

Universidade de Vigo

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

José Luis Campo Rivas

para a obtención do Título de Graduado en Enxeñaría Informática

Sistema de gestión de inventario mediante acceso remoto



Xullo, 2024

Traballo de Fin de Grao Nº: [REDACTED]

Titor/a: [REDACTED]

Área de coñecemento: Linguaxes e Sistemas Informáticos

Departamento: Informática

*“Para toda la gente que me ha acompañado,
en especial a mi madre.*

*Gracias a ti he aprendido a esforzarme
cada día al igual que tú lo haces.*

*Espero que te sientas orgulloso de mí y,
algún día, pueda devolverte todo lo que me has dado.”*

INDICE

INTRODUCCION	6
Presentación del problema	6
Estado del arte	6
OBJETIVOS	7
Objetivo principal	7
Objetivos específicos	7
RESUMEN	7
Metodología	8
PLANIFICACION Y SEGUIMIENTO	9
Planificación inicial	9
Seguimiento del proyecto	11
ARQUITECTURA	15
TECNOLOGÍAS USADAS	16
Frontend	16
Backend	17
Base de datos	17
Control de cambios	18
Aplicaciones externas	18
ESPECIFICACION Y ANÁLISIS DE REQUISITOS	18
Definición de las historias de usuario	18
DISEÑO DEL SOFTWARE	20
Estático	20
Dinámico	21
GESTIÓN DE DATOS E INFORMACIÓN	24
PRUEBAS LLEVADAS A CABO	26
Casos de prueba	26
PRINCIPALES APORTACIONES	30
CONCLUSIONES	31
Conclusiones Técnicas	31
Conclusiones Personales	32
VÍAS DE TRABAJO FUTURO	32
BIBLIOGRAFIA	33
ANEXO: MANUAL DE USUARIO	35

INDICE DE ILUSTRACIONES

Ilustración 1: Diagrama de Gantt del proyecto	10
Ilustración 2: Diagrama de componentes del frontend	20
Ilustración 3: Diagrama de componentes del backend	21
Ilustración 4: Flujo de ejecución del backend en cada petición.....	21
Ilustración 5: Diagrama de secuencia del frontend	22
Ilustración 6: Diagrama de secuencia del backend	23
Ilustración 7: Modelo Entidad-Relación de la base de datos	24
Ilustración 9: Página de bienvenida de la web	37
Ilustración 10: Formulario de inicio de sesión	37
Ilustración 11: Página principal de la web.....	38
Ilustración 12: Lista de elementos de la barra lateral	38
Ilustración 13: Ejemplo de notificación	39
Ilustración 14: Formulario para modificar datos de usuario.....	39
Ilustración 15: Página de gestión de usuarios.....	40
Ilustración 17: Formulario de creación de usuarios	40
Ilustración 16: Formulario de modificación de usuarios	41
Ilustración 18: Página de gestión de compras	41
Ilustración 19: Formulario de creación de compras	42
Ilustración 20: Formulario de modificación de compras	42
Ilustración 21: Formulario de registro de proveedores	42
Ilustración 22: Página de gestión de productos	43
Ilustración 24: Formulario de creación de productos	43
Ilustración 23: Formulario de modificación de productos	44
Ilustración 25: Página de gestión de ventas.....	44
Ilustración 26: Formulario de creación de ventas.....	45
Ilustración 27: Formulario de modificación de ventas.....	45
Ilustración 28: Formulario de registro de clientes	45

INDICE DE TABLAS

Tabla 1: Etapas del proyecto	10
Tabla 2: Tabla de base de datos "Users"	24
Tabla 3: Tabla de base de datos "Unit_measures"	24
Tabla 4: Tabla de base de datos "Product_categories"	25
Tabla 5: Tabla de base de datos "Products"	25
Tabla 6: Tabla de base de datos "Clients"	25
Tabla 7: Tabla de base de datos "Invoices"	25
Tabla 8: Tabla de base de datos "Invoice_lines"	25
Tabla 9: Tabla de base de datos "Notifications"	25
Tabla 10: Tabla de base de datos "Suppliers"	25
Tabla 11: Tabla de base de datos "Supplier_provides_product"	25
Tabla 12: Tabla de base de datos "Buys"	25
Tabla 13: Tabla de base de datos "Buy_contains_product"	26
Tabla 14: Prueba de inicio de sesión	26
Tabla 15: Prueba de la barra lateral	27
Tabla 16: Prueba de barra de notificaciones	27
Tabla 17: Prueba de gestión de datos propios	27
Tabla 18: Prueba de representación gráfica de datos	28
Tabla 19: Prueba de gestión de usuarios	28
Tabla 20: Prueba de gestión de productos	29
Tabla 21: Prueba de gestión de compras	29
Tabla 22: Prueba de gestión de ventas	30
Tabla 23: Prueba de cierre de sesión	30
Tabla 24: Prueba de generación de PDF's	30

INTRODUCCION

Presentación del problema

En la actualidad, un gran número de empresas emplean herramientas digitales para poder realizar procesos de manera eficaz. Gracias a esto, obtienen ventaja competitiva contra el resto, permitiendo situar su marca por encima de la competencia [\[1\]](#).

Para que esto se dé, las empresas deben disponer de software que aumente su productividad y eficacia, siempre y cuando sea asequible [\[1\]](#). De este modo, las compañías de gran tamaño (con mayor poder adquisitivo) pueden acceder fácilmente a estas herramientas, mientras que a las pequeñas y medianas empresas (PYMES) les genera numerosos desafíos [\[2\]](#).

Las PYMES se enfrentan a numerosos retos para poder acceder a herramientas digitales. El principal reto es el coste asociado a la integración y mantenimiento de estas herramientas, pero eso no es todo, ya que estos sistemas suelen ser complejos y su curva de aprendizaje también [\[3\]](#). Como solución a esta problemática, se contrata a personal especializado y cualificado para configurar y mantener estas herramientas, suponiendo todavía más costes a la empresa [\[2\]](#). Debido a esta dificultad, las pequeñas empresas deben buscar formas simples y económicas para usar herramientas digitales, pero que a su vez les aporte una gran productividad y les facilite el día a día.

Una de las formas más sencillas, para lograr esta digitalización, es incorporar sistemas que no modifiquen en exceso la forma de trabajar de la empresa, haciendo que la adopción de estas nuevas herramientas sea imperceptible [\[4\]](#) [\[5\]](#).

Estado del arte

Cuando una empresa desea digitalizarse, busca herramientas que les ayuden con tareas como: planificación de materiales necesarios, patrones de demanda, estacionalidad y otros elementos que afectan a la planificación del inventario [\[6\]](#).

Para gestionar todos estos servicios, existen múltiples opciones. Algunas de ellas se centran en servicios concretos, mientras que otras abarcan un gran número de ellos.

Aquellas aplicaciones que abarcan muchas funcionalidades acaban siendo poco intuitivas, y sus costes muy altos [\[7\]](#). Por estas razones, se ha decidido atajar un problema concreto de las empresas, gestionar su stock.

En cuanto a las herramientas ya existentes para gestionar el stock, podemos encontrar múltiples, algunas de ellas son [\[8\]](#):

- Dolibarr: aplicación personalizable y escalable que permite todo tipo de funcionalidades. Algunas de ellas son gratuitas, mientras que a otras solo se puede acceder mediante una suscripción de pago [\[9\]](#).
- Monstock: aplicación francesa que ofrece seguimiento y gestión de inventarios, además de incorporar inteligencia artificial para el reabastecimiento. También incorpora una suscripción mensual [\[10\]](#).
- Odoo Inventory: aplicación de código abierto, por lo tanto gratuita, que permite gestionar el stock de una empresa. Tiene costes añadidos si quieres incorporar aplicaciones adicionales, además de necesitar dispositivos hardware específicos para trabajar con esta aplicación (lectores, escáneres, balanzas, etc.) [\[11\]](#).

Todas estas herramientas tienen múltiples funcionalidades comunes: registro de compras y ventas, alertas de reabastecimiento o identificación única de productos. Todas ellas son muy

interesantes para ofrecer a las empresas, porque brindan flexibilidad y adaptabilidad a cualquier situación que éstas requieran.

Por estos motivos es necesario diseñar e implementar un sistema que las incluya, pero sin olvidar que tiene que adaptarse a las metodologías de trabajo existentes en las empresas.

Además, permiten adaptarse a los distintos métodos de gestión de inventario que implementan las empresas [12]:

- FIFO (*First In First Out*): en español se traduce como *El primero en entrar es el primero en salir*. Consiste en vender las existencias según el orden de llegada.
- LIFO (*Last In First Out*): en español significa *El último en entrar es el primero en salir*. El nombre es autoexplicativo.
- Método ABC: se basa en dividir el inventario en 3 categorías según el porcentaje de ventas que suponga. En la categoría A se encuentran los productos más vendidos y se manejan por unidades, en la categoría B están los productos vendidos a menudo y son tratados por lotes, y, por último, en la categoría C se encuentran los artículos menos vendidos y se gestionan mediante cajas.

OBJETIVOS

Objetivo principal

El objetivo principal del proyecto es implementar un sistema, mediante el cual, PYMES puedan gestionar de manera remota su inventario de forma cómoda y desde diferentes dispositivos. Para ello se desarrollará un sistema web genérico que pueda ser usado por empresas con distintas necesidades y formas de trabajar.

Objetivos específicos

- Facilitar la adopción tecnológica proporcionando una solución intuitiva y fácil de usar, que permita a las pequeñas empresas adoptar tecnología moderna sin enfrentarse a altos costes ni cambios radicales en su forma de trabajar.
- Diseñar e implementar un sistema que gestione los datos de inventario de forma consistente y que agilice las operaciones referentes a este.
- Diseño e implementación de una interfaz amigable y responsiva, que permita acceder rápidamente a las funciones y herramientas necesarias para gestionar su inventario, desde cualquier dispositivo.
- Desarrollar una infraestructura donde los datos de la empresa sean tratados con rigurosa seguridad, incorporando cifrado en datos sensibles y medidas robustas de control de acceso para garantizar confidencialidad e integridad.

RESUMEN

Para proveer a empresas de un sistema gestor de stock con acceso remoto, en este trabajo se realizará una aplicación web, es decir, un recurso que será accesible a través de un navegador y que ofrecerá todas las funcionalidades necesarias para cumplir con las características descritas del sistema: registro de compras y ventas, alertas de reabastecimiento o identificación única de productos.

Como en gran parte de las infraestructuras web, en este proyecto se empleará una arquitectura cliente-servidor que, mediante una conexión a Internet, permitirá a los usuarios acceder de manera remota.

Además, el servidor, que recogerá toda la lógica de negocio, se organizará mediante el patrón arquitectónico MVC (siglas de Modelo-Vista-Controlador). Con este patrón separaremos las

responsabilidades de la aplicación, haciéndola más escalable y mantenible. Con el objetivo de, en un futuro, poder añadir o modificar funcionalidades fácilmente según las necesidades del cliente.

Las funcionalidades que tendrá el sistema están pensadas para, en la mayoría de empresas, poder hacer frente a cualquier circunstancia y así registrarlo en el sistema. Para ello, se desarrollarán características genéricas que todo sistema de gestión de stock debe tener: gestionar compras y ventas, generación de alertas por desabastecimiento, autenticar y autorizar el acceso de usuarios, acceder al historial de transacciones y proveer información de forma que sea fácilmente analizable.

A todas estas funcionalidades se podrá acceder mediante una interfaz intuitiva y fácil de usar, para proveer a las empresas de un sistema que les permita llevar una gestión integral de su inventario.

Metodología

La metodología para organizar un proyecto es fundamental para poder completar todos los objetivos marcados. Para este trabajo se ha optado por usar la metodología ágil Scrum [\[13\]](#), ya que tiene unas características muy adecuadas para nuestro contexto.

Estas características se basan en el empirismo y el pensamiento Lean, es decir, en tomar decisiones adaptadas al entorno y centrándonos en lo esencial. Por esto mismo, emplea un enfoque iterativo e incremental, para optimizar la previsibilidad y controlar el riesgo [\[13\]](#).

Este enfoque iterativo permite dividir el tiempo de desarrollo del proyecto en breves espacios de tiempo, llamados Sprints [\[13\]](#). En nuestro proyecto se realizan iteraciones de 2 semanas, en la que se implementan las funcionalidades acordadas previamente.

Además, en esta metodología es fundamental que el equipo de trabajo sea pequeño. Con esto conseguimos que el equipo permanezca ágil a la vez que completa un trabajo significativo [\[13\]](#). Los distintos roles del equipo Scrum son los siguientes:

- **Dueño del producto:** persona responsable de maximizar el valor del producto resultante. Para ello: identifica el objetivo del producto, define el orden de implementación de las funcionalidades, delegar responsabilidades, etc.
En nuestro caso, este rol es ejercido conjuntamente entre el autor del TFG (José Luis Campo Rivas) y el tutor (Juan Francisco Gálvez G.).
- **Equipo de desarrollo:** personas que crean los aspectos definidos en un Sprint. Por cada incremento, crean un plan llamado Sprint Backlog. Este rol lo ejerce el autor del TFG.
- **Maestro Scrum:** responsable de aplicar Scrum eficazmente para ayudar conseguir los objetivos del proyecto. Además, sirve de apoyo al dueño del producto en cuestiones técnicas. El tutor del TFG representa este rol.

Una vez explicados los diferentes actores que participan durante el desarrollo, es necesario explicar las distintas etapas de este. En total, Scrum tiene 5 etapas [\[14\]](#):

- **Planificación del proyecto:** En esta etapa, se definen los conceptos previos al inicio del desarrollo del proyecto (objetivos, tecnologías, arquitectura, ...). Se crea el Product Backlog, una lista priorizada de todas las funcionalidades de la aplicación, normalmente se usan historias de usuario para ello.
- **Planificación de Sprint:** Antes de comenzar cada sprint, el equipo realiza una reunión de planificación de sprint. Durante esta reunión, el equipo selecciona un conjunto de

elementos del Product Backlog para trabajar durante el sprint y define las tareas necesarias para completarlos.

- **Sprint:** El sprint es un período corto de tiempo, típicamente de dos semanas [\[15\]](#), durante el cual el equipo implementa las funcionalidades seleccionadas del Product Backlog.
- **Revisión del Sprint:** Al final de cada sprint, se lleva a cabo una reunión de revisión del sprint en la que el equipo muestra el trabajo completado al Product Owner. También se analiza el proceso de trabajo durante el Sprint, para saber cómo mejorarlo.
- **Fin del proyecto:** Esta fase de la metodología Scrum busca entregar, una vez terminados los Sprints, el resultado final al cliente y poder acumular experiencia que puedas utilizar en próximos proyectos para mejorar la eficiencia del equipo Scrum. Para ello, se entrega el resultado final al cliente y una retrospectiva del proyecto.

PLANIFICACION Y SEGUIMIENTO

Planificación inicial

Durante el proyecto se busca seguir la metodología Scrum al completo, tal y como se expone en el resumen de la metodología.

Para ello se ha dividido el trabajo en las siguientes etapas:

- **Planificación del proyecto (31/01 - 05/03):** Durante esta etapa se limita el alcance del proyecto. Primero, se analiza el mercado actual de aplicaciones similares y, sirviendo esto como base, se procede a definir las funcionalidades que tendrá nuestra aplicación. Una vez se crea una idea general sobre el funcionamiento de la aplicación, se procede a definir detalladamente todas las funcionalidades, mediante historias de usuario. A partir de este punto, se analizan las distintas opciones tecnológicas para poder desarrollar la aplicación. Por último, se divide todo el trabajo previsto para el proyecto en Sprints, para obtener la planificación inicial del proyecto.
- **Ciclo de Sprints (05/03 - 14/05):** Una vez sabemos el orden de prioridad de las historias de usuario (Product Backlog), comenzamos a implementarlas. Esta implementación se hará en 5 incrementos de 2 semanas. Una vez se termine un Sprint, se realizará una reunión con el Maestro Scrum para sacar conclusiones y planificar el siguiente.
 - o **Sprint 1 (05/03 - 19/03):** En este primer Sprint, se desarrollará la base del sistema, incluyendo la configuración del entorno de desarrollo, la creación del repositorio de código y la configuración inicial del servidor backend y la base de datos. Se implementarán las primeras historias de usuario relacionadas con el registro y la autenticación de usuarios.
 - o **Sprint 2 (19/03 - 02/04):** Se centrará en la implementación de funcionalidades básicas de gestión de inventarios, como la creación, actualización y eliminación de productos. También se desarrollarán las interfaces de usuario necesarias para estas funcionalidades.
 - o **Sprint 3 (02/04 - 16/04):** En este Sprint, se ampliarán las funcionalidades de gestión de ventas y se añadirá la capacidad de generar informes básicos de ventas. Se empezará a trabajar en la integración de proveedores y en las funciones para gestionar las compras, asegurando que el flujo de trabajo sea coherente y funcional.
 - o **Sprint 4 (16/04 - 30/04):** Este Sprint se enfocará en la optimización y refactorización del código. Se mejorará la eficiencia y la simplicidad del sistema, asegurando que todas las funcionalidades implementadas hasta el momento sean robustas y eficientes. Se corregirá cualquier error detectado que no haya sido solventado antes.

- **Sprint 5 (30/04 - 14/05):** El último Sprint se dedicará a implementar las funcionalidades finales, como la generación de facturas en formato PDF y la mejora de la seguridad del sistema. Se realizarán las últimas pruebas y se asegurará que todo el sistema funcione correctamente antes de la fase de finalización del proyecto.
- **Fin del proyecto (14/05 - 28/05):** Esta última etapa se centra en cerrar cualquier brecha pendiente y asegurarnos que todo funciona perfectamente. Se actualiza la documentación pendiente y, además, se crea un manual de usuario que proporciona una orientación clara y precisa del funcionamiento del producto final.

Como el trabajo tiene asignadas 300 horas de dedicación, a cada una de las etapas se le han estipulado un número de horas necesarias para completar las tareas planteadas. Siguiendo el siguiente esquema:

ETAPA	HORAS [TOTAL]	COMIENZO	FIN
INICIO DEL PROYECTO	60 [60]	31/01	05/03
INTRODUCCIÓN	10		
ESTADO DEL ARTE	10		
OBJETIVOS	5		
HISTORIAS DE USUARIO	15		
ARQUITECTURA Y TECNOLOGÍAS	10		
PLANIFICACIÓN INICIAL	10		
SPRINT 1	35 [95]	05/03	21/03
SPRINT 2	35 [130]	21/03	04/04
SPRINT 3	35 [165]	04/04	18/04
SPRINT 4	35 [200]	18/04	02/05
SPRINT 5	35 [235]	02/05	16/05
FIN DEL PROYECTO	65 [300]	16/05	30/05
PRUEBAS	20		
MANUAL DE USUARIO	15		
VÍAS DE TRABAJO FUTURAS	15		
FINALIZAR DOCUMENTACIÓN	15		

Tabla 1. Etapas del proyecto

Como resultado de este reparto de tareas con sus correspondientes fechas, el diagrama de Gantt resultante sería el siguiente:

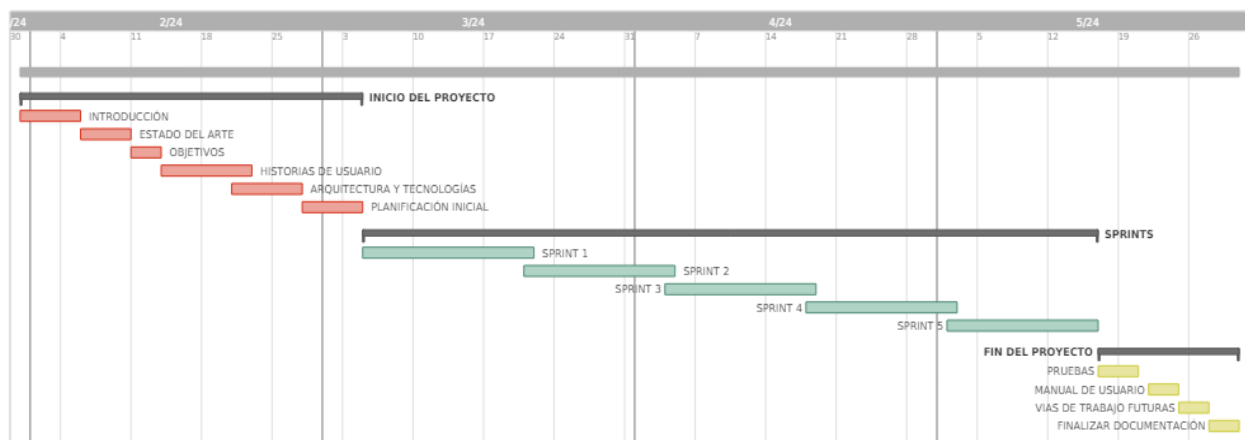


Ilustración 1: Diagrama de Gantt del proyecto

Seguimiento del proyecto

Como ya se ha explicado anteriormente, el desarrollo de la aplicación se divide en Sprints. En cada uno de ellos se implementan el mayor número de funcionalidades posibles, repitiendo este proceso hasta completar los tiempos establecidos en la planificación inicial.

Al comienzo de cada Sprint se realizó una reunión con el Product Owner y Scrum Máster para seleccionar las historias de usuario que se implementarán durante las siguientes dos semanas. En este apartado se recogerá la información asociada a cada Sprint, desde las historias de usuario que se implementarán hasta los posibles retrasos que puedan pasar, incluyendo la división en tareas de cada historia de usuario.

Sprint 1

Durante el primer Sprint, se realizaron tareas transversales a todas las historias de usuario, es decir, estas tareas contenían información relativa a numerosas historias de usuario.

Se comenzó realizando el diagrama Modelo Entidad-Relación para la base de datos, desde la definición de las entidades hasta sus atributos y relaciones.

A posteriori, se transformó este diagrama a un modelo relacional, es decir, a un conjunto de tablas (relaciones) usando SQL como lenguaje y, junto a la preparación del entorno de desarrollo para la base de datos, esta fue creada. Para facilitar el desarrollo a futuro, se crearon datos ficticios para adaptar el desarrollo de la aplicación a un entorno real desde el comienzo.

Una vez diseñada y creada la base de datos (mediante DDL, Lenguaje de Definición de Datos), se pudo comenzar con el desarrollo de la API y el frontend. Para ello se preparó el entorno de desarrollo (instalando las dependencias necesarias) y se implementaron las funcionalidades necesarias para las historias de usuario seleccionadas para este Sprint. Estas historias fueron seleccionadas con un objetivo común, obtener una gestión de usuarios al completo.

Estas historias de usuario son:

- HU-1: Como usuario, quiero iniciar sesión en la aplicación con mi cuenta para usar sus funcionalidades.
- HU-2: Como usuario, quiero actualizar los datos de mi cuenta.
- HU-6: Como gerente, quiero gestionar los usuarios del sistema.

La división de tareas durante este Sprint es la siguiente:

Para todas las historias de usuario:

- Identificar las entidades, los atributos y relaciones necesarias para poder representarlos en la base de datos
- Definir los permisos que tendrán los distintos tipos de usuarios: gestor de ventas, gestor de compras y gerente.

HU-1:

- Backend: Crear el sistema de generación de tokens JWT y rutas de la API para poder identificar las sesiones de los usuarios.
- Frontend: Mostrar al usuario un formulario donde pueda introducir sus credenciales.

HU-2:

- Backend: Crear una ruta que reciba los nuevos datos del usuario.
- Frontend: Mostrar un formulario cuando el usuario acceda a la información de su perfil, este formulario podrá modificar sus datos.

HU-6:

- Backend: Crear una ruta de la API, accesible solo para gerentes, que permita gestionar los datos de los usuarios.

- Frontend: Mostrar un listado, tan solo para los gerentes, de todos los usuarios del sistema. Con las opciones de crear, modificar y eliminar.

Durante este Sprint se ha aprendido mucho sobre el ritmo al que puede avanzar el proyecto, al avance durante estas dos semanas ha sido (en comparación con las expectativas) positivo, ya que se esperaba un menor avance del proyecto en este primer Sprint. Por lo tanto, para los siguientes Sprints se pueden tener una perspectiva más amplia de todo lo que se puede llegar a hacer.

Tiempo dedicado:

A este Sprint se le han dedicado 40 horas. El tiempo estimado era de 35 horas, pero debido a la alta disponibilidad de tiempo y la motivación inicial por comenzar el proyecto se ha decidido expandir el tiempo 5 horas más. Además, esto permite reducir la carga de trabajo futura.

Sprint 2

El trabajo realizado durante el segundo fue muy continuista con el primero. Se buscaron historias de usuario pendientes de implementar, en este caso fueron dos, y se procedió a su implementación tanto en el frontend como en el backend.

En el primer Sprint nos enfocamos en las secciones relativas a los usuarios, en este caso es referente a la gestión de compras y proveedores, incluyendo también la gestión de productos.

Historias de usuario de este Sprint:

- HU-3: Como gestor de compras, quiero gestionar los productos del inventario.
- HU-4: Como gestor de compras, quiero gestionar las compras.

División de tareas:

HU-3:

- Backend: Crear todas las rutas necesarias para poder consultar, insertar, modificar y eliminar productos de la base de datos.
- Frontend: Mostrar un listado de todos los productos con botones que permitan acceder a un formulario para modificarlos.
- Frontend: Mostrar un botón donde se despliegue un formulario para poder crear un nuevo producto.

HU-4:

- Backend: Crear las rutas necesarias para gestionar los datos referentes a las compras y proveedores.
- Frontend: Mostrar el listado de compras ordenadas por fecha (más recientes primero) y permitir al usuario gestionar los datos de cada una.
- Frontend: Mostrar un pequeño formulario donde el usuario pueda crear un nuevo proveedor.

El aprendizaje que se obtiene de este Sprint no es tan positivo como el anterior. La aplicación ha avanzado correctamente, a un buen ritmo; pero la gestión del tiempo no ha sido la ideal. Las horas dedicadas se han concentrado en pocos días (sábados y domingos) debido a carga de trabajo ajena al proyecto.

Pese a las complicaciones, se ha cumplido con las tareas marcadas para este Sprint y se añaden nuevas experiencias para poder organizar el trabajo futuro de la mejor forma posible.

Tiempo dedicado:

A este Sprint se le han dedicado 35 horas, se ha llegado las horas marcadas en la planificación inicial. Pese a todos los inconvenientes que ha habido durante estas 2 semanas se ha cumplido con el objetivo del Sprint.

Sprint 3

En el tercer Sprint se seleccionaron aquellas funcionalidades pendientes por implementar y se procedió a ello. Esto se hizo con el objetivo de obtener una aplicación funcional lo antes posible. Esto conlleva que, de cara al usuario, pueda verse una aplicación completa y que contenga todas las funcionalidades esperadas.

Aunque esto no es del todo así, porque durante estos tres Sprints se han ido dejando tareas más secundarias de lado, haciendo que la aplicación no sea todo lo sólida que se espera.

Por esto mismo, también se han centrado los esfuerzos en ese aspecto.

Historias de usuario para este Sprint:

- HU-5: Como gestor de ventas, quiero gestionar las ventas
- HU-7: Como gerente, quiero gestionar los productos, compras y ventas

Se han dividido en las siguientes tareas:

HU-5

- Backend: Crear las rutas necesarias para gestionar los datos referentes a las ventas y proveedores
- Frontend: Mostrar el listado de ventas ordenadas por fecha (más recientes primero) y permitir al usuario gestionar los datos de cada una.
- Frontend: Mostrar un pequeño formulario donde el usuario pueda crear un nuevo cliente.

HU-7

- Backend: Gestionar las rutas para que solo puedan acceder los usuarios acreditados para ello en función de sus responsabilidades.
- Frontend: Mostrar las funcionalidades adecuadas según el tipo de usuario.

El aprendizaje que se obtiene de este Sprint es referente a la importancia de priorizar tareas. Hasta ahora se han centrado los esfuerzos en hacer grandes desarrollos de la aplicación, haciendo que las funcionalidades más básicas ya estén desarrolladas.

Gracias a esto, en este Sprint se ha focalizado en tareas más específicas. Aquellas que el usuario no ve, pero que son muy necesarias.

Tiempo dedicado:

Durante estas dos semanas no se ha podido llegar a las horas asignadas en la planificación inicial, el total de horas ha sido de 30. Esto ha sido por causas ajenas al proyecto y a una mala gestión del tiempo por parte del autor.

A pesar de los inconvenientes, se lograron importantes avances en la implementación de las funcionalidades planificadas y estas circunstancias sirven de aprendizaje para futuras iteraciones.

Sprint 4

El enfoque de los anteriores Sprints ha sido implementar todas las funcionalidades de la aplicación, priorizando la entrega de un producto funcional. Gracias a esto, la aplicación cumple todos los requisitos exigidos, pero ahora es momento de mejorar su eficiencia y simpleza. Para ello, en este Sprint se han enfocado todos los esfuerzos en refactorizar el código ya creado.

Historias de usuario de este Sprint:

- Ninguna en concreto, tan solo se han mejorado las funcionalidades ya implementadas.

División de tareas:

- Detectar aquellas partes del proyecto cuyo rendimiento y resiliencia no es el esperado.
- Solucionar aquellas insuficiencias del proyecto que se han detectado. Concretamente:
 - o Mejorar la velocidad de carga de las páginas reduciendo las peticiones que se hacen al backend.
 - o Unificar los formularios para crear y modificar datos en un solo componente, reduciendo código repetitivo.

Durante este Sprint se ha puesto en manifiesto, y se ha aprendido mucho, sobre la importancia de tener una estrategia a largo plazo en el proyecto. El enfoque durante los primeros Sprints ha sido hacer que la aplicación cumpla con todos los objetivos (historias de usuario) descritas al comienzo del trabajo. Para posteriormente, hacer mejoras específicas que marcan la diferencia en cuanto a la experiencia de usuario.

Esta estrategia ha permitido simplificar y acelerar mucho el desarrollo a lo largo de todo el proyecto.

Tiempo dedicado:

A este Sprint se le han dedicado 37 horas, tal y como se ha definido en la planificación inicial. Gracias a lo aprendido en el anterior Sprint con respecto a la gestión de tiempo, el trabajo durante estas dos semanas ha sido muy constante haciendo que haya sido fácil llegar a las horas marcadas.

Sprint 5

En este Sprint se ha seguido el mismo enfoque que el Sprint anterior. Se ha mejorado la calidad del código y creado una aplicación más robusta, incluso se han detectado ciertas ineficiencias al renderizar el frontend y se han corregido.

Además, se ha completado el último requisito funcional previsto para este proyecto: generar facturas en formato PDF asociadas a cada venta. Por cada venta que se realiza, se genera un archivo PDF que recopila toda su información, almacenándose en una carpeta llamada *invoices* en el backend.

Historias de usuario de este Sprint:

- HU-5: Como gestor de ventas, quiero gestionar las ventas.
- Al igual que el Sprint anterior, mejorar las funcionalidades ya implementadas. Concretamente:
 - o Mostrar mensajes de error en el inicio de sesión, perfil y gestión de usuarios.
 - o Validar los datos introducidos por el usuario en el inicio de sesión, perfil y gestión de usuarios.

División de tareas:

- HU-5: Integrar un servicio que permita almacenar cada venta con toda su información.
- Mejorar aquellas partes del proyecto que sean ineficientes:
 - o Crear un componente único para todos los mensajes de error y éxito.
 - o Validar los datos de forma que estas mismas funciones puedan ser reutilizables en el resto de las páginas de la aplicación.

A lo largo de este último Sprint se han centrado los esfuerzos en finalizar y pulir todas las funcionalidades del proyecto. Se ha conseguido que el gestor de stock sea completamente funcional y eficiente, proporcionando una herramienta sólida y fiable para los usuarios finales.

La optimización del código y la corrección de ineficiencias en el frontend han resultado en una aplicación más fluida y con mejor rendimiento.

Tiempo dedicado:

Para este último Sprint se han necesitado 42 horas de trabajo. Esto hace que en la planificación inicial, la expectativa de horas necesarias para el desarrollo se haya desviado 9 horas con respecto al tiempo esperado.

Fin del proyecto

Una vez terminados los 5 Sprints del proyecto, se pasó a la etapa final del proyecto. Durante esta fase, se realizaron varias tareas clave para cerrar el proyecto de manera efectiva.

Pruebas

Se dedicaron las 20 horas planificadas para realizar pruebas exhaustivas del sistema. Esto incluyó pruebas de funcionalidad y usabilidad para asegurar que la aplicación cumpliera con todos los requisitos y fuera robusta y confiable. Las pruebas permiten identificar y corregir errores en caso de existir alguno, garantizando que la aplicación funcione de manera óptima en diferentes escenarios.

Manual de Usuario

Aunque inicialmente se habían planificado 15 horas para la elaboración del manual de usuario, se requirió más tiempo para completar un documento detallado y comprensible. Esta dedicación ocasionó retrasos, pero resultó en un manual de alta calidad. El tiempo dedicado final fueron 30 horas.

Vías de Trabajo Futuras

También se invirtieron las 15 horas estimadas en la redacción y planificación de las futuras mejoras y expansiones para el gestor de stock. Este apartado se centró en identificar áreas de crecimiento y proponer nuevas funcionalidades que podrían ser implementadas en futuras iteraciones. Las horas dedicadas a esta tarea fueron suficientes para delinear un plan claro y bien estructurado para el futuro desarrollo del sistema.

Finalizar Documentación

Se había estimado un total de 15 horas para completar la documentación final, que incluía el diseño de software, gestión de datos y conclusiones. Sin embargo, esta tarea se retrasó debido a la necesidad de hacerla lo más comprensible y detallada posible. Fue necesario actualizar continuamente la documentación para mantenerla en línea con el estado actual del proyecto, lo cual extendió el tiempo dedicado 10 horas más de lo previsto.

Resumiendo, en esta etapa final del proyecto se realizaron todas las tareas necesarias, aunque algunas requirieron más tiempo del inicialmente previsto. Las pruebas y la planificación de las vías de trabajo futuras se realizaron según lo planeado, pero la elaboración del manual de usuario y la finalización de la documentación se extendieron debido a la intención de hacer estos apartados lo más comprensibles y detallados posibles. A pesar de estos desafíos, se logró finalizar el proyecto con una documentación completa y un producto final que cumple con todos los objetivos establecidos.

ARQUITECTURA

Como en toda aplicación web, la arquitectura principal del sistema sigue el modelo cliente-servidor. Donde el navegador del usuario tomará el papel de cliente y, por otro lado, estará el servidor donde tengamos alojada nuestra aplicación.

Por lo tanto, separaremos las dos partes de la aplicación: cliente (aplicación front-end) y servidor (aplicación back-end o API, también conocida como *Application Programming Interface*). Ambas partes están implementadas con el mismo objetivo, pero cada una tiene distintas responsabilidades.

En el servidor se encuentra la lógica de nuestra aplicación. Donde se tratarán todas las peticiones de consultas o modificaciones de los datos de nuestro sistema. Para facilitar el desarrollo y su futuro mantenimiento, se ha decidido repartir las responsabilidades de la aplicación siguiendo el patrón arquitectónico Modelo-Vista-Controlador, también llamado MVC.

Debido a la separación entre frontend y backend, este patrón se modificará respecto del original. Esto será explicado más adelante. Las partes que conforman el backend son [\[16\]](#):

- **Modelo:** Es el encargado de interactuar con la base de datos. Para cada entidad que se maneje (productos, ventas, compras, usuarios, etc.), existirá un modelo específico para cada uno. Haciendo esto conseguimos compartimentar el código, haciéndolo mucho más mantenible en el largo plazo. Todos estos modelos se encuentran en la carpeta *models* del backend.
- **Controlador:** Define las diferentes rutas por las que podremos acceder a la API, al igual que en los modelos, existirá un controlador por cada entidad para un mejor desarrollo. Otra responsabilidad de cada controlador será hacer uso del modelo correspondiente, para gestionar las peticiones que reciba el servidor. Los controladores están en la carpeta *routes*.
- **Vista:** Aquí es donde se encuentra la diferencia con respecto al MVC original. En la vista se suelen presentar las distintas páginas HTML de nuestra aplicación. En nuestro caso eso se gestionará en el frontend, haciendo que la vista se represente mediante datos en formato JSON que se enviarán al cliente. En este caso no habrá ninguna carpeta que represente la vista, tan solo se enviarán los datos recibidos desde el modelo y, el framework escogido para el backend, se encargará de transformarlo en formato JSON.

En el cliente mostrarán los datos recibidos desde la API y se generarán peticiones para gestionarlos. Para ello se desarrolla una Single-Page Application (SPA), que consiste en un único conjunto de archivos HTML, CSS y JavaScript que se cargan una única vez en el cliente. Para dar la sensación al usuario de navegar entre distintas páginas se usa JavaScript, en cual carga todo el contenido HTML de la página de forma dinámica.

La principal ventaja de usar una SPA, en contrapartida con el MVC convencional, es la experiencia más fluida para los usuarios, es decir, no existen saltos bruscos entre las distintas páginas de la aplicación, ya que todo el contenido se carga desde el código JavaScript del cliente.

TECNOLOGÍAS USADAS

Para la elaboración del trabajo de fin de grado (TFG), se ha decidido usar una gran variedad de herramientas de software para desarrollar tanto el frontend como el backend de la aplicación, así como para gestionar la base de datos y controlar los cambios en el código. A continuación, se detalla el uso específico de cada una:

Frontend

HTML, CSS, JavaScript: Estos lenguajes son la base fundamental para la creación y estilización de la interfaz de usuario de la aplicación. Se utiliza HTML para estructurar el contenido, CSS para estilizarlo y JavaScript para agregar interactividad y funcionalidades dinámicas.

React 18.2 + Vite: Se opta por usar React como librería de desarrollo frontend, en conjunto con Vite, un empaquetador de código. Esta combinación proporciona una eficiente gestión del estado de la aplicación y una rápida capacidad de recarga durante el proceso de desarrollo, lo que permite una experiencia de desarrollo productiva.

Librerías específicas:

- **react-router-dom:** Se integra esta librería en el proyecto para gestionar la navegación dentro de la aplicación web. Con React-router-dom se pueden crear rutas y configurar el enrutamiento dinámico, lo que resulta esencial para garantizar una experiencia de usuario fluida.
- **Material UI:** Esta biblioteca proporciona una amplia gama de componentes y elementos de diseño predefinidos, así como iconos listos para usar. La utilización de Material UI agiliza el proceso de diseño e implementación de la interfaz de usuario, permitiendo crear una aplicación con una apariencia moderna, profesional y minimalista.

NPM: Se utiliza NPM como gestor de paquetes para instalar, gestionar y actualizar las dependencias de mi proyecto React. Esta herramienta facilita la integración de bibliotecas y módulos externos, simplificando el proceso de desarrollo y mantenimiento.

Backend

Python 3.12: Se decide usar Python como lenguaje de programación por su amplia gama de bibliotecas y su sintaxis clara y legible, lo que facilita el desarrollo y la mantenibilidad del backend de la aplicación.

FastAPI: Se utiliza FastAPI para crear la API REST del sistema. La elección de FastAPI se debe a su alto rendimiento y su sintaxis intuitiva, que permite desarrollar una API rápida y fácil de entender.

Librerías específicas:

- **pydantic:** Se usa pydantic para la validación de datos entrantes en las solicitudes a la API. Esta librería proporciona una forma sencilla y robusta de definir esquemas de datos y validar la entrada del usuario.
- **bcrypt:** Para garantizar la seguridad de las contraseñas almacenadas en la base de datos, se implementa un cifrado utilizando la librería bcrypt. Esto ayuda a proteger la información confidencial de los usuarios de la aplicación.
- **python-jose:** Se implementa la generación de tokens JWT para el proceso de autenticación de usuarios utilizando esta librería. Esta técnica proporciona una capa adicional de seguridad al sistema de autenticación de la aplicación.
- **pymysql:** Esta herramienta es usada para establecer la conexión con la base de datos MySQL, facilitando la comunicación entre el backend y la base de datos. Esto permite el almacenamiento y recuperación eficientes de datos.
- **fpdf2:** Permite generar ficheros PDF de forma rápida y sencilla. Además, ofrece una gran variedad de funcionalidades para insertar tablas, gráficas e imágenes.

Base de datos

MySQL: Se utiliza MySQL como sistema de gestión de bases de datos relacional para almacenar y estructurar los datos de mi aplicación.

XAMPP: Esta suite de software permite manejar la base de datos en un entorno local, facilitando el desarrollo y la depuración. Además, proporciona una interfaz gráfica donde se pueden observar los datos almacenados.

Control de cambios

Git + GitHub: Se utiliza Git como sistema de control de versiones para gestionar los cambios en el código del proyecto, y GitHub como plataforma de alojamiento remoto para poder acceder desde cualquier dispositivo.

Aplicaciones externas

Visual Studio Code: Se utiliza este entorno de desarrollo integrado (IDE) para programar el frontend de la aplicación debido a su flexibilidad y extensibilidad.

PyCharm: Se usa PyCharm como IDE para programar el backend de la aplicación, ya que es potente y posee un gran conjunto de herramientas para el desarrollo en Python.

Insomnia: Se emplea Insomnia como cliente REST para realizar solicitudes al backend de la aplicación y probar su funcionamiento sin necesidad de desarrollar el frontend.

Draw.io: Se utiliza Draw.io para crear el Modelo de Entidad-Relación (MER) de la base de datos y los diagramas de Gantt para la planificación y seguimiento del proyecto.

Word: Se usa Word para la redacción y elaboración de la documentación del trabajo de fin de grado.

En resumen, estas herramientas son fundamentales en el desarrollo, la gestión y la documentación del trabajo de fin de grado, permitiendo crear una aplicación web funcional y completa.

ESPECIFICACION Y ANÁLISIS DE REQUISITOS

Definición de las historias de usuario

1. **Como usuario, quiero iniciar sesión en la aplicación con mi cuenta para usar sus funcionalidades.**

Descripción: Cualquier usuario podrá acceder de manera remota a la aplicación para las operaciones que su rol le permita. Para ello se verificará su identidad con los datos de acceso (nombre de usuario y contraseña).

Precondiciones:

- El empleado debe poseer datos de acceso asociados a un usuario.

Postcondiciones:

- El empleado puede acceder a todas las funcionalidades del sistema disponibles para él.

2. **Como usuario, quiero actualizar los datos de mi cuenta.**

Descripción: Cualquier usuario podrá consultar los datos de su cuenta. Además, podrá modificarlos y/o eliminarlos.

Precondiciones:

- Haber iniciado sesión con un usuario válido.
- Acceder al apartado “Perfil”.

Postcondiciones:

- Se muestran los datos del usuario.
- Si se elimina la cuenta, se redirige a la página principal y se hace un borrado lógico.

3. **Como gestor de compras, quiero gestionar los productos del inventario.**

Descripción: Se mostrará una lista con todos los productos del inventario y se podrá acceder a la información específica de cada uno: código identificador, nombre, precio, coste número de

unidades, límite de alerta, proveedor. Además, podrá: insertar, modificar y eliminar los datos de los productos.

Precondiciones:

- Haber iniciado sesión con un usuario válido.
- Acceder al apartado “Productos”.

Postcondiciones:

- Lista de productos donde podrá acceder a la información de cada uno y modificarla.

4. Como gestor de compras, quiero gestionar las compras.

Descripción: Un Gestor de Compras podrá consultar y administrar compras de elementos del inventario. Cada una estará formada por una o varias líneas de compras que detallarán: el producto comprado, proveedor, coste, número de unidades.

Precondiciones:

- Haber iniciado sesión con un usuario válido.
- Acceder al apartado “Compras”.

Postcondiciones:

- Listado de compras donde podrá acceder a la información de cada una.
- Registro de la compra.
- El gestor de compras tendrá acceso a una representación gráfica de las compras.

5. Como gestor de ventas, quiero gestionar las ventas.

Descripción: Un Gestor de Ventas podrá consultar y administrar ventas de elementos del inventario. Cada una estará formada por una o varias líneas de factura que detallarán: el producto comprado, cliente, precio, número de unidades. Toda esta información se reunirá en una factura que será almacenada.

Precondiciones:

- Haber iniciado sesión con un usuario válido.
- Acceder al apartado “Ventas”.

Postcondiciones:

- Registro de la venta.
- La factura es almacenada.
- Si un producto tiene menos unidades de las deseadas, se genera una alerta a los gestores de compra y gerentes.
- El gestor de ventas tendrá acceso a una representación gráfica de las ventas.

6. Como gerente, quiero gestionar los usuarios del sistema.

Descripción: El gerente podrá acceder a la información de todos los usuarios y modificarla. Además, será el único que pueda crear usuarios. Estos usuarios pueden ser de cualquier tipo.

Precondiciones:

- Haber iniciado sesión con un usuario válido.
- Acceder al apartado “Usuarios”.

Postcondiciones:

- Listado de todos los usuarios activos junto con sus datos asociados.
- Registro de un nuevo usuario.

7. Como gerente, quiero gestionar los productos, compras y ventas.

Descripción: Los gerentes tendrán las funcionalidades disponibles para los gestores de compras y ventas. Por lo tanto, podrán gestionar productos, compras y ventas.

Precondiciones:

- Haber iniciado sesión con un usuario válido.

Postcondiciones:

- El gerente tendrá acceso a las funcionalidades de los gestores.

- El gerente tendrá acceso a una representación gráfica de las compras y ventas.

DISEÑO DEL SOFTWARE

Estático

Para poder entender las distintas funcionalidades del proyecto y como han sido implementadas es necesario ejemplificarlas de forma visual, de esta forma se podrán entender y mejorar las distintas relaciones entre nuestros componentes.

En este caso, se han usado JavaScript y Python como lenguajes de programación. Estos son lenguajes multiparadigma [17], es decir, que soportan múltiples paradigmas de programación (orientación a objetos, funcional, etc.).

En este proyecto se ha decidido hacer uso de la programación funcional como paradigma de diseño de software.

Para poder representar las relaciones que existen entre las distintas partes del sistema (tanto en el frontend como en el backend) se ha decidido crear diagramas de componentes [18].

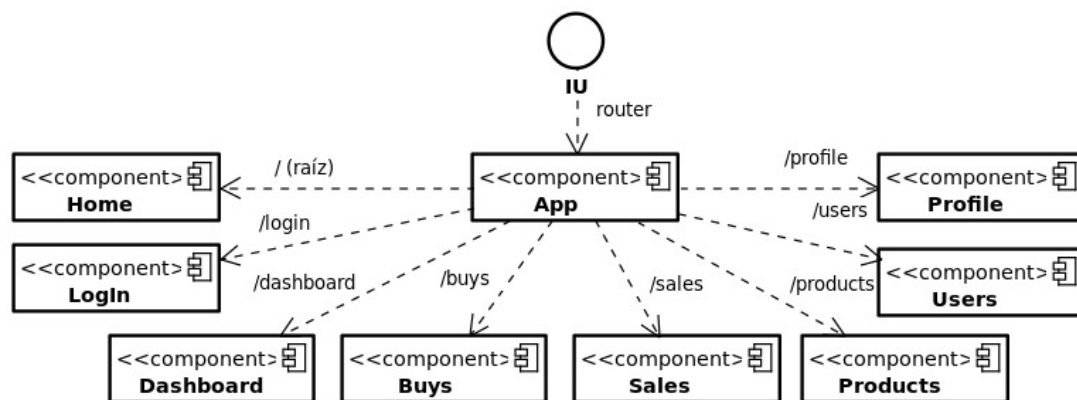


Ilustración 2: Diagrama de componentes del frontend

El diagrama anterior representa los componentes del frontend que se renderizan según la funcionalidad a la que acceda el usuario.

Comenzamos con el elemento principal, la interfaz de usuario (IU), como nos encontramos en un entorno web, estará representado por el navegador del usuario.

Esta interfaz hará uso del componente App, que sirve como enrutador para, según la ruta a la que acceda el usuario, mostrarle una página de la aplicación u otra.

Por ejemplo, si se accede a la ruta */profile* se mostrará un formulario con los datos del usuario que podrá cambiar si lo desea.

Toda la información referente al contenido de cada página se encuentra en el manual de usuario.

Con esta forma de aislar cada página, se consigue simplificar el desarrollo y permite gestionar cada parte de la aplicación de forma aislada.

Además, cada página hace uso de componentes externos, ya sea de la librería *Material UI* o aquellos que se encuentran en la carpeta *components* del frontend.

Por otro lado, tenemos el backend, que responderá a las diferentes peticiones que reciba.

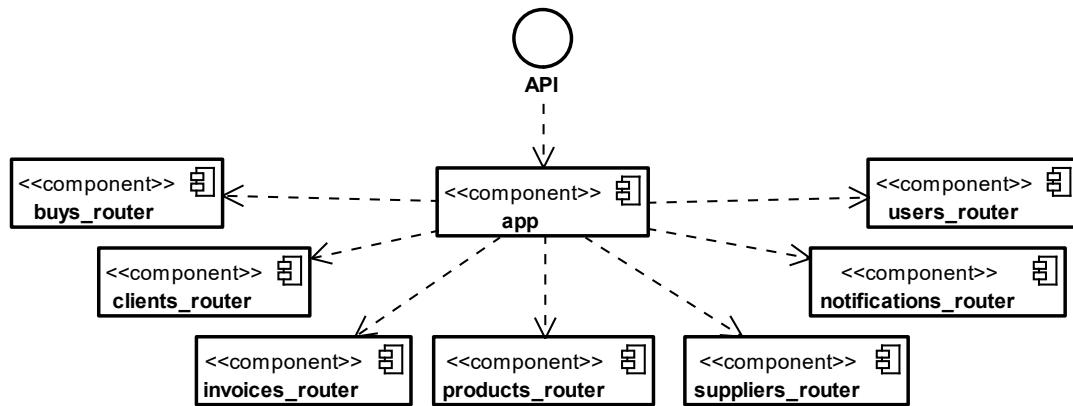


Ilustración 3: Diagrama de componentes del backend

En la jerarquía se encuentra la API definida para poder interactuar con el sistema. Esta API, como su propio nombre indica, es una interfaz que define los métodos mediante los cuales se pueden realizar operaciones en la aplicación.

Estas operaciones llegarán al componente *app* que contiene los diferentes *routers* que lo componen. Cada *router* define el conjunto de rutas y métodos que el cliente puede usar, todos ellos están relacionados con la entidad con la que se identifica, por ejemplo: *invoices_router* contiene las rutas y métodos referentes a la gestión de ventas/facturas.

A su vez, estos *routers* interactúan a más bajo nivel con los *models*, estos son los que tienen acceso a la base de datos de la aplicación. Existe un *model* por cada entidad (al igual que los *routers*), haciendo que no existan demasiadas interdependencias entre los distintos componentes. De esta forma, la cadena de dependencias queda de la siguiente manera:

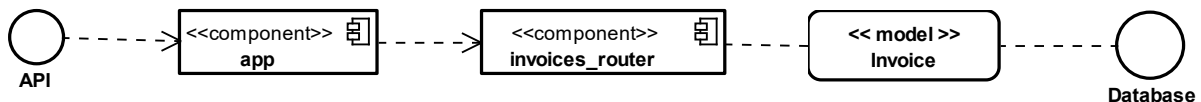


Ilustración 4: Flujo de ejecución del backend en cada petición

La API sería la parte de la aplicación más cercana al cliente y con la que interactúa. Cada petición que se reciba desencadenará una pila de llamadas en el orden que aparece en el diagrama anterior, resultado finalmente en una acción sobre la base de datos.

Dinámico

Una vez explicada la parte estática del diseño de software, es conveniente hacer lo mismo con la parte dinámica, ya que esta es la que permite entender cómo interactúan los distintos módulos de la aplicación y como se genera una pila de llamadas entre estos con cada interacción del usuario.

Por una parte, tenemos la secuencia de llamadas que se realiza en el frontend:

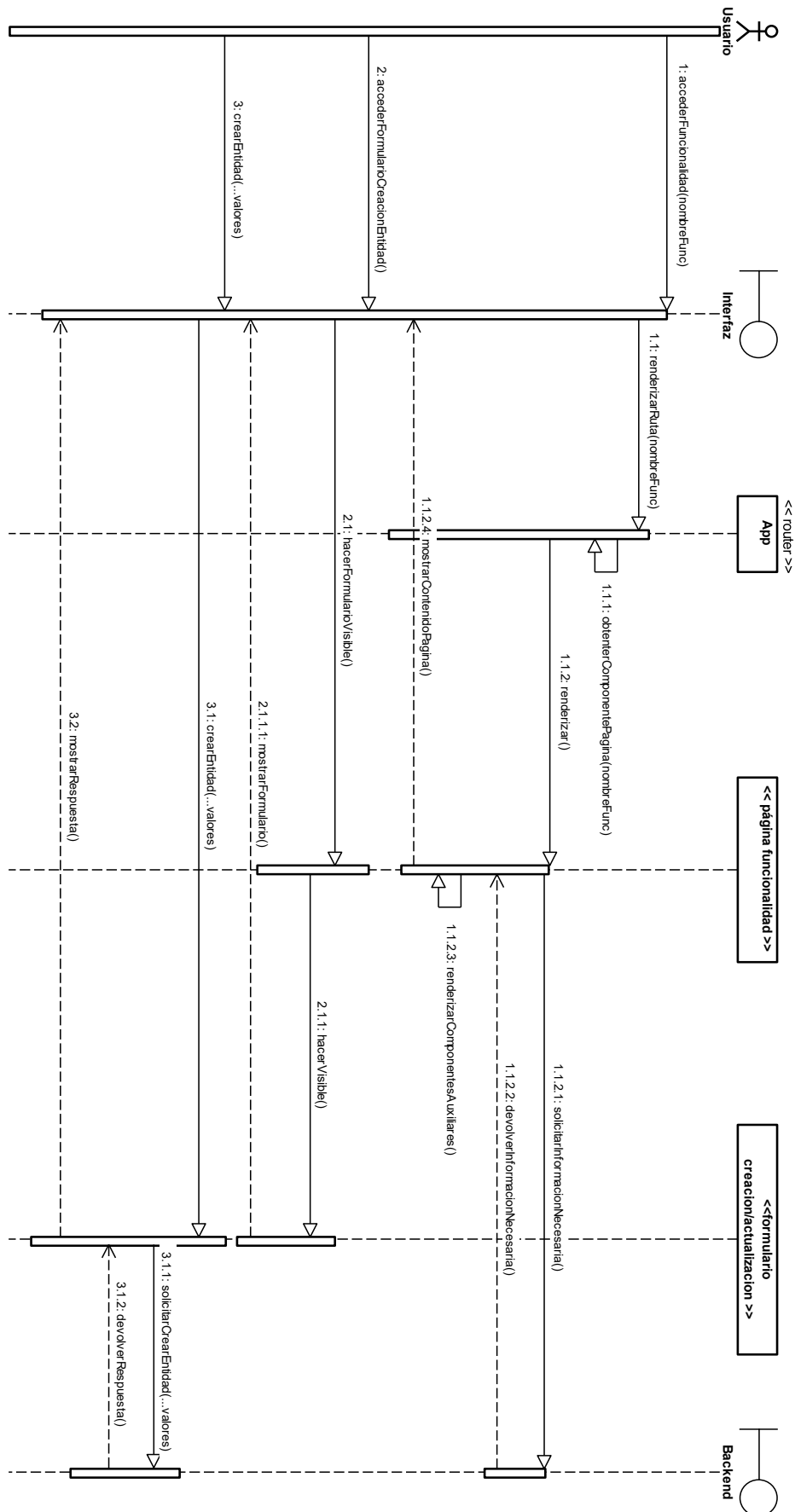


Ilustración 5: Diagrama de secuencia del frontend

En la primera solicitud del usuario se muestra la consecución de llamadas que surge cuando quiere acceder a una funcionalidad (por ejemplo: gestionar usuarios).

En el diagrama se observa como el componente *Interfaz* (en este caso el navegador web), llama al componente *App* que actúa como enrutador, seleccionando y renderizando la página que corresponda en función de la funcionalidad seleccionada.

Esto permite al usuario acceder a la página que ha solicitado en la primera llamada a la *Interfaz*.

A continuación, en el diagrama se muestran dos llamadas más, simbolizando un pequeño ejemplo de que sucedería si el usuario desea realizar una acción en la página seleccionada.

En primer lugar, el usuario solicita que se muestre el formulario para poder realizar la acción (por ejemplo: crear un nuevo usuario). A partir de la *Interfaz*, se crea la pila de llamadas necesaria para ello.

Cuando ya se ha mostrado el formulario, el usuario procede a introducir los valores necesarios en el formulario y genera otra acción para crear una nueva entidad (siguiendo con el ejemplo: crear un nuevo usuario).

Los valores introducidos por el usuario se transmiten desde el componente *formulario* al backend. A continuación, se recibe una respuesta, positiva o negativa, y según esta respuesta se muestra el mensaje correspondiente al usuario.

Una vez explicada la vista dinámica del frontend, se pasará a explicar la del backend.

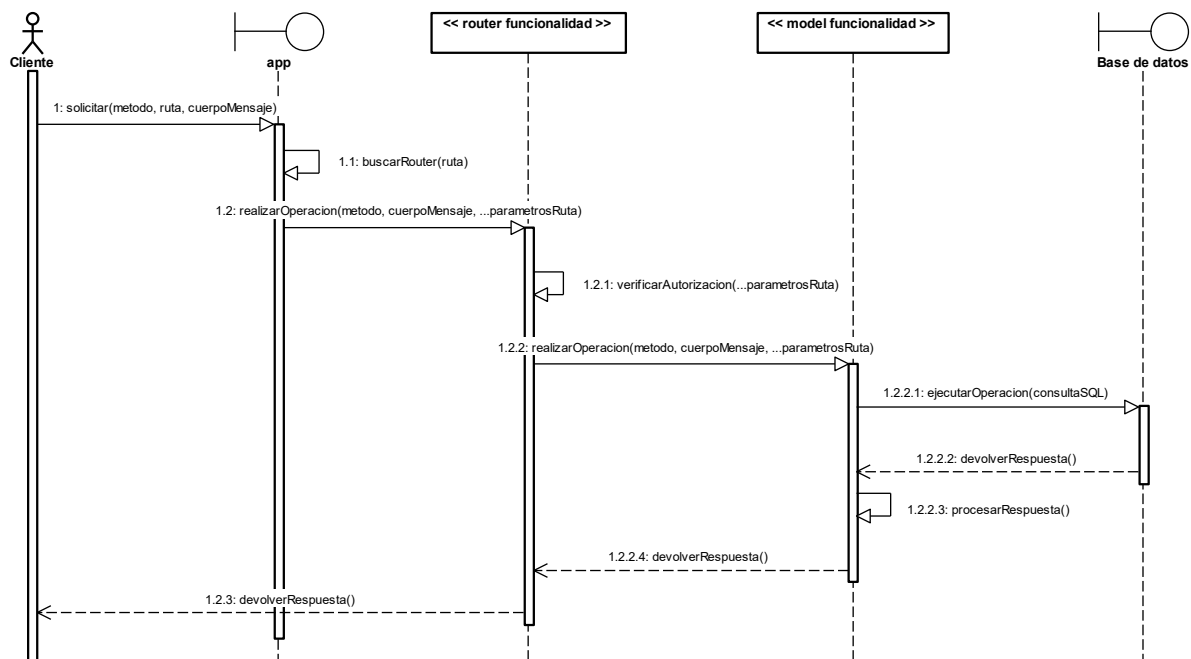


Ilustración 6: Diagrama de secuencia del backend

Desde el cliente se solicita una operación al backend, cada operación contiene una ruta (por ejemplo: */users*), un método (GET, POST, PUT, DELETE) y un cuerpo del mensaje en caso de que sea necesario.

El punto de entrada a la API es la instancia *app*. Esta instancia actúa como enrutador para seleccionar en que componente *router* está definida la ruta solicitada.

Una vez hecho esto, el *router* comprobará si el cliente que hizo la petición tiene los permisos necesarios para ello y llama al modelo, Este modelo es quien tiene acceso a la base de datos.

Cuando el modelo ya ha ejecutado la operación solicitada, genera una respuesta que finalmente se devolverá al cliente.

GESTIÓN DE DATOS E INFORMACIÓN

Como ya se ha mencionado en el apartado de *Tecnologías usadas*, los datos se almacenan en una base de datos relacional, concretamente el sistema gestor de base de datos usado es MySQL.

Una vez analizadas las historias de usuario y las funcionalidades de la aplicación, se ha definido mediante un Modelo Entidad-Relación (MER) las distintas entidades que se almacenarán en la base de datos, así como las relaciones entre ellas.

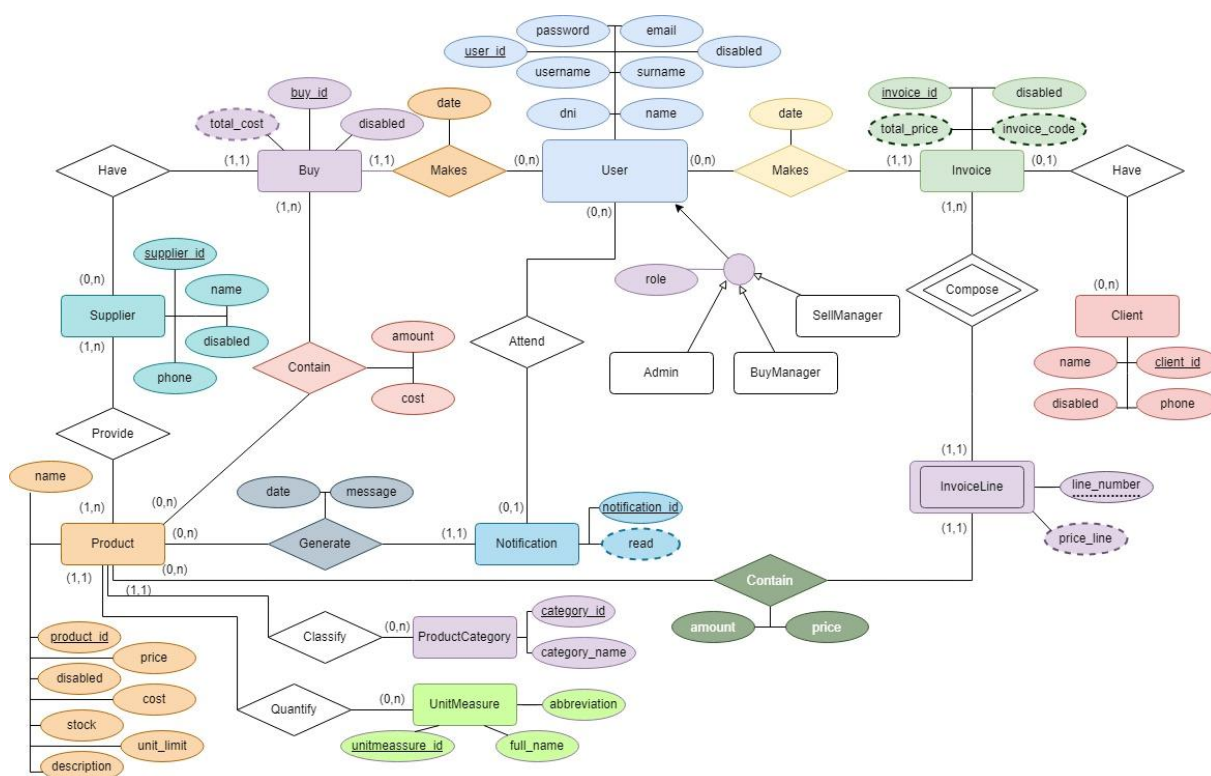


Ilustración 7: Modelo Entidad-Relación de la base de datos

En el MER anterior está representada toda la información que gestionará la aplicación. Una vez definido y representado toda esta información, se procedió a “traducir” todo este contenido de forma menos conceptual, acercándolo a un plano más técnico y de desarrollo. Para ello se han definido las diferentes tablas que tendrá la base de datos con sus atributos y relaciones:

Users	user_id	dni	username	password	name	surname	email	role	disabled
	Clave primaria	Único	Único				Único		

Tabla 2: Tabla de base de datos "Users"

Unit_measures	unit_measure_id	abbreviation	full_name
	Clave primaria		

Tabla 3: Tabla de base de datos "Unit_measures"

Product_categories	category_id	category_name
	Clave primaria	

Tabla 4: Tabla de base de datos "Product_categories"

Products	product_id	name	description	price	cost	stock	category_id	unit_measure_id	unit_limit	disabled
	Clave primaria						Clave foránea a Product_categories	Clave foránea a Unit_measures		

Tabla 5: Tabla de base de datos "Products"

Clients	client_id	name	phone	disabled
	Clave primaria			

Tabla 6: Tabla de base de datos "Clients"

Invoices	invoice_id	date	user_id	client_id	disabled
	Clave primaria		Clave foránea a Users	Clave foránea a Clients	

Tabla 7: Tabla de base de datos "Invoices"

Invoice_lines	invoice_id	line_number	product_id	amount	price
	Clave primaria		Clave foránea a Products		
	Clave foránea a Invoices				

Tabla 8: Tabla de base de datos "Invoice_lines"

Notifications	notification_id	date	message	product_id	user_id
	Clave primaria			Clave foránea a Products	Clave foránea a Users

Tabla 9: Tabla de base de datos "Notifications"

Suppliers	supplier_id	name	phone	disabled
	Clave primaria			

Tabla 10: Tabla de base de datos "Suppliers"

Supplier_provides_product	product_id	supplier_id
	Clave primaria	
	Clave foránea a Products	Clave foránea a Suppliers

Tabla 11: Tabla de base de datos "Supplier_provides_product"

Buys	buy_id	date	user_id	supplier_id	disabled
	Clave primaria		Clave foránea a Users	Clave foránea a Suppliers	

Tabla 12: Tabla de base de datos "Buys"

Buy_contains_product	buy_id	product_id	cost	amount
	Clave primaria			
	Clave foránea a Buys	Clave foránea a Products		

Tabla 13: Tabla de base de datos "Buy_contains_product"

Con este último diagrama se consigue simplificar la escritura del código SQL necesario para crear las distintas tablas que formarán la base de datos de la aplicación.

Este código SQL se encuentra en la carpeta *db* del proyecto, con el nombre *database.sql*.

Además, también se incluyen dos scripts adicionales (*example_data_buys.sql* y *example_data_invoices.sql*) para crear datos de prueba en caso de que se requiera para poder probar la aplicación en un entorno similar a uno real.

Esto no es todo, ya que en los objetivos del proyecto se especifica que la gestión de los datos debe hacerse con rigurosa seguridad, especialmente los datos sensibles. Para ello, todas las contraseñas de los usuarios se almacenan encriptadas.

El algoritmo usado se llama *bcrypt*, este permite aplicar una función hash a las contraseñas para que, en caso de sufrir un ataque malicioso, el atacante no pueda descifrar esas claves y así preservar la seguridad de los usuarios.

PRUEBAS LLEVADAS A CABO

Durante el desarrollo se ha hecho mucho hincapié en crear una aplicación fiable y de alta calidad, por lo tanto, se han hecho los esfuerzos necesarios para que, una vez se hubiese implementado una funcionalidad, esta fuese probada buscando posibles fallos.

A pesar de ser meticulosos durante el desarrollo, estos pequeños ensayos no son suficientes para demostrar que una aplicación es fiable. Por eso, una vez terminados todos los Sprints se han destinado numerosas horas de trabajo para probar todas las funcionalidades de manera exhaustiva.

Este proceso incluyó pruebas end-to-end [19] con el objetivo de comprobar que la aplicación funciona correctamente, de principio a fin. Esto se hizo simulando la interacción de un usuario real, detectando posibles problemas que se podrían dar durante el flujo general del sistema.

Casos de prueba

Prueba 1	Iniciar sesión	
Descripción	Probar que un usuario puede acceder a la página <i>Inicio de Sesión</i> e introducir un usuario y contraseña.	
PASOS		
Paso	Acción	Resultado
Acceder a la sección <i>Inicio de Sesión</i>	Se pulsa el botón <i>Iniciar sesión</i> de la pantalla principal.	Se muestra la página de inicio de sesión.
Introducir credenciales	Se introduce el usuario y contraseña correspondientes.	El usuario es validado y se redirige a la <i>Página Principal</i>
Conclusiones	La prueba ha sido exitosa.	

Tabla 14: Prueba de inicio de sesión

Prueba 2	Abrir barra lateral	
Descripción	Acceder a la barra lateral donde se encuentran las diferentes páginas a las que puede acceder un usuario.	
PASOS		
Paso	Acción	Resultado
Presionar botón de apertura	Se pulsa el botón que se encuentra situado arriba a la izquierda, una vez se ha iniciado sesión.	Se despliega la barra lateral donde se muestran las distintas páginas en función al tipo de usuario conectado.
Cierre de la barra lateral	Se pulsa en una parte externa a la barra lateral.	La barra lateral desaparece de la vista y vuelve a aparecer el botón de apertura.
Conclusiones	La prueba ha sido exitosa.	

Tabla 15: Prueba de la barra lateral

Prueba 3	Acceder a las notificaciones	
Descripción	Mostrar las notificaciones existentes relacionadas con el stock de sistema.	
PASOS		
Paso	Acción	Resultado
Presionar botón de apertura	Se pulsa el botón localizado arriba a la derecha, presente una vez se ha iniciado sesión.	Aparece una barra lateral donde se sitúa una notificación creada para realizar esta prueba.
Cierre de la sección de notificaciones	Se pulsa en la parte izquierda de la pantalla, fuera del área de la barra lateral.	La sección de notificaciones se oculta y aparece el botón de apertura.
Conclusiones	La prueba ha sido exitosa.	

Tabla 16: Prueba de barra de notificaciones

Prueba 4	Consulta y modificación de datos del propio usuario	
Descripción	Acceso a la información del usuario con el que se ha iniciado sesión y se prueba a modificar uno de los campos.	
PASOS		
Paso	Acción	Resultado
Acceso a la sección <i>Mi Perfil</i>	Se abre la barra lateral y se pulsa en la casilla llamada <i>Mi Perfil</i>	Se muestra una página que contiene una lista con todos los datos almacenados del usuario.
Habilitar modo edición	Se pulsa el botón situado en la parte superior de la lista.	Los campos que son modificables (excluyendo el rol) son habilitados para que el usuario pueda modificarlos.
Modificación de un campo	Se pulsa en el campo <i>apellidos</i> y se escribe el texto “Hernández”. Una vez hecho se pulsa el botón <i>Guardar cambios</i>	El campo queda registrado en el sistema.
Conclusiones	La prueba ha sido exitosa.	

Tabla 17: Prueba de gestión de datos propios

Prueba 5	Acceder a la gráfica con datos de compras y ventas	
Descripción	Cuando se visita la <i>Página Principal</i> se muestra una gráfica anual de los costes e ingresos por mes.	

PASOS		
Paso	Acción	Resultado
Acceso a la sección <i>Página Principal</i>	Se despliega la barra lateral y se pincha en la casilla llamada <i>Página Principal</i>	Se muestra una página que contiene una gráfica con los datos de 2023 referentes a costes e ingresos
Cambio de año	En el formulario desplegable superior se selecciona el año 2024	La gráfica cambia de valores mostrando los del año 2024, salvo aquellos meses en los que todavía no hay registros
Conclusiones	La prueba ha sido exitosa.	

Tabla 18: Prueba de representación gráfica de datos

Prueba 6	Gestionar usuarios	
Descripción	Validar que se puedan realizar operaciones de creación, modificación y borrado sobre los datos de los usuarios	
PASOS		
Paso	Acción	Resultado
Acceso a la sección <i>Gestión usuarios</i>	Se abre la barra lateral pulsando su botón y se pulsa en sobre <i>Gestión usuarios</i>	Se carga la página correspondiente a la gestión de usuarios y se muestra una tabla con botones para realizar las distintas operaciones.
Crear usuario	Se pulsa el botón situado en la parte superior de la tabla, se rellena el formulario emergente y se pulsa el botón <i>Crear usuario</i>	El usuario queda registrado y aparece un mensaje indicando que la acción ha sido exitosa.
Modificar usuario	Se busca la fila de la tabla correspondiente al usuario creado, se pulsa en el botón de editar y se cambia el campo <i>email</i>	El cambio queda registrado en el sistema y se muestra un mensaje con el mensaje “Usuario modificado”
Borrar usuario	Se vuelve a buscar la fila correspondiente al usuario modificado y se pulsa el botón de eliminar.	El usuario desaparece de la tabla y es borrado de la aplicación
Conclusiones	La prueba ha sido exitosa.	

Tabla 19: Prueba de gestión de usuarios

Prueba 7	Gestionar productos	
Descripción	Validar que se puedan realizar operaciones de creación, modificación y borrado sobre los datos de los productos	
PASOS		
Paso	Acción	Resultado
Acceso a la sección <i>Gestión productos</i>	Se abre la barra lateral pulsando su botón y se pulsa en sobre <i>Gestión productos</i>	Se carga la página correspondiente a la gestión de productos y se muestra una tabla con botones para realizar las distintas operaciones.
Crear producto	Se pulsa el botón situado en la parte superior de la tabla, se rellena el formulario emergente y se pulsa el botón <i>Crear producto</i>	El producto queda registrado y aparece un mensaje indicando que la acción ha sido exitosa.

Modificar producto	Se busca la fila de la tabla correspondiente al producto creado, se pulsa en el botón de editar y se cambia el campo <i>precio</i>	El cambio queda registrado en el sistema y se muestra un mensaje con el mensaje “Producto modificado”
Borrar producto	Se vuelve a buscar la fila correspondiente al producto modificado y se pulsa el botón de eliminar.	El producto desaparece de la tabla y es borrado de la aplicación
Conclusiones	La prueba ha sido exitosa.	

Tabla 20: Prueba de gestión de productos

Prueba 8	Gestionar compras	
Descripción	Validar que se puedan realizar operaciones de creación, modificación y borrado sobre los datos de las compras	
PASOS		
Paso	Acción	Resultado
Acceso a la sección <i>Gestión compras</i>	Se abre la barra lateral pulsando su botón y se pulsa en sobre <i>Gestión compras</i>	Se carga la página correspondiente a la gestión de compras y se muestra una tabla con botones para realizar las distintas operaciones.
Crear compra	Se pulsa el botón situado en la parte superior de la tabla, se rellena el formulario emergente y se pulsa el botón <i>Crear compra</i>	La compra queda registrada y aparece un mensaje indicando que la acción ha sido exitosa.
Modificar compra	Se busca la fila de la tabla correspondiente a la compra creada, se pulsa en el botón de editar y se cambia el campo <i>cantidad</i> del segundo producto introducido	El cambio queda registrado en el sistema y se muestra un mensaje con el mensaje “Compra modificada”
Borrar compra	Se vuelve a buscar la fila correspondiente a la compra modificada y se pulsa el botón de eliminar.	La compra desaparece de la tabla y es borrada de la aplicación
Conclusiones	La prueba ha sido exitosa.	

Tabla 21: Prueba de gestión de compras

Prueba 9	Gestionar ventas	
Descripción	Validar que se puedan realizar operaciones de creación, modificación y borrado sobre los datos de las ventas	
PASOS		
Paso	Acción	Resultado
Acceso a la sección <i>Gestión ventas</i>	Se abre la barra lateral pulsando su botón y se pulsa en sobre <i>Gestión ventas</i>	Se carga la página correspondiente a la gestión de ventas y se muestra una tabla con botones para realizar las distintas operaciones.
Crear venta	Se pulsa el botón situado en la parte superior de la tabla, se rellena el formulario emergente y se pulsa el botón <i>Crear venta</i>	La venta queda registrada y aparece un mensaje indicando que la acción ha sido exitosa.

Modificar venta	Se busca la fila de la tabla correspondiente a la venta creada, se pulsa en el botón de editar y se cambia el campo <i>precio</i> del primer producto introducido	El cambio queda registrado en el sistema y se muestra un mensaje con el mensaje “Venta modificada”
Borrar venta	Se vuelve a buscar la fila correspondiente a la venta modificada y se pulsa el botón de eliminar.	La venta desaparece de la tabla y es borrada de la aplicación
Conclusiones	La prueba ha sido exitosa.	

Tabla 22: Prueba de gestión de ventas

Prueba 10	Cerrar sesión	
Descripción	Comprobar que un usuario no puede realizar ninguna operación cuando ha cerrado sesión, en ningún caso podrá acceder hasta que no vuelva a iniciar sesión	
PASOS		
Paso	Acción	Resultado
Cerrar sesión	Se abre la barra lateral y se pulsa el botón <i>Cerrar sesión</i>	El usuario es redirigido a la página de <i>Inicio de sesión</i>
Intentar acceder a la sección <i>Gestión usuarios</i>	En la URL del navegador se sustituye la ruta <i>/login</i> por <i>/users</i>	El usuario es redirigido de vuelta a la página de <i>Inicio de sesión</i>
Conclusiones	La prueba ha sido exitosa.	

Tabla 23: Prueba de cierre de sesión

Prueba 11	Obtener PDF de una venta	
Descripción	Acceder a la carpeta donde se crean PDF's de todas las ventas creadas.	
PASOS		
Paso	Acción	Resultado
Acceder a la carpeta	Buscamos en la ruta <code>./backend/invoices</code> del proyecto.	Se puede observar como hay un PDF creado con los datos referentes a la venta creada en la Prueba 9
Conclusiones	La prueba ha sido exitosa.	

Tabla 24: Prueba de generación de PDF's

PRINCIPALES APORTACIONES

El desarrollo del gestor de stock ha cumplido con éxito los objetivos planteados, ofreciendo una herramienta robusta, segura y fácil de usar para la gestión remota de inventarios. A continuación, se detallan las principales aportaciones del trabajo realizado en relación con los objetivos marcados.

Para alcanzar el objetivo principal, se desarrolló una aplicación web utilizando React para el frontend, FastAPI para el backend y MySQL para la gestión de la base de datos. Esta combinación de tecnologías permite acceder al sistema desde cualquier dispositivo con conexión a Internet, ya sea una computadora de escritorio, una tableta o un teléfono móvil. La interfaz responsiva y el diseño modular facilitan la adaptación del sistema a las necesidades específicas de diferentes

empresas, permitiendo personalizar flujos de trabajo y funcionalidades según los requerimientos de cada usuario.

Teniendo en cuenta los objetivos específicos, se implementaron prácticas sólidas de gestión de datos utilizando MySQL, asegurando la integridad y consistencia de la información almacenada. Se diseñaron y optimizaron consultas SQL para garantizar operaciones rápidas y eficientes, reduciendo el tiempo de espera y mejorando la experiencia del usuario. La base de datos fue normalizada para evitar redundancias y asegurar la integridad referencial.

Interfaz amigable que permita acceder rápidamente a las funciones y herramientas necesarias para gestionar su inventario.

El frontend desarrollado con React proporciona una interfaz de usuario intuitiva y fácil de navegar. Se implementaron componentes visuales claros y coherentes, lo que permite a los usuarios gestionar su inventario de manera eficiente y sin complicaciones.

La seguridad fue una prioridad durante todo el desarrollo del proyecto. Se implementaron técnicas de cifrado para proteger los datos sensibles tanto durante la transmisión como en almacenamiento. Además, se diseñó un sistema de autenticación y autorización robusto utilizando tokens JWT para controlar el acceso a las diferentes partes del sistema.

Las contraseñas de los usuarios se almacenan utilizando un algoritmo de hash seguro (bcrypt), y se aplicaron prácticas de codificación segura para prevenir vulnerabilidades comunes como inyecciones SQL y ataques de Cross-Site Scripting (XSS).

Por lo tanto, el proyecto ha logrado cumplir con éxito todos los objetivos establecidos, resultando en un sistema de gestión de inventario robusto, seguro y accesible desde diferentes dispositivos. La aplicación no solo permite una gestión eficiente y segura de los datos, sino que también ofrece una experiencia de usuario fluida y amigable. Estas características hacen del gestor de stock una herramienta valiosa para empresas de diversos tamaños y sectores, facilitando la administración de inventarios de manera remota y eficaz.

CONCLUSIONES

Conclusiones Técnicas

El desarrollo de un gestor de stock utilizando tecnologías modernas como React, FastAPI y MySQL ha sido un proyecto desafiante que me ha permitido poner en práctica los conocimientos adquiridos a lo largo de estos cuatro años y explorar nuevas tecnologías y metodologías que son relevantes en el sector privado.

Durante el desarrollo de este proyecto, he podido aplicar de manera práctica muchos de los conceptos teóricos aprendidos durante mis estudios. Esto incluye desde la estructuración y normalización de bases de datos en MySQL hasta la creación de interfaces de usuario interactivas y responsivas con React. Además, la implementación de la lógica del negocio y la API en FastAPI me permitió reforzar mis habilidades en Python y en la gestión de servicios web.

Uno de los aspectos más importantes que he aprendido es la necesidad de planificar y pensar a largo plazo. Esto incluye la implementación de prácticas de código limpio, la utilización de patrones de diseño adecuados y la importancia de buscar en cada momento la solución óptima. Además, he aprendido a identificar y mitigar riesgos potenciales, como la gestión de dependencias y posibles fallos en la base de datos.

El desarrollo de software raramente sigue un camino recto, y este proyecto no fue la excepción. Durante el desarrollo, me encontré con varios problemas inesperados, desde errores de integración hasta problemas de rendimiento. Estos desafíos me enseñaron la importancia de la

resiliencia y la adaptabilidad. Aprendí a utilizar herramientas de debugging como las herramientas de desarrollo de Chrome, a buscar y aplicar soluciones en la documentación y comunidades en línea, y a iterar rápidamente sobre posibles soluciones.

Conclusiones Personales

Además del aprendizaje técnico, este proyecto ha sido fundamental para mi desarrollo personal y profesional. Trabajar de manera individual en el gestor de stock me ha permitido mejorar significativamente mis habilidades de autogestión y disciplina. La necesidad de planificar y llevar a cabo todas las etapas del proyecto de manera autónoma ha sido un gran reto que me ha enseñado la importancia de la organización y la gestión eficaz del tiempo.

A lo largo del proyecto, he aprendido a gestionar mi tiempo de manera eficiente, establecer prioridades y mantenerme enfocado en los objetivos a largo plazo. Estos aspectos no solo son cruciales para el desarrollo de software, sino también para cualquier desafío profesional que enfrente en el futuro. La experiencia adquirida en este proyecto me ha proporcionado una base sólida sobre la cual construir mi carrera en el sector tecnológico.

VÍAS DE TRABAJO FUTURO

El gestor de stock ha sido diseñado con una visión a largo plazo, asegurando su capacidad de adaptación y crecimiento, ha sido realizado teniendo en cuenta la escalabilidad y la flexibilidad, permitiendo así que futuras mejoras y expansiones sean fáciles de implementar.

Esto asegura que la aplicación pueda evolucionar conforme a las necesidades de los usuarios. Algunas de las funcionalidades adicionales que podrían incorporarse incluyen:

- Integración con Sistemas de Contabilidad: Añadir servicios para la gestión contable permitirá a las empresas llevar un control más detallado de sus finanzas. Esto incluye la automatización de facturas, gestión de pagos y cobros, y generación de informes financieros.
- Automatización de Procesos: Incorporar funcionalidades que automaticen procesos repetitivos, como la generación de órdenes de compra cuando el stock alcanza niveles bajos, pudiendo ahorrar tiempo y reducir errores humanos.

Digitalización de Más Áreas Empresariales

El gestor de stock puede expandirse para digitalizar más áreas dentro de una empresa, lo que puede mejorar la eficiencia y la productividad. Estas expansiones serían opcionales, permitiendo que las pequeñas y medianas empresas usen solo las funcionalidades que necesiten y, por lo tanto, paguen un precio más económico. Algunas posibles áreas de expansión incluyen:

- Gestión de Recursos Humanos: Implementar módulos para la gestión de empleados, seguimiento de horas trabajadas, y gestión de nóminas.
- Gestión de Clientes y Proveedores: Añadir herramientas para gestionar relaciones con clientes y proveedores, incluyendo CRM (Customer Relationship Management) y SRM (Supplier Relationship Management).

En resumen, el enfoque en la flexibilidad y la escalabilidad del gestor de stock asegura su relevancia y utilidad a largo plazo, permitiendo su evolución y adaptación continua en respuesta a las demandas cambiantes del mercado y las necesidades de los usuarios.

BIBLIOGRAFIA

- [1] Parviainen P., Tihinen M., Kääriäinen J. & Teppola S. (2022). *Tackling the digitalization challenge: how to benefit from digitalization in practice*. *International Journal of Information Systems and Project Management*, 5(1), 63–77. <https://doi.org/10.12821/ijispm050104>
- [2] Nilay Balkan. (2023, 27 diciembre). *Barriers to digitally transform SMEs in Europe*. European DIGITAL SME Alliance. <https://www.digitalsme.eu/barriers-to-digitally-transform-smes-in-europe/>
- [3] Arnesh Telukdarie, Thabile Dube, Simon Philbin, & Pretty Matjuta. (2022). *The opportunities and challenges of digitalization for SME's* [Article, University of Johannesburg & London South Bank University].
<https://www.sciencedirect.com/science/article/pii/S1877050922023432/pdf?md5=3250dbeb1dfdab5fac2e0c00649c645c&pid=1-s2.0-S1877050922023432-main.pdf>
- [4] Malkhasyan, G., & Savelyeva, O. (2023). Digitalization of Industry as Tool for Increasing Production Efficiency: Best Practices and New Solutions. En *Lecture notes in networks and systems* (pp. 2151-2157). https://doi.org/10.1007/978-3-031-21432-5_235
- [5] Kellermayr-Scheucher, M., Hörandner, L., & Brandtner, P. (2022). Digitalization at the Point-of-Sale in Grocery Retail - State of the Art of Smart Shelf Technology and Application Scenarios. *Procedia Computer Science*, 196, 77-84. <https://doi.org/10.1016/j.procs.2021.11.075>
- [6] *¿Cómo digitalizar el stock en el comercio pequeño y local?* (s. f.). Comercio AceleraPYME. <https://comercio.acelerapyme.gob.es/novedades-comercio/p/C3%ADIdora/%C2%BFc%C3%B3mo-digitalizar-el-stock-en-el-comercio-peque%C3%B1o-y-local%3F>
- [7] Meyer, J. (s. f.). *The Top 5 Reasons ERPs are Inefficient for Managing Resources*. <https://www.mosaicapp.com/post/the-top-5-reasons-erps-are-inefficient-for-managing-resources>
- [8] Aguirre, M. F. (2022, 21 diciembre). *Top 6 - Mejor software para control de inventarios en 2024*. appvizer.es. <https://www.appvizer.es/revista/organizacion-planificacion/gestion-de-stock/software-inventario>
- [9] Editor Dolibarr. (n.d.). *ERP & CRM Dolibarr - España y Latam - Información*. <https://www.dolibarr.es/index.php/erp-dolibarr>
- [10] Monstock. (n.d.). *Monstock - Programa informático de gestión de stocks y flujos*. Monstock. https://monstock.net/es_es
- [11] *El sistema de gestión de inventario de código abierto #1 | Odoo*. (n.d.). Odoo S.A. https://www.odoo.com/es_ES/app/inventory
- [12] MhEducation. (s. f.). *La gestión de stocks* [PDF]. <https://www.mheducation.es/bcv/guide/capitulo/8448612124.pdf>
- [13] Ken Schwaber & Jeff Sutherland. (2020). *The Scrum Guide. The definitive guide to scrum: the rules of the game*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

[14] Arrarte, A. (2023, September 4). Las 5 fases de la metodología Scrum para tus proyectos. *AlvaroArrarte.com*. <https://alvaroarrarte.com/fases-de-la-metodologia-scrum/>

[15] Atlassian. (n.d.). *¿Qué es scrum? [+ Cómo empezar] | Atlassian*. <https://www.atlassian.com/es/agile/scrum>

[16] Giménez, A. (n.d.). *Aplicaciones web con front-end y back-end desacoplado - Aunitz Giménez*. <https://www.aunitz.net/aplicaciones-web-front-end-back-end-desacoplado/>

[17] colaboradores de Wikipedia. (2024, May 21). *Paradigma de programación*. Wikipedia, La Enciclopedia Libre. https://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n

[18] *Diagrama de componentes. Teoría y ejemplos*. (2024, January 30). DiagramasUML.com. <https://diagramasuml.com/componentes/>

[19] Ise, M. (n.d.). *E2E Testing - Engineering Fundamentals Playbook*. <https://microsoft.github.io/code-with-engineering-playbook/automated-testing/e2e-testing/>

[20] *MySQL :: MySQL 8.4 Reference Manual :: 5 Tutorial*. (n.d.). <https://dev.mysql.com/doc/refman/8.4/en/tutorial.html>

ANEXO: MANUAL DE USUARIO

REQUISITOS MÍNIMOS DEL DISPOSITIVO	36
MANUAL DE INSTALACIÓN.....	36
MANUAL DE UTILIZACIÓN	37
Inicio de sesión.....	37
Acceso a gráficas	38
Acceso de la barra lateral.....	38
Acceso a las notificaciones.....	38
Perfil del usuario	39
Modificar datos del perfil.....	39
Gestión de usuarios.....	39
Crear usuario.....	40
Modificar usuario	40
Eliminar usuario.....	41
Gestión de compras	41
Crear compra.....	42
Modificar compra.....	42
Eliminar compra	42
Crear proveedor	42
Gestión de productos.....	43
Crear producto	43
Modificar producto	43
Eliminar producto.....	44
Gestión de ventas.....	44
Crear venta.....	44
Modificar venta	45
Eliminar venta	45
Crear cliente	45

Para facilitar el acceso al producto y poder usar todas sus funcionalidades, se detallará a continuación toda la información necesaria para disponer de esta aplicación web.

Este manual de usuario está enfocado para dos tipos de usuarios:

- Cientes: personas o empresas que quieren acceder al gestor de stock.
- Desarrolladores: personas que quieren hacer uso de la aplicación con fines didácticos o que desean tener su propia copia de la aplicación en su propio dispositivo.
 - o Si desean profundizar más, pueden consultar el fichero *README.md* que contiene el proyecto para tener unas instrucciones más específicas de como ejecutar la aplicación.

Requisitos mínimos del dispositivo

Los requisitos mínimos que debe tener su dispositivo para poder disfrutar del gestor de stock son:

- Cientes: disponer de un navegador web (Google Chrome, Safari, Microsoft Edge, Firefox, ...) con acceso a Internet.
- Desarrolladores: el dispositivo donde se ejecute la aplicación debe tener:
 - o CPU con, al menos, 2 núcleos.
 - o 4 GB de RAM.
 - o 2 GB de almacenamiento libre.
 - o Python con la versión 3.12 o superior
 - o MySQL con la versión 5.7
 - o NPM con la versión 9.8

Manual de instalación

En primer lugar, es necesario tener MySQL ejecutándose en segundo plano, para ello puede consultar la documentación oficial [\[20\]](#). Para crear las tablas que conforman el gestor de stock, hay que ejecutar los scripts que se encuentran en la carpeta *db* del proyecto, en este orden:

- *database.sql*
- *example_data_buys.sql*
- *example_data_invoices.sql*

Con estos archivos SQL se crean las tablas necesarias y se les añaden datos para facilitar su prueba y simular un entorno real.

A continuación ejecutaremos la API, para ello accederemos mediante terminal a la carpeta *backend* del proyecto y ejecutaremos el siguiente comando¹:

```
pip install -r requirements.txt
```

Este comando permite instalar las dependencias necesarias de Python. Por último, ejecutamos el servidor:

```
uvicorn main:app --reload
```

¹ Los comandos que se especifican están enfocados para el sistema operativo Windows, pueden variar si se usa Linux o MacOS.

Una vez hecho todo esto debemos tener la base de datos y el backend conectado. Ahora es momento de iniciar el frontend, para ello accedemos a la carpeta *frontend* del proyecto y ejecutamos:

```
npm run dev
```

Esto iniciará un servidor local, situado en la URL <http://localhost:5173> que será accesible desde el navegador.

Manual de utilización

En este apartado se detallarán las distintas funcionalidades de la aplicación web con la que el usuario podrá interactuar.



Ilustración 8: Página de bienvenida de la web

Esta es la página de bienvenida a la aplicación, en ella hay un botón de inicio de sesión para aquellas personas que tengan un usuario creado en la plataforma.

El formulario de inicio de sesión tiene el título 'INICIO SESIÓN' en azul. Contiene dos campos de entrada: 'Nombre de usuario' y 'Contraseña' (con un icono de ojo para alternar visibilidad). Debajo de los campos hay un botón 'INICIAR SESIÓN'.

Ilustración 9: Formulario de inicio de sesión

Cuando pulsamos el botón se mostrará una página que cuenta con un formulario para introducir sus credenciales. En caso de introducirlos correctamente, se redirigirá a la siguiente página.

Acceso a gráficas

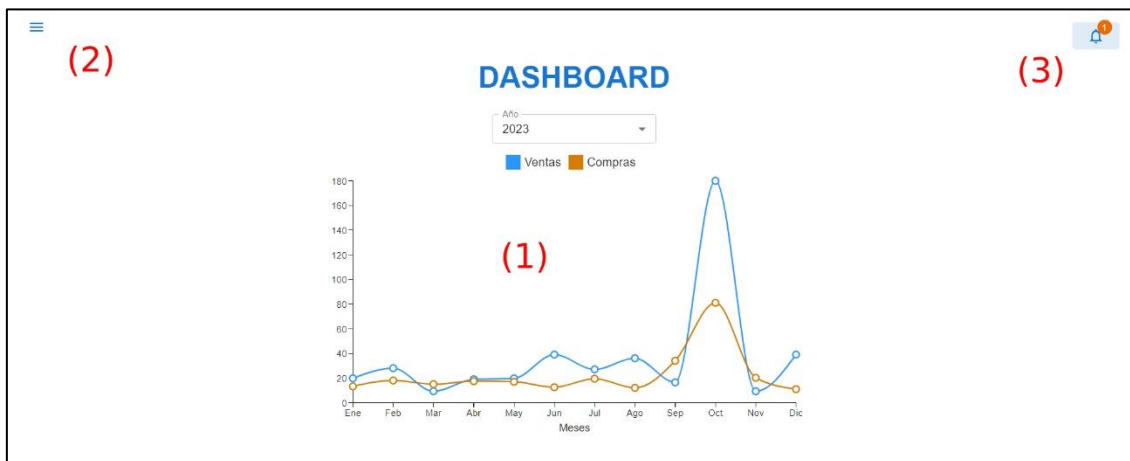


Ilustración 10: Página principal de la web

Una vez iniciada la sesión, se mostrará la página principal. En esta página se muestra una gráfica donde se muestran los datos mensuales de costes e ingresos (1) del año seleccionado. En caso de querer acceder a esta página desde otra diferente, se tendrá que acceder a la barra lateral y pulsar la opción *Página principal*.

Acceso de la barra lateral

Existen 2 botones, que estarán presentes en las todas las páginas de la aplicación. El primero de ellos (2) está relacionado con la barra lateral, cuando se pulsa se muestra lo siguiente:



Ilustración 11: Lista de elementos de la barra lateral

Esto es una lista con las distintas páginas a las que puede acceder el usuario, además de la opción para cerrar sesión.

Acceso a las notificaciones

En el segundo botón (3) de la página se muestra el botón que abre la sección de notificaciones:

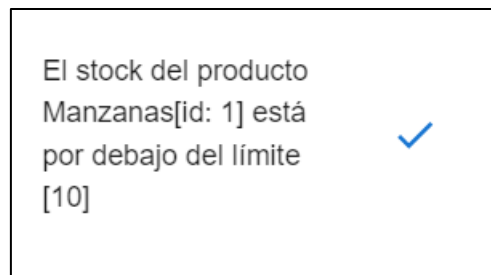


Ilustración 12: Ejemplo de notificación

En ella aparecerán notificaciones como la imagen anterior, que permite marcar como leída pulsando el botón derecho.

Perfil del usuario

En caso de querer acceder o modificar la información del perfil, se debe pulsar en la opción *Mi perfil* de la barra lateral.

A form titled 'MI PERFIL' in large blue letters. Below the title is a small icon of a pencil inside a square. The form contains several input fields: 'Nombre de usuario' with the value 'admin', 'Contraseña' with an eye icon, 'Nombre' with the value 'Admin', 'Apellidos' with the value 'Admin', 'DNI' with the value '12345678A', 'Email' with the value 'admin@gmail.com', and 'Rol' with the value 'ADMIN'. At the bottom is a blue button with the text 'GUARDAR CAMBIOS' and a right-pointing arrow.

Ilustración 13: Formulario para modificar datos de usuario

Una vez se accede a la página, como se puede observar en la imagen anterior, los datos del usuario aparecen en el formulario (salvo la contraseña por seguridad)

Modificar datos del perfil

Si se desea cambiar alguno de los datos del usuario, tan solo tendrá que modificarlos y pulsar el botón *Guardar cambios*.

Gestión de usuarios

Para acceder a la página de gestión de usuarios, es necesario pulsar la opción *Gestión usuarios* que aparece en la barra lateral.

Una vez pulsado, se mostrará la página mostrada a continuación:



Ilustración 14: Página de gestión de usuarios

En esta página aparece en la parte central una tabla donde cada fila representa un usuario (en este caso hay uno).

Crear usuario

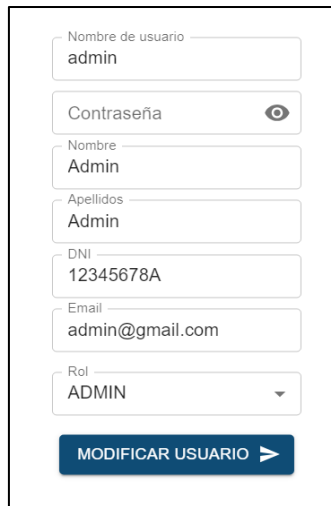
En caso de querer crear un usuario se pulsará el botón (4), el cual mostrará un formulario como el siguiente:

Ilustración 15: Formulario de creación de usuarios

Este formulario deberá cubrirse con la información correspondiente del usuario y, finalmente, pulsar el botón *Crear usuario*.

Modificar usuario

Para modificar un usuario, se pulsará el botón (5), que hará visible el siguiente formulario:



Formulario de modificación de usuarios. Campos: Nombre de usuario (admin), Contraseña (con icono de ojo), Nombre (Admin), Apellidos (Admin), DNI (12345678A), Email (admin@gmail.com), Rol (ADMIN). Botón: MODIFICAR USUARIO ➤.

Ilustración 16: Formulario de modificación de usuarios

En este formulario aparecerán los datos del usuario correspondiente a la fila seleccionada. Podrá modificarse la información que se desee y posteriormente pulsar el botón *Modificar usuario*.

Eliminar usuario

En caso de querer borrar un usuario, tan solo hay que pulsar el botón (6). Esto borrará el usuario perteneciente a la fila seleccionada.

Gestión de compras

Para acceder a esta funcionalidad, es necesario acceder a la barra lateral y pulsar la opción *Gestión compras*. Una vez hecho esto se mostrará la siguiente página:



Página de gestión de compras. Encabezado: COMPRAS. Botones: (7) +, (10) +. Tabla principal:

Compras (orden por fecha)	Número de productos	Proveedor	Gestor de compra	Total de compra	Acciones
14/5/2024, 17:05:55	5	Frutas y Verduras S.A.	admin	18.75 €	(8) (9)
Lineas de compra					
Producto	Cantidad	Coste	Coste total		
Manzana	3 kg	1 €/kg	3 €		
Pera	2 kg	0.75 €/kg	1.5 €		
Plátano	3 kg	1.5 €/kg	4.5 €		
Naranja	3 kg	1.25 €/kg	3.75 €		
Uva	3 kg	2 €/kg	6 €		
11/4/2024, 17:05:59	3	Drogueria S.L.	admin	32 €	(8) (9)
8/3/2024, 17:05:58	6	Panaderia S.L.	admin	19.5 €	(8) (9)

Ilustración 17: Página de gestión de compras

En ella se puede observar una tabla como elemento principal, cuyas filas son desplegadas. Cada fila muestra información general sobre la compra realizada y si se extiende, aparecerán los detalles de esa compra (productos, cantidad, coste y coste total).

Crear compra

Para crear una compra basta con pulsar el botón (7), esto hará visible el siguiente formulario donde se cubrirán los datos de la compra. Una vez hecho esto, hay que pulsar el botón *Crear compra*.



Formulario de creación de compras. Incluye un menú desplegable para 'Proveedor', un campo para 'Producto', un campo para 'Cantidad', y un campo para 'Coste unitario' con un icono de error rojo. Hay dos botones: 'AÑADIR PRODUCTO' (azul) y 'CREAR COMPRA' (oscuro azul).

Ilustración 18: Formulario de creación de compras

Modificar compra

Si se quiere modificar la información de una compra, se debe seleccionar la fila que le corresponda pulsar el botón (8). Cuando se pulse este botón aparecerá en pantalla un formulario con la información modificable de la compra.



Formulario de modificación de compras. Muestra una lista de productos con sus cantidades y costes unitarios. El proveedor seleccionado es 'Frutas y Verduras S.A.'. Los productos listados son: Manzana (3 unidades, coste 1), Pera (2 unidades, coste 0,75), Plátano (3 unidades, coste 1,5), Naranja (3 unidades, coste 1,25) y Uva (3 unidades, coste 2). Hay un botón 'MODIFICAR COMPRA' (oscuro azul).

Ilustración 19: Formulario de modificación de compras

Eliminar compra

Para eliminar una compra basta con pulsar el botón (9) de la fila correspondiente a la compra que se quiere borrar.

Crear proveedor

Como elemento adicional a la página de *Gestión compras*, está el botón (10), el cual mostrará un formulario para poder dar de alta nuevos proveedores en el sistema. Esto se hará completando el siguiente formulario:



Formulario de registro de proveedores. Incluye campos para 'Nombre' y 'Teléfono', y un botón 'CREAR PROVEEDOR' (azul).

Ilustración 20: Formulario de registro de proveedores

Gestión de productos

En caso de querer administrar la información de los productos de la aplicación, habrá que pulsar en la opción *Gestión productos* presente en la barra lateral. Después, se mostrará la siguiente página:

Nombre	Categoría	Precio de compra	Precio de venta	Existencias	Límite de alerta	Acciones
Manzana	Fruta y verdura	1 €	2 €	100 kg	10 kg	(12) (13)
Pera	Fruta y verdura	0.75 €	1.5 €	75 kg	7 kg	
Plátano	Fruta y verdura	1.5 €	3 €	50 kg	5 kg	
Naranja	Fruta y verdura	1.25 €	2.5 €	125 kg	12 kg	
Uva	Fruta y verdura	2 €	4 €	150 kg	15 kg	
Fresa	Fruta y verdura	2.5 €	5 €	100 kg	10 kg	

Ilustración 21: Página de gestión de productos


Crear producto

Si se desea crear un producto se tiene que pulsar el botón (11) y completar el formulario mostrado:

Ilustración 22: Formulario de creación de productos

Modificar producto

En caso de querer editar los datos de un producto, se tendrá que pulsar el botón (12) y modificar los datos que se desee del formulario:



Formulario de modificación de productos. Campos:

- Nombre del producto: Manzana
- Descripción: Manzanas de todo tipo
- Coste: 1
- Precio: 2
- Existencias: 100
- Límite de alerta: 10
- Unidades: kg
- Categoría: Fruta y verdura
- Botón: MODIFICAR PRODUCTO ➤

Ilustración 23: Formulario de modificación de productos

Eliminar producto

Si se desea eliminar, basta con pulsar el botón (13) de la fila del producto correspondiente.

Gestión de ventas

Si se desea gestionar las ventas de la empresa se accederá a la siguiente página pulsando la opción *Gestión ventas* de la barra lateral. Esto permitirá acceder a una página similar a esta:



VENTAS

(14) + (17)

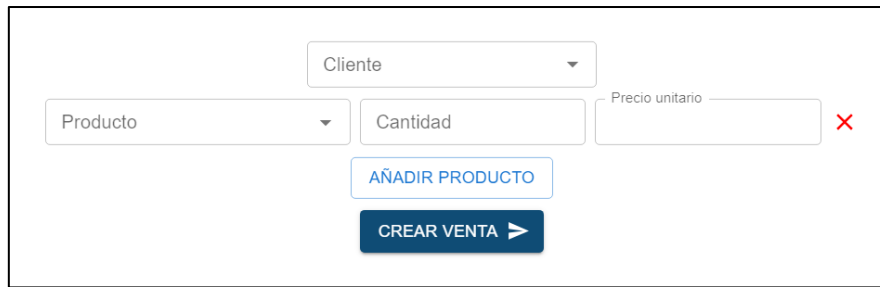
Ventas (orden por fecha)	Número de productos	Cliente	Gestor de venta	Total de venta	Acciones
6/5/2024, 4:10:40	3	Mercedes	admin	108 €	(15) (16)
Lineas de venta					
Producto	Cantidad	Precio	Coste total		
Ambientador	3 ud	16 €/ud	48 €		
Insecticida	2 ud	18 €/ud	36 €		
Fregasuelos	1 ud	24 €/ud	24 €		
5/4/2024, 13:24:00	3	Antonio	admin	16.5 €	
4/3/2024, 13:12:10	3	Pedro Perez	admin	36 €	

Ilustración 24: Página de gestión de ventas

La interfaz contiene una tabla para acceder a todas las ventas, mostrando tanto los detalles más generales (fecha, cliente, empleado que la registró, ...) como los más específicos (productos, cantidades, precios, ...).

Crear venta

Para crear una venta, es necesario pulsar los botones (14). A continuación se tendrá que completar el siguiente formulario:

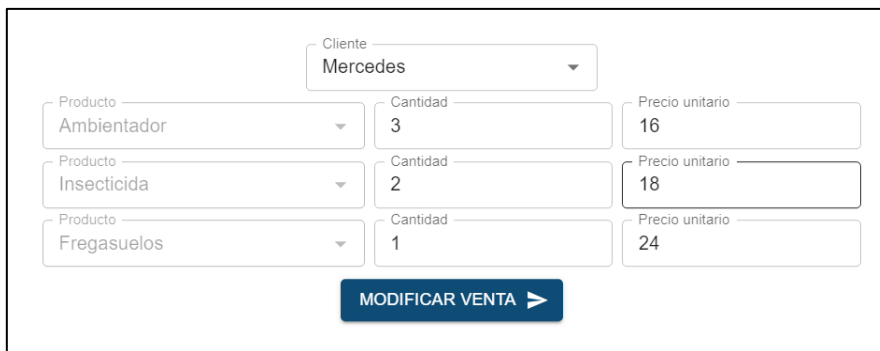


Formulario de creación de ventas. Incluye un campo desplegable para 'Cliente', un campo desplegable para 'Producto', un campo de texto para 'Cantidad', y un campo de texto para 'Precio unitario' con un icono de error rojo. Hay dos botones: 'AÑADIR PRODUCTO' y 'CREAR VENTA >'.

Ilustración 25: Formulario de creación de ventas

Modificar venta

En caso de querer modificar una venta existente, se tendrá que pulsar el botón (15) y editar los campos que se deseen.



Formulario de modificación de ventas. Incluye un campo desplegable para 'Cliente' con el valor 'Mercedes'. Hay tres filas de campos para 'Producto', 'Cantidad' y 'Precio unitario'. Los valores predefinidos son: Ambientador (3, 16), Insecticida (2, 18) y Fregasuelos (1, 24). Hay un botón 'MODIFICAR VENTA >'.

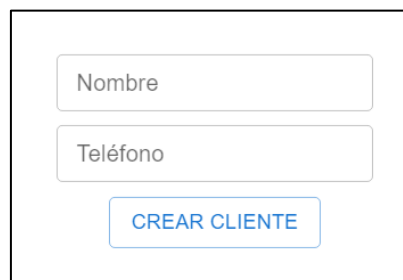
Ilustración 26: Formulario de modificación de ventas

Eliminar venta

En caso de querer eliminar una venta, tendrá que pulsarse el botón (16) correspondiente a la venta que se quiera borrar.

Crear cliente

Existe un botón (17) que permite registrar nuevos clientes mediante el siguiente formulario:



Formulario de registro de clientes. Incluye dos campos de texto: 'Nombre' y 'Teléfono'. Hay un botón 'CREAR CLIENTE'.

Ilustración 27: Formulario de registro de clientes