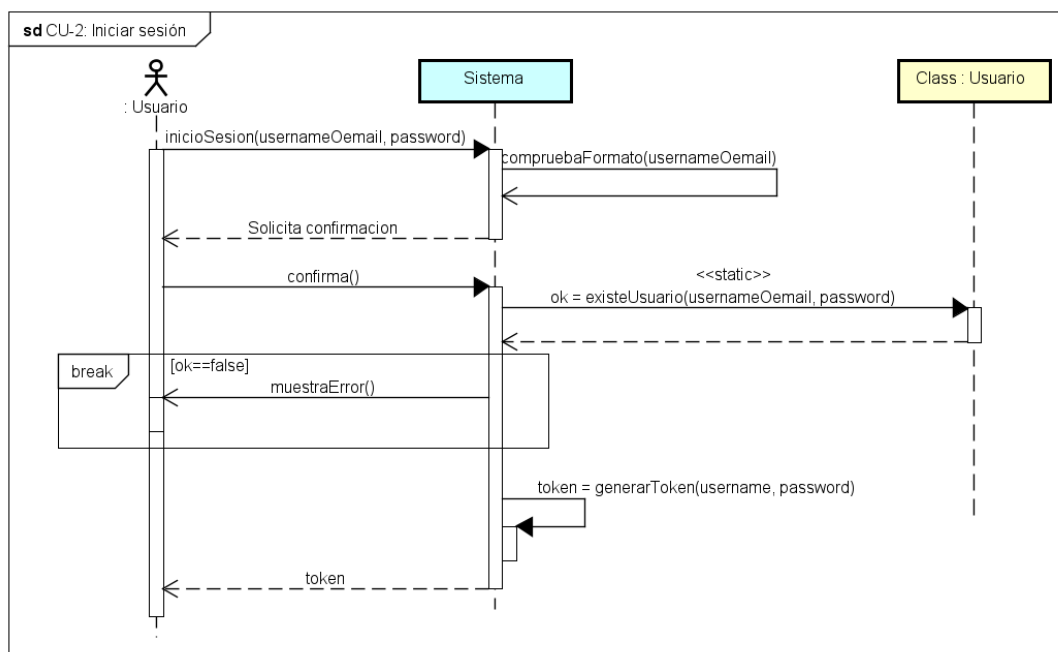


Figura 3.9: Secuencia en análisis del CU-2: Iniciar sesión

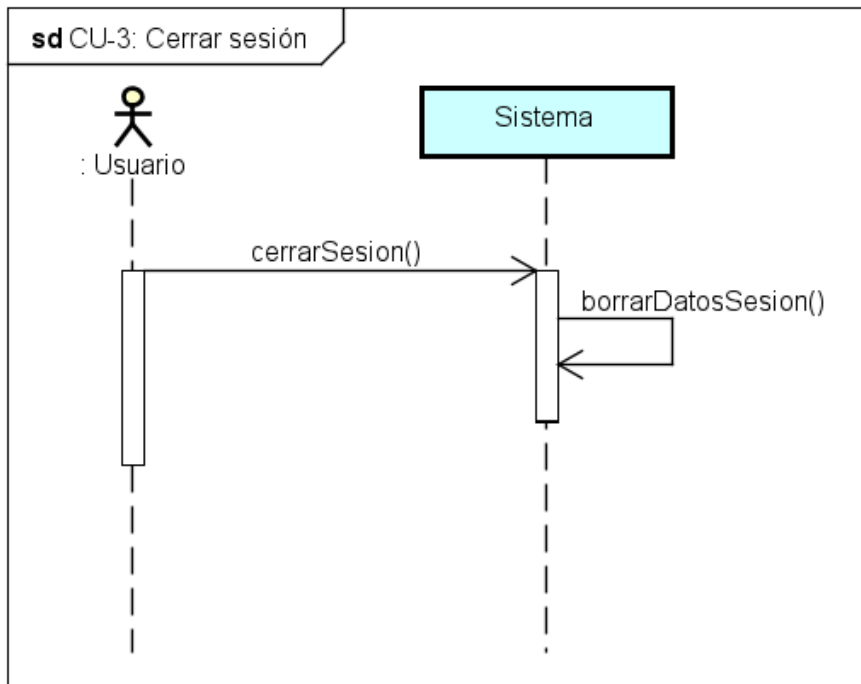


Figura 3.10: Secuencia en análisis del CU-3: Cerrar sesión

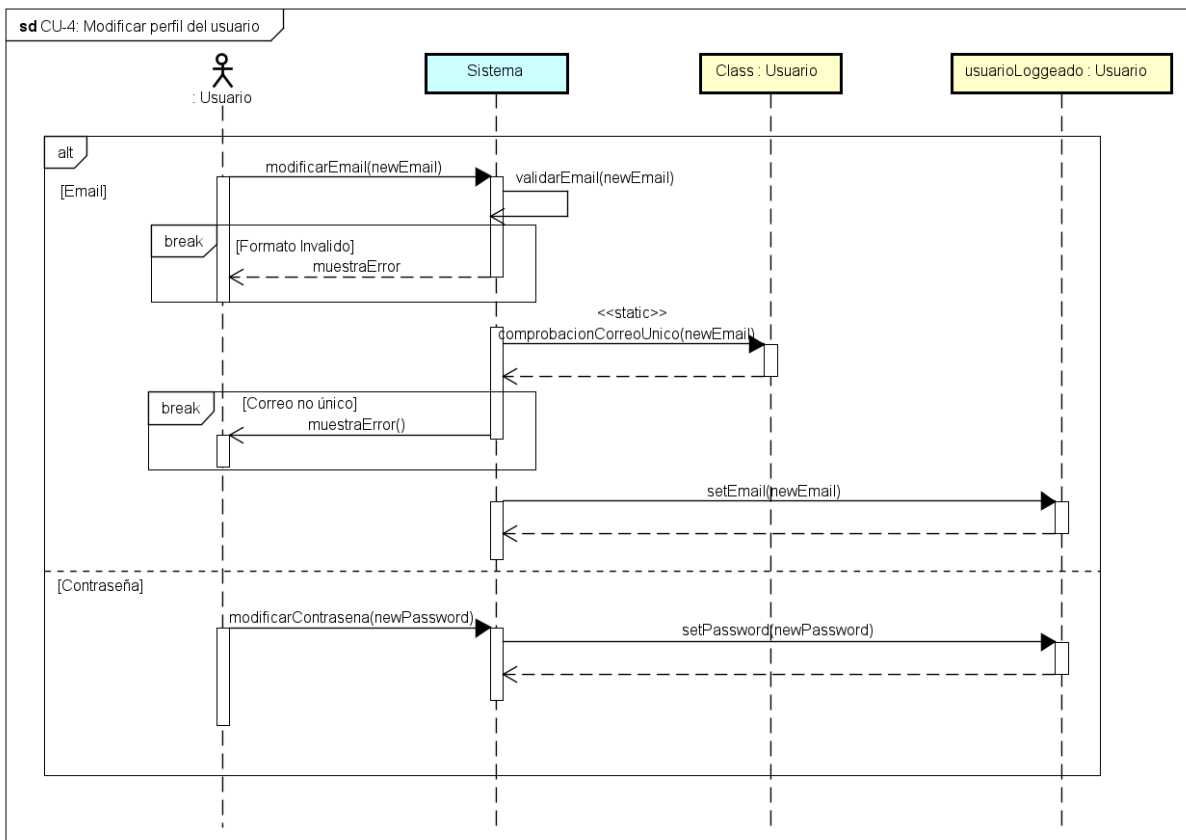


Figura 3.11: Secuencia en análisis del CU-4: Modificar perfil del usuario

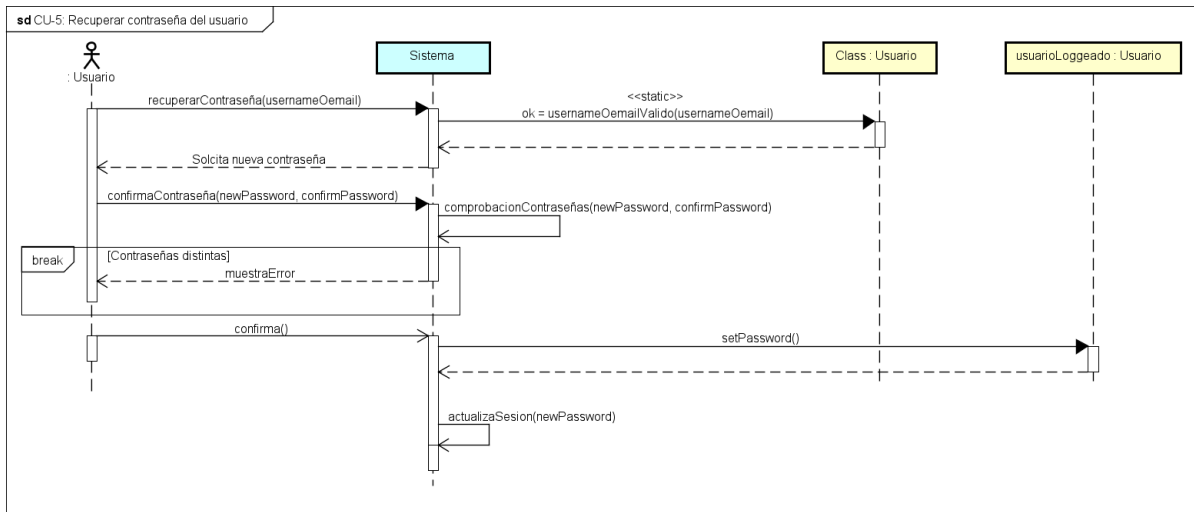


Figura 3.12: Secuencia en análisis del CU-5: Recuperar contraseña del usuario

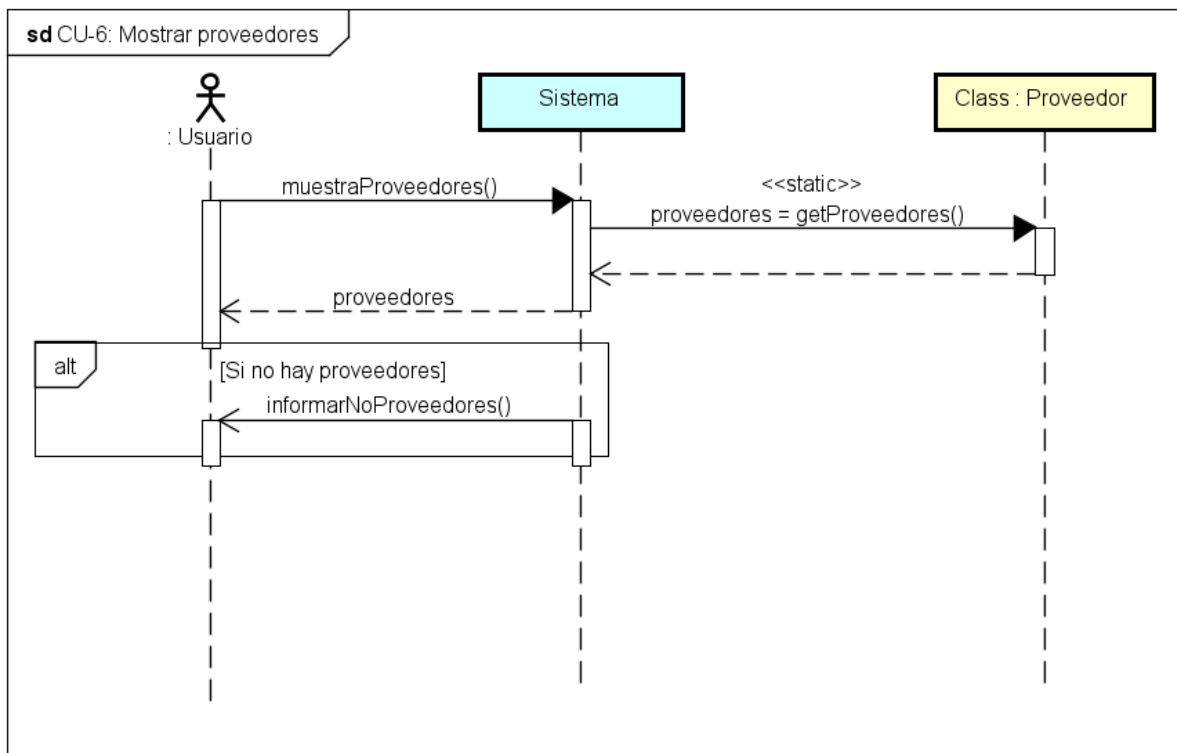


Figura 3.13: Secuencia en análisis del CU-6: Mostrar proveedores

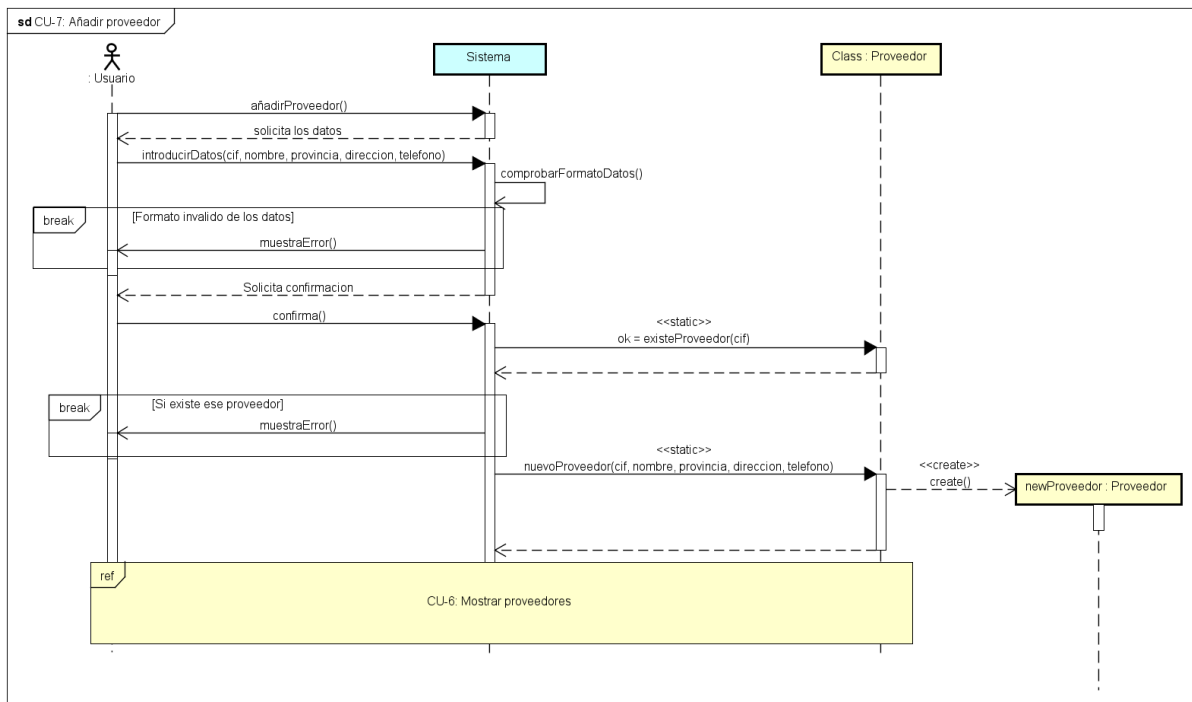


Figura 3.14: Secuencia en análisis del CU-7: Añadir proveedor

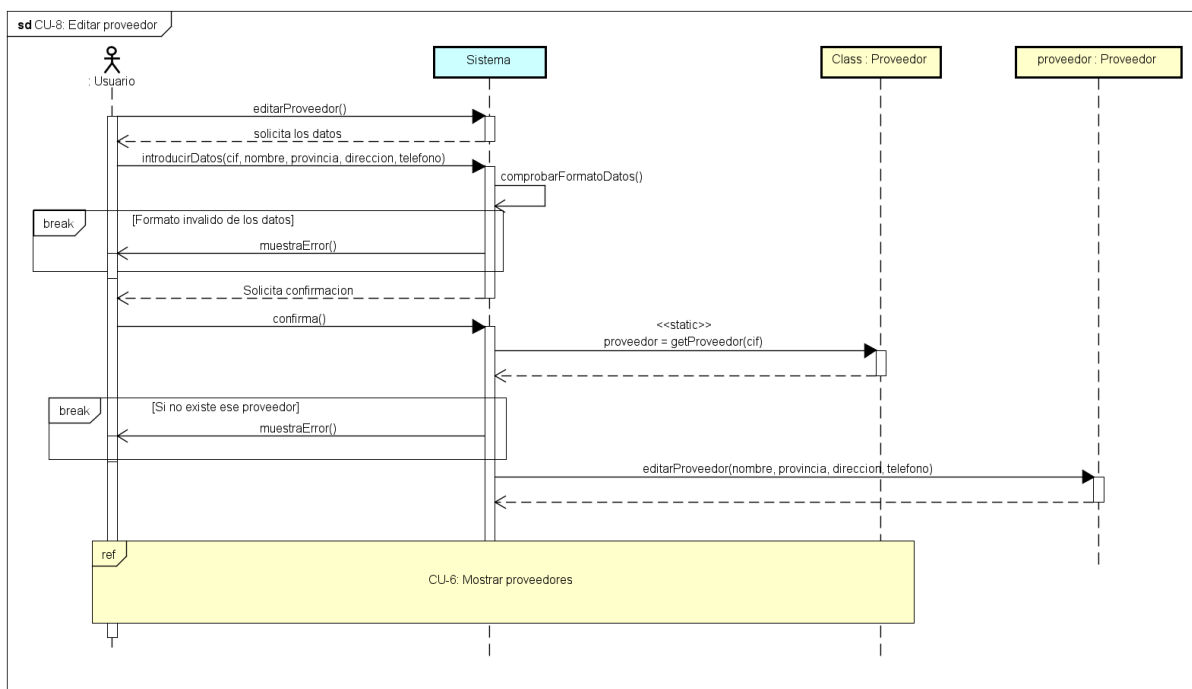


Figura 3.15: Secuencia en análisis del CU-8: Editar proveedor

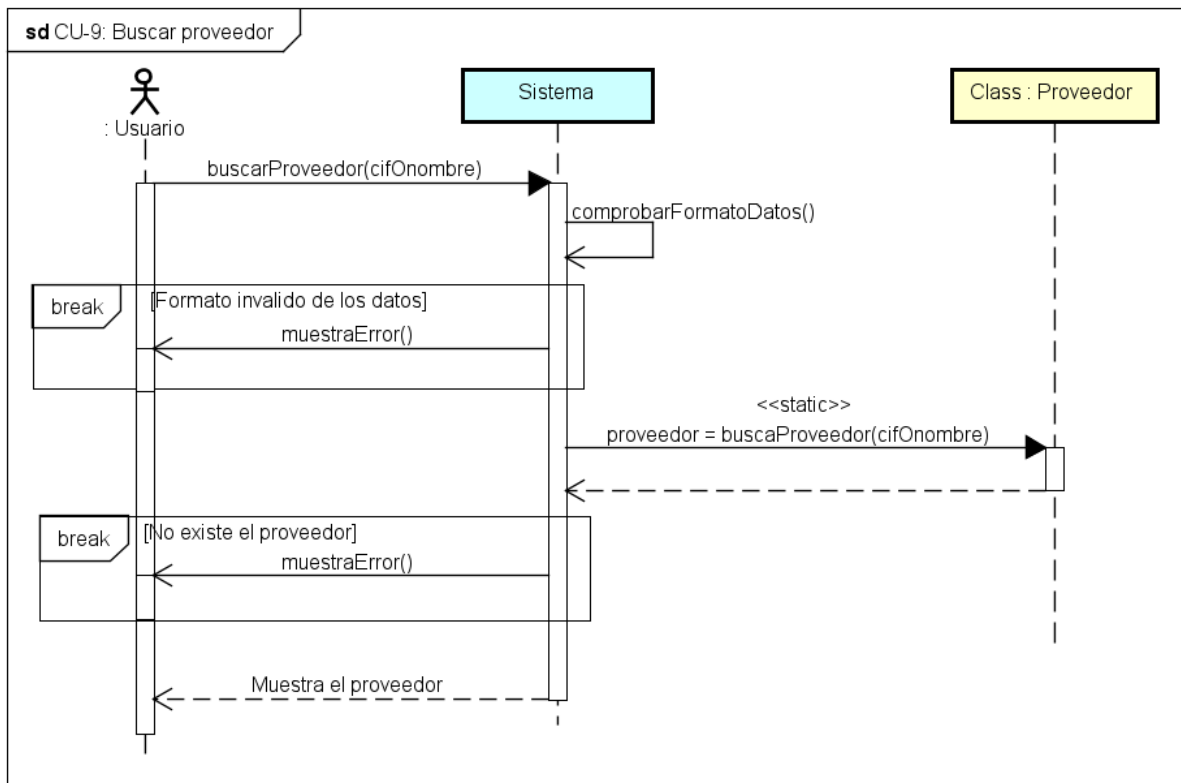


Figura 3.16: Secuencia en análisis del CU-9: Buscar proveedor

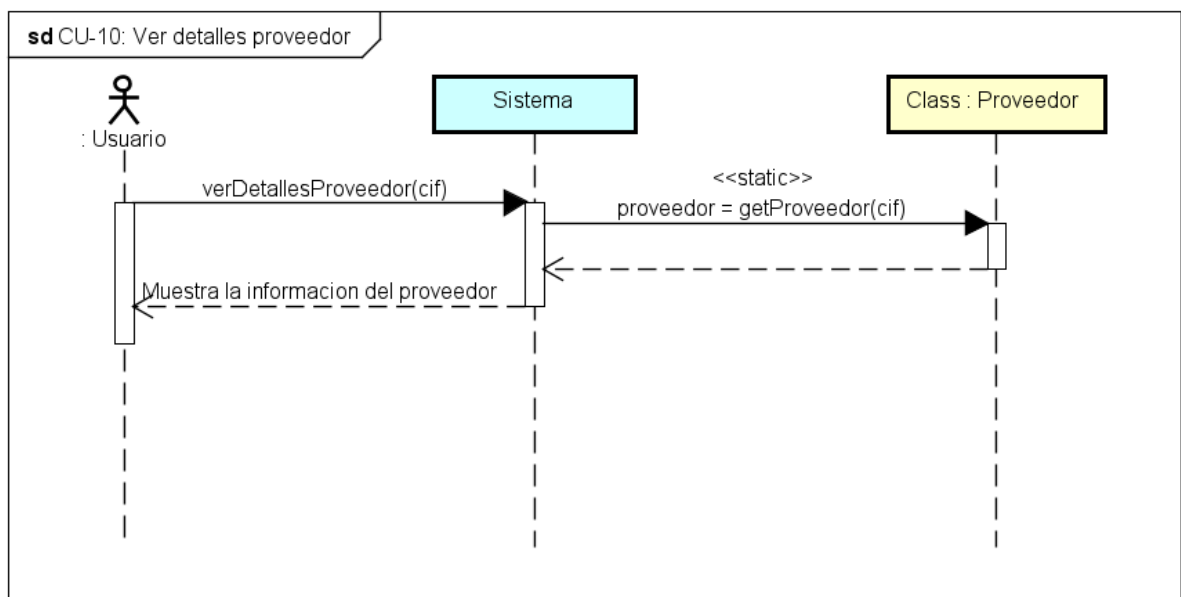


Figura 3.17: Secuencia en análisis del CU-10: Ver detalles proveedor

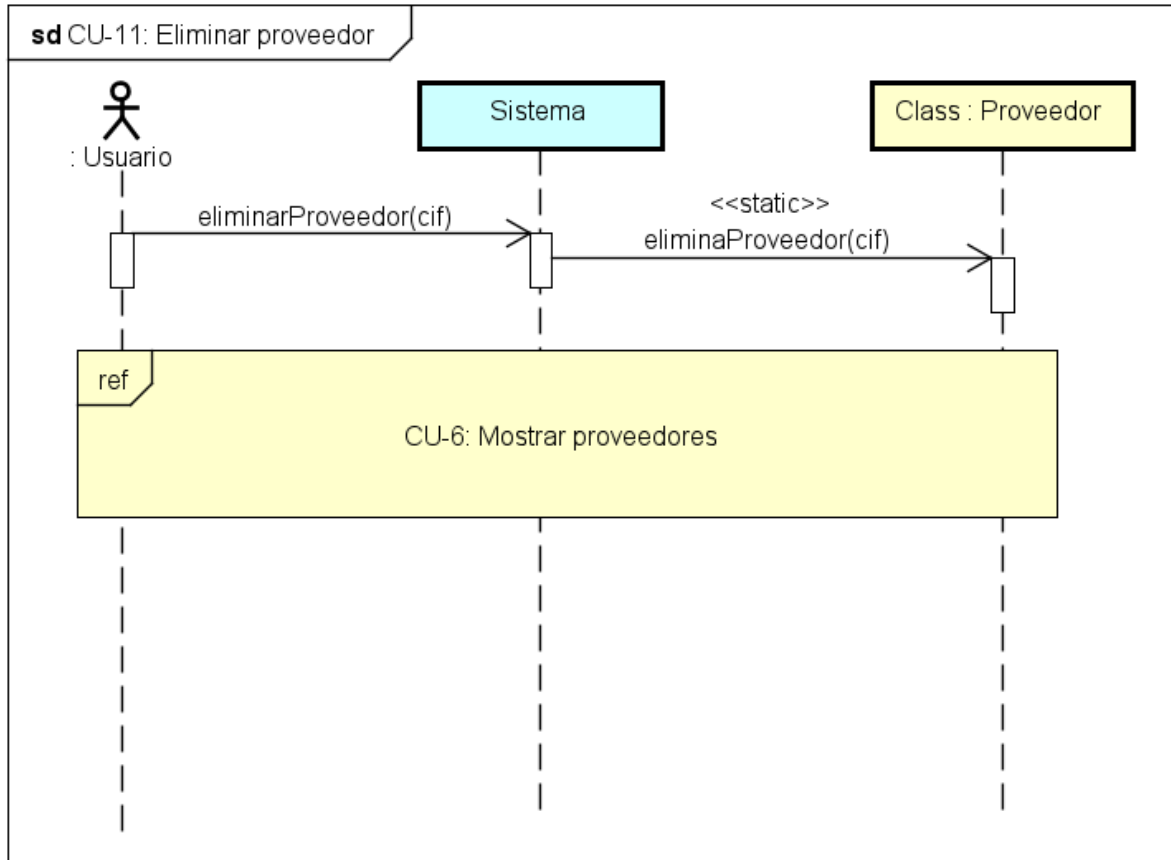


Figura 3.18: Secuencia en análisis del CU-11: Eliminar proveedor

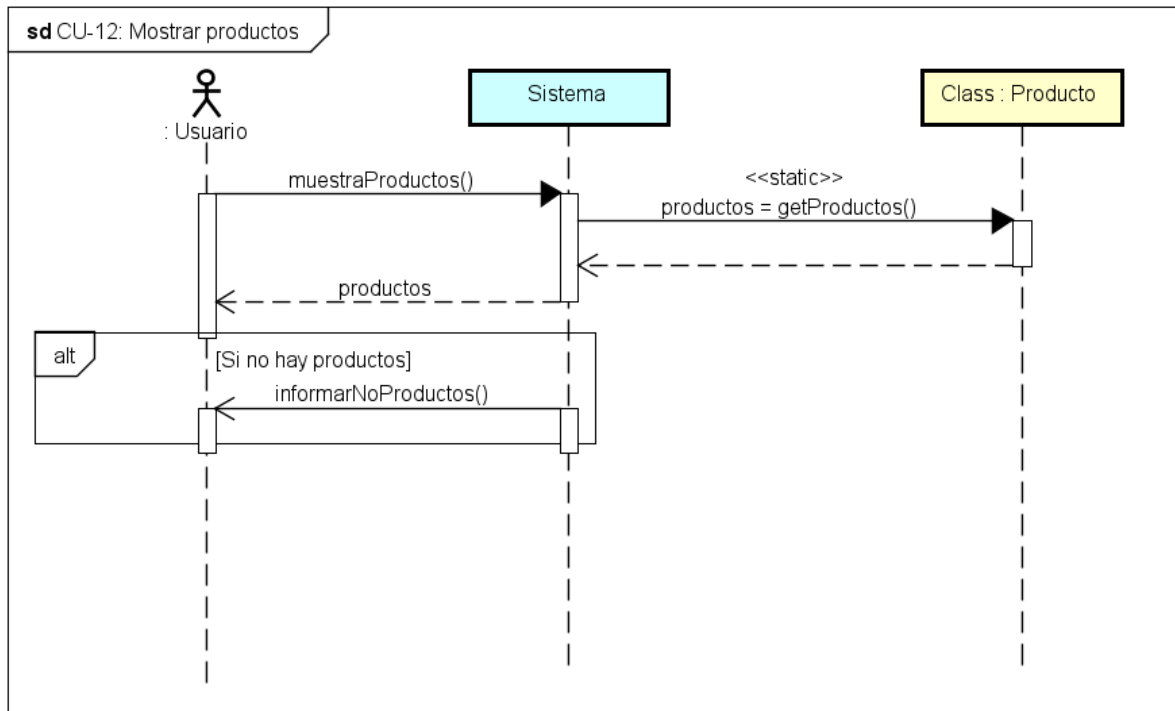


Figura 3.19: Secuencia en análisis del CU-12: Mostrar productos

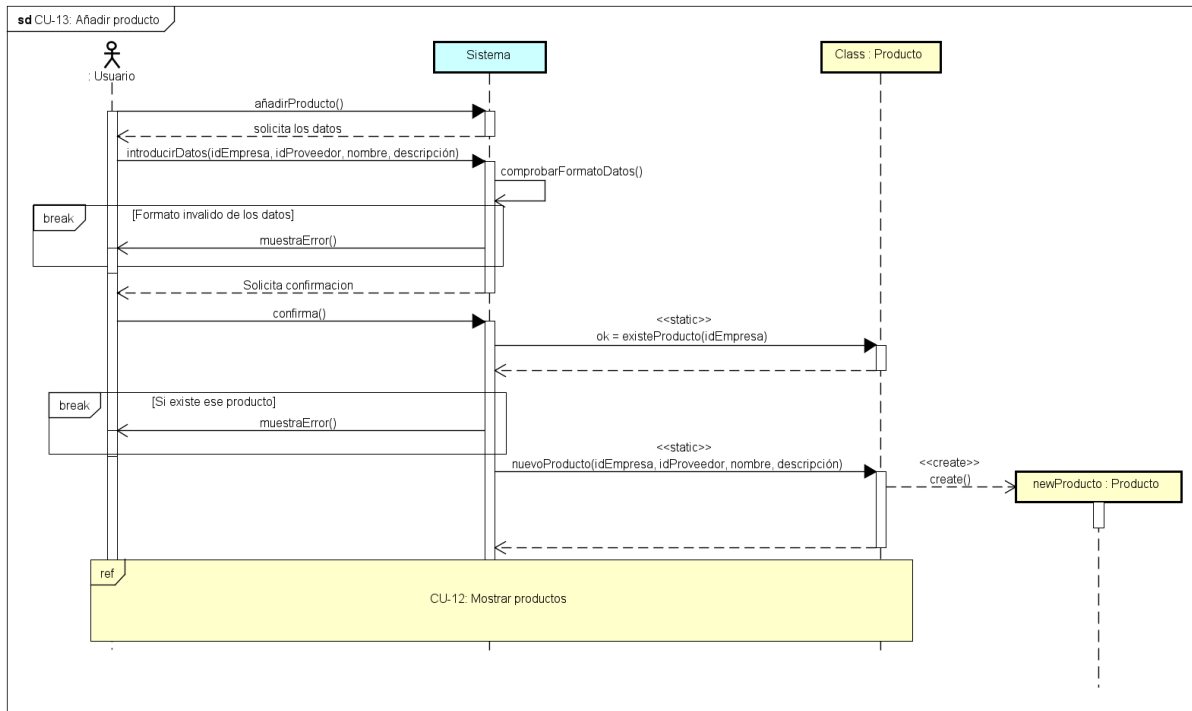


Figura 3.20: Secuencia en análisis del CU-13: Añadir producto

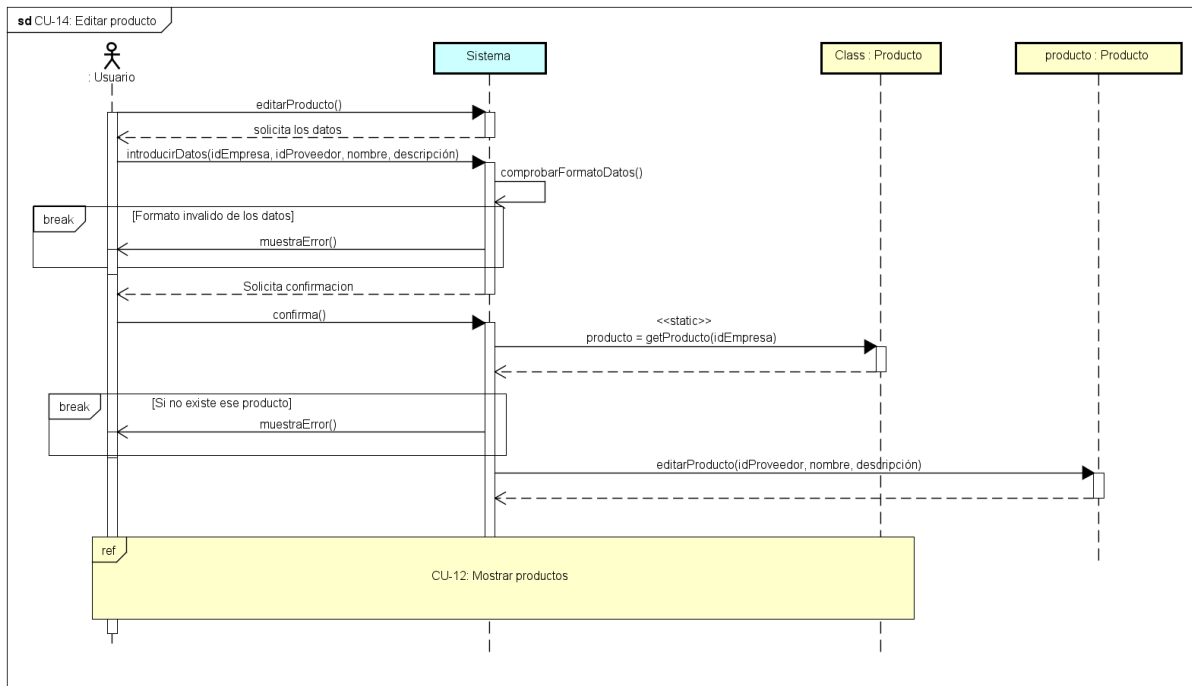


Figura 3.21: Secuencia en análisis del CU-14: Editar producto

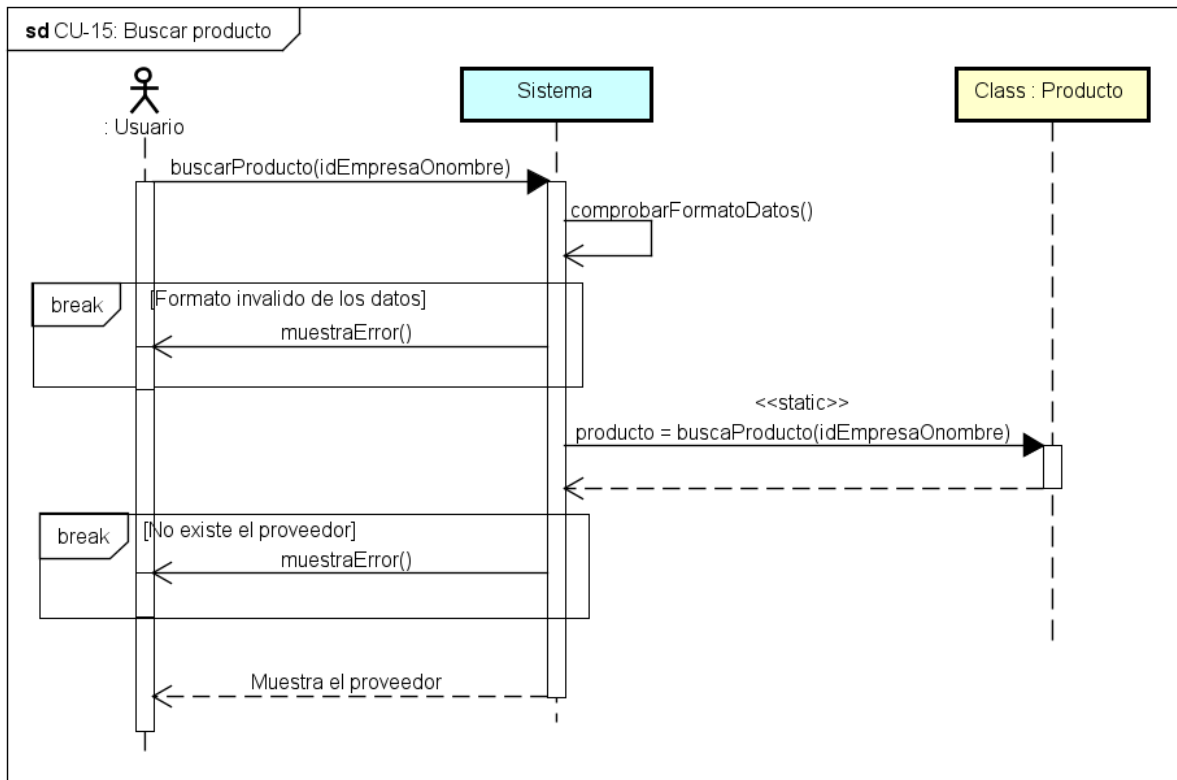


Figura 3.22: Secuencia en análisis del CU-15: Buscar producto

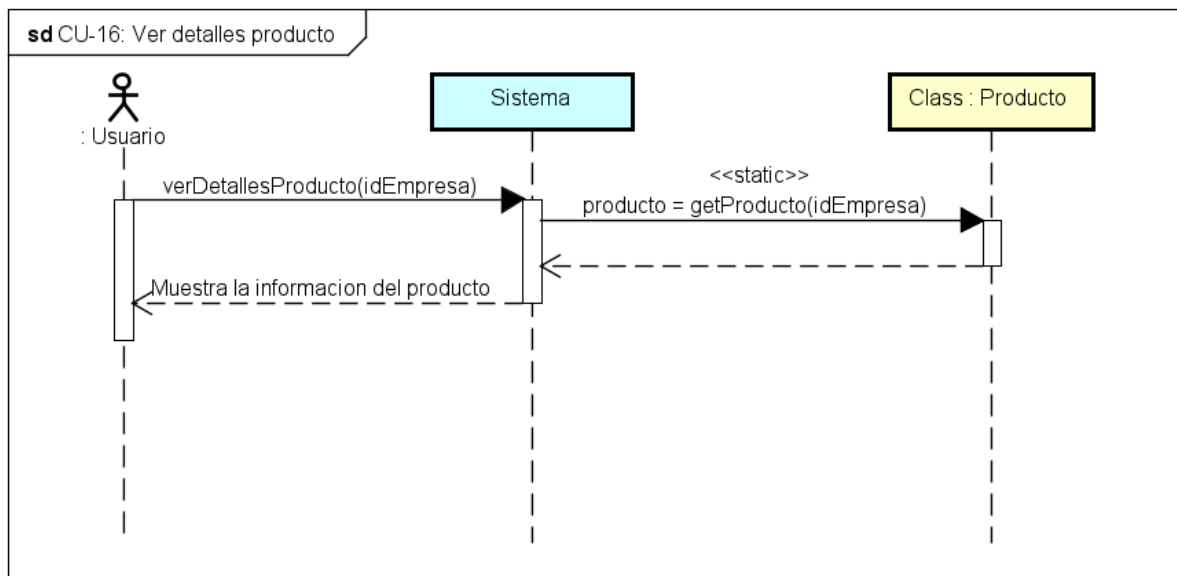


Figura 3.23: Secuencia en análisis del CU-16: Ver detalles producto

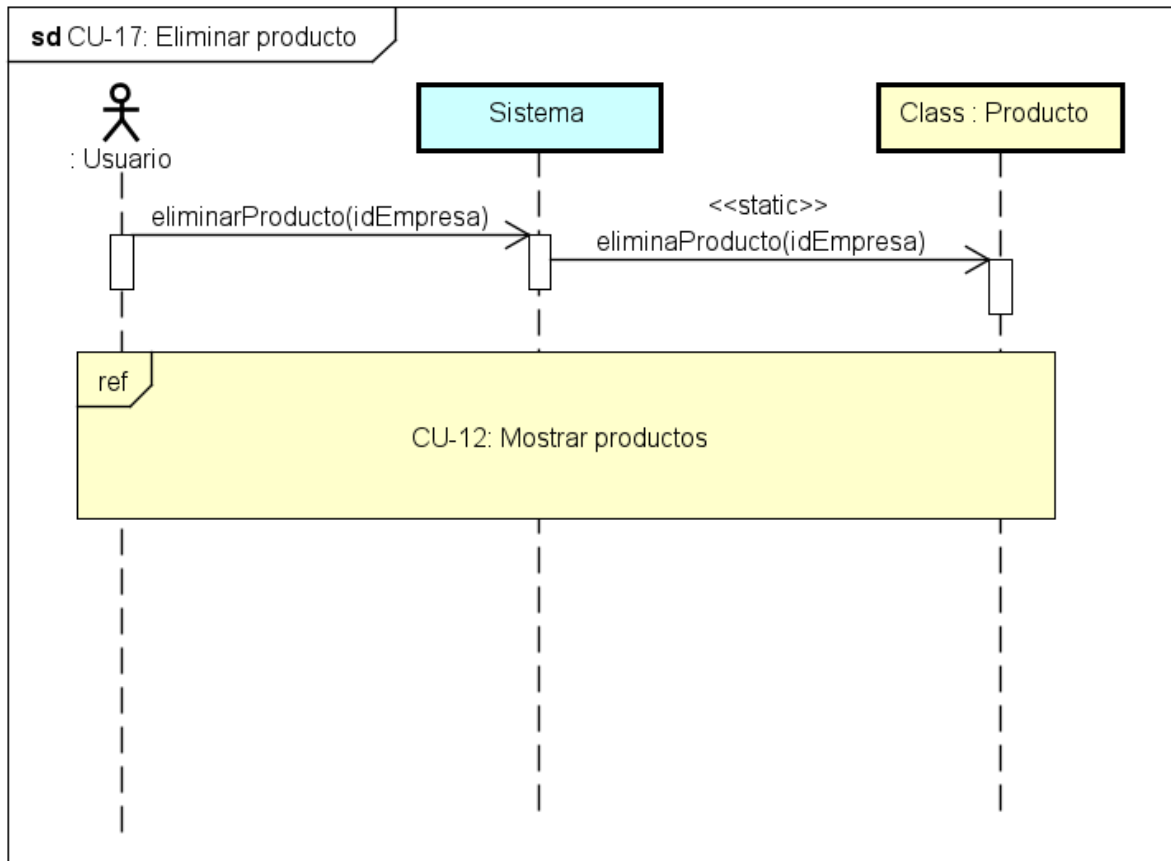


Figura 3.24: Secuencia en análisis del CU-17: Eliminar producto

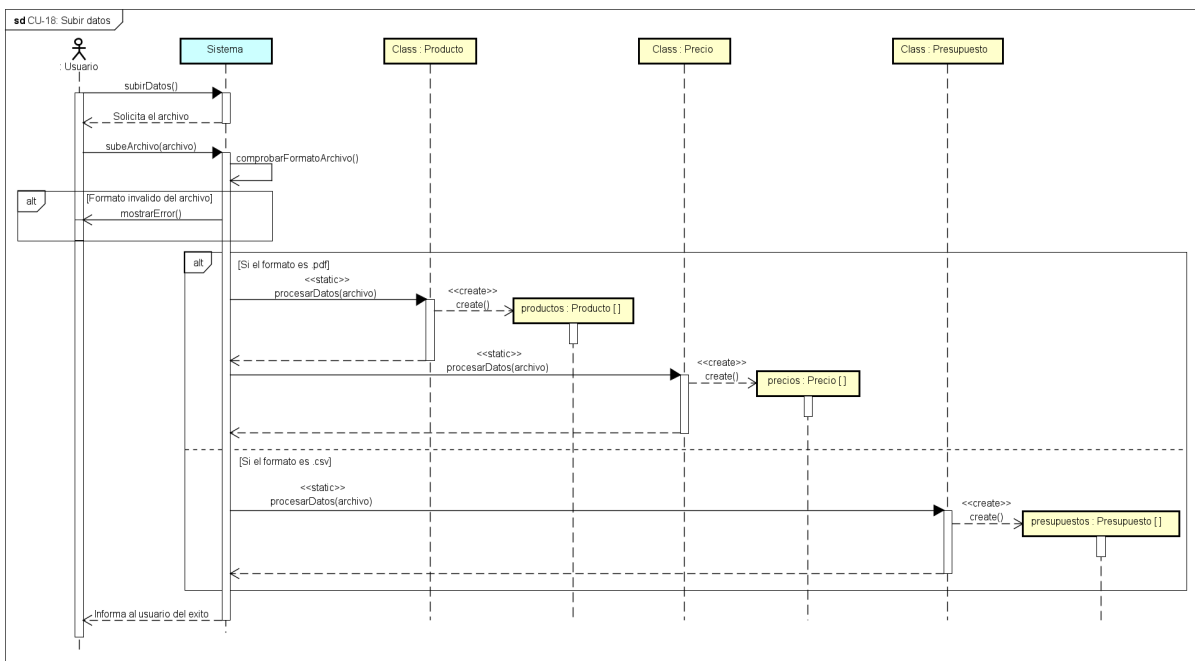


Figura 3.25: Secuencia en análisis del CU-18: Subir datos

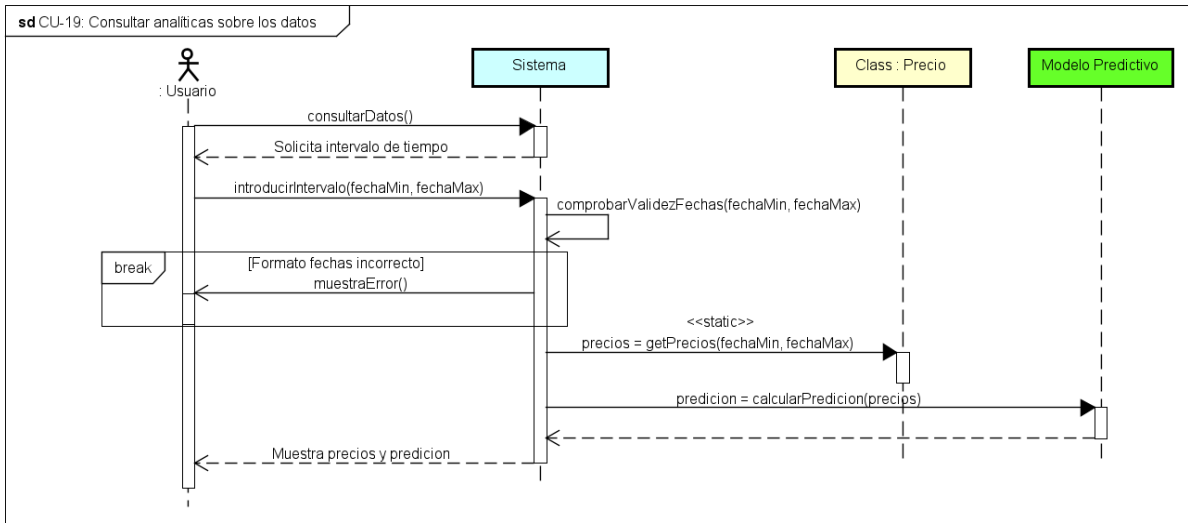


Figura 3.26: Secuencia en análisis del CU-19: Consultar analíticas sobre los datos

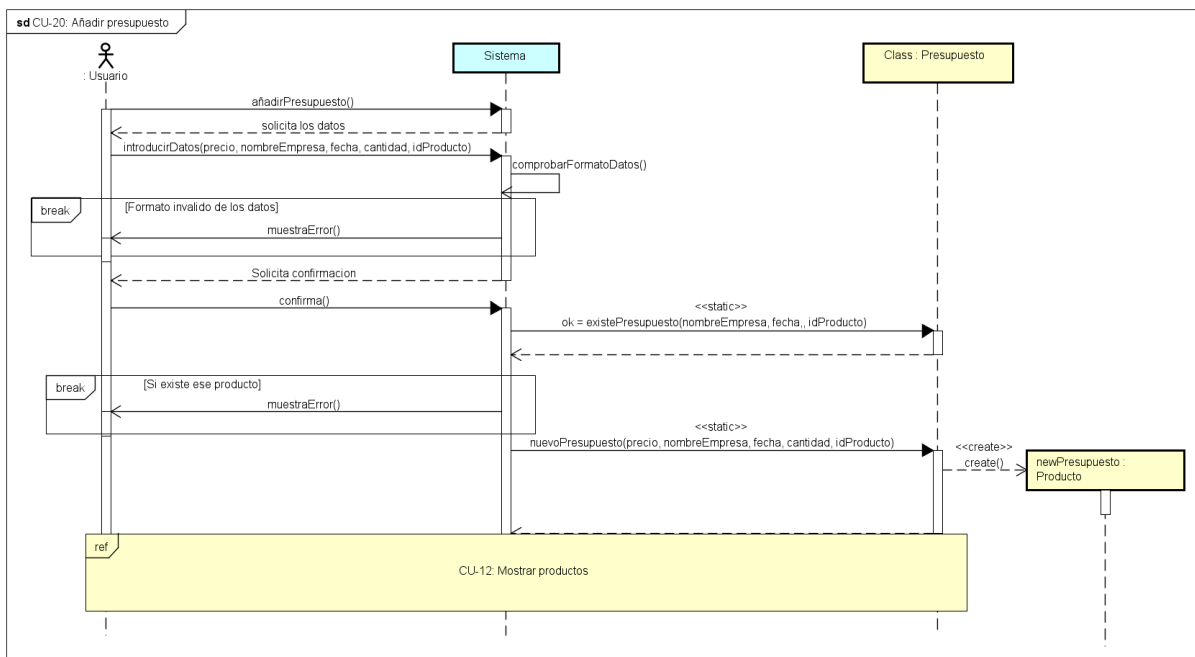


Figura 3.27: Secuencia en análisis del CU-20: Añadir Presupuesto

Capítulo 4

Diseño

El siguiente capítulo aborda la fase de diseño del proyecto software, explicando los frameworks utilizados, la arquitectura del proyecto, primero de forma general y después más detalladamente comentando cada una de las capas. Posteriormente se describen los patrones arquitectónicos utilizados, el diseño de la base de datos, el despliegue de la aplicación y alguna de las decisiones tomadas durante el diseño.

4.1. Arquitectura del sistema

4.1.1. Patrón arquitectónico Cliente-Servidor

La arquitectura Cliente-Servidor es uno de los modelos más utilizados en el desarrollo de aplicaciones. En esencia, esta arquitectura implica la interacción entre dos componentes principales: el cliente, que consume servicios, y el servidor, que proporciona dichos servicios. Esta comunicación entre cliente y servidor se lleva a cabo, generalmente, a través de la red.

Esta arquitectura se puede aplicar a una amplia gama de aplicaciones y escenarios. Por ejemplo, todas las aplicaciones web son ejemplos de aplicaciones cliente-servidor, al igual que esta aplicación. En este caso, el cliente sería el navegador web que utilizan los usuarios finales, mientras que el servidor sería la máquina que aloja la aplicación, base de datos.

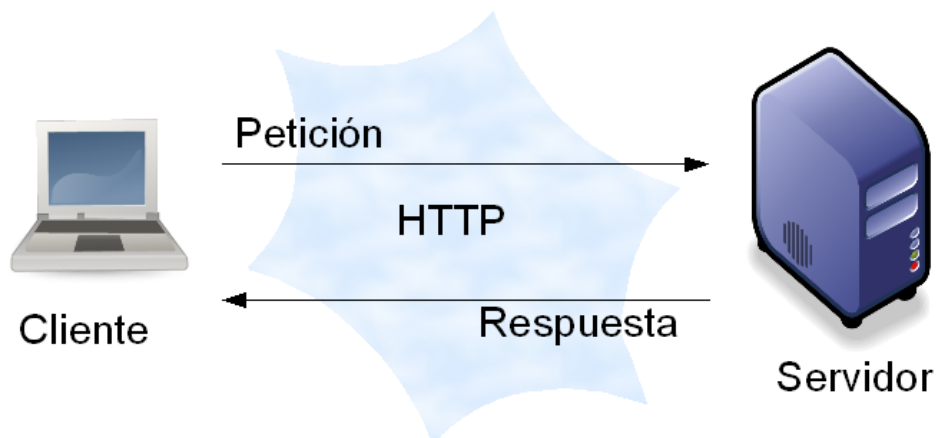


Figura 4.1: Representación gráfica del patrón Cliente-Servidor

Aplicaremos a esta arquitectura cliente-servidor un patrón arquitectónico de capas. Existen

hasta tres tipos diferenciados en capas, que son 2 capas, 3 capas y N capas. Estos niveles o capas vienen determinados en función de si existen otras aplicaciones o servidores que proporcionen recursos o servicios a la aplicación. No es objetivo de este estudio profundizar en los detalles de las diferentes ventajas e inconvenientes que surgen de las modificaciones de este patrón.

El marco de trabajo utilizado para el desarrollo de este TFG se basa en el Cliente-Servidor de 2 capas ya que esto permite dividir la complejidad en dos partes por lo que cada una por su lado es más fácil de entender y desarrollar que el sistema completo.

Las 2 capas están compuestas de la siguiente manera:

- **Capa de presentación:** En esta capa reside el Frontend y maneja la interfaz de usuario y la interacción con el usuario final, al mismo tiempo de consumir distintos servicios del Backend.
- **Capa de lógica de aplicación:** En esta capa el Backend proporciona servicios RESTful que son consumidos por el Frontend, al mismo tiempo que se encarga de realizar las operaciones en la base de datos.

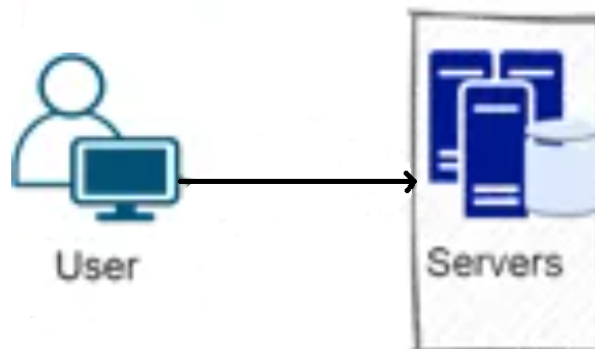


Figura 4.2: Cliente-Servidor de 2 capas

En la imagen anterior se puede apreciar un ejemplo de un cliente-servidor de 2 capas, donde en la primera capa se encuentra únicamente el cliente y en la segunda capa se encuentra el servidor con la base de datos.

4.1.2. Arquitectura del cliente

Para describir la arquitectura del cliente se van a emplear los diagramas propios de los estilos de documentación Modules Style y Uses Style. Forman parte del modelo de vistas “4+1” que nos permite representar y documentar arquitecturas software desde puntos de vistas diferentes, concurrentes y que informan sobre diferentes aspectos del sistema. En las Figuras 4.3, 4.4 y 4.5 se puede apreciar la arquitectura del cliente descrita utilizando este tipo de diagramas.

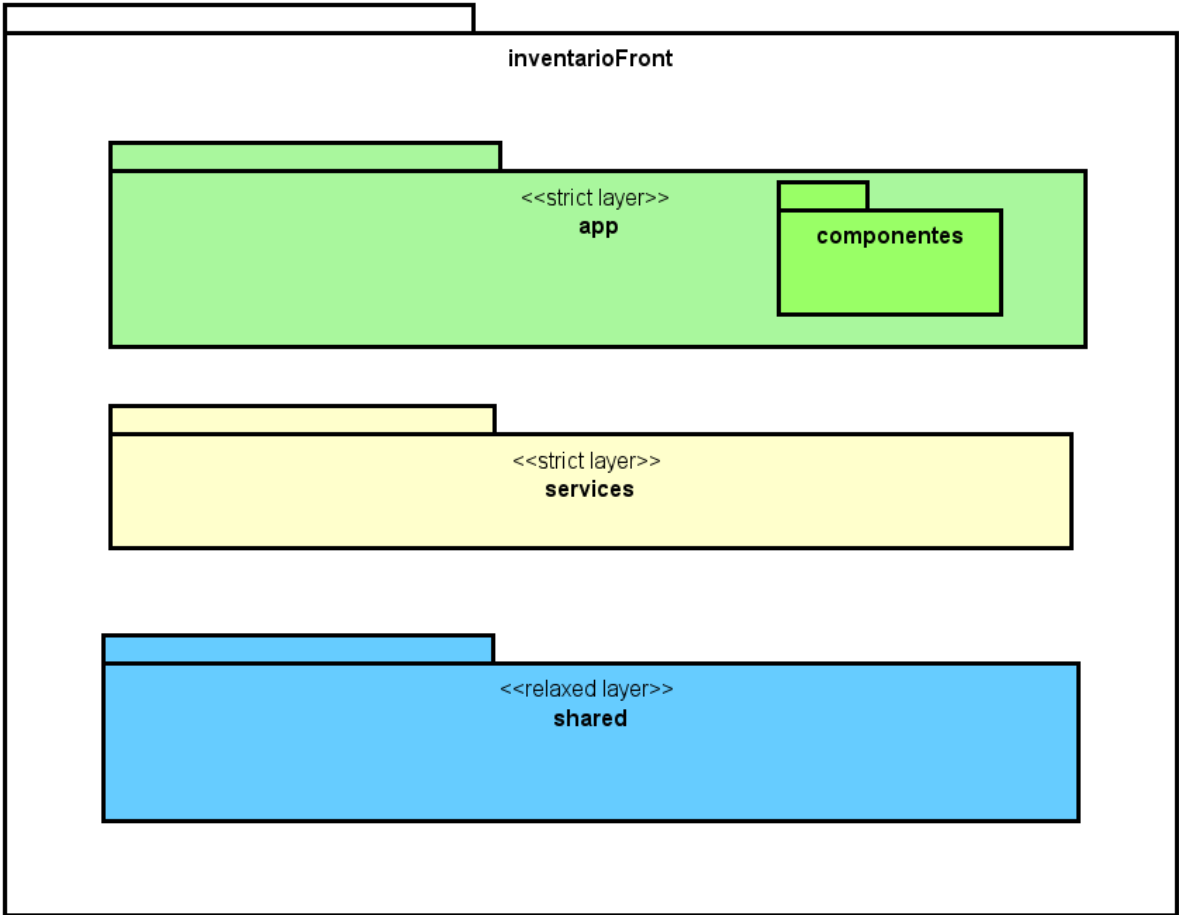


Figura 4.3: ModulesStyle Cliente Angular

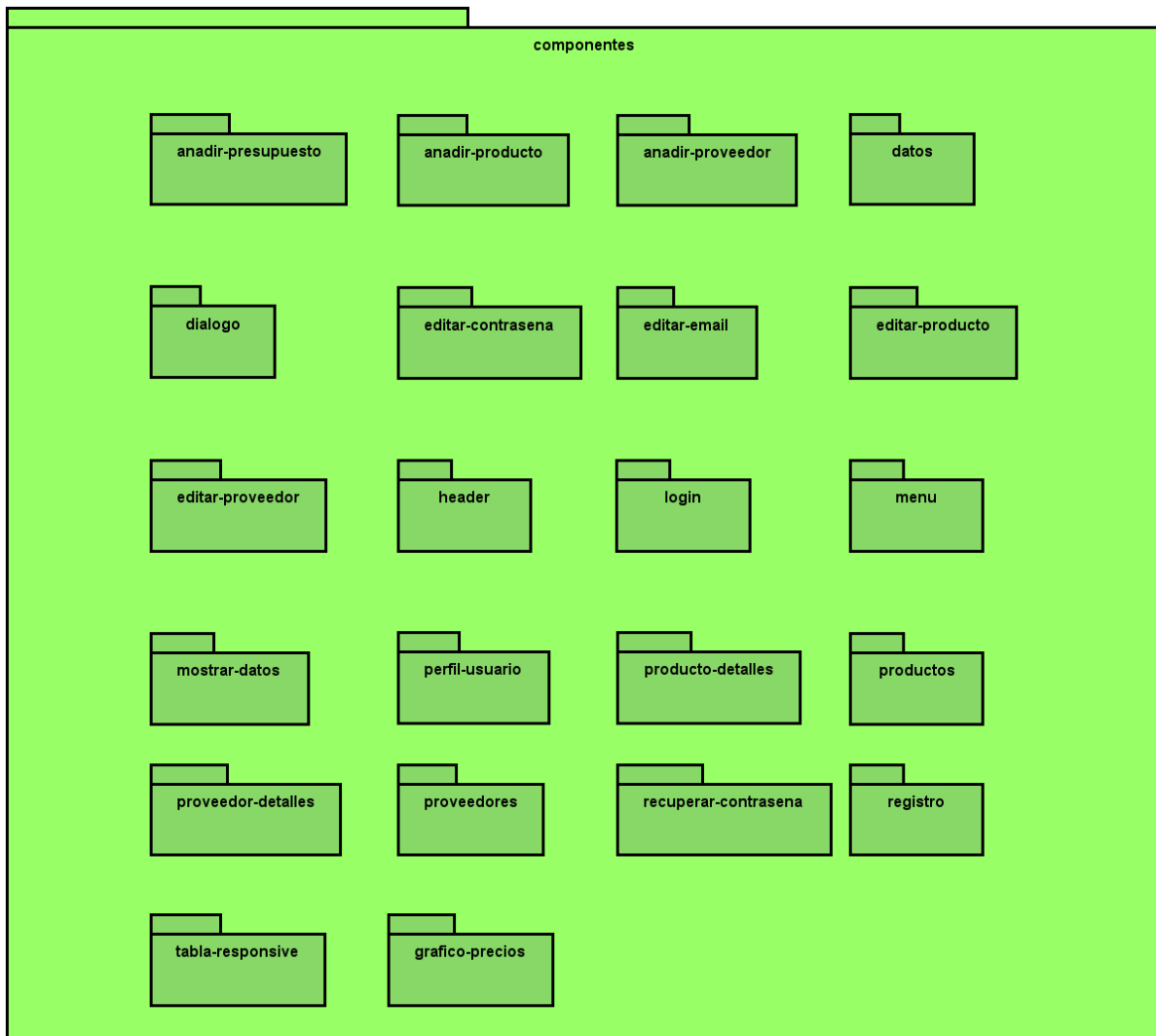


Figura 4.4: Modules Style Componentes

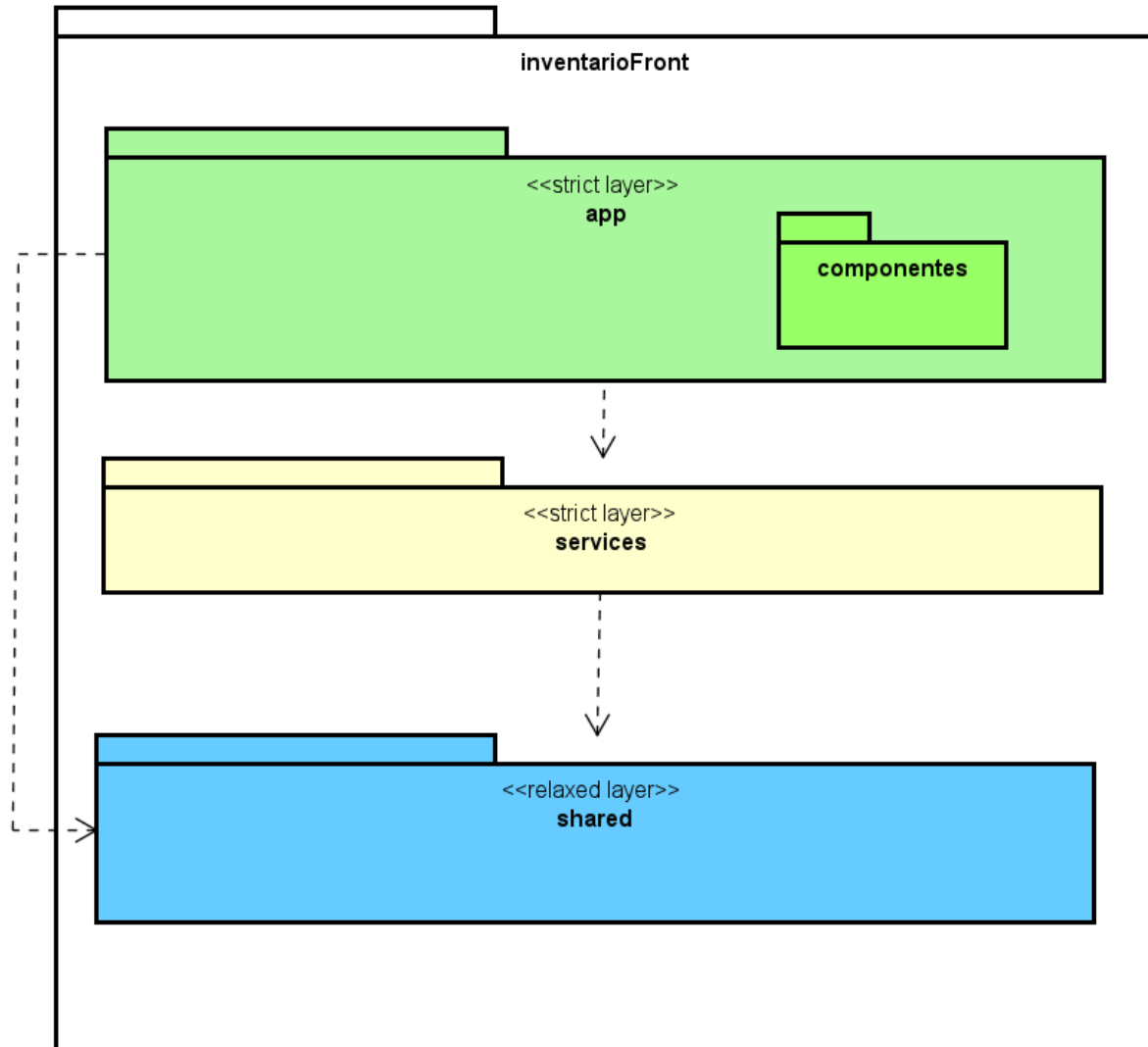


Figura 4.5: Uses Style Cliente Angular

4.1.3. Arquitectura del servidor

Para un servicio web RESTful implementado en Java Spring, como es el caso de esta aplicación, el patrón arquitectónico predominante es el Modelo-Vista-Controlador (MVC). Sin embargo, en el contexto de un servicio RESTful, la "vista" no se refiere a una interfaz de usuario renderizada, sino más bien al contenido que se envía como respuesta a las solicitudes HTTP, generalmente en formato JSON o XML, como se puede apreciar en la siguiente imagen:

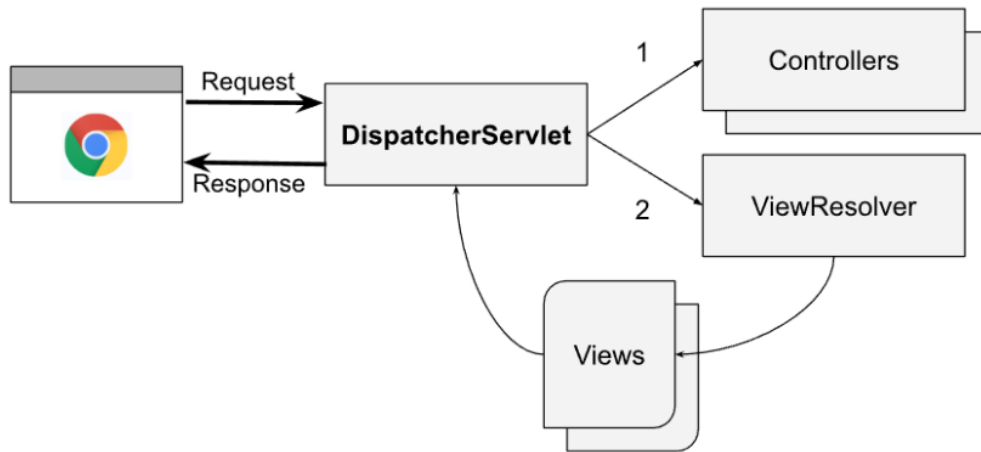


Figura 4.6: Ejemplo MVC Java Spring

Para describir la arquitectura del servidor se van a emplear los diagramas propios de los estilos de documentación Modules Style y Uses Style. Forman parte del modelo de vistas “4+1” que nos permite representar y documentar arquitecturas software desde puntos de vistas diferentes, concurrentes y que informan sobre diferentes aspectos del sistema. En las Figuras 4.7 y 4.8 se puede apreciar la arquitectura del servidor descrita utilizando este tipo de diagramas.

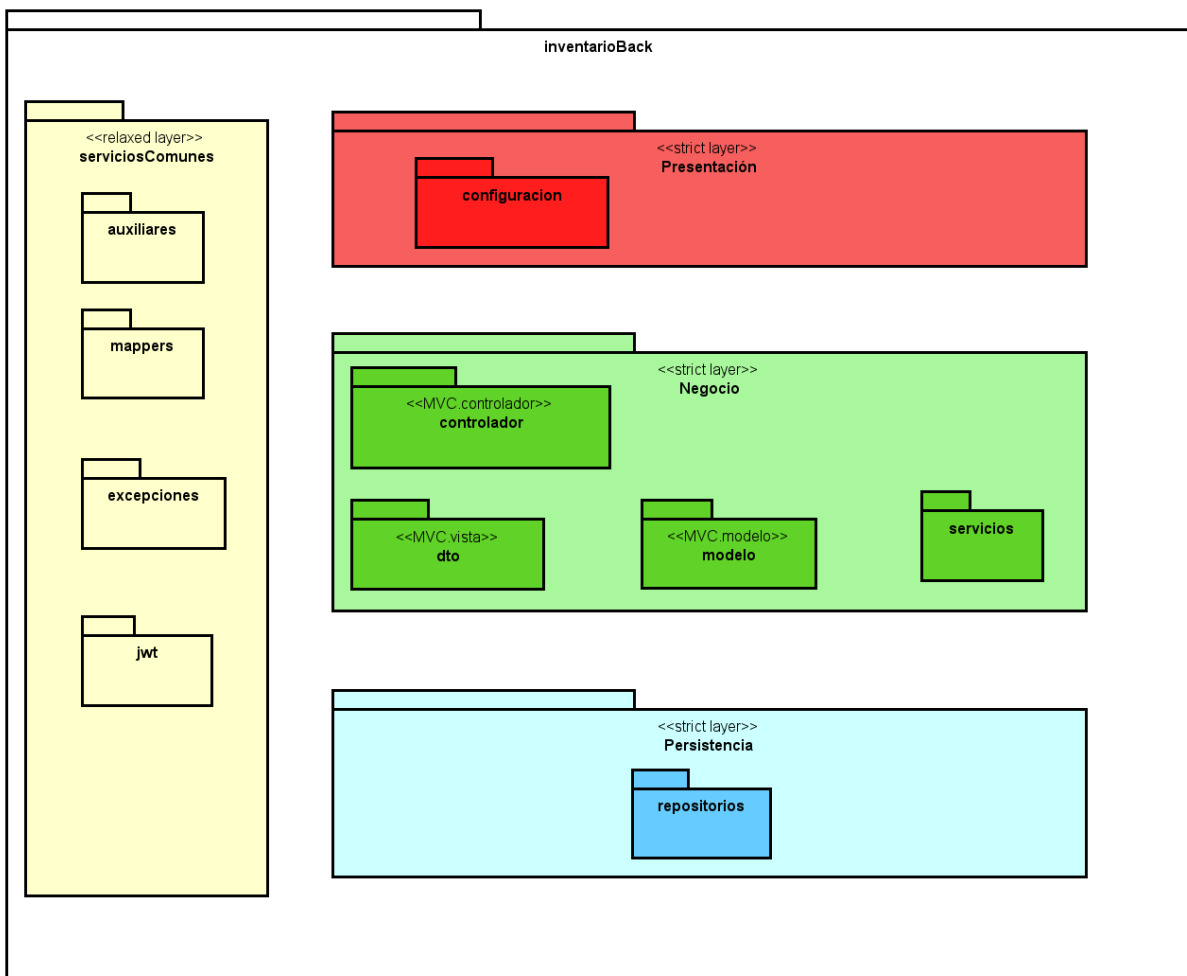


Figura 4.7: Modules Style Servidor Java Spring

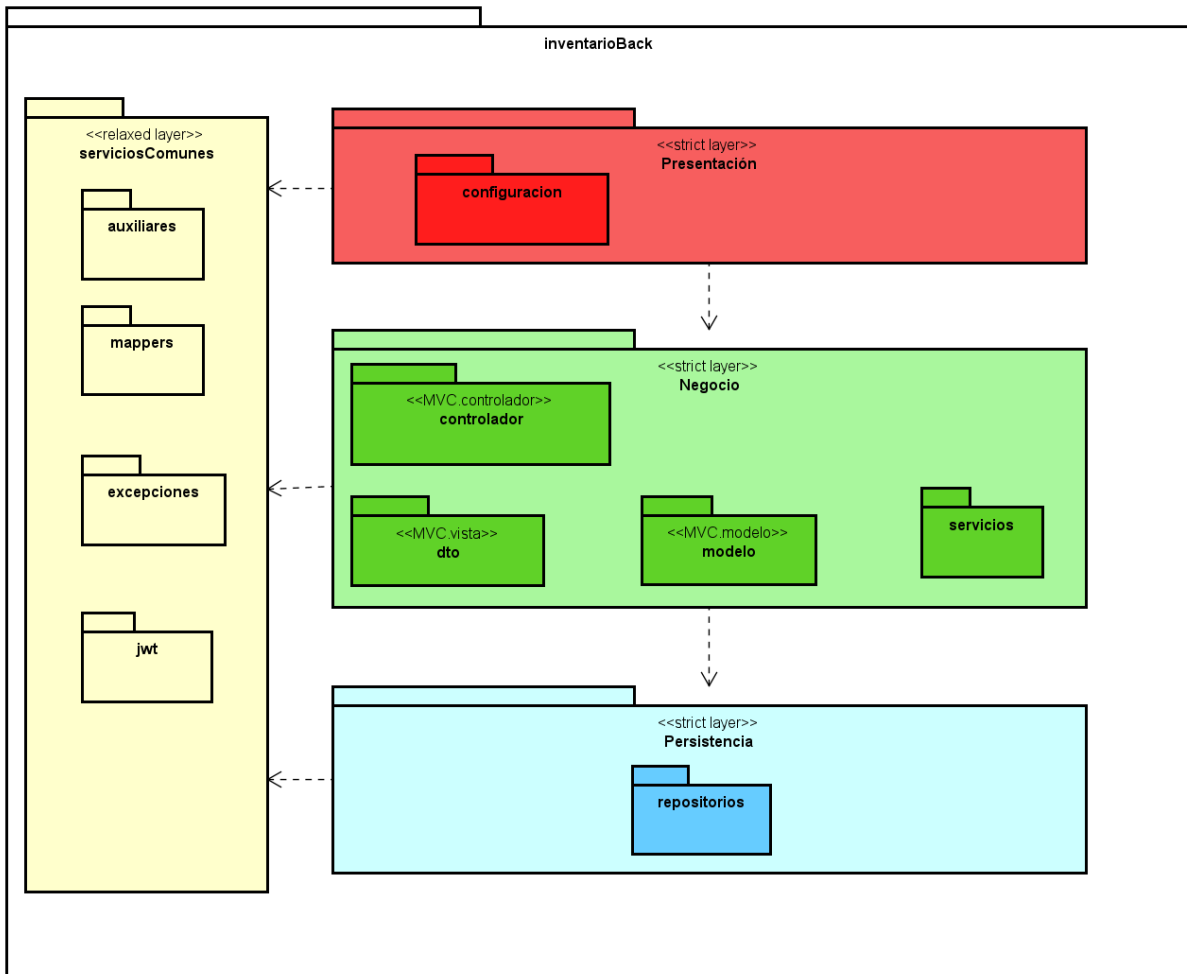


Figura 4.8: Uses Style Servidor Java Spring

4.2. Otros patrones empleados

Debido al empleo de Java Spring también ha sido necesario emplear otros patrones arquitectónicos como:

- Patrón Inyección de Dependencias
- Patrón Repositorio
- Patrón Servicio
- Patrón DAO/DTO
- Patrón Singleton

4.2.1. Patrón Inyección de Dependencias (Dependency Injection)

La Inyección de Dependencias (DI) es un patrón de diseño ampliamente utilizado en el marco de Spring para administrar las dependencias entre los componentes de una aplicación. En lugar de que un objeto cree o busque sus propias dependencias, Spring las proporciona externamente durante la inicialización del objeto.

En Spring, este enfoque se logra mediante la configuración de los componentes y sus dependencias en un contenedor de aplicaciones, que es administrado por el propio Spring. Cuando un

componente necesita una dependencia, Spring la inyecta en el componente durante su creación. Esto permite una mayor flexibilidad y facilita la gestión de las relaciones entre los diferentes componentes de la aplicación.

Dentro de la aplicación que se está analizando, se puede apreciar con el servicio de JWT o para el encriptador de la contraseña.

4.2.2. Patrón Repositorio (Repository Pattern)

El patrón Repositorio se utiliza para encapsular la lógica de acceso a datos y proporcionar una abstracción sobre la persistencia de datos. Este patrón es comúnmente utilizado en aplicaciones que necesitan interactuar con una base de datos u otro almacenamiento de datos. En el contexto de Spring, un repositorio generalmente consiste en una interfaz que define métodos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en una entidad de dominio específica. Estos métodos de repositorio se corresponden con operaciones comunes de acceso a datos, como guardar, buscar por identificador, buscar todos, actualizar y eliminar.

Para la implementación de este patrón en la aplicación se ha empleado Spring Data JPA, que es un subproyecto de Spring que simplifica aún más el uso del patrón Repositorio al proporcionar una implementación automática basada en interfaces.

4.2.3. Patrón Servicio (Service Pattern)

El patrón Servicio se utiliza para encapsular la lógica de negocio de una aplicación. Los servicios en Spring son componentes que contienen la lógica que no pertenece directamente a las capas de controlador ni de acceso a datos.

Los servicios en Spring se definen mediante clases anotadas con `@Service`. Estas clases encapsulan la lógica de negocio de la aplicación.

4.2.4. Patrón DAO/DTO

Para el acceso a base de datos se emplea el patrón DAO/DTO, siguiendo la siguiente figura:

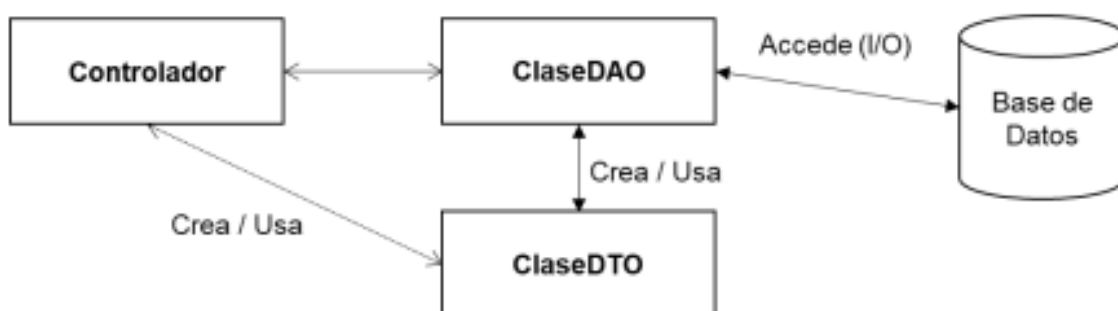


Figura 4.9: Patrón DAO/DTO

Donde los DAOs serán los repositorios que se crean siguiendo el patrón repositorio, los DTO serán los mapeos de la tabla correspondiente en la base de datos, realizados a través de Hibernate y la anotación `@Entity`. Habrá tantos DTOs como tablas tenga la base de datos. Y el controlador será el que accede a la base de datos invocando los métodos correspondientes de los DAOs. Dependiendo de si la operación es de lectura o escritura, se utilizan o crean objetos o colecciones de objetos DTO para interactuar con la base de datos.

4.2.5. Patrón Singleton

En Spring, el patrón Singleton se refiere a la creación de un solo objeto de un bean específico en el contexto de la aplicación Spring. Cuando un bean se declara como un singleton en Spring, Spring IoC Container crea exactamente una instancia de ese bean y la comparte a lo largo de la aplicación.

Para crear un *bean singleton* en Spring, simplemente se anota la clase con cualquier anotación que marque la clase como un componente administrado por Spring. Spring gestionará automáticamente la creación y el ciclo de vida del bean, asegurando que solo se cree una instancia del bean en el contexto de la aplicación.

4.3. Diagrama detallado de clases

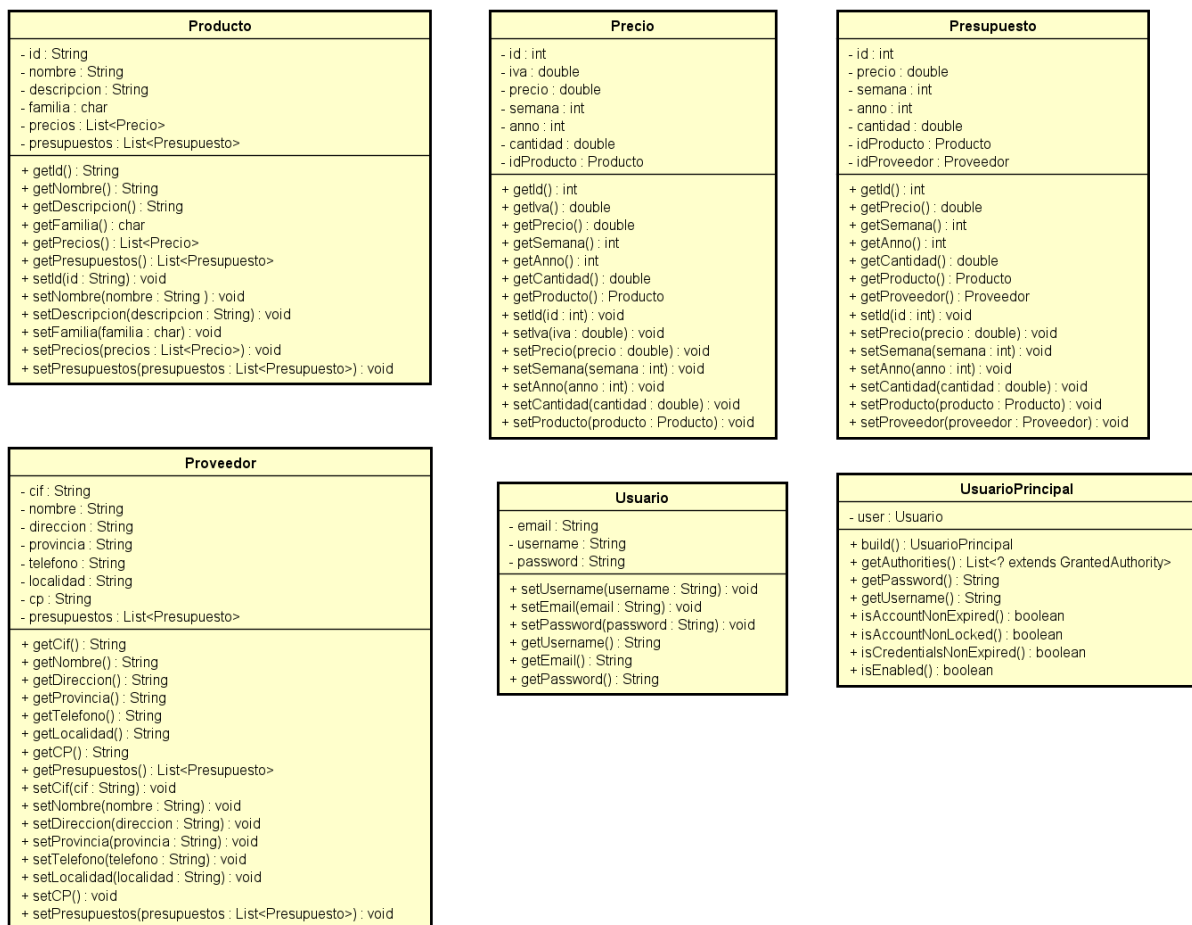


Figura 4.10: Modelo detallado de las clases de dominio

4.4. Modelo de datos

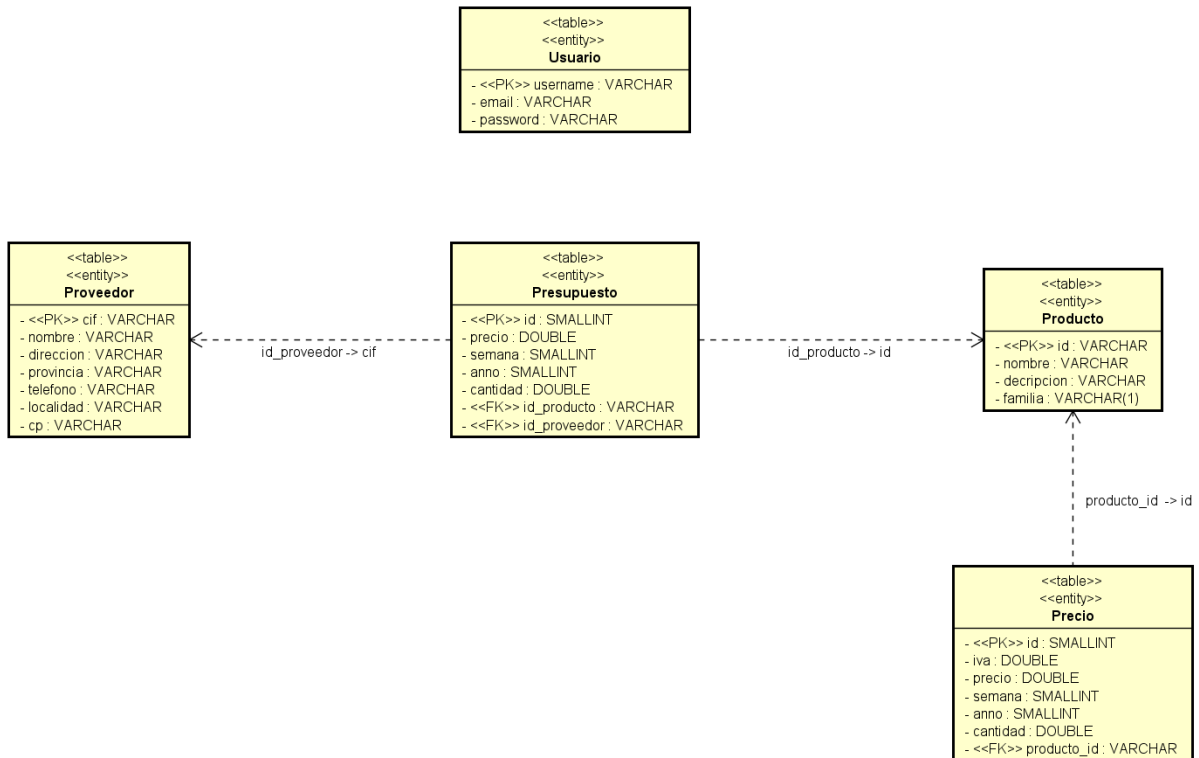


Figura 4.11: Modelo de datos

4.4.1. Diccionario de datos

A continuación se encuentran unas explicaciones sobre cada entidad de la base de datos y sus atributos.

Usuario: representa a los usuarios de la aplicación. Sus atributos son:

- **username** el nombre de usuario que le identificará de manera única. Tipo: VARCHAR.
- **email:** el email elegido por el usuario en el registro. Tipo: VARCHAR.
- **password:** la contraseña elegida por el usuario en el registro. Tipo: VARCHAR.

Proveedor: representa a la serie de proveedores de producto con los que trabaja el usuario. Sus atributos son:

- **cif:** el número de identificación fiscal que lo identificará de manera única. Tipo: VARCHAR.
- **nombre:** el nombre de empresa que tiene dicho proveedor. Tipo: VARCHAR.
- **direccion:** la calle y número de la empresa proveedora. Tipo: VARCHAR.
- **provincia:** provincia donde se encuentra la empresa proveedora. Tipo: VARCHAR.
- **telefono:** el número de teléfono con prefijo de la empresa proveedora. Tipo: VARCHAR.
- **localidad:** localidad donde se encuentra la empresa proveedora. Tipo: VARCHAR.

- **cp:** código postal de la localidad donde se encuentra la empresa proveedora. Tipo: VARCHAR.

Presupuesto: representa a la serie de precios recibidos por el usuario de parte del proveedor. Sus atributos son:

- **id:** el número de que lo identificará de manera única. Tipo: SMALLINT.
- **precio:** el importe que tiene el producto. Tipo: DOUBLE.
- **semana:** semana del año a la que corresponde el presupuesto. Tipo: SMALLINT.
- **anno:** año al que corresponde dicho presupuesto. Tipo: SMALLINT.
- **cantidad:** cantidad del producto suministrada por el proveedor. Tipo: DOUBLE.
- **idProducto:** identificador del producto sobre el que se recibe el presupuesto. Tipo: VARCHAR.
- **idProveedor:** identificador del proveedor que suministra el presupuesto. Tipo: VARCHAR.

Producto: representa a la serie de producto con los que comercializa el usuario. Sus atributos son:

- **id:** el identificador que tiene asignado en la empresa de manera única. Tipo: VARCHAR.
- **nombre:** el nombre que tiene el producto. Tipo: VARCHAR.
- **descripcion:** una pequeña aclaración sobre el producto. Tipo: VARCHAR.
- **familia:** familia animal a la que pertenece el producto. Tipo: VARCHAR de tamaño 1 debido a que es una única letra.

Precio: representa a la serie de precios que tiene asignados el usuario a los productos que tiene en inventario. Sus atributos son:

- **id:** el número de que lo identificará de manera única. Tipo: SMALLINT.
- **iva:** el impuesto sobre el valor añadido asignado que tiene dicho producto. Tipo: DOUBLE
- **precio:** el importe que tiene el producto. Tipo: DOUBLE.
- **semana:** semana del año a la que corresponde el precio. Tipo: SMALLINT.
- **anno:** año al que corresponde dicho precio. Tipo: SMALLINT.
- **cantidad:** cantidad del producto ofertada. Tipo: DOUBLE.
- **idProducto:** identificador del producto sobre el que se recibe el presupuesto. Tipo: VARCHAR.

4.5. Privacidad desde el diseño y por defecto

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

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

Las características fundamentales de este principio extraídas de análisis como el de Grupo Ático 34 [1] y en base a la información recogida en la guía de protección de datos publicada por la AEPD en 2020 [11] son:

- **Preventivo y pro-activo:** las acciones llevadas a cabo no deben orientarse en la mitigación de los riesgos, deben centrarse primordialmente en la reducción o eliminación.
- **Privacidad de forma predeterminada:** el usuario no debe solicitar que se tenga en cuenta su privacidad.
- **Privacidad como parte del diseño:** debe considerarse un elemento crucial en cualquier desarrollo que se lleve a cabo.
- **Funcionalidad completa:** la privacidad no debe interferir en la funcionalidad de la aplicación.
- **Seguridad de extremo a extremo:** se debe garantizar la seguridad e integridad de los datos en todo momento.
- **Visibilidad y transparencia:** se debe seguir un modelo transparente de tratamiento ofreciendo facilidades para su verificación y trazabilidad.
- **Respeto por la privacidad de los usuarios:** es necesario situar al usuario en el centro del diseño realizado.

A continuación, se describirán las medidas llevadas a cabo para garantizar estos principios en la aplicación.

Políticas de usuario

Antes de que un usuario se registre en la aplicación, es necesario que lea y acepte las condiciones de uso. Estas condiciones detallan cómo se manejarán los datos personales recopilados y las acciones que los usuarios pueden tomar para ejercer sus derechos de privacidad, como el "derecho al olvido". También se proporcionan los métodos de contacto con los administradores de la aplicación para posibles consultas o reclamaciones sobre estas condiciones o su aplicación.

Además, estas políticas establecen los términos que los usuarios deben cumplir para garantizar la seguridad de sus datos al utilizar la aplicación.

Tratamiento de los datos introducidos por el usuario

e ha buscado tener el número de datos mínimo del usuario en el sistema para así asumir la menor cantidad de riesgos posibles. Por lo tanto, solo se almacenan los datos identificativos [2], como correo y contraseña.

Además, dichos datos son tratados al darse de alta e iniciar sesión mediante la capa de seguridad de Spring, en conjunto con JWT para garantizar la autenticación y autorización

adecuadas. Este enfoque asegura que solo los usuarios autorizados puedan acceder a la aplicación y que sus datos estén protegidos durante todo el proceso de registro y sesión.

También se tiene en cuenta dicha privacidad en los datos de los archivos, debido a que pueden ser sensibles, solo se envían las rutas de los mismos y no la información que contienen.

4.5.1. Spring Security con JWT

Para la parte de seguridad de la aplicación se ha empleado Spring Security, de esta forma a través de roles se puede restringir el acceso a ciertas partes de la aplicación. Solo se emplea el rol de usuario en esta aplicación. Esto restringe el acceso de las URLs a las peticiones que contenga la palabra USER.

Esto se consigue a través del archivo de configuración en el que se pone la anotación `@EnableWebSecurity` para activar la seguridad web y utilizando el elemento `http` para definir la autenticación dentro del método `securityFilterChain()`.

Para que se pueda llevar a cabo es necesario tener implementada la autenticación de usuarios conteniendo los roles de cada usuario, para ello se ha implementado la interfaz `UserDetails` en la clase `UsuarioPrincipal` y también la interfaz `UserDetailsService` como servicio en nuestra aplicación.

Por otro lado, en los usuarios se ha encriptado la contraseña para que no se guarde en texto plano en la base de datos y aunque se tuviera acceso a ella no se pueda conseguir acceso a las cuentas de los usuarios, en el caso de esa contraseña haya sido utilizada para otras cuentas del usuario también seguirían seguras esas cuentas. Para el cifrado de las contraseñas se ha definido un método para crear un codificador de contraseñas en una aplicación Spring.

Para crear el codificador se crean una instancia del codificador `BCryptPasswordEncoder` para el algoritmo "bcrypt", que es una función de hashing de contraseñas que agrega una sal para evitar ataques con tablas arcoíris y es adaptable, lo que significa que puede aumentar el número de iteraciones para resistir ataques de fuerza bruta.

Para fortalecer la seguridad y mejorar la experiencia de autenticación de los usuarios, la aplicación utiliza JSON Web Tokens (JWT) tanto en el proceso de registro como en el inicio de sesión. JWT es un estándar abierto que define un formato compacto y autónomo para la transferencia segura de información entre partes como un objeto JSON.

Cuando un usuario inicia sesión con éxito, se genera un token JWT que contiene información de autenticación y autorización relevante, como el ID del usuario y sus roles. Este token se firma digitalmente con una clave secreta y se envía al cliente. El cliente almacena el token y lo envía junto con cada solicitud subsiguiente a la aplicación. Dicho token tiene validez de un día.

Al recibir una solicitud con un token JWT adjunto, el servidor verifica la autenticidad y validez del token mediante la verificación de la firma y la comprobación de la expiración del token. Si la verificación es exitosa, el servidor confía en la información contenida en el token para autorizar la solicitud del usuario.

4.6. Diseño de la interfaz de usuario

4.6.1. Usabilidad

La usabilidad es un aspecto fundamental en el diseño y desarrollo de aplicaciones, ya que determina lo fácil y agradable que es para los usuarios interactuar con el sistema. Una aplicación con alta usabilidad puede llegar a mejorar la satisfacción del usuario. Entre los atributos de usabilidad sobre los que se ha hecho incidencia en esta aplicación se encuentran la facilidad para el recuerdo, el tratamiento de errores y la eficiencia.

- **Facilidad para el recuerdo:** la aplicación ha sido diseñada para que los usuarios puedan recordar fácilmente cómo usarla, incluso después de un período de inactividad. Esto se

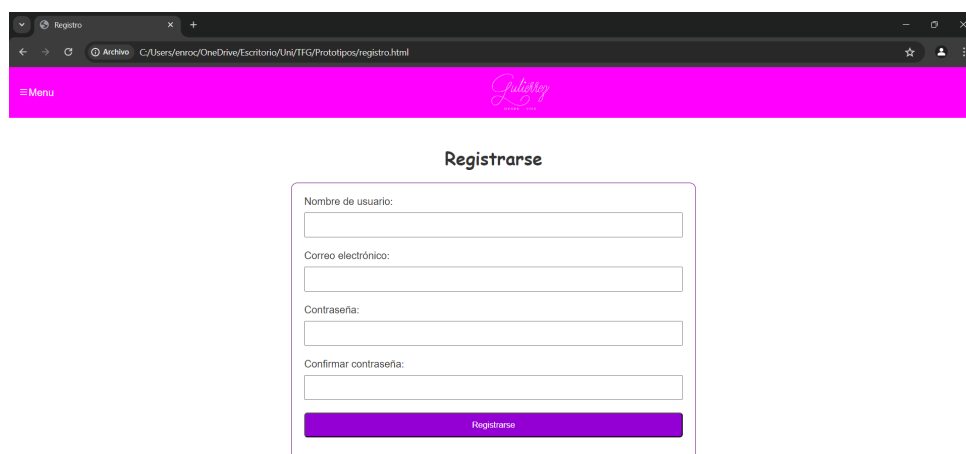
logra mediante una interfaz intuitiva y consistente, donde los iconos, menús y comandos son claros y predecibles.

- **Tratamiento de los errores:** se han integrado mensajes de error claros y constructivos que informan al usuario de lo que salió mal. Adicionalmente, se han desarrollado mecanismos de prevención de errores, como validaciones en tiempo real y confirmaciones antes de realizar acciones críticas, minimizando así la posibilidad de cometer errores.
- **Eficiencia:** se refleja en la rapidez y precisión con la que los usuarios pueden completar sus tareas. Esto se ha logrado optimizando los procesos internos y asegurando que las funciones más utilizadas sean fácilmente accesibles. La interfaz ha sido diseñada para minimizar el número de clics y entradas necesarias, permitiendo a los usuarios realizar sus tareas de manera ágil y sin complicaciones.

Para aplicar estos atributos de usabilidad se han seguido las guías de diseño de cada uno de ellos y se han aplicado en las etapas iniciales del TFG. Para ello, se diseñaron unos prototipos de la interfaz.

4.6.2. Prototipos de la interfaz

En las siguientes etapas de refinamiento, se fueron modificando para mejorar la claridad, la cohesión del diseño y la eficiencia.



El prototipo muestra una interfaz web con un encabezado azul que contiene un menú y el logo 'Juliana'. El título principal es 'Registrarse'. El formulario de registro incluye los siguientes campos:

- Nombre de usuario:
- Correo electrónico:
- Contraseña:
- Confirmar contraseña:

Debajo de los campos hay un botón azul con el texto 'Registrarse'.

Figura 4.12: Prototipo pantalla de registro

Figura 4.13: Prototipo pantalla de login

Figura 4.14: Prototipo pantalla de Añadir proveedor

CIF	Nombre	Dirección	Provincia	Teléfono
12345678A	Proveedor 1	Dirección 1	Madrid	123456789
87654321B	Proveedor 2	Dirección 2	Barcelona	987654321

Figura 4.15: Prototipo pantalla de listado de proveedores

4.7. Despliegue de la aplicación

En la figura siguiente se muestra el diagrama de despliegue de la aplicación. Como se puede apreciar y ya se ha comentado previamente, la aplicación está pensada para ser desplegada y empleada en local.

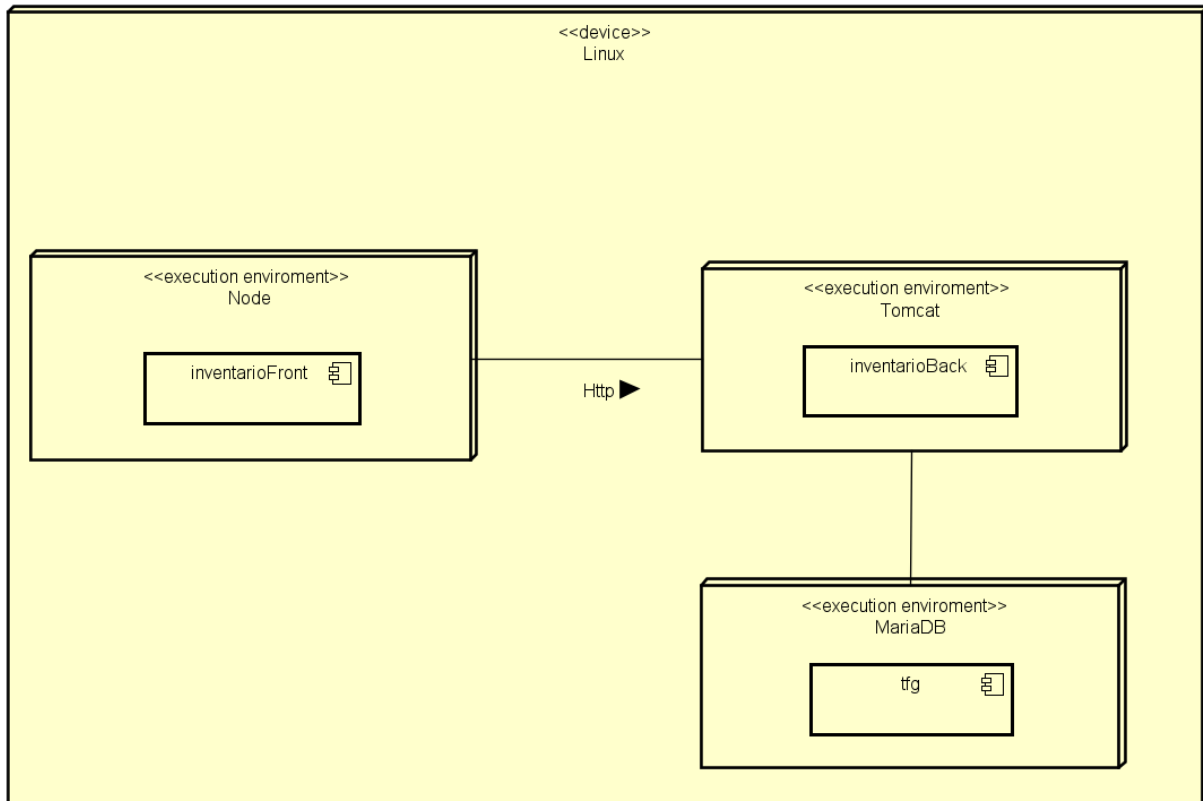


Figura 4.16: Diagrama de despliegue de la aplicación

4.8. Decisiones de Diseño

Las versiones de Java, Angular, Node y NPM son LTS y son más estables tanto en seguridad como en las demás funcionalidades ya que permitirán que el sistema esté en funcionamiento durante más tiempo.

Se ha tomado una decisión más de diseño relacionada con la importación de los archivos relacionados con los presupuestos recibidos por el usuario, todos los archivos deberán ser subidos en formato .xlsx y con un formato muy específico, que será explicado con detenimiento en el manual de usuario.

Capítulo 5

Implementación

Este capítulo aborda la fase de implementación del proyecto software, explicando las tecnologías utilizadas y con que herramientas. A mayores también se comenta la implementación del modelo predictivo.

5.1. Tecnologías utilizadas

5.1.1. Generación de diagramas

Astah

Es una de las herramientas de modelado de productos software empleadas durante la carrera. Con ella se han generado los artefactos de las diferentes etapas del desarrollo. Como ya se ha mencionado, es una herramienta muy conocida y no se va a profundizar en su uso. En cualquier caso y para cualquier duda, se puede consultar su página web [astah] donde podemos encontrar documentación y ejemplos de uso.

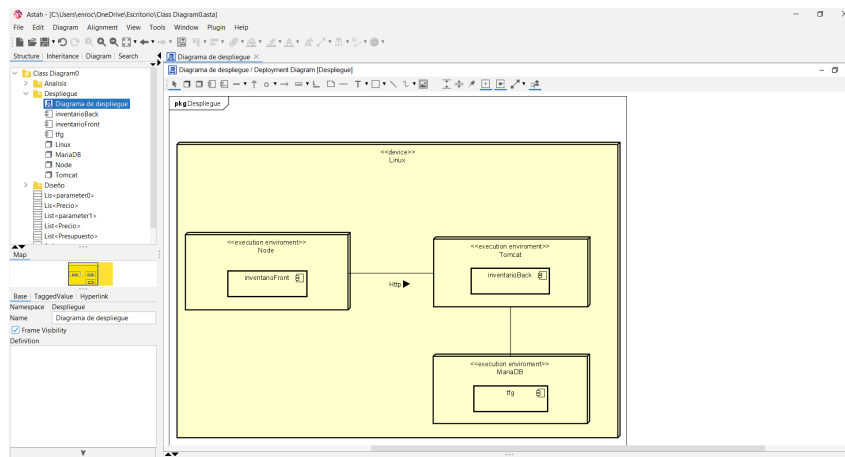


Figura 5.1: Interfaz de Astah

5.1.1.1. Creación de la memoria

Esta sección recogerá los programas utilizados para realizar este documento de memoria.

LaTeX

Es un sistema de composición de alta calidad; incluye características diseñadas para la producción de documentación técnica y científica. LaTeX es el estándar de facto para la comunicación y publicación de documentos científicos.[16]

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

Overleaf

Es una plataforma en línea que ofrece un entorno de edición colaborativa para documentos LaTeX. Permite a los colaboradores trabajar simultáneamente en el mismo documento y proporciona herramientas para la compilación y visualización en tiempo real. Además, al estar basado en la nube, elimina la necesidad de instalar y mantener software LaTeX.

5.1.1.2. Entorno de desarrollo: IntelliJ IDE

Es un entorno de desarrollo para Java. Esta herramienta desarrollada por JetBrains contiene todo lo necesario para la realización del desarrollo, pruebas y despliegue de la aplicación.

Se ha optado por este entorno debido a la simplicidad para la configuración del mismo, el empleo de git, la facilidad para aplicar Maven y la posibilidad de desarrollar HTML y TypeScript en él, gracias al empleo de sus extensiones.

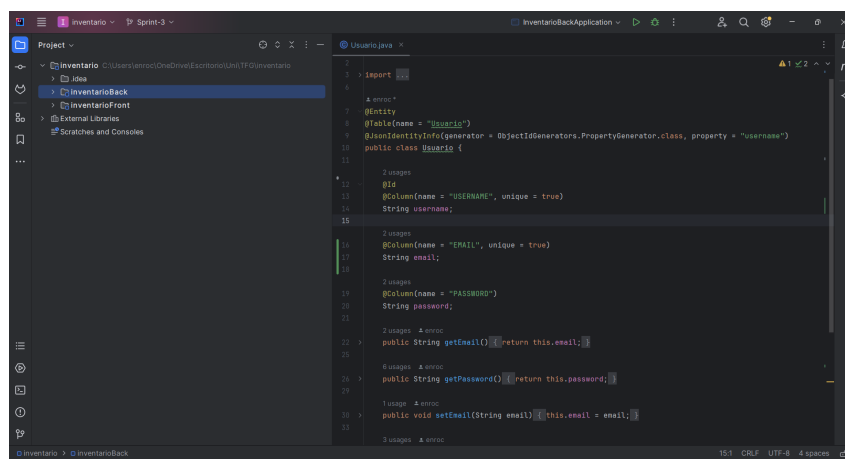


Figura 5.2: Interfaz de IntelliJ

5.1.2. Spring Framework

Es uno de los entornos más populares para creación de grandes proyectos Java, ya que facilita en gran medida su realización, no siempre fácil cuando se emplea la versión empresarial (JEE). Es un entorno integrado que permite configurar, crear y desplegar aplicaciones Java de distintos tipos.

Spring tiene los siguientes módulos (solo se describirán los más importantes para este proyecto):

- **Core container:** Es la parte fundamental del framework. Provee las herramientas fundamentales para la ejecución de una aplicación Java. Será donde reside la parte encargada de resolver las anotaciones y la inyección de dependencias, dependency injection (inclusión automática y sencilla de dependencias, sin necesidad de conocer su ubicación).
- **Web:** Permite crear controladores Web, tanto de vistas MVC como aplicaciones REST.

- **Acceso a datos:** Contiene todas las partes que facilitan el acceso a los datos de manera simple. Incluye abstracciones sobre JDBC (haciendo las operaciones transparentes al gestor de bases de datos), ORMs como JPA (Java Persistence API), etc.
- **Programación orientada a Aspectos (AOP).**
- **Instrumentación.**
- **Seguridad.** Spring Security es un marco de autenticación y control de acceso potente y altamente personalizable, que se centra en proporcionar tanto autenticación como autorización a aplicaciones Java.[14]

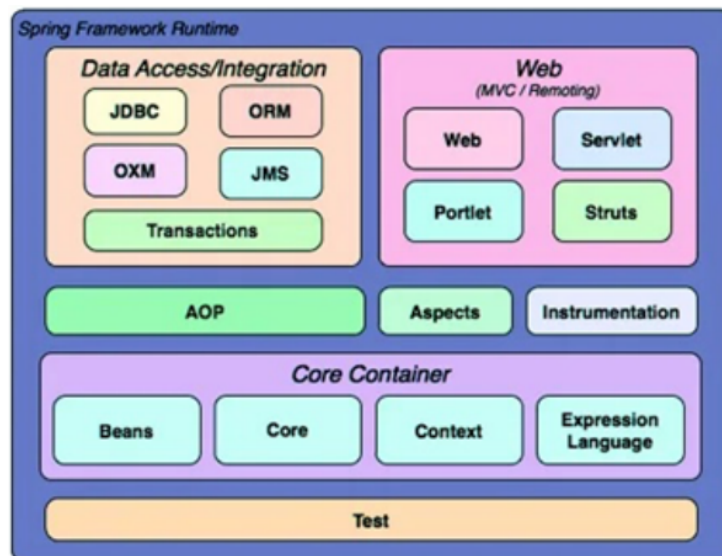


Figura 5.3: Módulos Spring Framework

5.1.2.1. SpringBoot

Para la inicialización del proyecto que implementará el Spring Framework, se empleará Springboot.

Ahora se comentarán las ventajas de usar Spring Boot para simplificar el proceso de creación de una aplicación con Spring:

- **Starters:** Proporciona una herramienta que permite inicializar la aplicación Spring. Permite configurar la aplicación, así como añadir de manera sencilla lo que necesitamos, en este caso el crear un controlador REST, acceder a una base de datos usando JDBC. Una vez añadidos, el starter proporciona todas las dependencias que sean necesarias, tanto de Spring como de terceros.
- **Contenedor de aplicaciones integrado:** Lo que permite ejecutar la aplicación compiladas en un archivo .jar. Esto permite desplegar y probar la aplicación de una manera muy sencilla.

5.1.3. Angular Framework

Angular es una plataforma que permite el desarrollo de aplicaciones web dinámicas en la parte del cliente usando HTML y Javascript, para ello sigue el enfoque Single-Page Application

(SPA), en el cuál el servidor sirve la primera página y a partir de ahí todo el procesamiento dinámico (creación de nuevas vistas) se realiza en el cliente.

Está basado en el uso de componentes. Esto permite encapsular mejor la funcionalidad, facilitando la construcción y el mantenimiento de las aplicaciones. El componente, más que una “pieza reutilizable”, son las piezas básicas mediante las que se construye la aplicación y que se pueden comunicar entre sí. El objetivo de cada componente es encapsular una determinada funcionalidad de la aplicación, incluso, se podría decir que, en general, encapsulan todo lo relacionado con una determinada vista; por lo cuál se podría llegar a pensar en ser reutilizados, pero no es el objetivo principal.

Por todo lo comentado anteriormente, se ha decidido que sea la tecnología empleada en el desarrollo de la aplicación, se empleará la versión Angular-Cli ya que facilita la creación de la aplicación Angular y el despliegue sobre Node.

5.1.4. Node.Js

Es un entorno multiplataforma, de código abierto creado para la capa de servidor, pero no también permite la ejecución de código Javascript, inicialmente creado para ser ejecutado en un navegador, sin necesidad de éste. En nuestro caso, se usará como servidor para la parte de desarrollo y pruebas de la aplicación, pero también como plataforma que proporciona las bibliotecas básicas para el desarrollo en Angular.

5.1.5. Tecnología de pruebas

En esta sección se analizará la herramienta empleada para probar el software.

5.1.5.1. Postman

Esta tecnología, que comenzó como una extensión del navegador Chrome, y que ahora cuenta ya con su propia aplicación de escritorio, ayudará en el desarrollo del back ofreciendo un entorno personalizable de trabajo en el que se pueden realizar todo tipo de tareas sobre las peticiones. Permite tanto automatizarlas y encadenarlas como analizar las peticiones y respuestas recibidas.

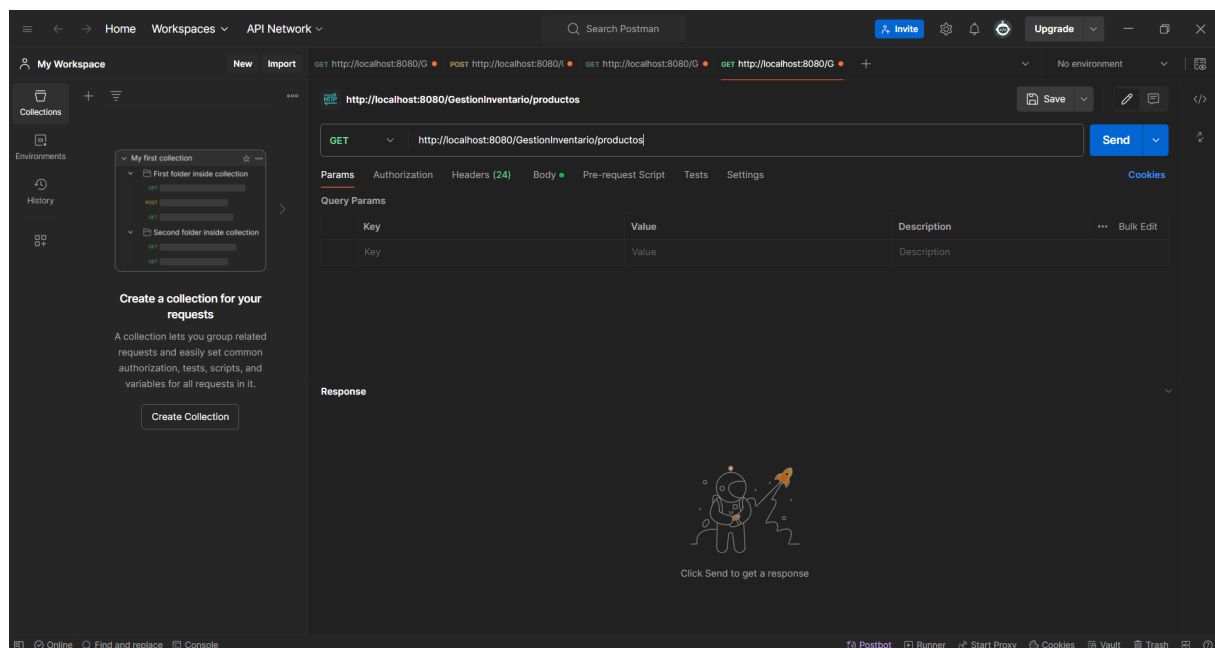


Figura 5.4: Interfaz de Postman

Para este desarrollo, el principal uso que se le ha dado es probar los distintos *endpoints* de nuestro backend y comprobar que los datos se almacenan y se recuperan de forma correcta.

5.1.6. Control de versiones

5.1.6.1. Git y Gitlab

Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software, creado por Linus Torvalds en 2005 para gestionar el desarrollo del kernel de Linux. Su capacidad para rastrear cambios en el código y coordinar el trabajo entre varios desarrolladores destaca, gracias a su modelo de ramificación flexible que permite el trabajo en paralelo sin interferencias. Almacena los cambios en un repositorio, facilitando la colaboración y gestión de versiones, siendo además de código abierto y con una comunidad activa que contribuye al desarrollo y brinda soporte.[6][7]

Se emplea con Gitlab debido a que facilita la gestión de los repositorios Git mediante el empleo de una interfaz muy intuitiva.

Se emplearán ambas en este proyecto para mantener un seguimiento de las versiones.

5.1.7. Sistema Gestor de Base de Datos

5.1.7.1. DBeaver

DBeaver es una herramienta de gestión de bases de datos, desarrollada sobre JAVA, potente que ofrece una interfaz gráfica de usuario (GUI) para interactuar con diversas bases de datos. Es de código abierto y es compatible con una amplia variedad de sistemas de gestión de bases de datos, incluyendo MySQL, PostgreSQL, SQLite, Oracle, SQL Server y muchos más.

Ha sido elegida porque facilita las tareas de administración y desarrollo de bases de datos al proporcionar funcionalidades avanzadas como la ejecución de consultas SQL, la visualización y edición de datos, la creación y modificación de esquemas, y la generación de informes. Además, posee una interfaz muy intuitiva.

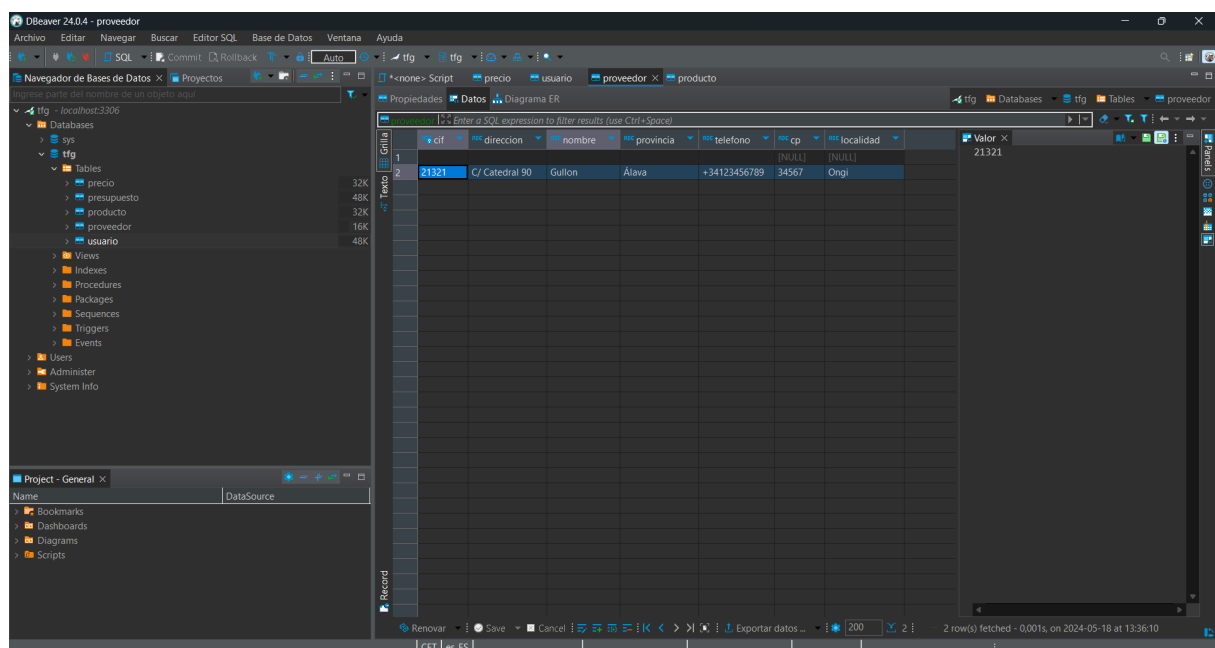


Figura 5.5: Interfaz de DBeaver

5.2. Modelo predictivo de los datos

El modelo ARIMA estacional (Autoregressive Integrated Moving Average) es una herramienta poderosa en el análisis y la predicción de series temporales que exhiben patrones estacionales. Este modelo se basa en la combinación de componentes de autorregresión (AR), integración (I) y promedio móvil (MA) para modelar la estructura temporal de los datos.

Cuando los datos muestran variaciones regulares o cíclicas a lo largo del tiempo, conocidas como patrones estacionales, el ARIMA estacional se convierte en una extensión crucial del modelo ARIMA básico. Por ejemplo, en contextos como ventas minoristas, donde las ventas pueden aumentar durante períodos de vacaciones o festivos, es esencial capturar y predecir estos patrones estacionales para una planificación efectiva.

5.2.1. Proceso de Implementación

Selección de Parámetros Óptimos

La función `seleccionarMejorModelo` realiza una búsqueda exhaustiva de combinaciones de parámetros del modelo ARIMA. Esto incluye ajustar los parámetros p (orden del modelo autoregresivo), d (grado de diferenciación), q (orden del modelo de media móvil), P (orden del componente autoregresivo estacional), D (grado de diferenciación estacional), Q (orden del componente de media móvil estacional) y s (periodo de la estacionalidad). El objetivo es minimizar el Criterio de Información Bayesiano (BIC), que evalúa la calidad del modelo balanceando el ajuste y la complejidad.

Entrenamiento del Modelo

Una vez seleccionados los mejores parámetros, el modelo se entrena utilizando la función `entrenarYPredecir`. Aquí, se utiliza regresión lineal múltiple para ajustar el modelo ARIMA estacional a los datos de la serie temporal. La ecuación de predicción resultante combina términos autoregresivos, de media móvil y sus equivalentes estacionales, permitiendo generar predicciones precisas para el siguiente período de tiempo. La ecuación predictiva empleada en el modelo es:

$$\text{prediccion} = \beta_0 + \sum_{i=1}^p \beta_i \cdot X_{AR,i} + \sum_{j=1}^q \theta_j \cdot X_{MA,j} + \sum_{k=1}^P \phi_k \cdot X_{AR_estacional,k} + \sum_{l=1}^Q \theta_l \cdot X_{MA_estacional,l}$$

Figura 5.6: Ecuación predictiva

Dónde: β_0 es el término constante, $X_{AR,i}$ son los valores autoregresivos, $X_{MA,j}$ son los valores de media móvil, $X_{AR_estacional,k}$ son los valores autoregresivos estacionales, $X_{MA_estacional,l}$ son los valores de media móvil estacionales.

Evaluación del Modelo con BIC

El cálculo del BIC, realizado por la función `calcularBIC`, es crucial para la validación del modelo. Este criterio penaliza modelos más complejos y favorece aquellos que mejor se ajustan a los datos observados. La ecuación del BIC utiliza la Suma de los Cuadrados de los Residuos (RSS) y el número de parámetros en el modelo (k), que incluye términos autoregresivos, de media móvil y estacionales, además de un término constante.

$$\text{BIC} = n \cdot \ln \left(\frac{\text{RSS}}{n} \right) + k \cdot \ln(n)$$

Figura 5.7: Ecuación BIC

Aplicaciones Prácticas

Una vez entrenado y validado, el modelo ARIMA estacional proporciona predicciones, siguiendo la ecuación que pueden utilizarse para tomar decisiones informadas. Por ejemplo, en gestión de inventarios, estas predicciones ayudan a planificar y ajustar los niveles de stock en función de las tendencias históricas identificadas. Esto no solo optimiza los recursos, sino que también permite anticiparse a cambios estacionales en la demanda.

Mejoras y Consideraciones

También se han llevado a cabo mejoras significativas, como la interpolación de datos mediante la implementación de técnicas como `interpolarValor`. Esta mejora permite manejar datos faltantes o índices fuera del rango, asegurando así que el modelo pueda operar con conjuntos de datos incompletos o con valores ausentes.

Además, se ha fortalecido la robustez del modelo con el uso de métodos de regresión como los Mínimos Cuadrados Ordinarios (OLS), implementado en el método `ajustarOLS`, y Ridge implementado en el método `ajustarRidge`.

Mínimos Cuadrados Ordinarios (OLS): Este método proporciona estimaciones no sesgadas de los coeficientes del modelo de regresión lineal al minimizar la suma de los cuadrados de los residuos. La fórmula para calcular los coeficientes β en OLS es:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

donde X es la matriz de diseño de tamaño $n \times (p + 1)$, y es el vector de respuestas observadas, y $(X^T X)^{-1}$ es la inversa de la matriz de productos cruzados de X .

Regularización Ridge: Para abordar la multicolinealidad en los datos, Ridge introduce una penalización controlada por el parámetro λ , que modifica la matriz de productos cruzados $X^T X$ añadiendo λI :

$$\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

donde I es la matriz identidad. Esta técnica ayuda a estabilizar los coeficientes del modelo y mejorar su capacidad de generalización a nuevos datos.

Gracias a estas mejoras se asegura una implementación robusta de modelos de regresión lineal, capaz de manejar desafíos como la multicolinealidad y la presencia de datos faltantes o ruidosos.

Capítulo 6

Pruebas

6.1. Ejecución de las pruebas

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

- **Pruebas automatizadas:** programadas empleando software específico para ello. Son pruebas que necesitan un gran dominio de las herramientas empleadas debido a su alta complejidad pero permiten ejecutar una gran cantidad de pruebas de manera rápida.
- **Pruebas manuales:** realizadas por los propios desarrolladores analizando las respuestas obtenidas.

Las pruebas de software se dividen en dos categorías principales: funcionales y no funcionales. Estas categorías reflejan los requisitos planteados durante la fase de análisis y desarrollo.

6.1.1. Pruebas Funcionales

Estas pruebas evalúan si el software cumple con las funcionalidades requeridas. Se centran en verificar que las características específicas funcionen correctamente. Algunos tipos de pruebas funcionales son:

- **Pruebas Unitarias:** Evalúan componentes individuales del software de forma aislada. Detectan fallos en su funcionamiento que podrían afectar la integración con otros elementos del sistema.
- **Pruebas de Integración:** Evalúan cómo interactúan diferentes partes del sistema cuando se combinan. Se buscan posibles fallos en la comunicación y el traspaso de datos.
- **Pruebas de Aceptación:** Simulan escenarios del mundo real para evaluar la satisfacción del cliente con los resultados ofrecidos por el sistema.

6.1.2. Pruebas No Funcionales

Estas pruebas se centran en aspectos que no están directamente relacionados con las funcionalidades, como la seguridad, escalabilidad y rendimiento:

- **Pruebas de Rendimiento:** Evalúan la velocidad, capacidad y estabilidad del software bajo diferentes condiciones.
- **Pruebas de Seguridad:** Verifican que el sistema esté protegido contra amenazas y vulnerabilidades.
- **Pruebas de Usabilidad:** Evalúan la experiencia del usuario y la facilidad de uso.

6.2. Pruebas de aceptación por cada caso de uso

Durante el desarrollo del TFG se han ido realizando pruebas unitarias de caja negra y pruebas de integración. En las pruebas unitarias de caja negra se han ido detectando errores por cada método donde había una discordancia entre el resultado esperado y el resultado obtenido. Se ha aplicado lo que se ha enseñado en la asignatura de programación orientada a Objetos.

Al finalizar cada caso de uso de forma completa, se han ido realizado pruebas de aceptación por cada uso. A continuación, se describen una parte de estas pruebas de aceptación por cada caso de uso:

CP1	Registro en el sistema
Entrada	Con la sesión cerrada y en la pantalla principal (Inicio de sesión) cambiamos a la pantalla de Registro. Añadimos el nombre “admin1234”, el correo alguien@example.com y “admin1234” como contraseña, aceptamos los términos y condiciones y se envía el formulario.
Resultado Esperado	Se informará por pantalla al usuario de que su registro ha sido exitoso y se le redirigirá a la pantalla de Inicio de sesión.

Tabla 6.1: Caso de Prueba 1

CP2	Registro sin aceptar los términos y condiciones del sistema
Entrada	Con la sesión cerrada y en la pantalla principal (Inicio de sesión) cambiamos a la pantalla de Registro. Añadimos el nombre “admin1234”, el correo alguien@example.com y “admin1234” como contraseña y no se aceptan los términos y condiciones.
Resultado Esperado	No se activará el botón de envío de formulario hasta que no se acepten los términos y condiciones.

Tabla 6.2: Caso de Prueba 2

CP3	Inicio de sesión correcto
Entrada	Con la sesión cerrada y en la pantalla principal (Inicio de sesión). Añadimos el nombre de usuario “admin1234” y “admin1234” como contraseña y se envía el formulario.
Resultado Esperado	El sistema permite acceder a la funcionalidad total de la aplicación al usuario.

Tabla 6.3: Caso de Prueba 3

CP4	Inicio de sesión de usuario no registrado
Entrada	Con la sesión cerrada y en la pantalla principal (Inicio de sesión). Añadimos un nombre de usuario de una cuenta no registrada y cualquier valor como contraseña y se envía el formulario.
Resultado Esperado	El sistema informará al usuario del error y le enviará a la pantalla de registro.

Tabla 6.4: Caso de Prueba 4

CP5	Mostrar los proveedores almacenados en el sistema
Entrada	En la pantalla del menú principal se seleccionará la opción de proveedores y se le redirige a la pantalla de proveedores.
Resultado Esperado	El usuario podrá ver los proveedores almacenados e interactuar con ellos.

Tabla 6.5: Caso de Prueba 5

CP6	Añadir proveedor
Entrada	En la pantalla de proveedores se pulsa el botón con el texto Añadir, se abre un desplegable que solicita los datos del proveedor, se introducen el CIF, el nombre, la dirección, se selecciona la provincia entre las de España, la localidad, el CP, el prefijo y el número de teléfono del proveedor y se pulsa el botón con el texto Guardar que envía el formulario.
Resultado Esperado	El usuario podrá ver el nuevo proveedor en el listado con los demás proveedores almacenados e interactuar con ellos.

Tabla 6.6: Caso de Prueba 6

CP7	Cancelar añadir proveedor
Entrada	En la pantalla de proveedores se pulsa el botón con el texto Añadir, se abre un desplegable que solicita los datos del proveedor, se introduce alguno de los datos o ninguno y se pulsa el botón con el texto Cancelar.
Resultado Esperado	El desplegable se cerrará y se volverá a la pantalla con la lista de proveedores y el nuevo proveedor no aparecerá con los demás proveedores almacenados.

Tabla 6.7: Caso de Prueba 7

CP8	Editar proveedor
Entrada	En la pantalla de proveedores se pulsa el botón con el icono del lápiz, se abre un desplegable que muestra los datos del proveedor y permite cambiarlos, se cambia alguno de los datos o ninguno y se pulsa el botón con el texto Guardar que envía el formulario.
Resultado Esperado	El usuario podrá ver los datos del proveedor cambiados en el listado de proveedores.

Tabla 6.8: Caso de Prueba 8

CP9	Ver datos proveedor
Entrada	En la pantalla de proveedores se pulsa el botón con el icono del ojo, se abre un desplegable que muestra los datos del proveedor, se pulsa el botón con la X.
Resultado Esperado	El desplegable se cerrará y se volverá a la pantalla con la lista de proveedores.

Tabla 6.9: Caso de Prueba 9

CP10	Eliminar proveedor
Entrada	En la pantalla de proveedores se pulsa el botón con el icono de la basura que envía la solicitud.
Resultado Esperado	El proveedor será eliminado de la base de datos y por tanto del listado.

Tabla 6.10: Caso de Prueba 10

CP11	Mostrar los productos almacenados en el sistema
Entrada	En la pantalla del menú principal se seleccionará la opción de productos y se le redirige a la pantalla de productos.
Resultado Esperado	El usuario podrá ver los productos almacenados e interactuar con ellos.

Tabla 6.11: Caso de Prueba 11

CP12	Añadir producto
Entrada	En la pantalla de productos se pulsa el botón con el texto Añadir, se abre un desplegable que solicita los datos del producto, se introducen el id, el nombre, la descripción, la familia y se envía el formulario.
Resultado Esperado	El usuario podrá ver el nuevo producto en el listado con los demás productos almacenados e interactuar con ellos.

Tabla 6.12: Caso de Prueba 12

CP13	Cancelar añadir producto
Entrada	En la pantalla de productos se pulsa el botón con el texto Añadir, se abre un desplegable que solicita los datos del producto, se introduce alguno de los datos o ninguno y se pulsa el botón con el texto Cancelar.
Resultado Esperado	El desplegable se cerrará y se volverá a la pantalla con la lista de productos y el nuevo producto no aparecerá con los demás productos almacenados.

Tabla 6.13: Caso de Prueba 13

CP14	Editar producto
Entrada	En la pantalla de productos se pulsa el botón con el icono del lápiz, se abre un desplegable que muestra los datos del producto y permite cambiarlos, se cambia alguno de los datos o ninguno y se pulsa el botón con el texto Guardar que envía el formulario.
Resultado Esperado	El usuario podrá ver los datos del producto cambiados en el listado de productos.

Tabla 6.14: Caso de Prueba 14

CP15	Ver datos producto
Entrada	En la pantalla de productos se pulsa el botón con el icono del ojo, se abre un desplegable que muestra los datos del producto, se pulsa el botón con la X.
Resultado Esperado	El desplegable se cerrará y se volverá a la pantalla con la lista de productos.

Tabla 6.15: Caso de Prueba 15

CP16	Eliminar producto
Entrada	En la pantalla de productos se pulsa el botón con el icono de la basura que envía la solicitud.
Resultado Esperado	El producto será eliminado de la base de datos y por tanto del listado.

Tabla 6.16: Caso de Prueba 16

CP17	Subir datos
Entrada	En la pantalla de datos se pulsa el botón con el texto de Subir Datos que abre un desplegable que solicita los archivos a subir y al pulsar el botón Confirmar, se envían.
Resultado Esperado	El usuario será informado de que los datos se han procesado de manera correcta.

Tabla 6.17: Caso de Prueba 17

CP18	Añadir presupuesto
Entrada	En la pantalla de datos se pulsa el botón con el texto Introducir Datos, se abre un desplegable que solicita los datos del presupuesto, se introducen el id del Producto, el nombre del Proveedor, el precio, la cantidad, la fecha y se envía el formulario.
Resultado Esperado	El usuario será informado sobre el éxito de la operación.

Tabla 6.18: Caso de Prueba 18

CP19	Editar contraseña del perfil
Entrada	En la pantalla donde se muestran los datos del perfil, se pulsa el icono del lápiz situado al lado de la contraseña y esto lleva a la pantalla de cambio de contraseña, donde se introduce la nueva contraseña 2 veces y se envía.
Resultado Esperado	El usuario será informado sobre el éxito de la operación.

Tabla 6.19: Caso de Prueba 19

CP20	Editar email del perfil
Entrada	En la pantalla donde se muestran los datos del perfil, se pulsa el icono del lápiz situado al lado del email y esto lleva a la pantalla de cambio de email, donde se introduce el nuevo email 2 veces y se envía.
Resultado Esperado	El usuario será informado sobre el éxito de la operación.

Tabla 6.20: Caso de Prueba 20

CP21	Recuperar contraseña del perfil
Entrada	En la pantalla de inicio de sesión se pulsa el texto de no recordar la contraseña y esto lleva a la pantalla de recuperación de contraseña, donde se introduce la nueva contraseña 2 veces y se envía.
Resultado Esperado	El usuario será informado sobre el éxito de la operación y se le llevará a la pantalla de inicio de sesión.

Tabla 6.21: Caso de Prueba 21

CP22	Consultar tendencias de precios
Entrada	En la pantalla de datos se solicita un intervalo de tiempo entre el que se mostrarán los datos, se introducen 2 fechas válidas y se envía el formulario.
Resultado Esperado	El usuario será redirigido a una pantalla con una gráfica mostrando la tendencia de los precios en dicho intervalo.

Tabla 6.22: Caso de Prueba 22

CP23	Cerrar sesión
Entrada	En la pantalla donde se muestran los datos del perfil, se pulsa el icono de cierre de sesión o en cualquier pantalla en el botón del header.
Resultado Esperado	La información del usuario de la sesión será eliminada y el será redirigido a la pantalla de inicio de sesión.

Tabla 6.23: Caso de Prueba 23

6.3. Resultados de las pruebas

Debido a que en la mayoría de las pruebas el resultado obtenido ha sido el esperado, en esta sección se comentarán solo en las cuáles la salida no ha sido la esperada.

CP19	Editar contraseña del perfil
Salida obtenida	Se realiza correctamente el cambio de contraseña pero no se permite el inicio de sesión una vez realizado.
Resultado final	Corregido.
Detalles	Se ha codificado la contraseña previo a su actualización en la BBDD.

Tabla 6.24: Resultados de ejecución del Caso de Prueba 19

CP21	Recuperar contraseña del perfil
Salida obtenida	Se realiza correctamente la recuperación de contraseña pero no se permite el inicio de sesión una vez realizado.
Resultado Esperado	Corregido.
Detalles	Se ha codificado la contraseña previo a su actualización en la BBDD.

Tabla 6.25: Resultados de ejecución del Caso de Prueba 21

6.3.1. Pruebas de usabilidad

Para la evaluación de la usabilidad de esta aplicación se ha creado un guión con ligeras variaciones para que distintos usuarios que no participaron en el desarrollo de la aplicación prueben la aplicación y proporcionen su feedback. Se incidirá en evaluar la facilidad de recuerdo, tratamiento de errores y eficiencia.

Se llevarán a cabo pruebas de usabilidad en las que se pondrán a prueba todos los casos de uso y los posibles errores que se puedan dar. A continuación se detallarán los resultados obtenidos:

Se han realizado pruebas con un usuario de mediana edad que no es nativo digital y podría enfrentar dificultades adicionales al aprender a utilizar la aplicación. El tiempo empleado para completar la prueba fue de aproximadamente 20 minutos, logrando finalizarla sin encontrar ningún problema significativo. Se le proporcionó el manual de usuario y algunas indicaciones adicionales para facilitar la realización de la prueba.

Se han realizado pruebas con un cliente y otra persona de mediana edad, ambas nativas digitales. Ambos completaron la prueba en menos de 10 minutos sin encontrar ningún problema significativo. Consultaron el manual de usuario cuando fue necesario y encontraron suficiente ayuda para completar la prueba exitosamente.

Se ha realizado una prueba a un usuario joven, de 23 años, familiarizado con la tecnología pero sin conocimientos informáticos. Este usuario completó la prueba en un tiempo que osciló entre los 7 y 9 minutos y no se le tuvo que dar ningún apunte ni obtuvo ninguna alerta de error.

En conclusión, todos los usuarios elogiaron la facilidad de uso de la aplicación, destacando lo intuitiva, rápida y sencilla que resultó su utilización. También valoraron positivamente la utilidad de recibir notificaciones claras sobre fallos y errores que surgieron durante el uso.

Capítulo 7

Conclusiones

7.1. Objetivos conseguidos

- Se ha realizado una aplicación web de productos, precios y presupuestos de proveedores.
- Se ha conseguido la gestión del inventario diseñando una base de datos relacional optimizada para almacenar información detallada sobre productos y proveedores.
- Se ha logrado aplicar privacidad desde el diseño sobre los datos del usuario mediante Spring Security y JSON Web Tokens (JWT) para autenticar y autorizar usuarios.
- Se ha desarrollado el modelo predictivo de aprendizaje automático ARIMA estacional para analizar datos históricos de precios y generar predicciones sobre tendencias futuras.
- Se ha realizado una visualización gráfica de los precios de los productos y sus tendencias en cada una de las semanas dentro de un periodo de tiempo establecido.
- Se ha conseguido que la aplicación web sea fácil de utilizar analizando los atributos de usabilidad más adecuados y aplicando las guías de diseño de forma que se ha construido un interfaz intuitivo y sencillo.

7.2. Análisis personal

Este proyecto ha supuesto todo un reto personal y una gran posibilidad para aprender sobre nuevas tecnologías y nuevos temas sobre los que había oído hablar y que han cogido gran relevancia en los últimos años.

Como ya he comentado el proyecto ha supuesto un gran reto personal pero también tecnológico al emplear tecnologías totalmente nuevas para mí pero que me pueden ayudar a desarrollarme en el ámbito profesional, también a lidiar con los problemas debido a todos los que han ido surgiendo durante el desarrollo. En cuanto al ámbito personal, ha sido un reto debido a la compaginación de trabajo y el desarrollo de este proyecto, lo cual me ha ayudado a aprender a gestionar mejor el tiempo y organizarme. En líneas generales, estoy contento con el producto desarrollado.

Bibliografía

- [1] Grupo Ático 34. *Privacidad desde el diseño y por defecto ¿Qué es?* (Última consulta: 05/05/2024). 2022. URL: <https://protecciondatos-lopd.com/empresas/privacy-by-design/>.
- [2] Grupo Adaptalia. *Categorías de datos personales RGPD*. (Última consulta: 05/05/2024). 2021. URL: <https://www.grupoadaptalia.es/blog/categorias-de-datos-personales-rgpd/>.
- [3] Software Advice. *Optimus Price*. (Último Acceso: 18/02/2024). URL: <https://www.softwareadvice.ie/software/348385/optimus-price>.
- [4] Asana. *Metodologías Agile*. (Última consulta: 27/02/2024). 2022. URL: <https://asana.com/es/resources/agile-methodology>.
- [5] Atlassian. *SCRUM*. (Última consulta: 27/02/2024). 2013. URL: <https://www.atlassian.com/es/agile/scrum#continued>.
- [6] Scott Chacon. *Git: An Oral History*. (Última consulta: 14/03/2024). 2015. URL: <https://www.wired.com/2015/01/programming-society-destroying-story-git/>.
- [7] Scott Chacon y Ben Straub. *Pro Git*. 2nd. Apress, 2014.
- [8] Bob Hughes. *Software Project Management*. McGraw Hill, 2009.
- [9] Project Management Institute. *Software Project Management*. Project Management Institute, 2017.
- [10] Minderest. *Minderest - Dynamic Pricing & Price Optimization*. (Última consulta: 18/02/2024). URL: <https://sonitron.net/price-tracking-gfk-minderest/>.
- [11] Agencia Española de Protección de Datos. *Guía de protección de datos por defecto*. (Última consulta: 05/05/2024). URL: <https://www.aepd.es/guias/guia-proteccion-datos-por-defecto.pdf>.
- [12] Odoo S.A. *Odoo*. (Última consulta: 18/02/2024). URL: https://www.odoo.com/es_ES/blog/odoo-news-5/introduciendo-odoo-11-455.
- [13] SAP. *SAP Business One*. (Última consulta: 18/02/2024). URL: <https://axalphaconsulting.com/general/interfaz-usuario-sap-business-one-10-mejorada/>.
- [14] Spring Security. *Spring Security*. (Última consulta: 14/03/2024). URL: <https://spring.io/projects/spring-security>.
- [15] Ian Sommerville. *Software Engineering*. Pearson, 2016. Cap. 4, págs. 101-134.
- [16] LaTeX Project Team. *LaTeX Project*. (Última consulta: 14/03/2024). URL: <https://www.latex-project.org/>.
- [17] PRINCE2 Wiki. *PRINCE2 Wiki en español*. (Última consulta: 18/02/2024). 2023. URL: <https://prince2.wiki/es/tematicas/riesgo/>.

Anexos

Manual de instalación

Requisitos Previos

Antes de comenzar, es necesario tener una instalación de Linux Ubuntu con un servidor web Apache en el ordenador donde se vaya a instalar esta aplicación web. Hay que asegurarse de tener instaladas las siguientes herramientas:

- Java 17.0.8 (Oracle JDK)
- Node.js v18.19.1
- NPM 10.2.4
- Angular CLI 16.2.8
- MariaDB 10.11.7
- Maven (para la primera opción)

A continuación descargar el contenido del repositorio: <https://gitlab.inf.uva.es/enrmart/tfg> y una vez descargado su contenido, seguir los pasos enumerados a continuación.

Instalación Principal

Esta opción será exclusivamente realizada desde un terminal de Linux por línea de comandos Linux.

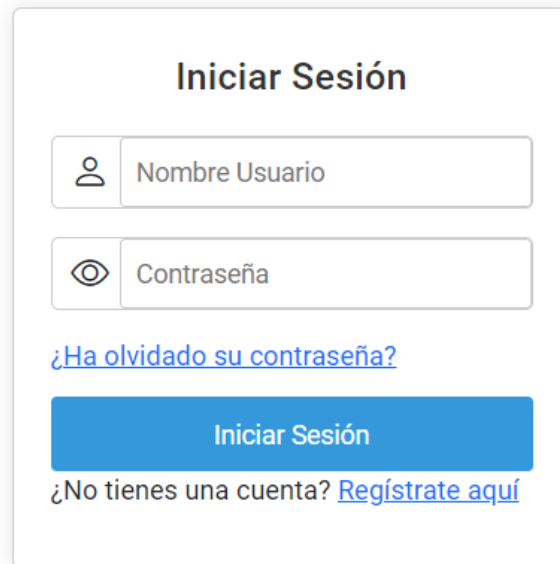
1. Instalación de Dependencias del servidor: Situarse en el directorio de inventario y navegar al directorio `inventarioBack`, una vez ahí ejecutar el comando: `mvn clean install`
2. Iniciar el servidor: Ejecutar el comando: `mvn spring-boot:run`
3. Instalación de Dependencias del cliente: Situarse en el directorio de inventario y navegar al directorio `inventarioFront`, una vez ahí ejecutar el comando: `npm install`
4. Iniciar el cliente: Ejecutar el comando: `ng serve`
5. Abrir el navegador y acceder a la URL: `http://localhost:4200`

Manual de Usuario

Este anexo contiene las instrucciones para el uso de la aplicación. Se presentará toda la funcionalidad implementada mediante imágenes ilustrativas capturadas de la propia aplicación.

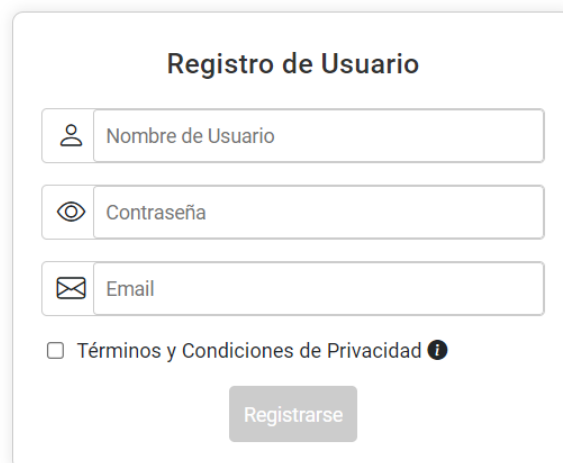
Acceso a la aplicación

En las Figuras 7.1 y 7.2 podemos ver los formularios de registro e inicio de sesión. En el registro deberemos cumplimentar el formulario, leer los términos de uso de la aplicación y aceptarlos para poder continuar. Una vez realizado este proceso se le llevará a la pagina de Inicio de Sesión, dónde se necesitará el nombre de usuario y la contraseña para iniciar sesión, una vez iniciada estaremos dentro de la aplicación. En la pantalla de Inicio de sesión también se puede apreciar la redirección a la recuperación de contraseña.



Formulario de Inicio de Sesión. El título es "Iniciar Sesión". Hay un campo de texto con un ícono de usuario y el texto "Nombre Usuario". Debajo hay un campo de texto con un ícono de ojo y el texto "Contraseña". Debajo de los campos hay un enlace azul que dice "¿Ha olvidado su contraseña?". Debajo del enlace hay un botón azul que dice "Iniciar Sesión". Debajo del botón hay un enlace azul que dice "¿No tienes una cuenta? Regístrate aquí".

Figura 7.1: Login



Formulario de Registro de Usuario. El título es "Registro de Usuario". Hay un campo de texto con un ícono de usuario y el texto "Nombre de Usuario". Debajo hay un campo de texto con un ícono de ojo y el texto "Contraseña". Debajo de los campos hay un campo de texto con un ícono de correo electrónico y el texto "Email". Debajo del campo de texto hay un checkbox con el texto "Términos y Condiciones de Privacidad" y un ícono de información. Debajo del checkbox hay un botón gris que dice "Registrarse".

Figura 7.2: Registro

Recuperación de contraseña

En las Figuras 7.3 y 7.4 podemos ver las pantalla de recuperación de contraseña. En la cuál se solicita el nombre de usuario primero y la nueva contraseña y su repetición después, una vez

todos los datos están introducidos y ambas contraseñas coinciden se permitirá pulsar el botón que realizará el cambio de contraseña. Una vez realizado será redirigido a la pantalla de inicio de sesión.

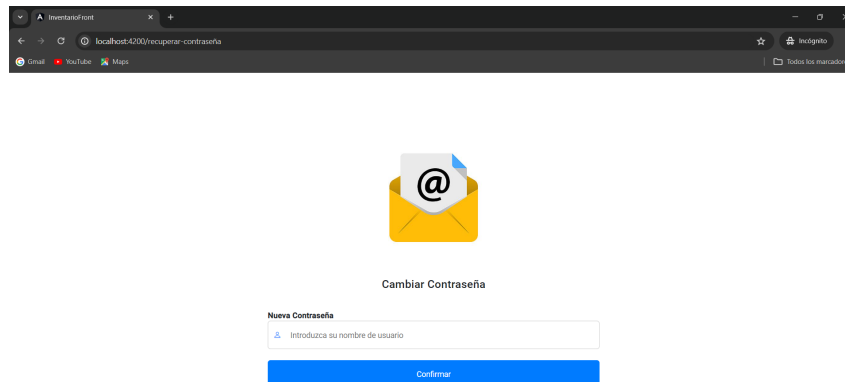


Figura 7.3: Nombre de Usuario para la recuperación de contraseña

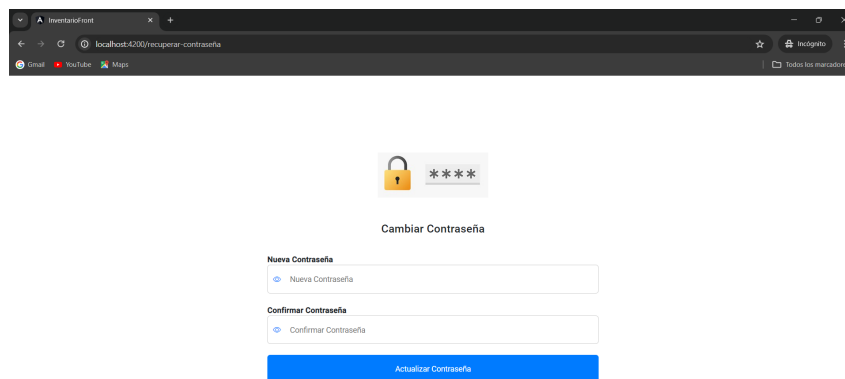


Figura 7.4: Contraseñas en la recuperación

Navegación por la aplicación

La aplicación tiene un menú superior en el cuál se encuentran las distintas secciones para poder moverse entre ellas, también cuenta con el icono de la empresa para volver al menú principal y el nombre del usuario, el cuál al pulsar sobre él permitirá consultar la información del usuario y cerrar sesión, como se puede apreciar en la siguiente imagen.



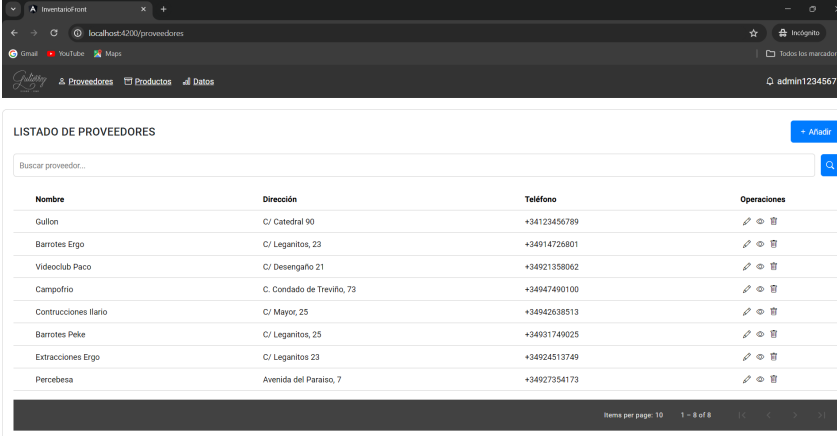
Figura 7.5: Header

Listado de proveedores

En la siguiente imagen se muestra la página donde reside todo el funcionamiento relacionado con los proveedores, en el centro se muestran el nombre, dirección y teléfono de todos los proveedores almacenados en el sistema en formato lista. Al lado de cada uno de ellos se muestran las

distintas operaciones que se puede llevar a cabo sobre ellos ojo (ver los detalles del proveedor), lápiz (editar la información del proveedor) y papelera (eliminar el proveedor).

A mayores, se puede apreciar en la parte superior derecha el botón Añadir Proveedor que permite añadir la información relevante de un proveedor. Y el buscador que nos permitirá buscar al proveedor por su CIF o nombre.

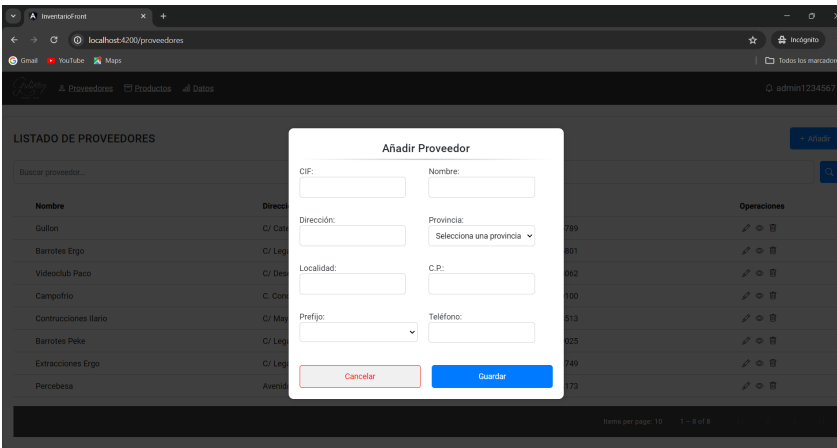


Nombre	Dirección	Teléfono	Operaciones
Gullon	C/ Catedral 90	+34123456789	
Barrotes Ergo	C/ Leganitos, 23	+34914726801	
Videoclub Paco	C/ Desengaño 21	+34921358062	
Campofrio	C. Condado de Treviño, 73	+34947490100	
Contrucciones Ilario	C/ Mayor, 25	+34942638513	
Barrotes Peke	C/ Leganitos, 25	+34931749025	
Extracciones Ergo	C/ Leganitos 23	+34924513749	
Percebesa	Avenida del Paraíso, 7	+34927354173	

Figura 7.6: Página principal de los proveedores

Añadir proveedor

La imagen 7.7 corresponde al formulario que se despliega al quiere añadir el proveedor. En el se piden distintos datos, que una vez introducidos, serán validados y una vez sean correctos se permitirá pulsar el botón guardar que almacenará el proveedor en el sistema.



Añadir Proveedor

CIF: Nombre:

Dirección: Provincia:

Localidad: C.P.:

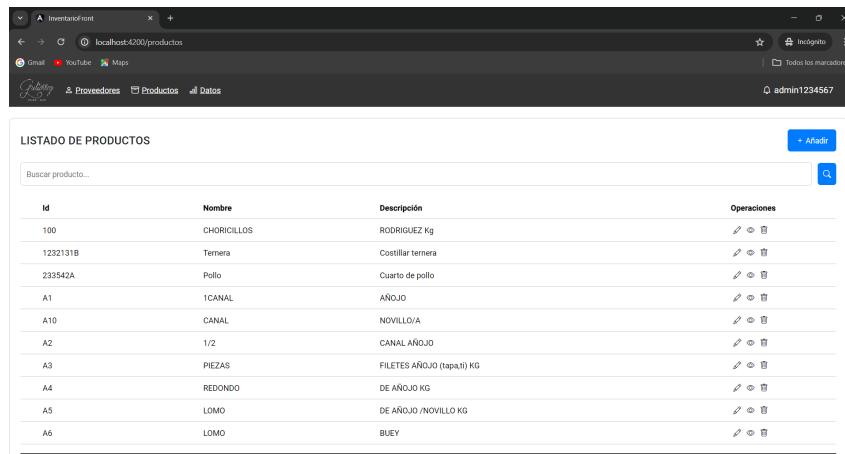
Prefijo: Teléfono:

Figura 7.7: Formulario de Añadir Proveedor

Listado de productos

En la siguiente imagen se muestra la página donde reside todo el funcionamiento relacionado con los productos, en el centro se muestran el id, nombre y descripción de todos los productos almacenados en el sistema en formato lista. Al lado de cada uno de ellos se muestran las distintas operaciones que se puede llevar a cabo sobre ellos ojo (ver los detalles del producto), lápiz (editar la información del producto) y papelera (eliminar el producto).

A mayores, se puede apreciar en la parte superior derecha el botón Añadir Producto que permite añadir la información relevante de un producto. Y el buscador que nos permitirá buscar al producto por su ID o nombre.























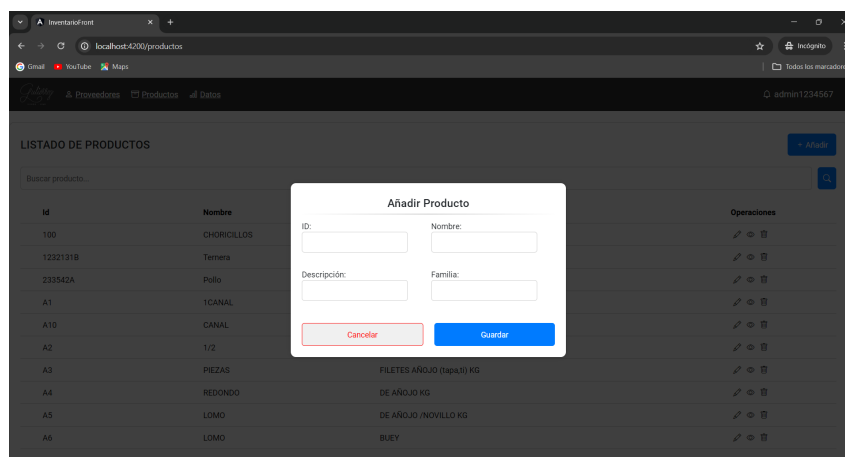
Id	Nombre	Descripción	Operaciones
100	CHORICILLOS	RODRIGUEZ Kg	 
1232131B	Ternera	Costillar ternera	 
233542A	Pollo	Cuarto de pollo	 
A1	TCANAL	AÑOJO	 
A10	CANAL	NOVILLO/A	 
A2	1/2	CANAL AÑOJO	 
A3	PIEZAS	FILETES AÑOJO (tapa) KG	 
A4	REDONDO	DE AÑOJO KG	 
A5	LOMO	DE AÑOJO /NOVILLO KG	 
A6	LOMO	BUEY	 

Figura 7.8: Página principal de los productos

Añadir producto

La imagen 7.9 corresponde al formulario que se despliega al quiere añadir el producto. En el se piden distintos datos(id, nombre, descripción y familia), que una vez introducidos, serán validados y una vez sean correctos se permitirá pulsar el botón guardar que almacenará el producto en el sistema.



Añadir Producto

ID:

Nombre:

Descripción:

Familia:

Cancelar
Guardar

Figura 7.9: Formulario de Añadir Producto

Información de perfil

En la siguiente imagen se puede ver la página donde se muestra la información del perfil que tiene una sesión iniciada en ese momento. También se aprecia un la modificación de los distintos datos a los cuales se accede si se pulsa el botón del lápiz de alguno de dichos campos. También existe el botón de cerrar sesión.

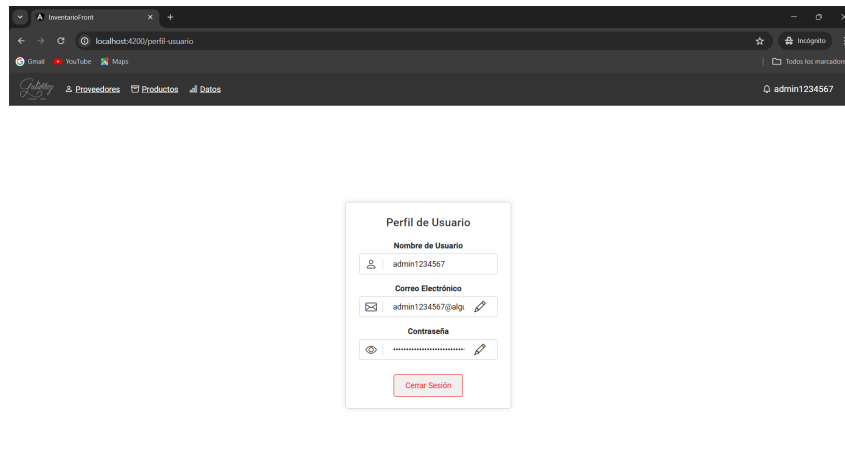


Figura 7.10: Pagina de perfil de Usuario

Consultar los datos

Las imágenes siguientes muestran la pantalla 7.11 donde se selecciona el periodo de tiempo entre el cual se quiere comprobar los precios o presupuestos que existen en el sistema para el producto indicado. La cuál una vez se han introducido los datos nos lleva a la pantalla 7.12 en la que se muestran los precios o presupuestos de dentro del periodo de tiempo indicado, más una semana extra que es la calculada por el modelo predictivo.

También se pueden ver 2 botones uno para Introducir Datos y otro para Subir Datos.

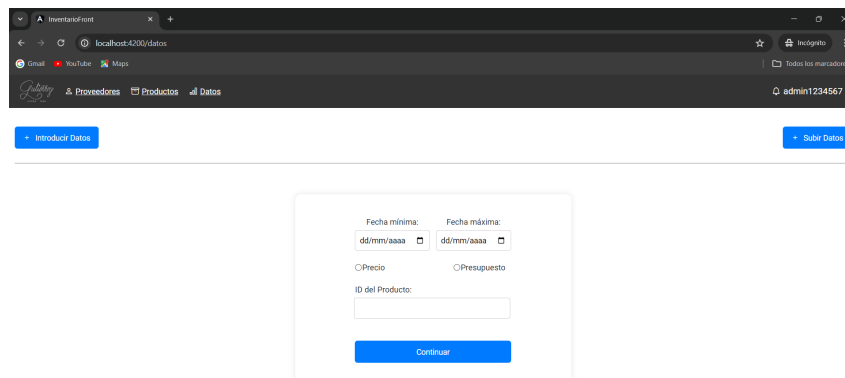


Figura 7.11: Página principal de los datos

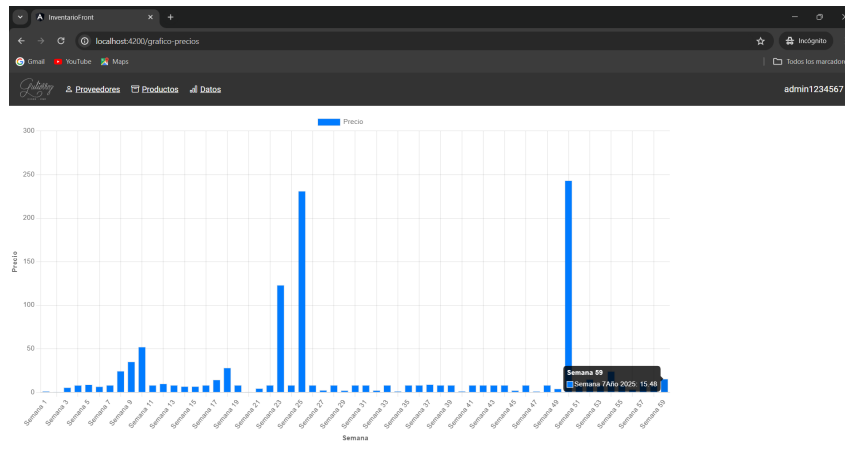


Figura 7.12: Página donde se muestran los datos

Subir datos

En la imagen siguiente se aprecia el formulario para subir archivos, ya sea mediante Drag N' Drop o mediante la selección del mismo pulsando en el botón Seleccionar.

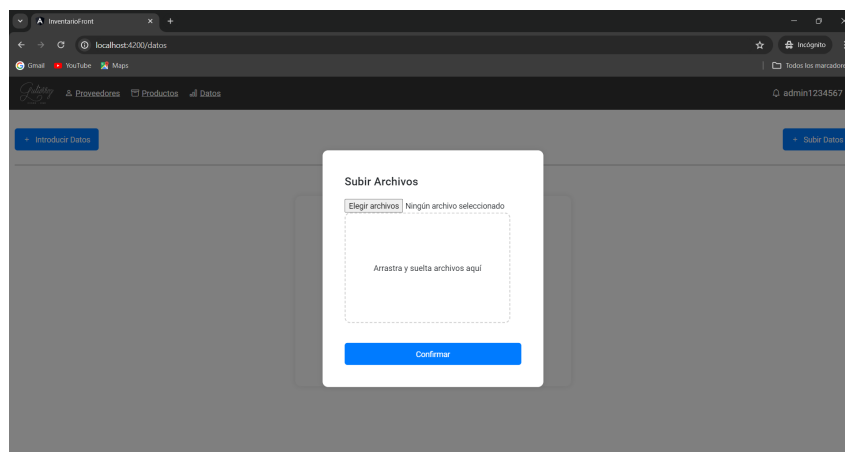


Figura 7.13: Formulario de Subir Archivos

Nota importante: El archivo se debe encontrar en una carpeta llamada `archivos` que se encuentre en el directorio padre de la carpeta `inventario`. También los archivos introducidos deben seguir un formato específico. Los archivos PDF deben seguir el formato de PDF de precios proporcionado por el cliente. Y el formato de los archivos sobre presupuestos debe ser `.xlsx` y de formato interno debe ser el de la siguiente imagen y solo se debe usar la HOJA 1 (los datos no son reales).

TARIFA DE PRECIOS						
18-mar-24						
CÓDIGO	AÑOJO HEMBRA		CÓDIGO	AÑOJO MACHO		
10001	MEDIA CANAL mas de 130 kg.....	5,85 €	9000	DELANTERO.....	5,35 €	
10002	MEDIA CANAL 3trozos.....menos de 130kg.....	5,85 €	9001	BOLA.....	6,80 €	
10003	DELANTERO.....	5,35 €	9002	CHULETA CON SOLOMILLO.....	10,10 €	
10004	TRASERO.....	6,80 €	9003	FALDA CON COSTILLA.....	5,70 €	
10005	BOLA hembra.....	6,46 €				
10006	CHULETA CON SOLOMILLO.....	12,60 €				
10007	FALDA.....	5,20 €				
				FILETES		
			9004	BABILLA (TITO).....	8,50 €	
			9005	CADERA.....	8,70 €	
			9006	TAPA.....	9,00 €	
			9007	ESPALDILLA.....	7,90 €	
			9008	CONTRA con Picaña.....	8,75 €	
			9009	LOMO ALTO.....	10,00 €	
			9010	LOMO BAJO.....	10,30 €	
			9011	SOLOMILLO.....	24,10 €	
			9012	PICAÑA.....	8,80 €	
			9013	REDONDO.....	8,60 €	

Figura 7.14: Formato interno archivo .xlsx

Introducir datos

La imagen 7.15 corresponde al formulario que se despliega al quiere añadir el un presupuesto. En el se piden distintos datos(nombre proveedor, id interno del producto, precio, cantidad y fecha), que una vez introducidos, serán validados y una vez sean correctos se permitirá pulsar el botón guardar que almacenará el presupuesto en el sistema.

The screenshot shows a web browser window with the URL 'localhost:4200/datos'. The page title is 'InventarioFront'. There are navigation links for 'Proveedores', 'Productos', and 'Datos'. A modal form titled 'Añadir Presupuesto' is displayed in the center. The form contains the following fields:

- Nombre Proveedor:
- ID Producto:
- Precio:
- Cantidad:
- Fecha:

At the bottom of the form are two buttons: 'Cancelar' (red) and 'Guardar' (blue).

Figura 7.15: Formulario de Añadir Presupuesto