



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN SISTEMAS DE INFORMACIÓN

Aplicación web para la gestión de grandes almacenes

Estudiante: Pablo Trillo Valeiro

Dirección: Luis Andrés González Ares

A Coruña, setembro de 2019.

A mis padres y a mi abuelo

Agradecimientos

En primer lugar, agradecer a Luis Andrés González Ares su infinita paciencia, su confianza y todo el apoyo y ayuda que me ha proporcionado para poder realizar este TFG.

Agradecer también a mis padres su sacrificio, gracias a ellos ha sido posible estudiar lo que quería y gracias a su apoyo y confianza he logrado llegar a este punto.

A mis compañeros de facultad a lo largo de estos años, por todas las veces que pensamos que no llegaríamos al final pero cogíamos fuerzas para seguir adelante, a mis compañeros de trabajo Emilio y Miguel por su inestimable ayuda y a mis compañeros de piso por los buenos momentos.

A mi novia por todo su apoyo y paciencia.

Muchas gracias a todos.

Resumen

En la actualidad, toda empresa que pretenda ser competitiva en el mercado debe necesariamente adoptar tecnologías que le ayuden a conseguir más ventas y a controlar sus procesos internos realizando el menor esfuerzo posible. Actualmente es muy complicado encontrar empresas, por pequeñas que sean, que no tengan informatizados sus sistemas, puesto que el gasto en tiempo y dinero que supone la gestión manual de los citados procesos provocaría una clara desventaja frente a su competencia.

Este proyecto se presenta como un programa de gestión para pequeñas y medianas empresas que necesiten una herramienta para gestionar sus empleados, productos y ventas. La mayoría de softwares comerciales de este tipo están diseñados para grandes empresas y esto implica que su coste sea demasiado elevado para que una PYME pueda afrontarlo.

En concreto, el proyecto se enfoca en la gestión de unos grandes almacenes: una empresa que cuenta con uno o varios centros de venta, con varias plantas en cada centro y a su vez con varias secciones en cada planta.

El proyecto proporcionará las herramientas necesarias para gestionar al personal de la empresa, desde el responsable del centro hasta el vendedor, pasando por los responsables de planta y sección, los posibles cambios de puesto que se realicen y las ventas que registren. Permitirá también gestionar los distintos centros, sus plantas y sus secciones, y los productos que se venden en cada uno de estos lugares. Por último, proveerá de una herramienta de análisis de ventas que permitirá a los responsables tener información detallada y actualizada sobre el desempeño de sus empleados.

Es importante destacar que este proyecto se llevará a cabo en forma de aplicación web, lo que permitirá el acceso a los usuarios desde cualquier dispositivo que tenga un navegador de internet.

Abstract

Nowerdays, every company that wants to be competitive in the market must necessarily adopt technologies that help it achieve more sales and control its internal processes by making the least possible effort. It is unthinkable that today there are companies, however small, that do not have their systems computerized, since the expense in time and money that would be the manual management of the aforementioned processes would cause a clear disadvantage compared to their competition.

This project is presented as a management program for small and medium enterprises that need a tool to manage their employees, products and sales. Most commercial software of this

type are designed for large companies and this implies that its cost is too high for a SME to face.

Specifically, the project focuses on the management of department stores: a company that has one or several sales centers, with several plants in each center and in turn with several sections in each plant.

The project must provide the necessary tools to manage the company's staff, from the head of the center to the seller, through those responsible for plant and section, the possible changes of position that are made and the sales that register. It will also allow to manage the different centers, their plants and sections, and the products that are sold in each of these places. Finally, it will provide a sales analysis tool that will allow those responsible to have detailed and updated information on the performance of their employees.

It is important to note that this project will be carried out in the form of a web application, which will allow access to users from any device that has an internet browser.

Palabras clave:

- Gestión empleados
- Análisis ventas
- Aplicación web
- Java
- Spring
- Bootstrap

Keywords:

- Employee management
- Sales analysis
- Web application
- Java
- Spring
- Bootstrap

Índice general

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Organización de la memoria	2
2	Estado del arte	5
2.1	Alternativas	5
2.1.1	SAP Business One	5
2.1.2	Namely HR	5
2.1.3	Simple HRM	7
2.2	Conclusiones	7
3	Fundamentos tecnológicos	9
3.1	Tecnologías y herramientas de la parte servidor	9
3.2	Tecnologías y herramientas de la parte web	10
3.3	Otras herramientas utilizadas	10
4	Metodología	11
4.1	Proceso Unificado de Desarrollo de Software	11
5	Planificación y costes	13
5.1	Planificación	13
5.2	Coste	14
5.2.1	Coste del material	14
5.2.2	Coste de los recursos humanos	14
6	Análisis de requisitos	19
6.1	Actores	19
6.2	Casos de Uso	20

6.2.1	Gestión de empleados	20
6.2.2	Gestión de Lugares	20
6.2.3	Gestión de Productos	22
6.2.4	Análisis de Ventas	24
6.3	Casos de uso de cada actor	25
6.3.1	Casos de uso del rol Administrador	25
6.3.2	Casos de uso del rol Jefe de Centro	25
6.3.3	Casos de uso del rol Jefe de Planta	26
6.3.4	Casos de uso del rol Jefe de Sección	26
6.3.5	Casos de uso del rol Vendedor	26
7	Diseño	31
7.1	Diseño de los datos	31
7.2	Diseño de la arquitectura del sistema	32
8	Implementación	37
8.1	Implementación del servidor	37
8.1.1	El patrón DAO	37
8.1.2	El patrón DTO	40
8.1.3	El patrón Facade	40
8.2	Implementación del cliente	48
9	Pruebas	65
9.1	Pruebas unitarias	65
9.2	Pruebas de aceptación	66
9.3	Pruebas de integración	66
9.4	Otras pruebas	67
10	Conclusiones	69
11	Líneas futuras	71
	Bibliografía	73

Índice de figuras

2.1	Captura de la página principal de SAP B1 en distintos dispositivos	6
2.2	Captura de la página principal de Namely HR	6
2.3	Captura de la página principal de Simple HRM	7
4.1	Proceso Unificado de Desarrollo de Software	12
5.1	Diagrama de Gantt del proyecto	15
5.2	Diagrama de Gantt correspondiente a la iteración 1	16
5.3	Diagrama de Gantt correspondiente a la iteración 2	17
5.4	Diagrama de Gantt correspondiente a la iteración 3	17
6.1	Jerarquía de actores	20
6.2	Casos de uso pertenecientes al bloque Gestión de Empleados	21
6.3	Casos de uso pertenecientes al bloque Gestión de Lugares	22
6.4	Casos de uso pertenecientes al bloque Gestión de Productos	23
6.5	Casos de uso pertenecientes al bloque Análisis de Ventas	24
6.6	Casos de uso de los bloques Gestión Empleados y Gestión Lugares del rol Administrador	25
6.7	Casos de uso de los bloques Gestión Productos y Análisis Ventas del rol Administrador	27
6.8	Casos de uso de los bloques Gestión Empleados y Gestión Lugares del rol Jefe de Centro	27
6.9	Casos de uso de los bloques Gestión Productos y Análisis Ventas del rol Jefe de Centro	28
6.10	Casos de uso de los bloques Gestión Empleados y Gestión Lugares del rol Jefe de Planta	28
6.11	Casos de uso de los bloques Gestión Planta y Análisis Ventas del rol Jefe de Planta	29

6.12	Casos de uso del rol Jefe de Sección	29
6.13	Casos de uso del rol Vendedor	30
7.1	Diagrama Entidad-Relación	33
7.2	Funcionamiento del patrón MVC	34
7.3	Arquitectura del sistema	35
8.1	Diagrama de paquetes de la parte servidor	37
8.2	Diagrama de clases del paquete "entities"	39
8.3	Diagrama de clases del paquete "dto"	41
8.4	Diagrama de clases de EmpleadoService	44
8.5	Diagrama de clases de LugarService	45
8.6	Diagrama de clases de ProductoService	46
8.7	Diagrama de clases de VentaService	47
8.8	Captura de pantalla donde se muestran las gráficas de Ránking de ventas Em- pleados	50
8.9	Captura de pantalla donde se muestran las gráficas de Análisis de ventas por Centro	51
8.10	Captura de pantalla donde se muestran las gráficas de Análisis de ventas por Planta	52
8.11	Captura de pantalla donde se muestran las gráficas de Análisis de ventas por Seccion	53
8.12	Captura de pantalla donde se muestra la pestaña Gestión de Empleados del usuario "pablo@app.com"	54
8.13	Captura de pantalla donde se muestra la pestaña Gestión de Empleados del usuario "sergio@app.com"	55
8.14	Captura de pantalla donde se muestra la pestaña Gestión de Lugares del usua- rio "pablo@app.com"	56
8.15	Captura de pantalla donde se muestra la pestaña Gestión de Lugares del usua- rio "sergio@app.com"	57
8.16	Captura de pantalla donde se muestra la pestaña Gestión de Productos del usuario "pablo@app.com"	58
8.17	Captura de pantalla donde se muestra la pestaña Gestión de Productos del usuario "sergio@app.com"	59
8.18	Captura de pantalla donde se muestra la pestaña Gestión de Productos del usuario "pedro@app.com"	60
8.19	Captura de pantalla donde se muestra la pestaña Registrar Venta del usuario "miguel@app.com"	61

ÍNDICE DE FIGURAS

8.20	Captura de pantalla donde se muestra la aplicación en una tablet	62
8.21	Captura de pantalla donde se muestra la aplicación en un smartphone	63
9.1	Captura de pantalla de una petición GET a la url /productos	66

Índice de tablas

5.1	Detalle del coste por tareas del analista-programador.	14
5.2	Detalle del coste por tareas del jefe de proyecto.	17
5.3	Coste total del proyecto.	18
6.1	Descripción de los casos de uso de Gestión de Empleados.	21
6.2	Descripción de los casos de uso de Gestión de Lugares.	22
6.3	Descripción de los casos de uso de Gestión de Lugares.	23
6.4	Descripción de los casos de uso de Análisis de Ventas.	24

Introducción

EN el primer capítulo de esta memoria se explicará de forma detallada la motivación de este proyecto y los objetivos a alcanzar, así como la organización de la misma.

1.1 Motivación

La informatización de los procesos internos de gestión de las empresas es un factor determinante en el éxito de cualquier compañía. No se concibe que a día de hoy existan grandes empresas que no utilicen sofisticados programas de gestión para controlar sus recursos, pero no ocurre lo mismo con las empresas más pequeñas, debido a que no pueden asumir el coste que les supondría pagar las licencias de dichos programas. Es por esto que deben recurrir a aplicaciones más sencillas que no siempre cubren la totalidad de funcionalidades necesarias.

Este proyecto nace con el objetivo de cubrir esa demanda: pequeñas y medianas empresas dedicadas al sector del comercio que aspiran a expandirse y convertirse en unos grandes almacenes, y que no poseen un software para la gestión de sus recursos o que disponen de distintas aplicaciones que no terminan de cubrir todas sus necesidades. Es por ello que la aplicación desarrollada constará de varios módulos o apartados que cubran las necesidades referentes a la gestión de sus empleados y sus sucursales, de los productos que venda y que a su vez proporcione análisis de las ventas que realizan sus empleados.

Un punto importante de este proyecto es que se trata de una aplicación web, con un servidor centralizado donde se almacena toda la información a la que acceden los distintos usuarios, por lo que sería posible la gestión de múltiples sucursales en distintas localizaciones dado que todas estarían accediendo a los mismos datos, cosa que evita los errores y dificultades técnicas de tener los datos de cada sucursal físicamente en la misma.

En definitiva, el proyecto se presenta como una alternativa viable a pequeñas y medianas empresas de venta al por menor que pretendan crecer y expandirse hasta convertirse en unos grandes almacenes.

1.2 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación web que permita la gestión de los procesos internos de una empresa, de manera sencilla e intuitiva, relativos a:

- **Gestión de Empleados:** este módulo proporcionará las funcionalidades para la gestión de los usuarios, los responsables de cada lugar de trabajo tendrán en su mano las herramientas necesarias para gestionar a sus subordinados.
- **Gestión de Lugares:** este módulo permitirá a los responsables de cada lugar tener el control de los lugares subordinados al suyo.
- **Gestión de Productos:** en este módulo están encuadradas las funcionalidades relativas a la gestión de productos por parte de los usuarios.
- **Análisis de Ventas:** este módulo proporcionará a los usuarios información acerca de las ventas que se produzcan en los distintos lugares relativos a su nivel de jerarquía.
- **Registro de Ventas:** este módulo proporcionará a los usuarios la capacidad de registrar ventas en el sistema.

1.3 Organización de la memoria

El presente documento se ha dividido en los siguientes capítulos:

- **Capítulo 1: Introducción**
Se introduce la motivación del proyecto y los objetivos a cumplir al finalizar su desarrollo.
- **Capítulo 2: Estado del arte**
Se valoran las distintas alternativas existentes en el mercado a este proyecto.
- **Capítulo 3: Fundamentos tecnológicos**
Se indican las tecnologías que se han utilizado a lo largo del desarrollo del proyecto.
- **Capítulo 4: Metodología**
Se explica la metodología seguida para llevar a cabo este proyecto.
- **Capítulo 5: Planificación y costes**
Se indica la planificación del proyecto y el coste del mismo.

- **Capítulo 6: Análisis de requisitos**
Se describen las diferentes funcionalidades que permite la aplicación y se detallan los actores y sus casos de uso.
- **Capítulo 7: Diseño**
Se describe el diseño de los datos y de la arquitectura del sistema.
- **Capítulo 8: Implementación**
Se explica cómo se desarrolló el proyecto.
- **Capítulo 9: Pruebas**
Se indican las distintas pruebas realizadas.
- **Capítulo 10: Conclusiones**
Se explican las conclusiones obtenidas al finalizar el proyecto.
- **Capítulo 11: Líneas futuras**
Se indican las líneas futuras de desarrollo.

Estado del arte

EN este capítulo se destacarán algunas de las aplicaciones existentes hoy en día en el mercado que guardan relación con el proyecto llevado a cabo.

2.1 Alternativas

2.1.1 SAP Business One

SAP Business One es un ERP distribuido por la compañía alemana SAP enfocado a pequeñas y medianas empresas.

La multinacional SAP es líder en el mercado de software para empresas, pero la mayoría de sus productos están enfocados a grandes corporaciones. El coste de sus productos provocaba que las PYMEs no pudieran afrontar el gasto que suponía la adopción de sus productos, por eso decidieron lanzar SAP B1, herramienta enfocada a ese nicho de mercado.

Entre las funcionalidades que incluyen destacan: Gestión de contabilidad, gestión de ventas y clientes, gestión de compras e inventario y analíticas e informes.

En la figura 2.1 podemos ver la pantalla principal de SAP Business One

2.1.2 Namely HR

Namely HR es una aplicación de pago desarrollada por la compañía Namely.

Esta aplicación integra varias funcionalidades, las cuales permiten la gestión de los empleados, nóminas, gestión de talento y una plataforma de noticias internas de la empresa, que permite a los empleados estar al tanto de la actualidad corporativa.

En la figura 2.2 podemos ver la pantalla principal de Namely HR.



Figura 2.1: Captura de la página principal de SAP B1 en distintos dispositivos

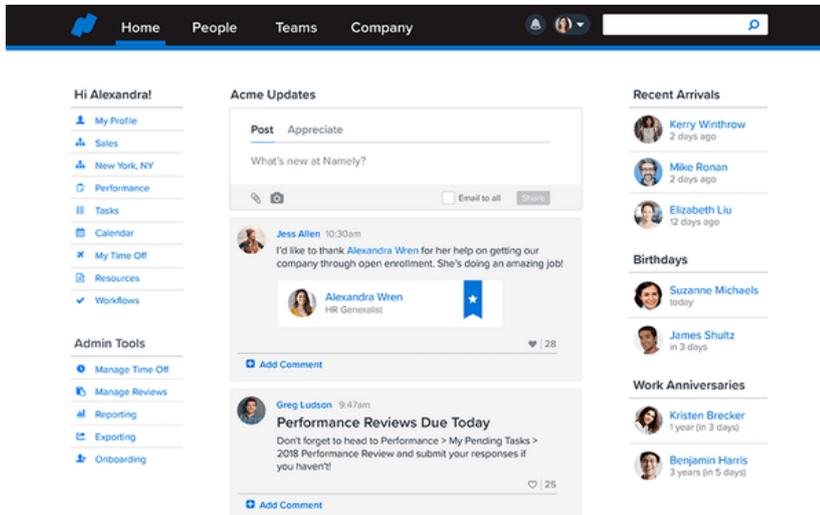


Figura 2.2: Captura de la página principal de Namely HR

2.1.3 Simple HRM

Esta aplicación de código abierto permite gestionar la mayoría de los procesos relativos a la administración de los empleados de una empresa: vacaciones, salarios, beneficios, etc.

Esta herramienta destaca por ser intuitiva, fácil de usar y gratuita. Esto permite que sea posible implantarla en pequeñas y medianas empresas que no pueden afrontar un gran gasto en un software de este tipo.

En la figura 2.3 podemos ver la pantalla principal de Simple HRM.

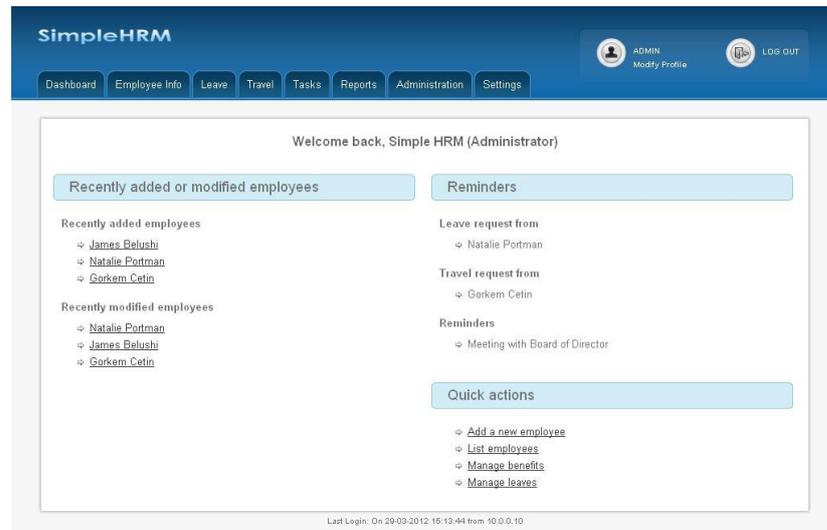


Figura 2.3: Captura de la página principal de Simple HRM

2.2 Conclusiones

Tras analizar las principales alternativas existentes en el mercado a este proyecto, se llega a las siguientes conclusiones:

- Las dos primeras herramientas (SAP B1 y Namely HR) son aplicaciones de pago con un coste elevado: en el caso de SAP B1 el coste ronda los 90\$ por usuario al mes, y el de Namely HR 20\$ al mes. Cabe destacar que la herramienta de SAP es bastante mas completa que la de Namely, de ahí la diferencia.
- SAP B1 es una herramienta muy completa, demasiado para lo que se necesita, puesto que incluye multitud de funcionalidades que puede que no interesen a la empresa y que aumentan su coste. El hecho de que sea una herramienta compleja dificulta también el aprendizaje de los usuarios.

- Namely HR es una buena aplicación para la gestión de los recursos humanos, pero no provee de un módulo para la gestión de stock y ventas ni tampoco para la gestión de lugares de trabajo. Esto hace que no cubra nuestras necesidades.
- Simple HRM destaca por ser gratuita, pero esto también tiene sus defectos: un simple vistazo a la interfaz de la aplicación nos indica que se trata de una herramienta demasiado simple y anticuada; esto sumado a que no incluye las funcionalidades de análisis de ventas y gestión de productos y lugares nos reafirma en la decisión de descartarla.

Una vez comprobado que no existen alternativas viables en el mercado, se decide la realización de este proyecto para conseguir:

- Una interfaz simple e intuitiva con la que los usuarios puedan familiarizarse desde el primer minuto.
- Permitir la gestión de empleados, lugares y productos en una única aplicación, sin necesidad de instalación de módulos de pago.
- Que la aplicación sea web, para poder acceder a ella desde cualquier lugar y equipo.

Fundamentos tecnológicos

EN este capítulo se indican las distintas tecnologías y herramientas que se han utilizado para desarrollar este proyecto. La principal razón de la elección de estas tecnologías es que son de código abierto y gratuitas, pero aportan prestaciones destacadas.

3.1 Tecnologías y herramientas de la parte servidor

- **MySQL:** sistema de Gestión de Bases de Datos Relacionales (SGBDR) más popular del mundo, utilizado por aplicaciones web muy importantes tales como Facebook y Twitter. Entre sus principales características destacamos su alto rendimiento y su seguridad.[1]
- **Java:** lenguaje de programación más utilizado en el mundo, esto provoca que haya una enorme cantidad de librerías y frameworks disponibles que nos facilitan el desarrollo. Podemos citar que grandes aplicaciones como Twitter y Netflix utilizan el lenguaje JAVA en su parte servidor.[2]
- **Spring:** framework diseñado para facilitar el desarrollo de aplicaciones Java. [3]
- **Spring Boot:** herramienta de Spring que simplifica las tareas de configuración de aplicaciones web. [4]
- **Spring Data:** herramienta de Spring que simplifica las tareas de acceso a datos. [5]
- **JPA:** hramework que nos permite establecer una correlación entre una base de datos relacional (MySQL) y un sistema orientado a objetos (Java). [5]
- **Maven:** herramienta para la gestión de dependencias de un proyecto Java.
- **Spring Security:** framework de seguridad que gestiona la autenticación y la autorización. [5]

3.2 Tecnologías y herramientas de la parte web

- **JavaScript:** lenguaje de programación utilizado en clientes web. [6]
- **HTML5:** lenguaje de marcado utilizado para el desarrollo de páginas web. [7]
- **CSS:** lenguaje de diseño gráfico que se utiliza para dar estilo a una página web. [7]
- **jQuery:** librería de JavaScript que permite simplificar la manera de interactuar con los documentos HTML. [7]
- **Thymeleaf:** motor de plantillas que renderiza documentos HTML desde el servidor. [8]
- **Bootstrap:** librería de JavaScript que facilita la maquetación y diseño de sitios web. [9]
- **Datatables:** plugin jQuery que permite crear y manipular tablas.[10]
- **ChartJs:** librería JavaScript que proporciona herramientas para la visualización de datos. [11]

3.3 Otras herramientas utilizadas

- **Eclipse:** IDE de código abierto y multiplataforma para desarrollar aplicaciones.
- **DBVisualizer:** herramienta multiplataforma utilizada para gestionar bases de datos relacionales. Se trata de una herramienta de pago, pero tiene utilidades disponibles gratuitamente.
- **Postman:** herramienta que permite la creación de peticiones REST a APIs.
- **JUnit:** framework utilizado para testear proyectos Java. [12]
- **LaTeX:** sistema de composición de textos orientado a la creación de documentos escritos.
- **Dia:** herramienta que permite crear diagramas.
- **WhiteStar UML:** herramienta que permite crear diagramas. [13]
- **SmartSheet:** herramienta utilizada para crear los diagramas de Gantt.

Metodología

Para la realización de este proyecto se ha utilizado el Proceso Unificado de Desarrollo de Software (PUDS) [14]. En este capítulo se explicará brevemente en que consiste.

4.1 Proceso Unificado de Desarrollo de Software

El desarrollo de una aplicación informática es un desafío complejo que conlleva grandes riesgos, es por ello que debe de realizarse siguiendo una metodología que minimice la posibilidad de cometer errores.

El Proceso Unificado de Desarrollo de Software (PUDS) es una de las metodologías más utilizadas en desarrollo de software, se caracteriza por estar dirigido por los casos de uso, estar centrado en la arquitectura y ser iterativo e incremental:

- **Dirigido por casos de uso:** Los casos de uso son las distintas acciones que los usuarios pueden llevar a cabo en el sistema. El hecho de dirigir el desarrollo de la aplicación por esas acciones implica que los requisitos funcionales y las iteraciones estén basados en dichos casos de uso.
- **Centrado en la arquitectura:** La arquitectura del sistema es una parte crítica del mismo que puede acarrear unos costes enormes si hay errores en su concepción. Es por ello que debe de establecerse de manera que se minimicen los riesgos.
- **Iterativo e incremental:** El PUDS consta de cuatro fases: **inicio, elaboración, construcción y transición**. Cada una de ellas dividida en una serie de interacciones compuestas de las subfases: **análisis de requisitos, diseño, implementación y pruebas**. El resultado de cada iteración es un incremento de las funcionalidades del sistema. En la figura 4.1 podemos apreciar las distintas fases y sus subfases con la carga de trabajo de cada una.

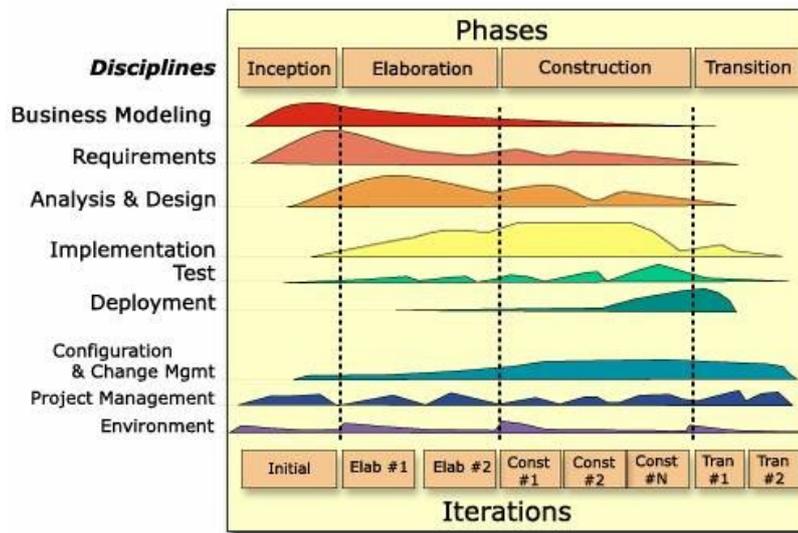


Figura 4.1: Proceso Unificado de Desarrollo de Software

Planificación y costes

EN este capítulo de la memoria se detalla el proceso seguido para la realización de este proyecto y el coste del mismo.

5.1 Planificación

La primera fase del proyecto consistió en un análisis previo de requisitos en la que se definieron los distintos módulos o apartados de los que consta el proyecto: un módulo que engloba las funcionalidades relativas a la **gestión de empleados**, otro para las funcionalidades relativas a la **gestión de lugares**, un tercer apartado para la **gestión de productos** y un último que recoge las funcionalidades de **análisis de ventas**.

A continuación se procedió a la elección de las herramientas y tecnologías a utilizar para el desarrollo de la aplicación. Una vez seleccionadas se procedió a realizar un estudio más detallado y una fase de aprendizaje de aquellas que eran desconocidas para el desarrollador.

Una vez definidos los distintos módulos y analizadas las herramientas a utilizar, se procedió a la planificación de tres iteraciones. En la primera de ellas se implementaron las funcionalidades recogidas en los apartados de gestión de empleados y gestión de lugares, se tomó la decisión de realizarlas en la misma iteración debido a la fuerte dependencia existente entre ellas. En esta primera iteración también se llevó a cabo el diseño de la arquitectura del sistema. En la segunda iteración se desarrollaron las funcionalidades relativas al módulo de gestión de productos. La tercera y última de las iteraciones abarca las funcionalidades del módulo de análisis de ventas. La documentación del proyecto se realizó al mismo tiempo que se desarrollaban las distintas iteraciones.

En la figura 5.1 puede verse el diagrama de Gantt general del proyecto, y en las figuras 5.2, 5.3 y 5.4 los diagramas correspondientes a cada iteración.

Debido a que la realización de este proyecto se compaginó con un trabajo a tiempo completo de 8h, la dedicación diaria será de 3h.

A pesar de que la realización de este proyecto se haya dividido en tres iteraciones, los capítulos de esta memoria "Análisis de requisitos", "Diseño", "Desarrollo" y "Pruebas" agrupan las funcionalidades de las tres iteraciones para una mayor claridad a la hora de leerla.

5.2 Coste

Para calcular el coste de este proyecto se tienen en cuenta dos aspectos, el coste del material empleado y el coste de los recursos humanos.

5.2.1 Coste del material

El coste del material empleado en el proyecto se reduce al valor del equipo informático en el que se ha desarrollado la aplicación, dado que se ha utilizado software libre para la realización del mismo. La única herramienta de pago utilizada ha sido SmartSheet, para la elaboración de los diagramas de Gantt, que cuenta con un período de prueba de 30 días, por lo que no ha sido necesario pagar. El equipo donde se ha desarrollado es un ordenador portátil HP Pavillion con Windows 8.1, procesador Intel Core i7 y 12 GB de memoria RAM, cuyo precio es de 800 euros.

5.2.2 Coste de los recursos humanos

En la realización del proyecto han intervenido dos recursos humanos: el alumno, ejerciendo el rol de analista-programador, y el tutor, con un rol de jefe de proyecto. Para elaborar la estimación de coste del proyecto se asume que el salario de un jefe de proyecto es de 50€/hora y el de un analista-programador es de 30€/hora. Como ya se indicó en el apartado anterior, la jornada laboral es de 3 horas, en los días que coinciden el desarrollo de las iteraciones con la elaboración de la documentación, se ha destinado 1h a la documentación y 2 al desarrollo. En la tabla 5.1 se indica el desglose el coste de las tareas realizadas por el analista mientras que en la tabla 5.2 se muestra el coste de las tareas realizadas por el jefe de proyecto.

Tarea	Días	Horas	Coste
Análisis previo de requisitos	8	24	720€
Elección y estudio de herramientas	13	39	1170€
Iteración 1	54	116	3480€
Iteración 2	16	32	960€
Iteración 3	21	42	1260€
Documentación 91 107	3210€		

Tabla 5.1: Detalle del coste por tareas del analista-programador.

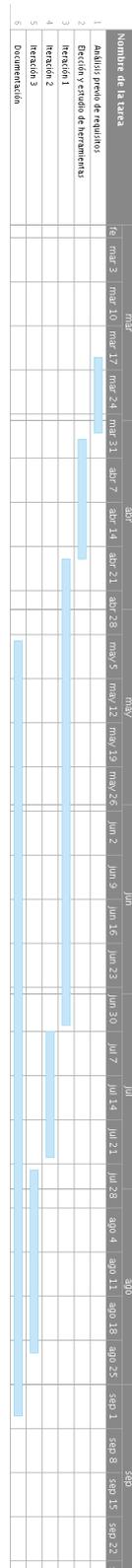


Figura 5.1: Diagrama de Gantt del proyecto

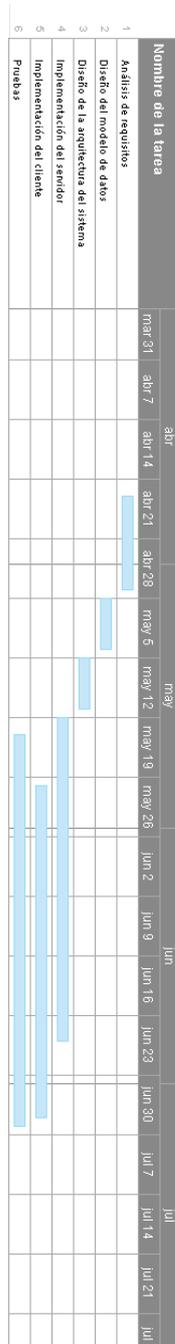


Figura 5.2: Diagrama de Gantt correspondiente a la iteración 1

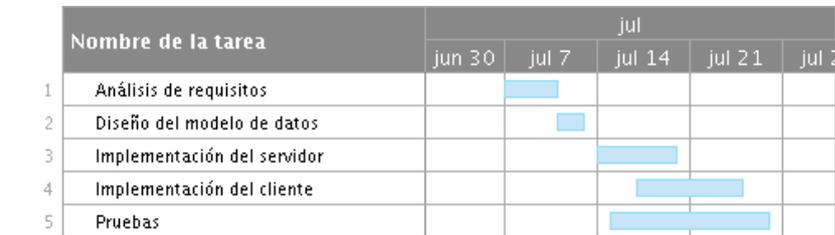


Figura 5.3: Diagrama de Gantt correspondiente a la iteración 2



Figura 5.4: Diagrama de Gantt correspondiente a la iteración 3

Tarea	Días	Horas	Coste
Análisis previo de requisitos	3	9	450€
Elección y estudio de herramientas	1	3	150€
Iteración 1	1	3	150€
Iteración 2	1	3	150€
Iteración 3	1	3	150€

Tabla 5.2: Detalle del coste por tareas del jefe de proyecto.

El coste total del proyecto, sumando el coste del material y el de los recursos humanos se indica en la tabla 5.3

Recurso	Coste
Material	800€
Analista-programador	10800€
Jefe de proyecto	1050€
Coste total	12650€

Tabla 5.3: Coste total del proyecto.

Análisis de requisitos

EN este capítulo se detallarán los requisitos que debe satisfacer la aplicación. Para ello se realizará un estudio de las funcionalidades necesarias y se identificarán los casos de uso que podrán llevar a cabo los distintos actores.

6.1 Actores

Los actores representan los distintos roles que pueden adoptar los usuarios al interactuar con el sistema, en nuestro caso un usuario solo podrá tener un rol activo a la vez, pero su rol puede cambiar a lo largo del tiempo. A continuación se describen los distintos tipos de actores:

- **Administrador:** Se trata del rol con mayor número de funcionalidades en el sistema, ya que tiene control sobre todos los lugares gestionados y todos los empleados asignados a esos lugares.
- **Jefe de Centro:** Representa el rol de responsable de un centro, sus funcionalidades están restringidas a lo relativo al centro que tenga asignado, a las plantas y secciones que dependen de dicho centro y a los empleados asignados a estos lugares.
- **Jefe de Planta:** Representa el rol de responsable de una planta, similar al Jefe de Centro pero con funcionalidades restringidas a su planta asignada y las secciones que dependen de la misma, así como a los empleados asignados a las mencionadas secciones.
- **Jefe de Sección:** Representa el rol de responsable de una sección, similar al Jefe de Centro y al Jefe de Planta, pero con funcionalidades restringidas a su sección asignada y a los vendedores de esa sección.
- **Vendedor:** Representa el rol de vendedor, sus funcionalidades se reducen a registrar ventas.

En la figura 6.1 se muestra la jerarquía existente entre los distintos actores.

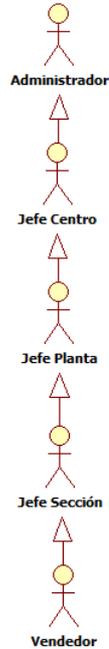


Figura 6.1: Jerarquía de actores

6.2 Casos de Uso

Los casos de uso definen las distintas acciones que podrán llevar a cabo los usuarios, en este proyecto se han agrupado en cuatro bloques que se detallan a continuación:

6.2.1 Gestión de empleados

En este apartado se agrupan las acciones referentes a la gestión de empleados. Las funcionalidades que tendrá disponibles un usuario estarán siempre limitadas a los distintos empleados que se encuentren bajo su mando. En la tabla 6.1 se muestra una descripción resumida de cada caso de uso, mientras que en la figura 6.2 podemos ver el diagrama que refleja los citados casos.

6.2.2 Gestión de Lugares

En este bloque se recogen las funcionalidades relativas a la gestión de los lugares por parte de los empleados. Al igual que en el caso de la gestión de empleados, un usuario solo podrá ejecutar acciones sobre los lugares que se encuentren bajo su escala de mando. En la tabla

Caso de Uso	Descripción
Ver Empleados Subordinados	El usuario registrado obtiene la lista de empleados que tiene a su mando.
Alta Jefe Centro	Se realiza el alta de un nuevo empleado con el rol "Jefe Centro" y se le asocia un centro.
Alta Jefe Planta	Se realiza el alta de un nuevo empleado con el rol "Jefe Planta" y se le asocia una planta.
Alta Jefe Sección	Se realiza el alta de un nuevo empleado con el rol "Jefe Sección" y se le asocia una sección.
Alta Vendedor	Se realiza el alta de un nuevo empleado con el rol "Vendedor" y se le asocia una sección.
Baja Empleado	Se da de baja un empleado
Cambio Puesto Empleado	Se cambia el rol del empleado, por lo que se le asigna un nuevo rol y un nuevo lugar asociado

Tabla 6.1: Descripción de los casos de uso de Gestión de Empleados.



Figura 6.2: Casos de uso pertenecientes al bloque Gestión de Empleados

6.2 se muestra una descripción resumida de cada caso de uso, mientras que en la figura 6.3 podemos ver el diagrama que refleja los citados casos.

Caso de Uso	Descripción
Ver Lugares Subordinados	El usuario registrado obtiene la lista de lugares que tiene bajo su mando.
Alta Centro	Se realiza el alta de un nuevo centro.
Alta Planta	Se realiza el alta de una nueva planta, asociada a un centro existente.
Alta Sección	Se realiza el alta de una nueva sección, asociada a una planta.
Baja Centro	Se da de baja un centro, lo que provoca la baja de las plantas y secciones asociadas a dicho centro y de los empleados asociados a esos lugares.
Baja Planta	Se da de baja una planta, lo que provoca la baja de las secciones asociadas a dicha planta y de los empleados asociados a esos lugares.
Baja Sección	Se da de baja una sección y a los empleados asociados a dicha sección.

Tabla 6.2: Descripción de los casos de uso de Gestión de Lugares.



Figura 6.3: Casos de uso pertenecientes al bloque Gestión de Lugares

6.2.3 Gestión de Productos

Este apartado reúne las funcionalidades asociadas a la gestión de productos. En la tabla 6.3 se describe brevemente cada caso de uso y en la figura 6.4 se muestra el diagrama correspondiente a los citados casos.

Caso de Uso	Descripción
Alta Producto	Se da de alta un nuevo producto en el sistema.
Baja Producto	Se da de baja un producto existene en el sistema, lo que provoca que desaparezca de todas las secciones en las que estuviera dado de alta.
Cambio Precio Producto	Se realiza un cambio de precio a un producto registrado en el sistema.
Ver Productos disponibles en Sistema	Se obtiene un listado de todos los productos disponibles en el sistema.
Registrar Entrada Stock en Sección	Se registra la entrada de stock de un determinado producto en una sección concreta.
Ver productos/stock en mi lugar asociado	Obtiene un listado de los productos y del stock asociados al lugar del usuario registrado.
Registrar Venta	Se registra una venta de n productos previamente registrados en una sección.

Tabla 6.3: Descripción de los casos de uso de Gestión de Lugares.



Figura 6.4: Casos de uso pertenecientes al bloque Gestión de Productos

6.2.4 Análisis de Ventas

Este bloque difiere ligeramente de los anteriores puesto que entre sus funcionalidades no se incluye ninguna relativa a la gestión. Se trata de un módulo que proporciona a los usuarios información relativa a las ventas que se producen en el sistema. El nivel de dicha información dependerá del rol que posea el usuario: un Jefe de centro podrá consultar las ventas relativas tanto a los centros como a las plantas y las secciones, pero en cambio un usuario de menor nivel como puede ser un Jefe de sección solo podrá acceder a las ventas de las secciones. En la tabla 6.4 se describe brevemente cada caso de uso y en la figura 6.5 se muestra el diagrama correspondiente a dichos casos.

Caso de Uso	Descripción
Consultar Ranking top 3 empleados	Se obtiene dos gráficas que muestran el importe de las ventas y el número de artículos vendidos de los tres empleados que más hayan vendido.
Consultar ventas de centros	Se obtienen dos gráficas que muestran las ventas en importe y el número de artículos vendidos de todos los centros.
Consultar ventas de plantas por importe	Se obtienen dos gráficas que muestran las ventas en importe y el número de artículos vendidos de todas las plantas.
Consultar ventas de secciones por importe	Se obtienen dos gráficas que muestran las ventas en importe y el número de artículos vendidos de todas las secciones.

Tabla 6.4: Descripción de los casos de uso de Análisis de Ventas.



Figura 6.5: Casos de uso pertenecientes al bloque Análisis de Ventas

6.3 Casos de uso de cada actor

Como se indica en el primer apartado de este capítulo, una de las particularidades de este proyecto radica en la existencia de múltiples roles organizados jerárquicamente. A continuación se detalla que casos de uso tiene asignados cada actor.

6.3.1 Casos de uso del rol Administrador

El rol “Administrador” tiene disponibles la mayoría de funcionalidades existentes, en los apartados de gestión de empleados, gestión de lugares y análisis de ventas podrá realizar cualquiera de las acciones disponibles en el sistema. Por el contrario, en el módulo de gestión de productos, tendrá restringido el acceso a “Registrar Entrada Stock en Sección” y “Ver productos/stock en mi lugar asociado”. En las figuras 6.6 y 6.7 podemos apreciar las relaciones entre este rol y los casos de uso que tiene a su disposición.

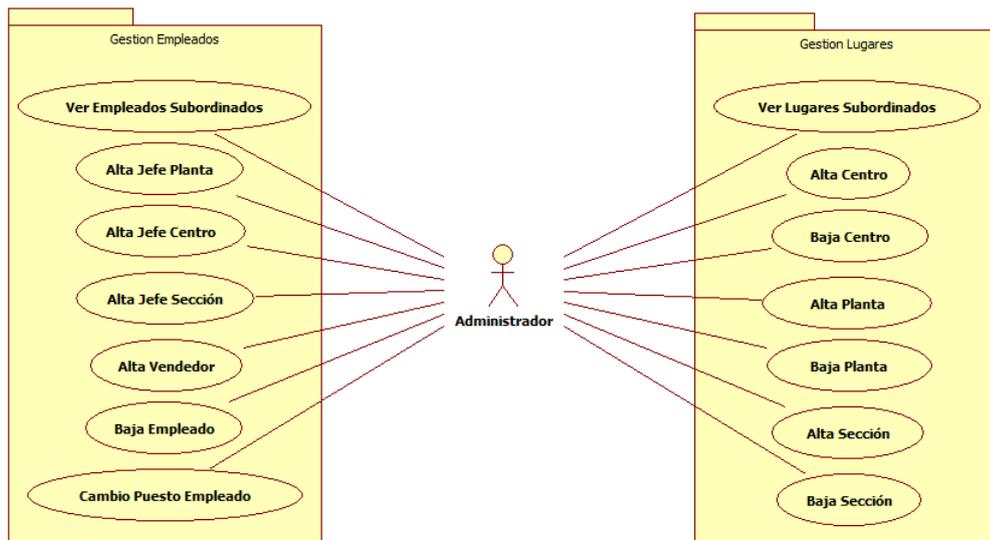


Figura 6.6: Casos de uso de los bloques Gestión Empleados y Gestión Lugares del rol Administrador

6.3.2 Casos de uso del rol Jefe de Centro

El rol “Jefe de centro” tendrá restringidas las funcionalidades de los módulos gestión empleados y gestión lugares a los empleados y lugares subordinados a él. En cuanto a los casos de uso del apartado de gestión de productos, “Registrar Entrada Stock en Sección” y “Ver productos/stock en mi lugar asociado”, también estarán acotados a los lugares bajo su mando. Al igual que el administrador, podrá ver información de las ventas de todos los lugares junto con

el ránking de empleados. En las figuras 6.8 y 6.9 podemos apreciar las relaciones entre este rol y los casos de uso que tiene a su disposición.

6.3.3 Casos de uso del rol Jefe de Planta

El rol “Jefe de planta” verá sus casos de uso reducidos con respecto al jefe de centro. Las funcionalidades de los apartados de gestión de empleados y lugares estarán restringidos a los lugares y empleados que se encuentren bajo su jurisdicción, al igual que los casos de uso “Registrar Entrada Stock en Sección” y “Ver productos/stock en mi lugar asociado” del módulo de gestión de productos. A diferencia del administrador y del jefe de centro, el jefe de planta no podrá consultar las ventas de los centros. En las figuras 6.10 y 6.11 podemos apreciar las relaciones entre este rol y los casos de uso que tiene a su disposición.

6.3.4 Casos de uso del rol Jefe de Sección

El rol “Jefe de sección” no tiene a su disposición ninguna funcionalidad del módulo de gestión de lugares. En lo relativo a los casos de uso del módulo de gestión de empleados, verá sus acciones acotadas a gestionar los vendedores que tenga bajo su mando. En cuanto a los casos de uso del apartado de gestión de productos, “Registrar Entrada Stock en Sección” y “Ver productos/stock en mi lugar asociado”, estarán acotados a la sección en la que se encuentre. Este rol solo podrá obtener información de las ventas de las secciones y del ránking de los tres mejores empleados. En la figura 6.12 podemos apreciar las relaciones entre este rol y los casos de uso que tiene a su disposición.

6.3.5 Casos de uso del rol Vendedor

El rol “Vendedor” es el que menos caso de uso puede realizar. Sus funcionalidades están reducidas a los apartados de gestión de productos y análisis de ventas, en el primer de ellos cabe destacar el caso de uso “Registrar Venta”, puesto que es el único rol con capacidad para realizarlo. En el módulo de análisis de ventas comparte las dos acciones posibles del jefe de sección: consultar las ventas por secciones y el ránking de tres mejores empleados. En la figura 6.13 se indican los distintos casos de uso disponibles para este rol.

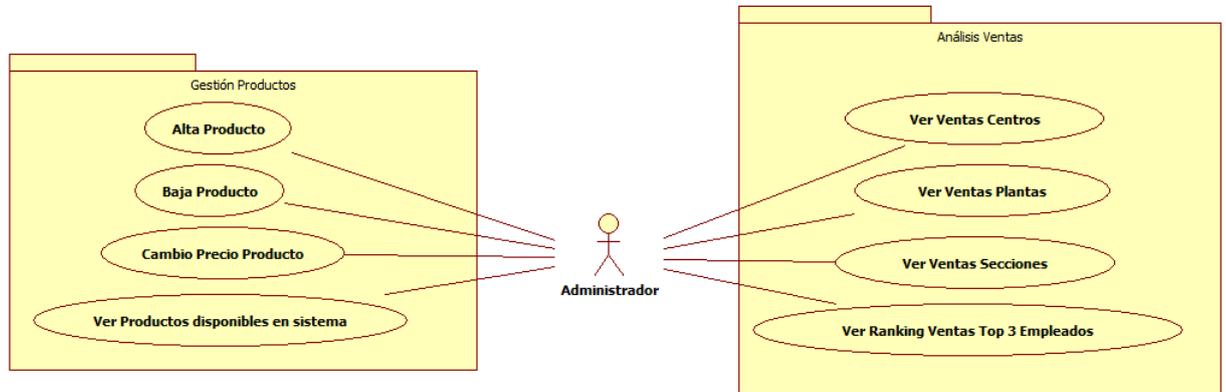


Figura 6.7: Casos de uso de los bloques Gestión Productos y Análisis Ventas del rol Administrador

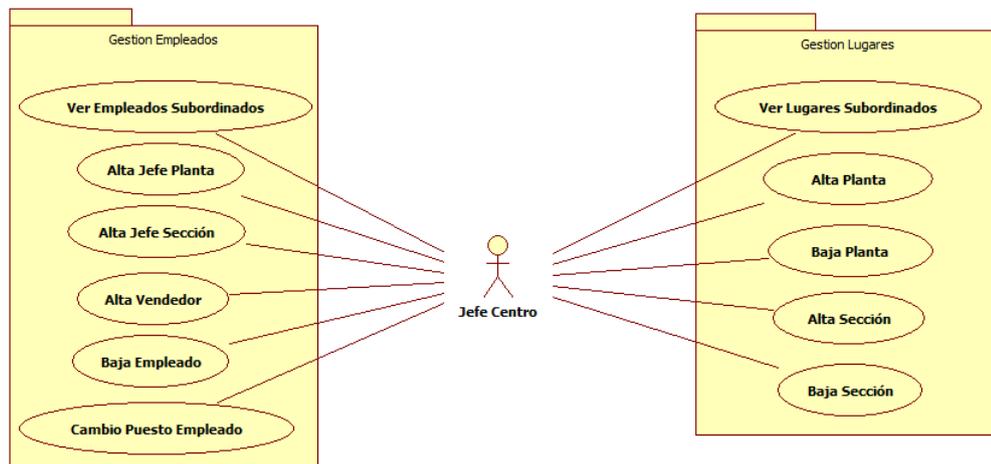


Figura 6.8: Casos de uso de los bloques Gestión Empleados y Gestión Lugares del rol Jefe de Centro

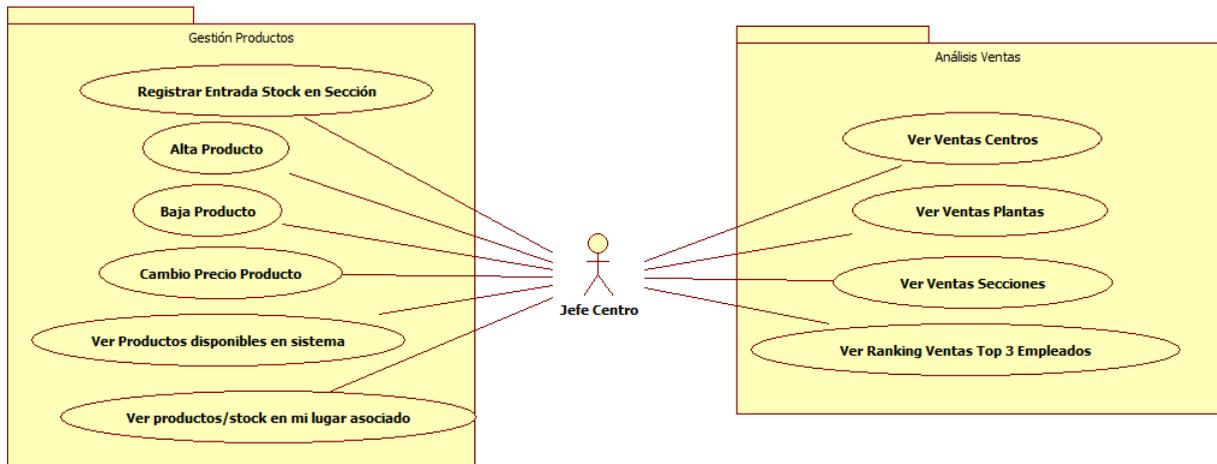


Figura 6.9: Casos de uso de los bloques Gestión Productos y Análisis Ventas del rol Jefe de Centro

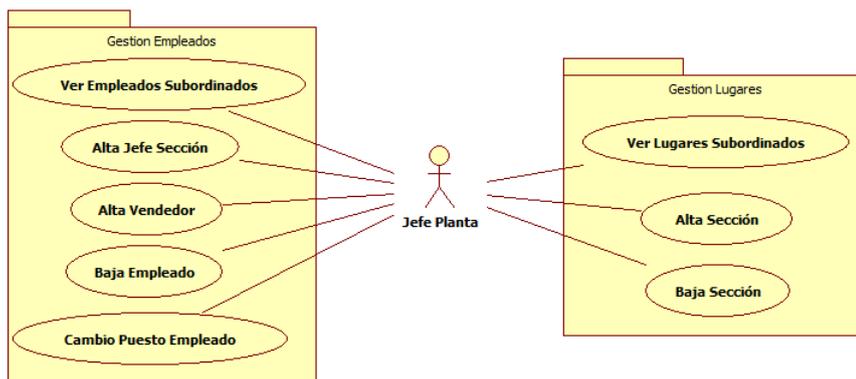


Figura 6.10: Casos de uso de los bloques Gestión Empleados y Gestión Lugares del rol Jefe de Planta

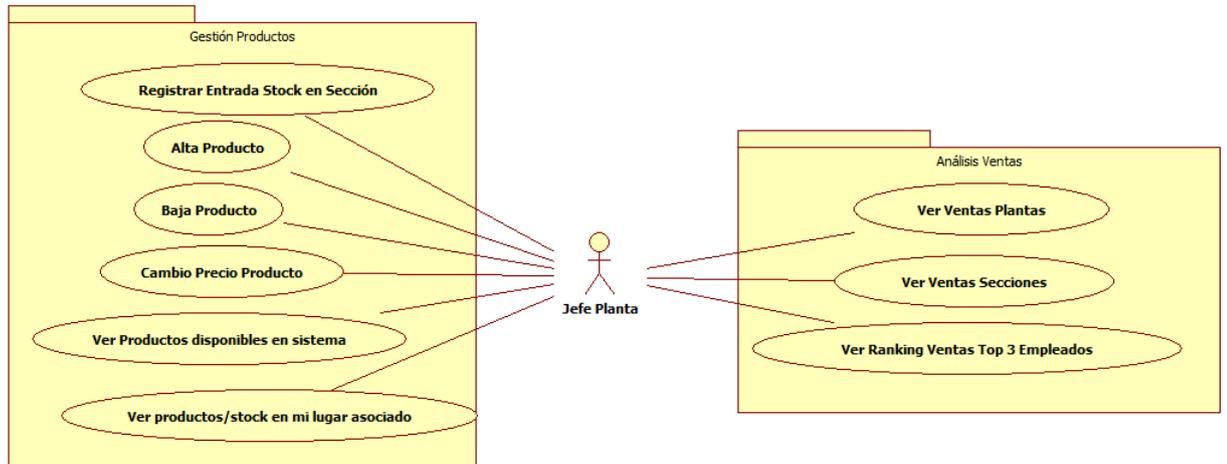


Figura 6.11: Casos de uso de los bloques Gestión Planta y Análisis Ventas del rol Jefe de Planta

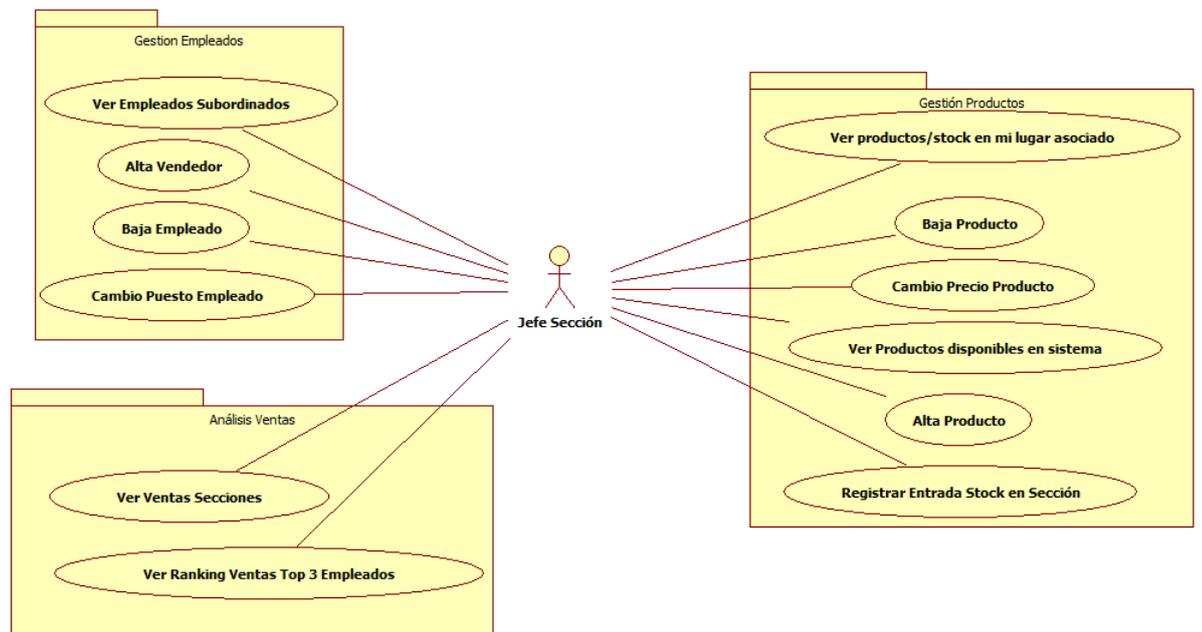


Figura 6.12: Casos de uso del rol Jefe de Sección

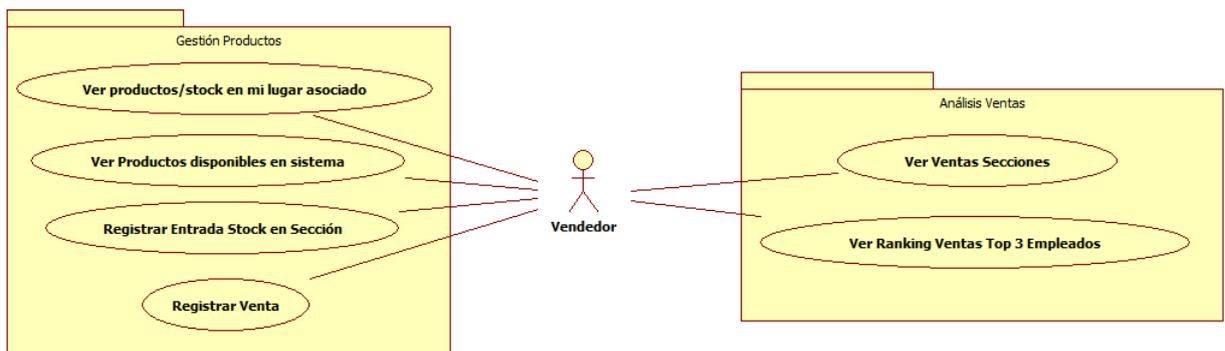


Figura 6.13: Casos de uso del rol Vendedor

Capítulo 7

Diseño

UNA vez finalizado en análisis, se realiza el diseño de la aplicación. Este capítulo estará dividido en dos apartados: Diseño de los datos y Diseño de la arquitectura.

7.1 Diseño de los datos

El diseño de los datos es la primera parte del apartado de diseño, en ella se modela la estructura de la base de datos que persistirá la información del sistema.

El mayor reto en este apartado consistió en dotar a la aplicación la capacidad de gestionar los distintos cambios de rol y de lugar que se producen entre los empleados. Un ejemplo es el caso de un empleado que tiene parte del rol de “Vendedor” en la sección 1 de la planta 1 del centro 1, cambia su rol a “Vendedor” de la sección 2, planta 2, centro 2 y vuelve a cambiar su rol a “Jefe Sección” de la sección 1, planta 1 centro 1. La dificultad no está solo en permitir que esto sea posible, si no en registrar correctamente dónde se han producido las ventas que realizó dicho empleado.

Las entidades diseñadas para lograr el desarrollo de este proyecto son las siguientes:

- **Empleado:** representa al usuario de la aplicación, se relaciona con Histórico Puesto.
- **Rol:** entidad que representa los distintos roles existentes en la aplicación, se relaciona con Histórico Puesto y consigo misma para representar la jerarquía de roles existente.
- **Lugar:** representa los distintos lugares existentes, se relaciona con Histórico Puesto y consigo misma para representar la jerarquía de roles existente. Esta entidad es extendida por las entidades Centro, Planta y Sección
- **Histórico Puesto:** entidad sirve de nexo entre Empleado, Rol y Lugar, representa la asignación de un rol en un lugar a un empleado en un determinado momento.

- **Centro:** extiende a la entidad Lugar y representa un centro.
- **Planta:** extiende a la entidad Lugar y representa una planta.
- **Sección:** extiende a la entidad Lugar, representa una sección y se relaciona con Producto indicando que la disponibilidad de éste en la sección.
- **Producto:** representa un producto gestionado en el sistema, se relaciona con Línea Venta.
- **Venta:** representa una venta realizada por un empleado con un Rol en un Lugar, se relaciona con Línea Venta, puesto que una venta se compone de líneas de venta.
- **Línea Venta:** entidad débil que representa una línea de venta dentro de una venta, es decir, cada producto que se vende en una venta.

En la figura 7.1 se muestra el modelo entidad-relacion de la aplicación.

7.2 Diseño de la arquitectura del sistema

Esta aplicación se ha desarrollado siguiendo las directrices del patrón arquitectónico **Modelo-Vista-Controlador (MVC)** combinado con la arquitectura **Cliente-Servidor**. Dicho patrón separa la lógica de la aplicación de la lógica de la interfaz en tres componentes bien definidos:

- **Modelo:** es el encargado de la lógica de negocio y el acceso a la capa de datos.
- **Vista:** se trata de la representación visual de los datos en un formato adecuado para que el usuario pueda interactuar con ellos.
- **Controlador:** procesa los distintos eventos realizados por el usuario en la vista y le envía peticiones al modelo.

En la figura 7.2 se muestra el funcionamiento de este patrón, y en la figura 7.3 se indica como se ha combinado la arquitectura **Cliente-Servidor** con el ya mencionado patrón **MVC**.

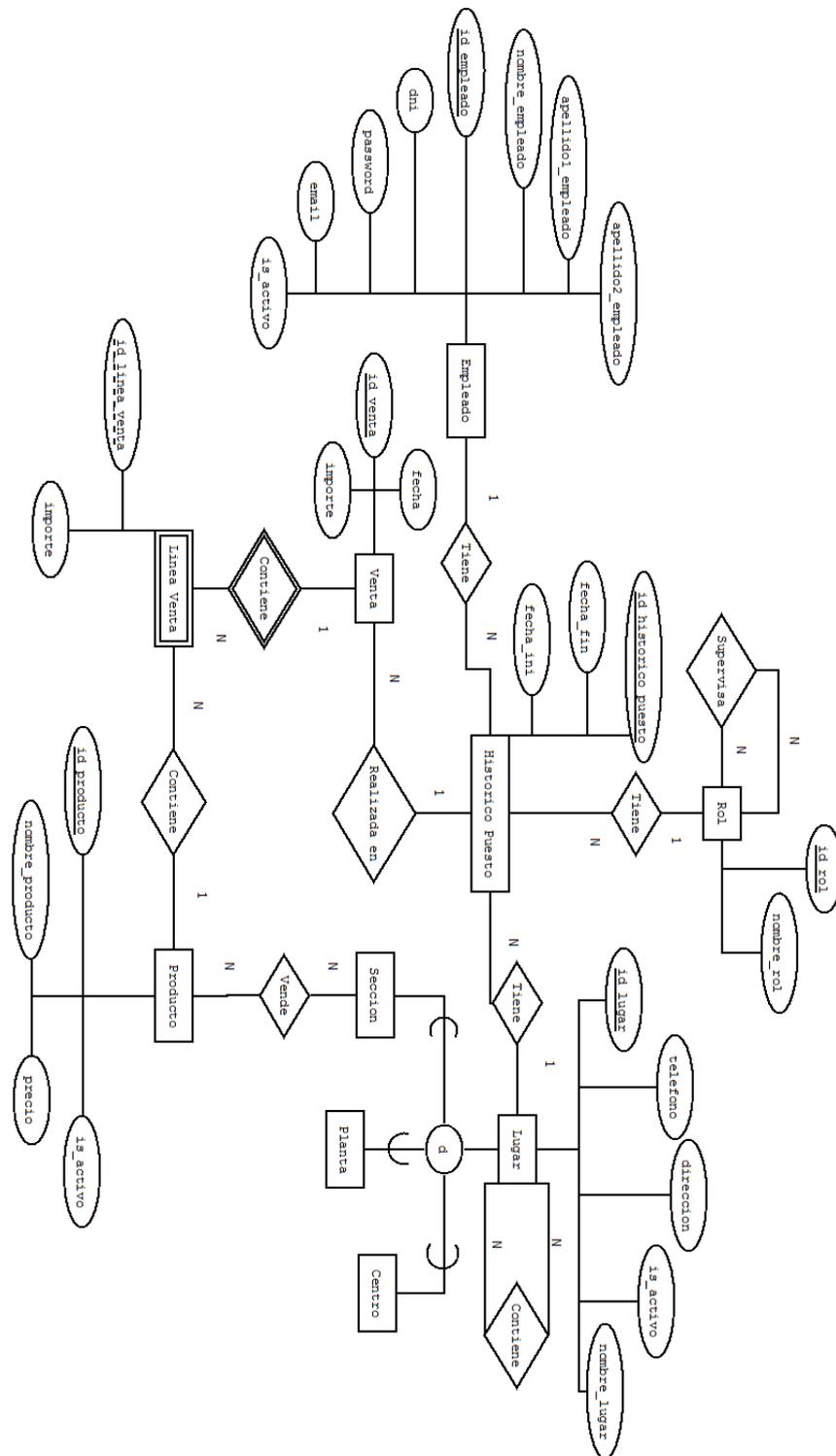


Figura 7.1: Diagrama Entidad-Relación

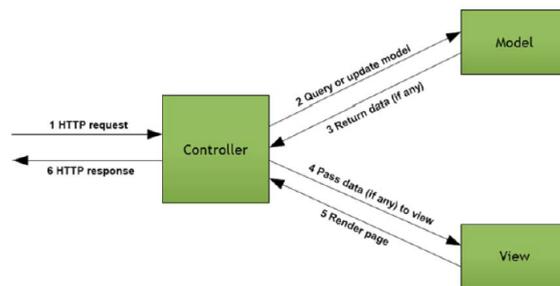


Figura 7.2: Funcionamiento del patrón MVC

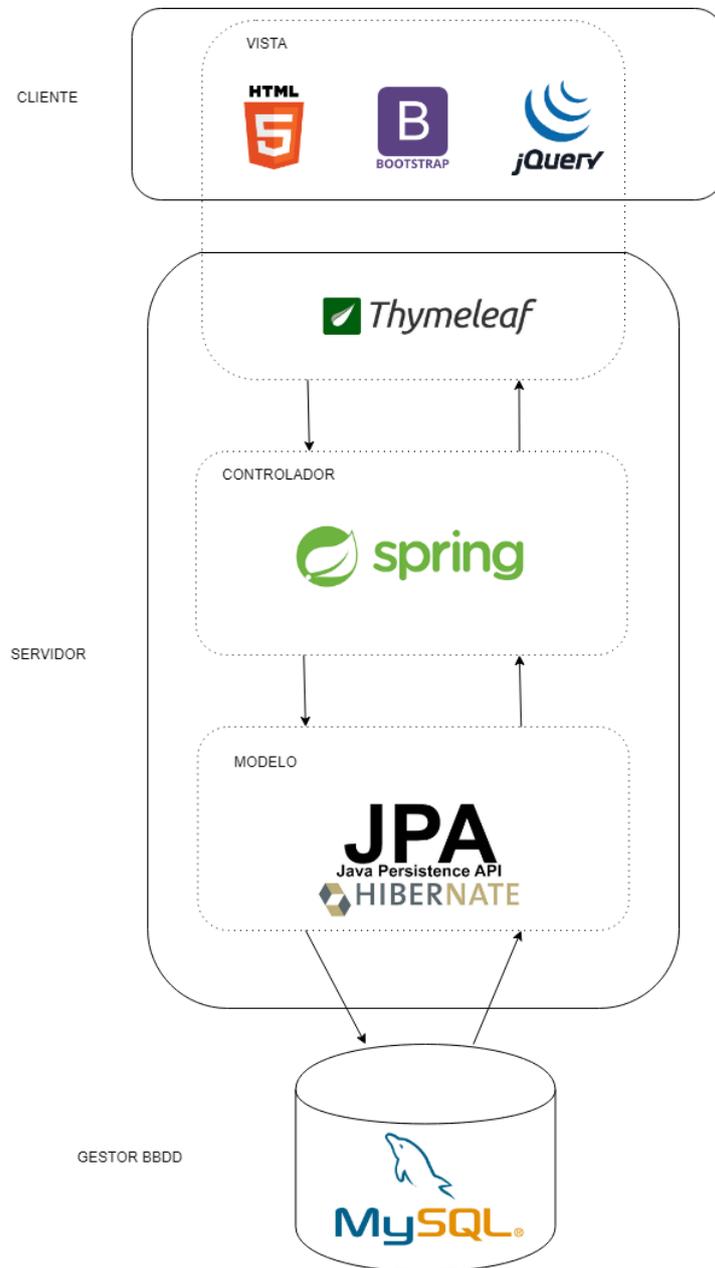


Figura 7.3: Arquitectura del sistema

Implementación

EN este capítulo se explica cómo se ha llevado a cabo el desarrollo de la aplicación, consta de dos apartados: implementación del servidor e implementación del cliente.

8.1 Implementación del servidor

La parte servidor se ha desarrollado siguiendo la estructura **Controller-Service-DAO**, tal y como se muestra en el diagrama de paquetes de la figura 8.1

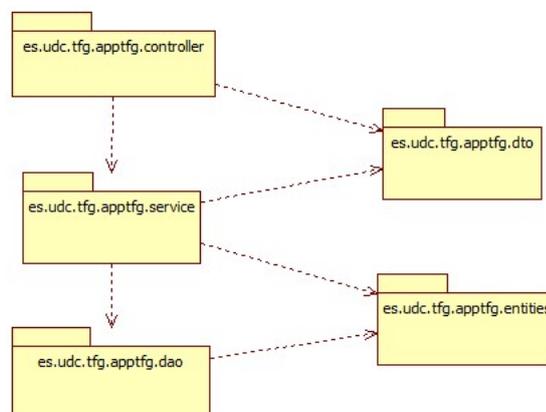


Figura 8.1: Diagrama de paquetes de la parte servidor

Esta implementación se basa en el uso de tres patrones: **el patrón DAO (Data Access Object)**, **el patrón DTO (Data Transfer Object)** y **el patrón Facade**.

8.1.1 El patrón DAO

El patrón DAO es un patrón que nos permite encapsular el acceso a la base de datos a través de la aplicación, es decir, proporciona los métodos necesarios para insertar, actualizar, borrar

y consultar información desde la aplicación a la base de datos de una manera transparente, lo que permite aislar la capa de lógica de negocio de la capa de acceso a datos. A su vez, este patrón permite una abstracción que oculta el gestor de base de datos a la aplicación, por lo que sería posible cambiar entre distintos gestores como MySQL y Oracle sin necesidad de cambios en el código.

En este proyecto se ha utilizado el framework Spring Data para facilitar el desarrollo de este apartado. Por un lado se han mapeado las clases que representan a las distintas entidades como se muestra en el código a continuación:

```
1 @Entity
2 @Table(name = "empleado")
3 public class Empleado {
4
5     @Id
6     @GeneratedValue(strategy=GenerationType.IDENTITY)
7     @Column(name = "id_empleado")
8     private Integer empleadoId;
9
10    @Column(name = "nombre_empleado")
11    private String nombreEmpleado;
12
13    @Column(name = "apellido1_empleado")
14    private String apellido1Empleado;
15
16    @Column(name = "apellido2_empleado")
17    private String apellido2Empleado;
18
19    @Column(name = "email_empleado")
20    private String emailEmpleado;
21
22    @Column(name="dni_empleado")
23    private String dniEmpleado;
24
25    @Column(name="password")
26    private String password;
27
28    @Column(name="is_activo")
29    private Boolean isActive;
30
31    @OneToMany
32    @JoinColumn(name = "id_empleado")
33    private List<HistoricoPuesto> puestos;
34    ...
```

Mientras que en las clases DAO se ha implementado la interfaz JpaRepository, la cual nos

proporciona los métodos CRUD relativos a la entidad relacionada y nos simplifica enormemente la codificación.

```

1 @Repository
2 public interface RolDAO extends JpaRepository<Rol, Integer>{
3
4     @Query(value = "" +
5         "select r.id_rol, r.nombre_rol " +
6         " from empleado e " +
7         " join historico_puesto hp " +
8         " on e.id_empleado = hp.id_empleado and hp.fecha_fin IS NULL
9         " +
10        "join rol r on hp.id_rol = r.id_rol "
11        + " WHERE e.id_empleado = :idEmpleado", nativeQuery = true)
12        Rol findRolActualByIdEmpleado(@Param("idEmpleado") final Integer
13        idEmpleado);
14    }

```

En la figura 8.2 se muestra el diagrama de clases de las entidades persistentes existentes en la aplicación.

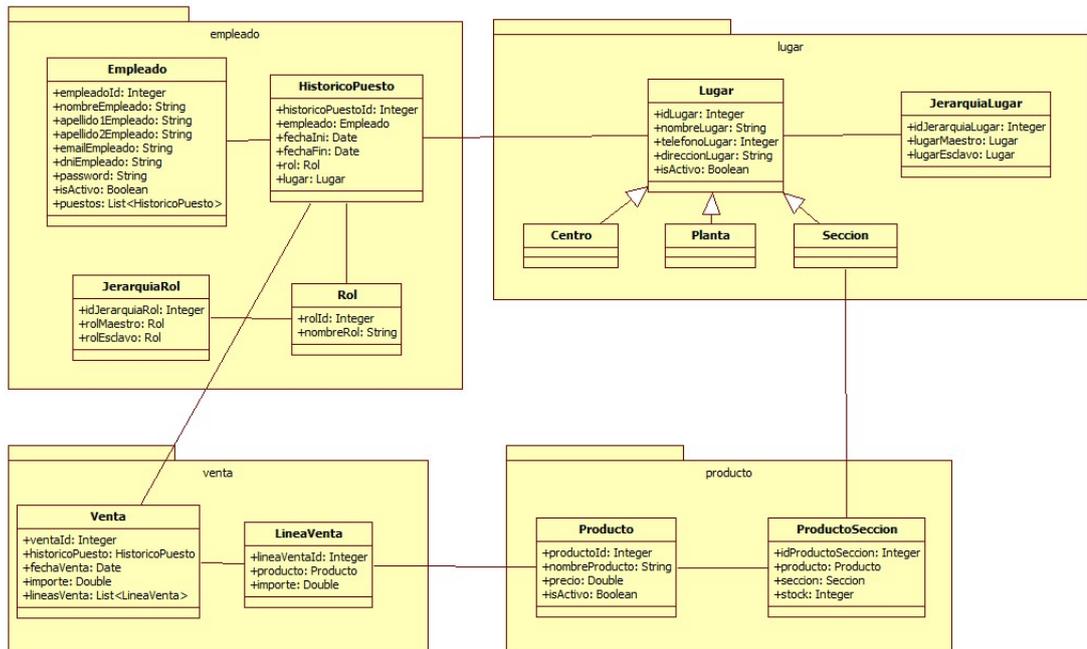


Figura 8.2: Diagrama de clases del paquete "entities"

8.1.2 El patrón DTO

El patrón DTO permite optimizar la transferencia de datos a través de las distintas capas de la aplicación. Su finalidad es la de crear un objeto plano con una serie de atributos que puedan ser enviados o recuperados del servidor en una sola invocación, de tal forma que un DTO puede contener información de una única tabla o de varias, representadas en una clase simple que únicamente cuenta con atributos, getters y setters. Es importante destacar que las clases DTO deben implementar la interfaz `Serializable`, lo que permite que las instancias de dicha clase se puedan convertir en una serie de bytes para enviarlas a través de la red y volver a reconstruirlas una vez recibidas, tal y como se muestra en el siguiente ejemplo:

```
1 import java.io.Serializable;
2
3 public class RolDto implements Serializable {
4
5     private static final long serialVersionUID =
6         -5355873372767429908L;
7
8     private Integer idRol;
9
10    private String nombreRol;
11    ...
```

En la figura 8.3 se indican las clases existentes en el paquete "dto" del proyecto.

8.1.3 El patrón Facade

El patrón Facade o Fachada se utiliza para estructurar un sistema en capas, utilizando las fachadas como punto de entrada a cada nivel para agrupar funcionalidades. En esta aplicación, los distintos casos de uso están implementados en clases `Service` que siguen el comportamiento establecido por este patrón: las distintas clases `Service` utilizan los `DAOs` para acceder a la información y los `DTOs` para transferirla.

En este proyecto, las clases que implementan las interfaces `Service` cuentan con las anotaciones **@Service** y **@Transactional**. La primera anotación le indica a Spring que esta clase es un objeto disponible para ser usado por el controlador, mientras que la segunda anotación indica que cada método de la clase ha de ser ejecutado como si fuera una transacción de base de datos. Los `DAOs` utilizados en las clases `service` se anotan con **@Autowired** para que Spring los inicialice automáticamente. A continuación se muestra un ejemplo de las anotaciones utilizadas:

```
1 @Service
2 @Transactional
```

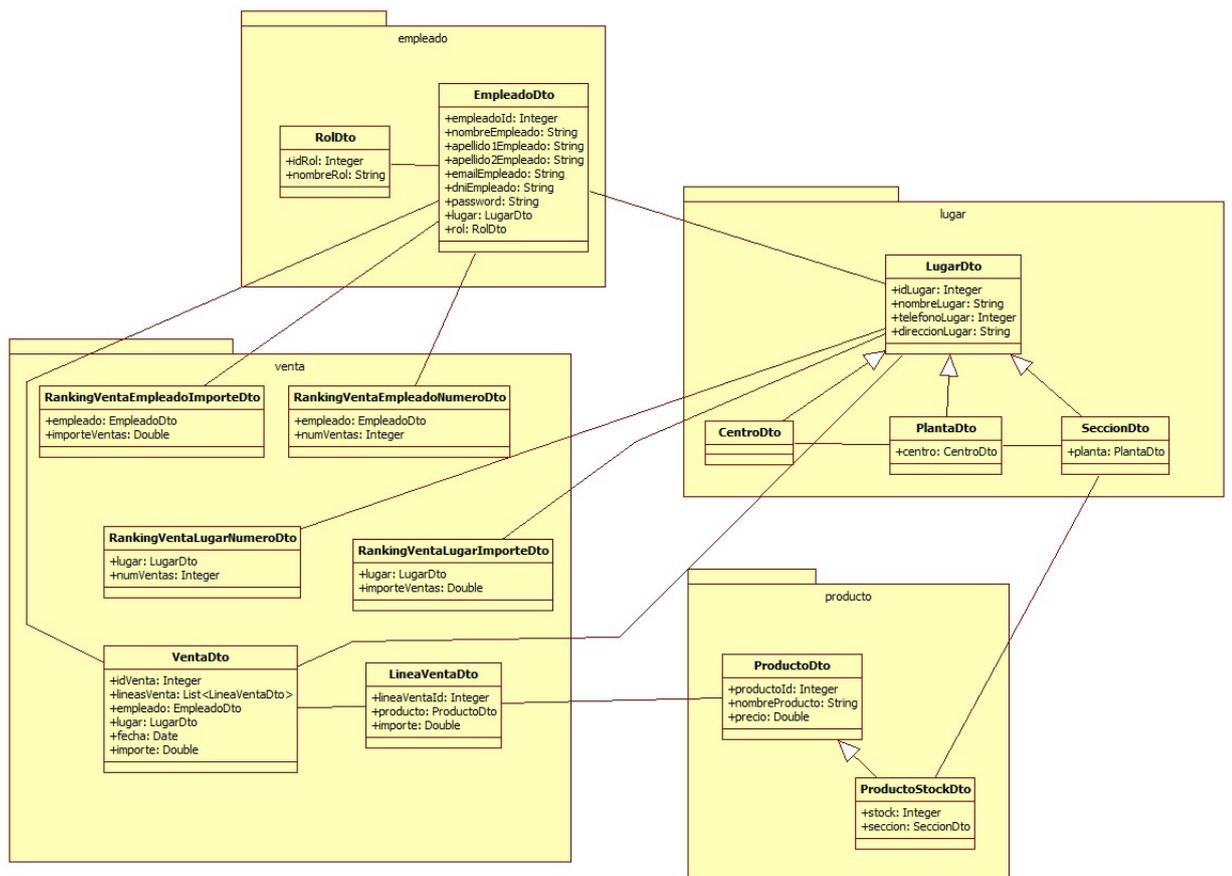


Figura 8.3: Diagrama de clases del paquete "dto"

```

3 public class EmpleadoServiceImpl implements EmpleadoService {
4
5     @Autowired
6     private EmpleadoDAO empleadoRepository;
7
8     @Autowired
9     private HistoricoPuestoDAO historicoPuestoRepository;
10
11    @Autowired
12    private RolDAO rolRepository;
13
14    @Autowired
15    private LugarDAO lugarRepository;
16
17    @Autowired
18    private JerarquiaRolDAO jerarquiaRolRepository;
19
20    ...

```

En las figuras 8.4, 8.5, 8.6 y 8.7 podemos ver los diagramas de clases de los distintos service.

Para finalizar el apartado, en el paquete controller se han creado dos controladores “Rest-Controller” y “HomeController”. El primero de ellos cuenta con los métodos necesarios para realizar peticiones REST desde el cliente devolviendo un String que se convierte en JSON con los datos obtenidos de los servicios, tal y como se puede ver en el siguiente extracto del código:

```

1 @RestController
2 public class RestController {
3
4     @Autowired
5     private EmpleadoService empleadoService;
6
7     @Autowired
8     private LugarService lugarService;
9
10    @Autowired
11    private VentaService ventaService;
12
13    @Autowired
14    private ProductoService productoService;
15
16    @RequestMapping(value = "/getDataChartRankingNumVentasTop3Emp")
17    public String loadDataRankingNum() {
18        final List<RankingVentaEmpleadoNumeroDto> rankingList =
19            this.ventaService.getTop3EmpleadoNumeroVentas();
20        final JSONArray jsonLabels = new JSONArray();
21        final JSONArray jsonVentas = new JSONArray();

```

```
21     for (final RankingVentaEmpleadoNumeroDto r : rankingList) {
22         jsonLabels.add(r.getNombreEmpleado());
23         jsonVentas.add(r.getNumeroVentas());
24     }
25     final JsonObject chartData = new JsonObject();
26     chartData.add("labels", jsonLabels);
27     chartData.add("ventas", jsonVentas);
28
29     final JsonObject chartDataDad = new JsonObject();
30     chartDataDad.add("chartData", chartData);
31
32     final ArrayList<Object> dataSet = new ArrayList<Object>();
33     dataSet.add(chartData);
34
35     return dataSet.toString();
36 }
37 ...
```

Por el contrario, el “HomeController” se encarga de proporcionar al cliente las vistas renderizadas, obteniendo los distintos datos de los servicios y devolviendo la ruta al fichero HTML en el proyecto a mostrar en conjunto con los datos necesarios para su procesado:

```
1 @Controller
2 public class HomeController {
3
4     @Autowired
5     private EmpleadoService empleadoService;
6
7     @Autowired
8     private LugarService lugarService;
9
10    @Autowired
11    private VentaService ventaService;
12
13    @Autowired
14    private ProductoService productoService;
15
16    @GetMapping("/user")
17    public String userIndex(final Model model) {
18        final List<ProductoDto> productos =
19            this.controller.getAllProductos();
20        model.addAttribute("listProductos", productos);
21
22        ...
23
24        return "user/index";
25    }
26 }
```

24 }

El objeto "listProductos" es accesible desde el fichero HTML "user/index" de la siguiente manera:

```

1 ...
2
3 <script th:inline="javascript">
4   var listProductos = [[${listProductos}]];
5 </script>
6
7 ...
8 }
    
```

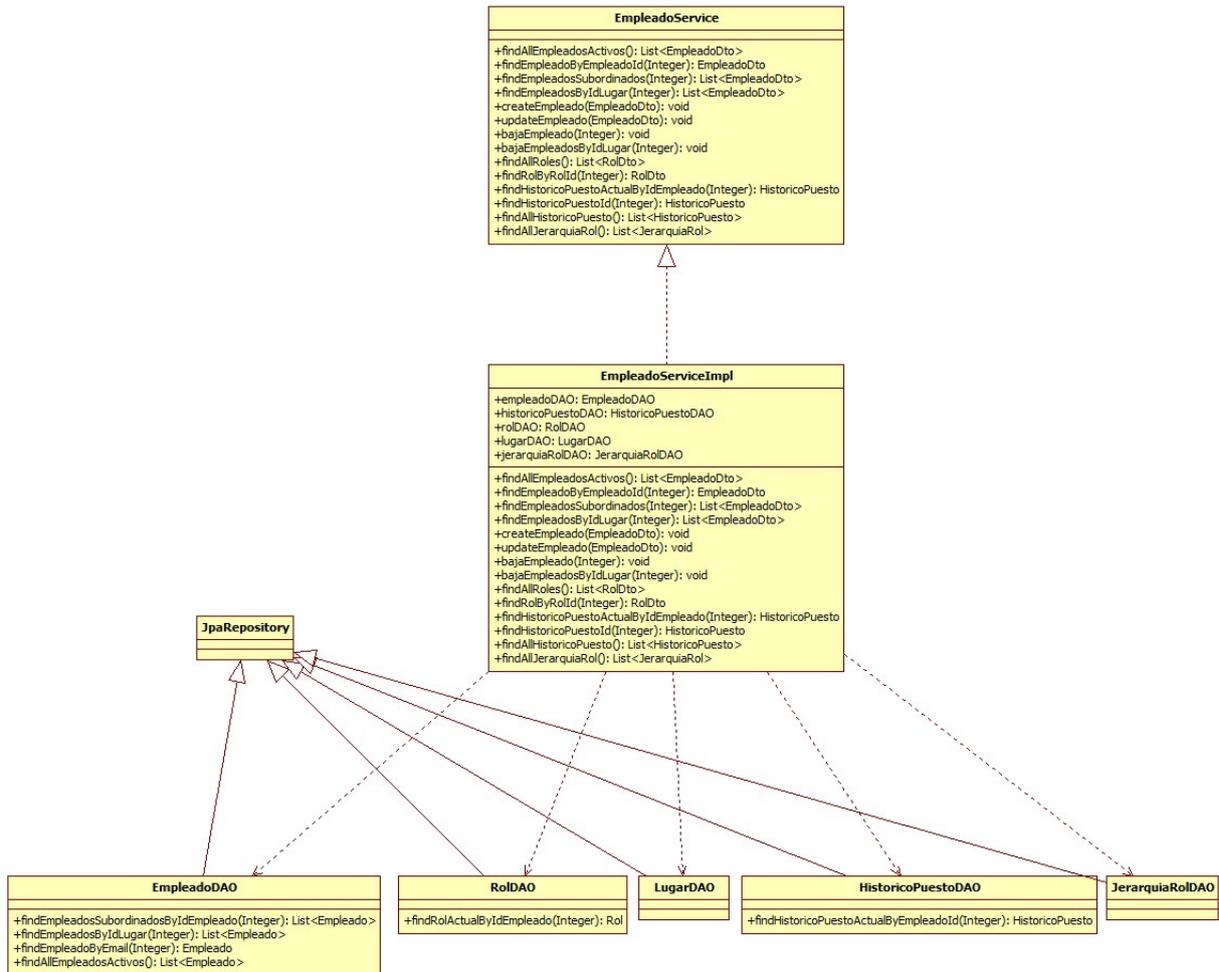


Figura 8.4: Diagrama de clases de EmpleadoService

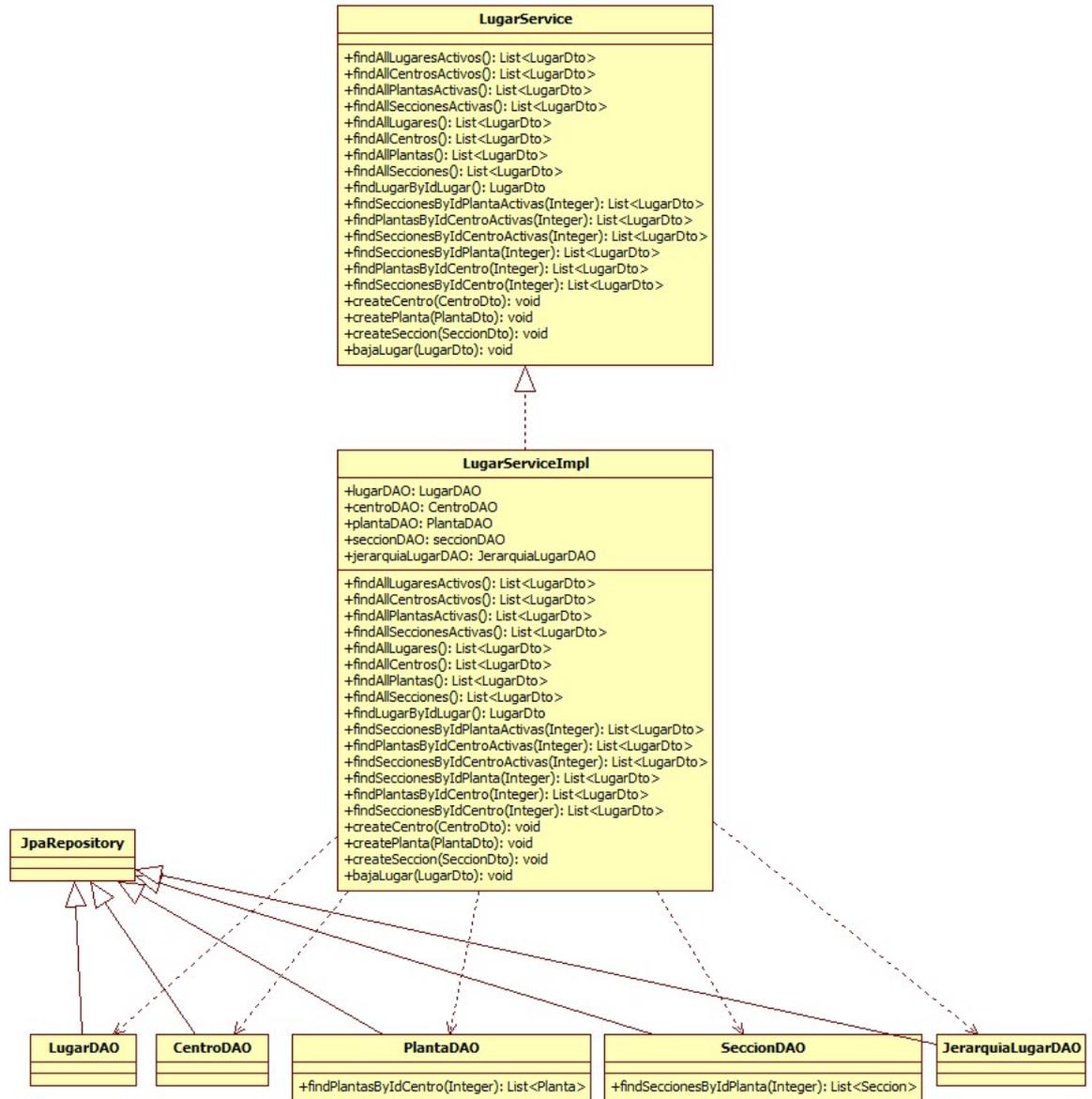


Figura 8.5: Diagrama de clases de LugarService

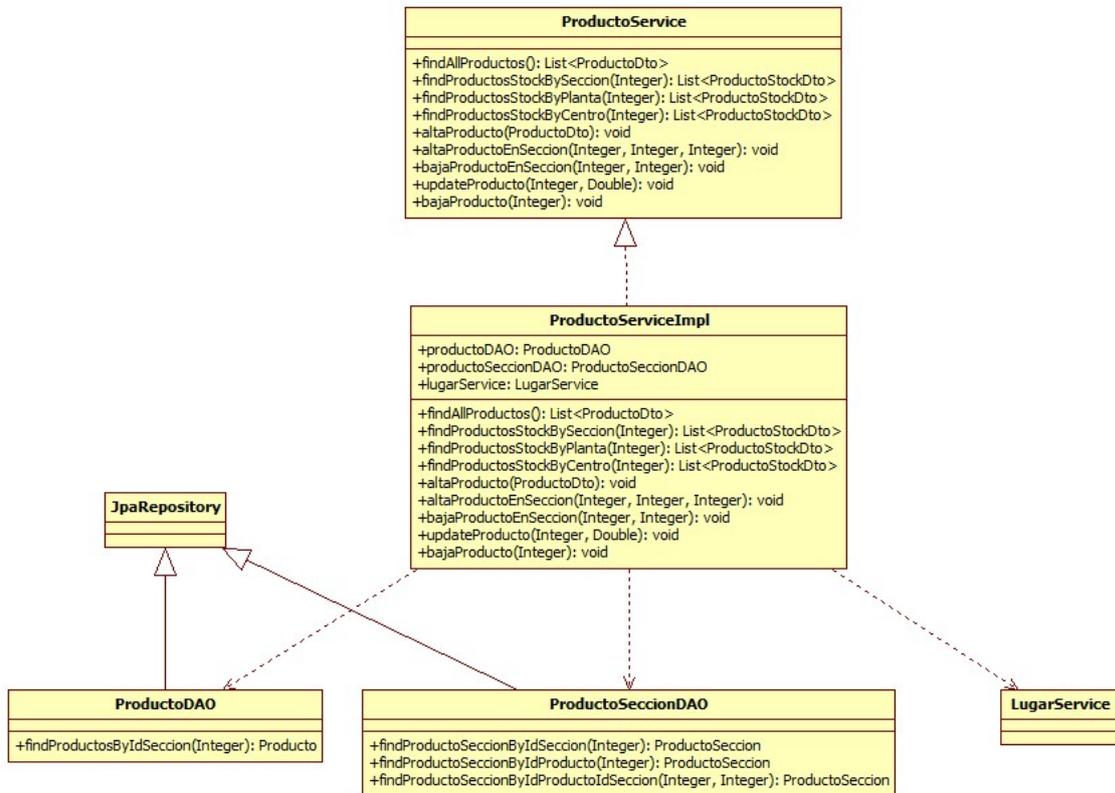


Figura 8.6: Diagrama de clases de ProductoService

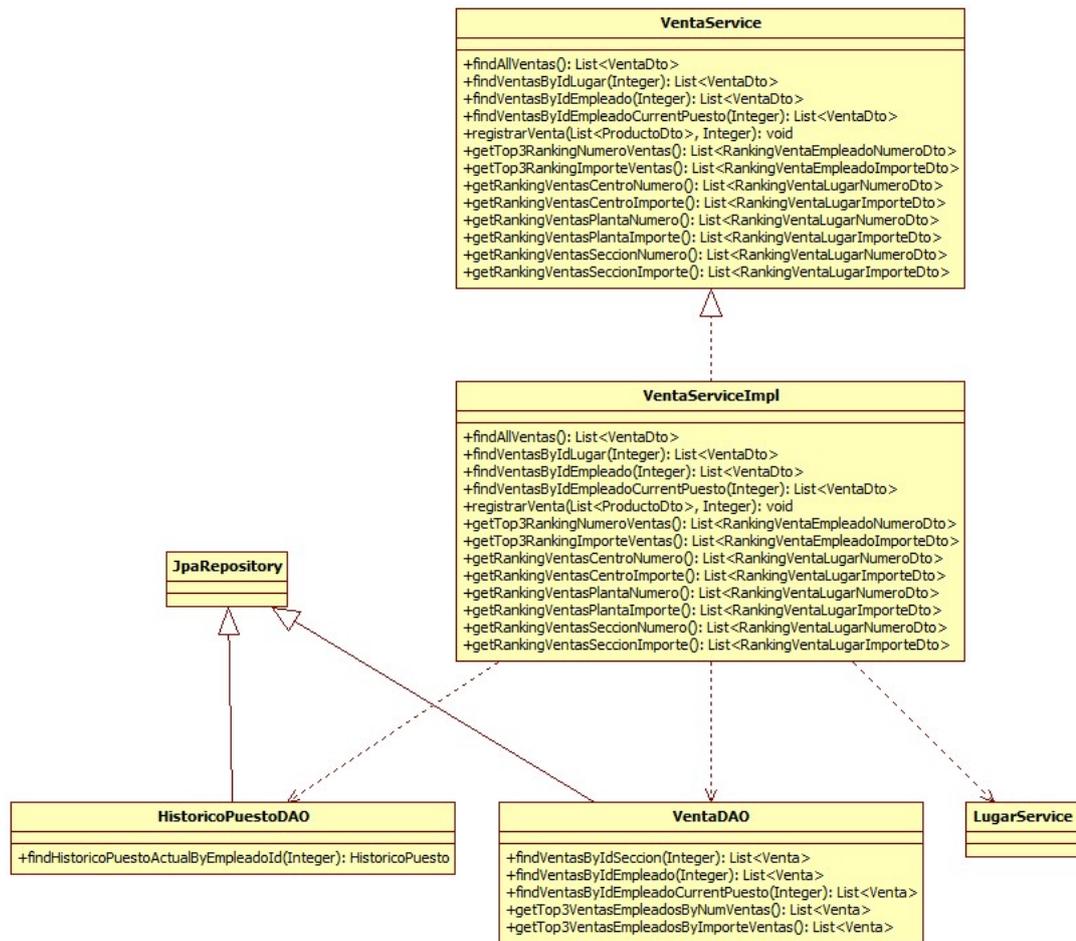


Figura 8.7: Diagrama de clases de VentaService

8.2 Implementación del cliente

La aplicación cliente se ha desarrollado en una sola página, siguiendo el modelo **Single-page application (SPA)**, con el objetivo de proporcionar una experiencia más fluida a los usuarios. En este tipo de aplicaciones, los códigos HTML, JavaScript y CSS se cargan al visitar por primera vez la página.

La interacción de datos entre el servidor y el cliente se ha realizado utilizando Thymeleaf tal y como se mostraba en el apartado anterior, pero también mediante peticiones REST ejecutadas de forma asíncrona utilizando AJAX, como se puede apreciar en el siguiente ejemplo en el que se muestra la llamada para registrar el alta de un empleado:

```

1 function altaEmpleado() {
2   $.ajax({
3     type: "post",
4     url: 'createEmpleado',
5     data: $("#formAltaEmpleado").serialize(),
6     success: function () {
7       $('#modalAltaEmpleado').modal('hide');
8       reloadTableEmpleado();
9     }
10  });
11 }

```

El diseño de la página web está conformado por un footer común para las páginas de login e index y un cuerpo que en la página principal está conformado por cinco pestañas, cada una de ellas contiene uno de los distintos módulos ya citados, a excepción de el módulo de productos, que está dividido en dos pestañas. El rol del usuario logeado, determinará que pestañas aparecerán visibles.

Pestañas de la aplicación:

- **Análisis de ventas:** esta pestaña contiene las gráficas relativas al análisis de las ventas, divididas en cuatro grupos: Ránking de ventas Empleados 8.8, Análisis por Centros 8.9, Análisis por Plantas 8.10 y Análisis por Secciones 8.11. El rol del usuario determinará las gráficas que estarán visibles, tal y como se puede ver en las figuras indicadas anteriormente, podemos ver que en la captura 8.9 el usuario logeado tiene el puesto **JEFE CENTRO** y tiene disponibles las cuatro gráficas, mientras que en la captura 8.10 el usuario logeado tiene asignado el puesto **JEFE PLANTA**, y las gráficas de Análisis por Centro no le aparecen disponibles. En esta última captura también se puede apreciar el comportamiento del componente gráfico que permite eliminar los valores que no queramos mostrar para un análisis más concreto, tal y como se ha hecho en la gráfica **IMPORTE DE VENTAS POR PLANTA**, eliminando las plantas del centro **CCMARI-**

NEDA del resultado.

- **Gestión Empleados:** esta pestaña está conformada por un botón que permite registrar altas de empleados en el sistema y una tabla en la que se muestran los empleados subordinados al usuario registrado. Para cada empleado subordinado el usuario puede cambiarlo de puesto o darlo de baja. En la figura 8.12 se muestra una captura de pantalla de dicha pestaña para el usuario logeado “pablo@app.com”, con rol Jefe Centro en **CCMARINEDA**, en la que se puede apreciar que tiene cuatro empleados subordinados entre ellos el usuario con email “sergio@app.com”, que posee el rol Jefe Planta en **PLANTA 1 - CCMARINEDA**. En la figura 8.13 se muestra la captura de pantalla de la misma pestaña para este usuario, y se puede apreciar que entre sus subordinados obviamente desaparece él mismo pero tampoco se encuentra el “miguel@app.com”, puesto que es un vendedor asignado a una sección de una planta distinta.
- **Gestión Lugares:** esta pestaña contiene una tabla con los lugares gestionados por el usuario y una serie de botones que permiten registrar nuevos lugares en la aplicación. El tipo de lugares que el usuario puede registrar viene determinado por su rol, esto lo podemos apreciar en las capturas de pantalla 8.14 y 8.15. La primera pertenece al usuario “pablo@app.com” y en ella se muestra el botón “Alta Planta”, mientras que en la segunda, perteneciente al usuario “sergio@app.com”, este botón no aparece. También se pueden apreciar menos lugares bajo la gestión de este segundo usuario.
- **Gestión Productos:** esta pestaña contiene dos tablas: en la primera de ellas aparecen los productos disponibles en el lugar gestionado por el usuario y el stock que tienen, y en la segunda aparece todos los productos registrados en el sistema. En la parte superior de la pantalla se encuentran dos botones, “Alta nuevo producto” y “Registrar entrada de stock”. El primero permite dar de alta un nuevo producto en el sistema, mientras que el segundo registrar la entrada de stock de un determinado producto en una de las secciones que tenga asignadas el usuario logeado. En la figura 8.16 se muestra una captura de pantalla del usuario “pablo@app.com”, donde se indica que en el lugar asociado a dicho usuario, en este caso **CCMARINEDA**, constan 201 y 200 unidades de los productos “PANTALON LANA” y “PANTALON VAQUERO” respectivamente. En la figura 8.17 se muestra una captura de pantalla del usuario “sergio@app.com”, subordinado a “pablo@app.com”, en la que se indica el lugar asociado a este usuario, **PLANTA 1 - CCMARINEDA** y el stock de los productos asociados a este lugar: 100 unidades de “PANTALON LANA” y 100 de “PANTALON VAQUERO”. Las 101 y 100 unidades restantes se encuentran asignadas a la **PLANTA 2 - CCMARINEDA**, tal y como se muestra en la figura 8.18, en la que se muestra una captura de pantalla del usuario “pedro@app.com”, responsable de dicha planta.

- **Registro de ventas:** esta pestaña tan sólo estará disponible para los usuarios que posean el rol “Vendedor”. En la figura 8.19 se muestra una captura de pantalla del usuario “miguel@app.com” que ha iniciado el proceso de registro de una venta.

Para terminar este capítulo, el desarrollo del cliente se ha realizado siguiendo la técnica de diseño web **responsive**, utilizando componentes adaptables al tamaño y disposición de la pantalla que permiten su correcta visualización en cualquier dispositivo. En las figuras 8.20 y 8.21 se adjuntan capturas de pantalla de la aplicación en una tablet y un smartphone respectivamente.

AppGestion

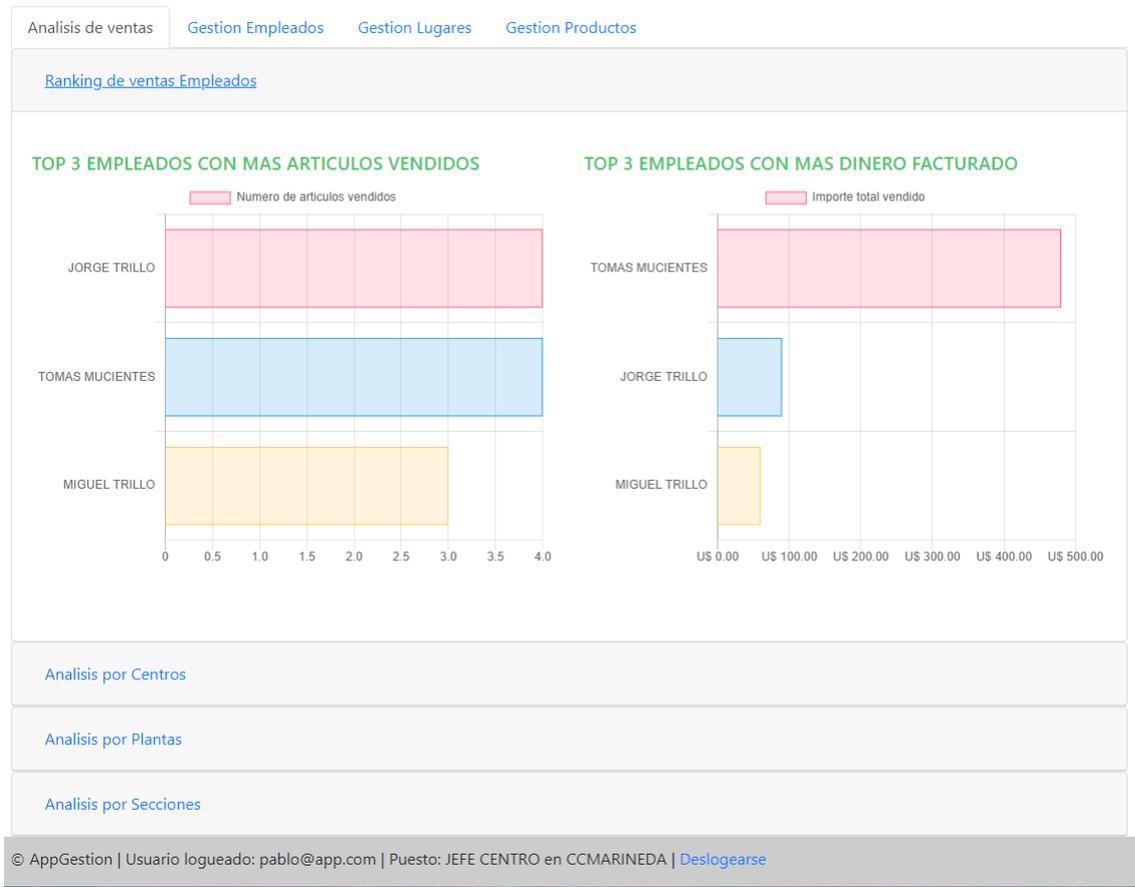


Figura 8.8: Captura de pantalla donde se muestran las gráficas de Ránking de ventas Empleados

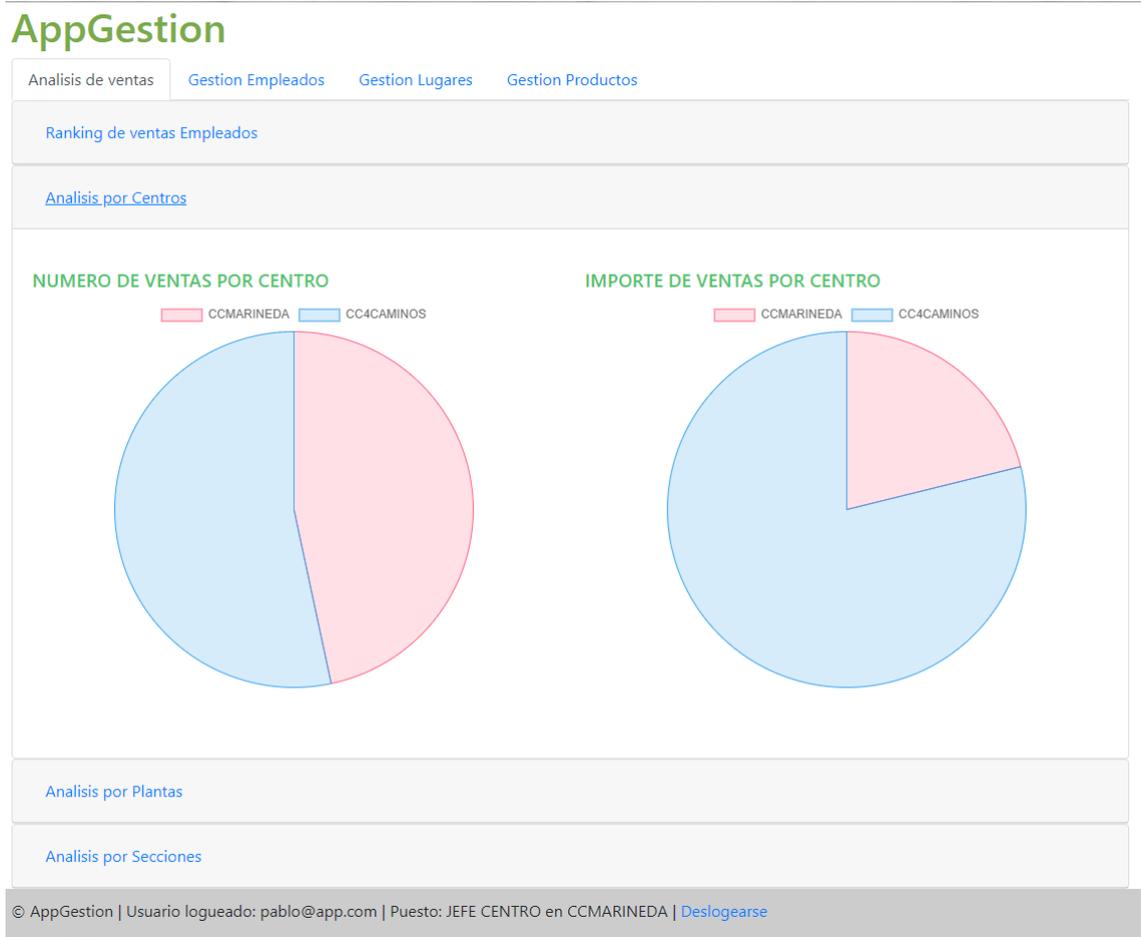


Figura 8.9: Captura de pantalla donde se muestran las gráficas de Análisis de ventas por Centro

AppGestion

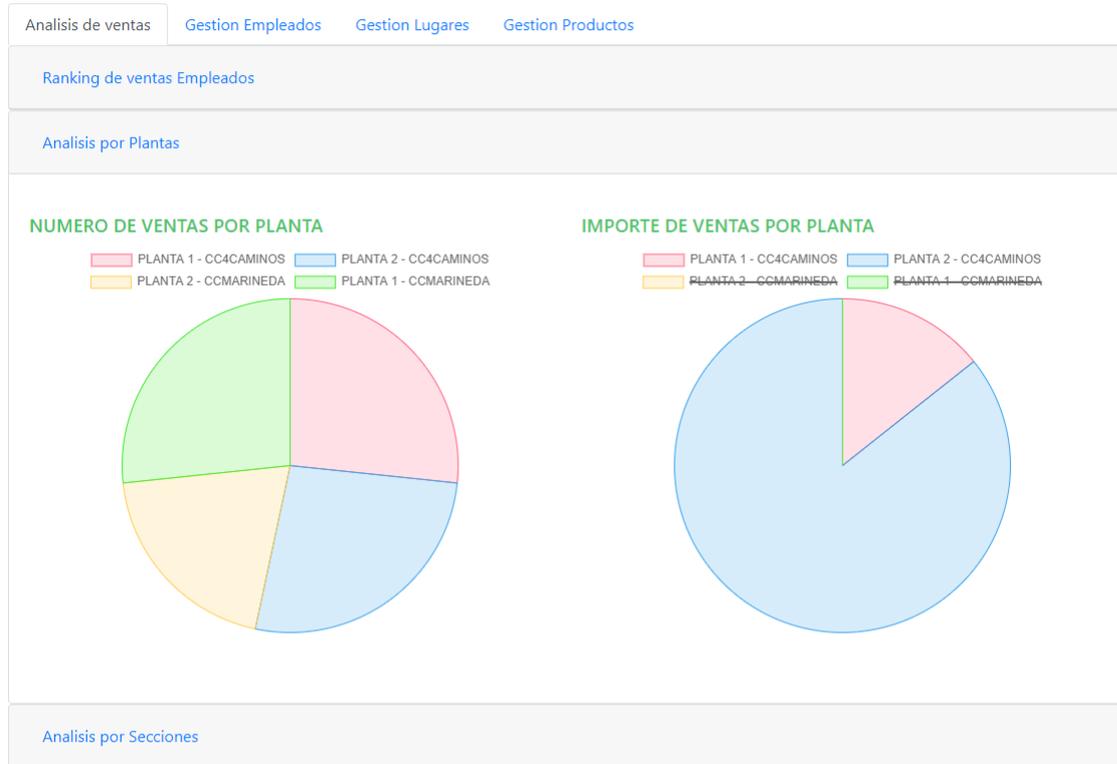


Figura 8.10: Captura de pantalla donde se muestran las gráficas de Análisis de ventas por Planta

AppGestion

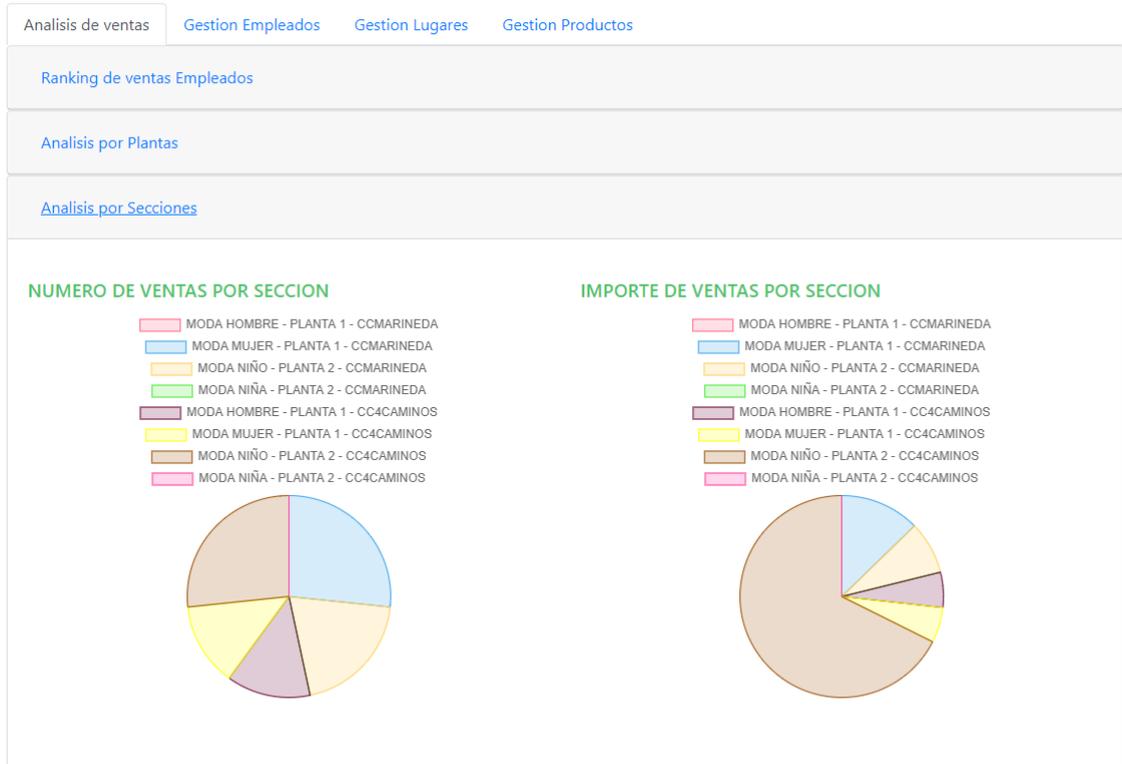


Figura 8.11: Captura de pantalla donde se muestran las gráficas de Análisis de ventas por Seccion

AppGestion

[Análisis de ventas](#)
[Gestion Empleados](#)
[Gestion Lugares](#)
[Gestion Productos](#)

Alta Empleado

Lista de subordinados

Mostrando 10 elementos

Buscar:

ID	NOMBRE	APELLIDO 1	APELLIDO 2	EMAIL	DNI	ROL	LUGAR		
2	SERGIO	TRILLO	VALEIRO	sergio@app.com	000000002A	JEFE PLANTA	PLANTA 1 - CCMARINEDA	Cambiar de Puesto	Dar de baja
3	NURIA	TRILLO	VALEIRO	nuria@app.com	000000003A	JEFE SECCION	MODA HOMBRE - PLANTA 1 - CCMARINEDA	Cambiar de Puesto	Dar de baja
4	JORGE	TRILLO	VALEIRO	jorge@app.com	000000004A	VENDEDOR	MODA MUJER - PLANTA 1 - CCMARINEDA	Cambiar de Puesto	Dar de baja
5	MIGUEL	TRILLO	VALEIRO	miguel@app.com	000000004A	VENDEDOR	MODA NIÑO - PLANTA 2 - CCMARINEDA	Cambiar de Puesto	Dar de baja

Mostrando 1 a 4 de 4 elementos

Anterior

1

Siguiente

© AppGestion | Usuario logueado: pablo@app.com | Puesto: JEFE CENTRO en CCMARINEDA | [Deslogearse](#)

Figura 8.12: Captura de pantalla donde se muestra la pestaña Gestión de Empleados del usuario “pablo@app.com”

AppGestion

[Análisis de ventas](#)
[Gestión Empleados](#)
[Gestión Lugares](#)
[Gestión Productos](#)

Alta Empleado

Lista de subordinados

Mostrando 10 elementos

Buscar:

ID	NOMBRE	APELLIDO 1	APELLIDO 2	EMAIL	DNI	ROL	LUGAR		
3	NURIA	TRILLO	VALEIRO	nuria@app.com	000000003A	JEFE SECCION	MODA HOMBRE - PLANTA 1 - CCMARINEDA	Cambiar de Puesto	Dar de baja
4	JORGE	TRILLO	VALEIRO	jorge@app.com	000000004A	VENDEDOR	MODA MUJER - PLANTA 1 - CCMARINEDA	Cambiar de Puesto	Dar de baja

Mostrando 1 a 2 de 2 elementos

[Anterior](#)
[1](#)
[Siguiente](#)

Figura 8.13: Captura de pantalla donde se muestra la pestaña Gestión de Empleados del usuario “sergio@app.com”

AppGestion

[Análisis de ventas](#) [Gestión Empleados](#) [Gestión Lugares](#) [Gestión Productos](#)

[Alta Planta](#) [Alta Sección](#)

Lugares gestionados

Mostrando 10 elementos

Buscar:

ID	NOMBRE	DIRECCION	TELEFONO	
3	PLANTA 1 - CCMARINEDA	CALLE MARINEDA SN	981000002	Dar de baja
4	PLANTA 2 - CCMARINEDA	CALLE MARINEDA SN	981000003	Dar de baja
7	MODA HOMBRE - PLANTA 1 - CCMARINEDA	CALLE MARINEDA SN	981000006	Dar de baja
8	MODA MUJER - PLANTA 1 - CCMARINEDA	CALLE MARINEDA SN	981000007	Dar de baja
9	MODA NIÑO - PLANTA 2 - CCMARINEDA	CALLE MARINEDA SN	981000008	Dar de baja
10	MODA NIÑA - PLANTA 2 - CCMARINEDA	CALLE MARINEDA SN	981000009	Dar de baja

Mostrando 1 a 6 de 6 elementos

[Anterior](#) [1](#) [Siguiente](#)

Figura 8.14: Captura de pantalla donde se muestra la pestaña Gestión de Lugares del usuario “pablo@app.com”

AppGestion

[Análisis de ventas](#) [Gestion Empleados](#) [Gestion Lugares](#) [Gestion Productos](#)

Alta Sección

Lugares gestionados

Mostrando 10 elementos

Buscar:

ID	NOMBRE	DIRECCION	TELEFONO	
7	MODA HOMBRE - PLANTA 1 - CCMARINEDA	CALLE MARINEDA SN	981000006	Dar de baja
8	MODA MUJER - PLANTA 1 - CCMARINEDA	CALLE MARINEDA SN	981000007	Dar de baja

Mostrando 1 a 2 de 2 elementos

Anterior

1

Siguiente

Figura 8.15: Captura de pantalla donde se muestra la pestaña Gestión de Lugares del usuario “sergio@app.com”

AppGestion

[Análisis de ventas](#)
[Gestion Empleados](#)
[Gestion Lugares](#)
[Gestion Productos](#)

Alta nuevo producto

Registrar entrada de stock

Productos y stock existentes en CCMARINEDA

Mostrando 10 elementos

Buscar:

ID	NOMBRE	PRECIO	STOCK	
1	PANTALON LANA	19.95	201	Dar de baja
2	PANTALON VAQUERO	29.95	200	Dar de baja

Mostrando 1 a 2 de 2 elementos

Anterior

1

Siguiente

Productos disponibles en el sistema

Mostrando 10 elementos

Buscar:

ID	NOMBRE	PRECIO		
1	PANTALON LANA	19.95	Cambiar de precio	Dar de baja
2	PANTALON VAQUERO	29.95	Cambiar de precio	Dar de baja
3	PANTALON PANA	9.95	Cambiar de precio	Dar de baja
4	CHAQUETA LANA	15.95	Cambiar de precio	Dar de baja
5	CHAQUETA VAQUERA	25.95	Cambiar de precio	Dar de baja

© AppGestion | Usuario logueado: pablo@app.com | Puesto: JEFE CENTRO en CCMARINEDA | [Deslogearse](#)

Figura 8.16: Captura de pantalla donde se muestra la pestaña Gestión de Productos del usuario “pablo@app.com”

AppGestion

[Análisis de ventas](#)
[Gestion Empleados](#)
[Gestion Lugares](#)
[Gestion Productos](#)

[Alta nuevo producto](#)
[Registrar entrada de stock](#)

Productos y stock existentes en PLANTA 1 - CCMARINEDA

Mostrando 10 elementos Buscar:

ID	NOMBRE	PRECIO	STOCK	
1	PANTALON LANA	19.95	100	Dar de baja
2	PANTALON VAQUERO	29.95	100	Dar de baja

Mostrando 1 a 2 de 2 elementos
[Anterior](#)
[1](#)
[Siguiete](#)

Productos disponibles en el sistema

Mostrando 10 elementos Buscar:

ID	NOMBRE	PRECIO		
1	PANTALON LANA	19.95	Cambiar de precio	Dar de baja
2	PANTALON VAQUERO	29.95	Cambiar de precio	Dar de baja
3	PANTALON PANA	9.95	Cambiar de precio	Dar de baja
4	CHAQUETA LANA	15.95	Cambiar de precio	Dar de baja
5	CHAQUETA VAQUERA	25.95	Cambiar de precio	Dar de baja

© AppGestion | Usuario logueado: sergio@app.com | Puesto: JEFE PLANTA en PLANTA 1 - CCMARINEDA | [Deslogearse](#)

Figura 8.17: Captura de pantalla donde se muestra la pestaña Gestión de Productos del usuario “sergio@app.com”

AppGestion

[Análisis de ventas](#)
[Gestión Empleados](#)
[Gestión Lugares](#)
[Gestión Productos](#)

Alta nuevo producto

Registrar entrada de stock

Productos y stock existentes en PLANTA 2 - CCMARINEDA

Mostrando 10 elementos

Buscar:

ID	NOMBRE	PRECIO	STOCK	
1	PANTALON LANA	19.95	101	Dar de baja
2	PANTALON VAQUERO	29.95	100	Dar de baja

Mostrando 1 a 2 de 2 elementos

Anterior

1

Siguiente

Productos disponibles en el sistema

Mostrando 10 elementos

Buscar:

ID	NOMBRE	PRECIO		
1	PANTALON LANA	19.95	Cambiar de precio	Dar de baja
2	PANTALON VAQUERO	29.95	Cambiar de precio	Dar de baja
3	PANTALON PANA	9.95	Cambiar de precio	Dar de baja
4	CHAQUETA LANA	15.95	Cambiar de precio	Dar de baja
5	CHAQUETA VAQUERA	25.95	Cambiar de precio	Dar de baja

© AppGestion | Usuario logueado: pedro@app.com | Puesto: JEFE PLANTA en PLANTA 2 - CCMARINEDA | [Desloguearse](#)

Figura 8.18: Captura de pantalla donde se muestra la pestaña Gestión de Productos del usuario “pedro@app.com”

AppGestion

[Análisis de ventas](#)

[Registrar Venta](#)

[Gestion Productos](#)

Nueva Venta

Introduzca el ID del artículo y su cantidad	PANTALON VAQUERO (29.95 \$)	▼	1	Add
ARTICULO: PANTALON LANA (19.95 \$) CANTIDAD: 1				
ARTICULO: PANTALON VAQUERO (29.95 \$) CANTIDAD: 1				
Confirmar Venta				

Figura 8.19: Captura de pantalla donde se muestra la pestaña Registrar Venta del usuario “miguel@app.com”

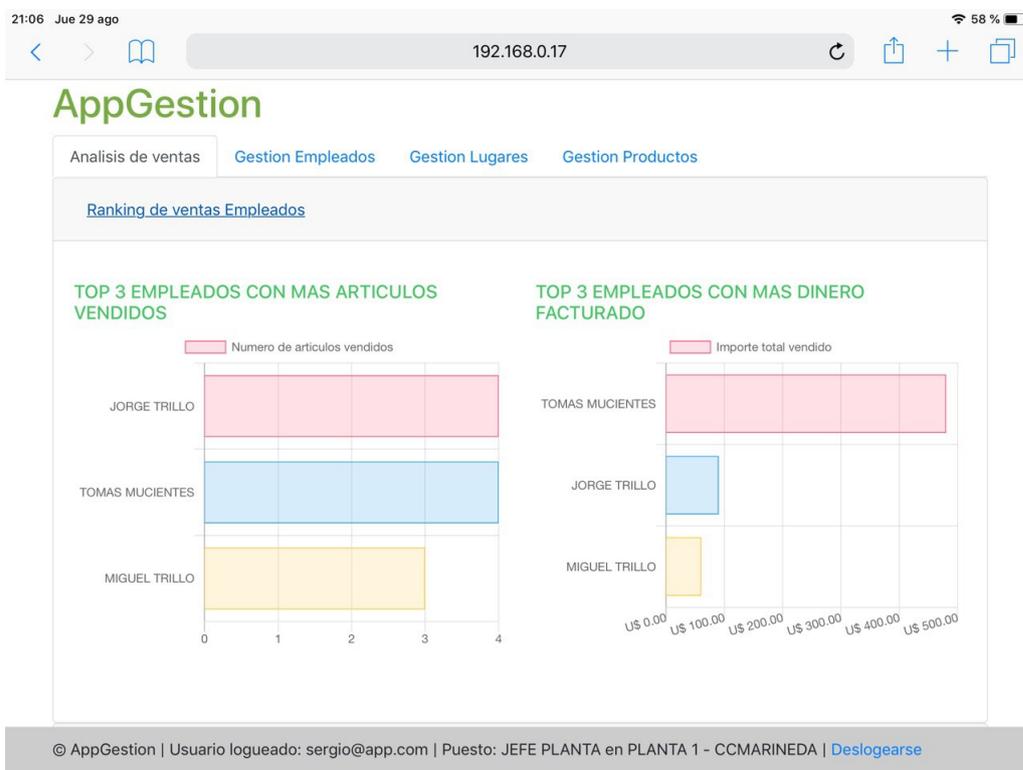


Figura 8.20: Captura de pantalla donde se muestra la aplicación en una tablet



Figura 8.21: Captura de pantalla donde se muestra la aplicación en un smartphone

EN este capítulo se indican las pruebas que se han realizado a lo largo del desarrollo de la aplicación.

En un proyecto de desarrollo de software pueden aparecer errores en cualquiera de las etapas del ciclo de vida, algunos de ellos incluso permanecen sin ser descubiertos, de ahí la importancia de las pruebas.

9.1 Pruebas unitarias

En esta aplicación se han realizado pruebas unitarias en dos de las tres capas existentes de la parte servidor (DAOs y controlador), abarcando la mayoría de métodos implementados, centrándose especialmente en aquellos de mayor complejidad puesto que son propensos a errores. Se han utilizado las herramientas JUnit, para las pruebas unitarias del código relativo a los DAOs y Postman para las peticiones publicadas en la API Rest. A continuación se incluyen ejemplos de una prueba de cada capa.

Ejemplo de pruebas de métodos en un dao con la herramienta JUnit:

```
1 @RunWith(SpringRunner.class)
2 @DataJpaTest
3 @AutoConfigureTestDatabase(replace = Replace.NONE)
4 public class ProductoTest {
5
6     @Autowired
7     ProductoDAO productoDao;
8
9     @Test
10    public void updateProductoDAO() {
11        final Producto producto = this.productoDao.findById(1).get();
12        producto.setPrecio(99.99);
13        this.productoDao.save(producto);
14        assertEquals(this.productoDao.findById(1).get(), producto);
15    }
16 }
```

```

15 }
16
17 ...
18 }

```

Ejemplo de prueba de métodos del controlador REST con la herramienta Postman en la figura 9.1.

The screenshot shows a Postman interface for a GET request to `localhost:8080/productos`. The response is a JSON array of 6 product objects. The status is 200 OK, with a response time of 572ms and a size of 707 B.

```

1  [
2  {
3    "productoId": 1,
4    "nombreProducto": "PANTALÓN LANA",
5    "precio": 19.95
6  },
7  {
8    "productoId": 2,
9    "nombreProducto": "PANTALÓN VAQUERO",
10   "precio": 29.95
11  },
12  {
13   "productoId": 3,
14   "nombreProducto": "PANTALÓN PANA",
15   "precio": 9.95
16  },
17  {
18   "productoId": 4,
19   "nombreProducto": "CHAQUETA LANA",
20   "precio": 15.95
21  },
22  {
23   "productoId": 5,
24   "nombreProducto": "CHAQUETA VAQUERA",
25   "precio": 25.95
26  },
27  {
28   "productoId": 6,
29   "nombreProducto": "CHAQUETA PANA",
30   "precio": 5.95
31  }
32 ]

```

Figura 9.1: Captura de pantalla de una petición GET a la url `/productos`

9.2 Pruebas de aceptación

Al final de cada una de las tres iteraciones se realizaron pruebas de aceptación por parte del alumno y el director de proyecto sobre los módulos finalizados. Estas pruebas se realizan sobre el sistema de manera manual abarcando todos los requisitos establecidos.

9.3 Pruebas de integración

Se han realizado pruebas de integración entre el servidor y el cliente de la aplicación en cada una de las iteraciones, para comprobar que la comunicación entre ellos era la correcta.

9.4 Otras pruebas

No se estimó oportuno la realización de otro tipo de pruebas, como podrían ser pruebas de carga, puesto que la aplicación no ha sido desplegada en un servidor público.

Conclusiones

EL objetivo de este proyecto es el desarrollo de una aplicación web para la gestión de unos grandes almacenes. Se ha desarrollado un software que provee a los usuarios de las herramientas necesarias para la gestión de empleados, lugares y productos, análisis y registro de ventas. Todo ello con una interfaz amigable e intuitiva que reduce enormemente la curva de aprendizaje media de una aplicación de este tipo.

Se han cumplido los objetivos especificados en la sección 1.2, desarrollando una aplicación web que permite a los usuarios gestionar los procesos relativos a:

- **Gestión de Empleados:** se ha desarrollado un módulo que dota al usuario de las funcionalidades necesarias para la gestión de sus subordinados, permite dar de baja, modificar el puesto de trabajo o dar de alta a cualquier empleado bajo su jerarquía de mando.
- **Gestión de Lugares:** se ha desarrollado un módulo que permite al usuario gestionar los lugares que se encuentran bajo su jerarquía de mando, realizando altas de nuevos lugares y bajas de lugares ya existentes.
- **Gestión de Productos:** se ha implementado un módulo que proporciona al usuario las herramientas necesarias para dar de alta o de baja nuevos productos en el sistema, así como registrar la entrada de stock en cualquiera de las secciones de las que sea responsable o se encuentre asignado.
- **Análisis de Ventas:** se ha desarrollado un módulo de análisis que muestra al usuario unas gráficas interactivas con datos en línea de las ventas registradas en el sistema, con un alto nivel de detalle .
- **Registro de Ventas:** se ha implementado un módulo que permite a los usuarios con el rol Vendedor registrar ventas en el sistema.

En cuanto a los conocimientos adquiridos a lo largo del desarrollo de este trabajo, destacan la planificación de un proyecto, la estimación de costes del mismo, el aprendizaje de nuevas

tecnologías y la aplicación de una metodología de desarrollo.

En resumen, la experiencia ha resultado de lo más satisfactoria, no sólo por los conocimientos adquiridos a lo largo del desarrollo de este proyecto, sino por el esfuerzo y la dedicación durante estos meses que se ven recompensados al alcanzar los objetivos que se marcaron en un principio.

Líneas futuras

EN el último capítulo de esta memoria, se exponen algunas de líneas de trabajo futuras que podrían llegar a incluirse en la aplicación para dotarla de un mayor número de funcionalidades.

- Internacionalización de la aplicación, puesto que la versión entregada tan sólo está disponible en castellano.
- Proporcionar una herramienta de generación de informes de ventas en formato PDF.
- Añadir geolocalización a los lugares y permitir visualizarlos en visores de mapas interactivos (por ej. Google Maps).

Bibliografía

- [1] R. A. Elmasri and S. B. Navathe, *Fundamentos de Sistemas de Bases de Datos*. Addison Wesley, 2007.
- [2] R. Cadenhead, *Java 8*. Anaya, 2014.
- [3] C. Walls, *Spring in Action*. Manning, 2018.
- [4] —, *Spring Boot in Action*. Manning, 2015.
- [5] V. autores, “Baeldung,” 2019. [En línea]. Disponible en: <https://www.baeldung.com/>
- [6] L. Atencio, *The Joy of JavaScript*. Manning, 2019.
- [7] V. autores, “W3schools. the world’s largest web developer site,” 2019. [En línea]. Disponible en: <https://w3schools.com/>
- [8] —, “Thymeleaf,” 2019. [En línea]. Disponible en: <https://www.thymeleaf.org/>
- [9] —, “Bootstrap. the most popular html, css and js library in the web,” 2019. [En línea]. Disponible en: <https://getbootstrap.com/>
- [10] —, “Datatables. table plug-in for jquery,” 2019. [En línea]. Disponible en: <https://datatables.net/>
- [11] —, “Chart.js. simple yet flexible javascript charting for designers and developers,” 2019. [En línea]. Disponible en: <https://chartjs.org/>
- [12] V. Massol, *jUnit in Action*. Manning, 2004.
- [13] J. Rumbaugh, I. Jacobson, and G. Boock, *El lenguaje unificado de modelado. Manual de referencia*. Addison Wesley, 2006.
- [14] I. Jacobson, G. Booch, and J. Rumbaugh, *El proceso unificado de desarrollo de software*, 2000.

