



Universidad Valladolid

Escuela de Ingeniería Informática
TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención Ingeniería de software

**Aplicación para la gestión de stock
de medianas y pequeñas empresas.**

Alumno: Mario Rodríguez Quintana
Tutor: Irene Lavín Perrino

Agradecimientos

Estos últimos años de paso por la Universidad de Valladolid me han enseñado lo necesario para poder valerme por mí solo en el entorno laboral, además de haberme aportado conocimientos que son incluso útiles en el ámbito personal, tanto académicos como de carácter personal.

Este proyecto es el fin de una etapa de mi vida, y desde aquí quiero agradecer a todas las personas que me han acompañado a lo largo de este camino, tanto profesores como compañeros con los que he podido compartir mucho tiempo durante todos estos años y aprender de ellos tanto en buenos como en malos momentos.

Agradecer también a la tutora de este proyecto su ayuda tanto en la realización del proyecto como de la corrección de esta memoria, que ha puesto empeño e interés durante todos estos últimos meses para que el proyecto llegase a buen puerto.

Resumen

En el mercado existen muchas alternativas para gestionar una tienda, siempre que se haga referencia a la parte digital de la misma, es decir a cómo está construida y con qué tecnologías, casi todas ellas forman parte de los denominados ERP, o sistema de planificación de los recursos empresariales.

Estos sistemas suelen ser muy grandes, complejos y con muchísimas funciones, por lo tanto, están basados en servicios de suscripción mensual. Sin embargo, la mayoría de las tiendas sobre todo pequeñas o pequeños negocios online no necesitan tanta funcionalidad.

La propuesta del presente TFG es un sistema de gestión de stock para las tiendas, en las que puedan registrar con un escáner físico todo lo que entra y tener una descripción de las características, tallas, precio, etc. De manera adicional se dispondrá de un histórico tanto de cuando se edita, se registra una nueva entrada, etc. Con todo ello se consigue tener toda la información relativa al stock de un vistazo.

El sistema estará implementado mediante una aplicación web, realizada con Spring Boot y Angular junto con la construcción de una API REST. La propia empresa podrá crear una cuenta donde tendrá la posibilidad de dar de alta a los empleados que gestionan la aplicación. Asimismo, a través de un formulario, los empleados podrán introducir los datos necesarios para dar de alta y gestionar los productos de la empresa. La persistencia se gestionará con una base de datos que refleja las entidades relacionadas con esta aplicación, como pueden ser la empresa, empleados, productos, etc. junto con otras para la propia gestión de la aplicación.

Abstract

The market is full of alternatives of managing the digital part of a store, most of these alternatives are usually part of big software known as ERP or Enterprise Resource Planning.

These resources are usually big and complex systems, that have lots of functionality and are most of the time subscription-based. But a small business does not need that much functionality.

The proposal of this project is a stockage system for these types of stores, in which they can scan new products and register that item and its features in a database. They will also have the possibility to record when the item was sold, edit prices for the items, register new entries... So, the store can have all the stock information centralized in a simple way.

The service will be made in a website, that is made with Spring Boot and Angular through a REST API. The store will be able to register employees for the store, and they will be able to control all the products of the store. The persistence of the system will be made with a database with all the entities of the application, for example the employees, the store, the products... As well as other data needed for the management of the system.

Tabla de contenidos

1	INTRODUCCIÓN.....	1
1.1	INTRODUCCIÓN	1
1.2	MOTIVACIÓN.....	1
1.3	OBJETIVOS	2
2	ESTADO DEL ARTE	3
2.1	COMPUTACIÓN EN LA NUBE	3
2.1.1	<i>Modelos/implementaciones</i>	3
2.2	ANGULAR.....	5
2.2.1	<i>Principales características</i>	5
2.3	SPRING BOOT	6
2.4	API REST	7
2.4.1	<i>Arquitectura RESTful</i>	7
2.5	MYSQL.....	8
2.6	CONTROL DE VERSIONES	8
2.6.1	<i>Git</i>	8
2.7	IMPLICACIÓN EN EL PROYECTO	8
2.8	LINGUAJES DE PROGRAMACIÓN	9
2.8.1	<i>TypeScript</i>	9
2.8.2	<i>Java</i>	9
3	HERRAMIENTAS UTILIZADAS.....	10
3.1	GITLAB.....	10
3.2	ASTAH	11
3.3	MYSQL WORKBENCH.....	12
3.4	VISUAL STUDIO CODE	12
3.5	ECLIPSE	13
3.6	OFFICE 365.....	14
3.7	POSTMAN	14
3.8	AWS.....	15
4	PLAN DE DESARROLLO	16
4.1	METODOLOGÍA	16
4.2	SCRUM.....	16
4.3	PLANIFICACIÓN.....	17
4.3.1	<i>Calendario detallado</i>	19
4.4	RIESGOS Y SU GESTIÓN	25
4.4.1	<i>Tipos de riesgos</i>	27
4.4.2	<i>Riesgos conocidos</i>	27
4.5	IMPREVISTOS EN PLANIFICACIÓN	31
4.6	ANÁLISIS DE COSTES.....	31
4.6.1	<i>Hardware</i>	31
4.6.2	<i>Software</i>	31
4.6.3	<i>Personal</i>	31
4.6.4	<i>Coste inicial proyecto</i>	32
5	ANÁLISIS DEL PROYECTO.....	33
5.1	REQUISITOS	33
5.1.1	<i>Requisitos funcionales</i>	33
5.1.2	<i>Requisitos no funcionales</i>	34

5.2	HISTORIAS DE USUARIO	34
6	DISEÑO.....	42
6.1	CASOS DE USO.....	42
6.1.1	<i>Diagrama casos de uso.....</i>	42
6.1.2	<i>Detalle casos de uso</i>	44
6.2	MODELO CONCEPTUAL	49
6.2.1	<i>Modelo inicial.....</i>	49
6.2.2	<i>Modelo final.....</i>	49
6.3	PATRONES DE DISEÑO	50
6.3.1	<i>Patrón DAO.....</i>	50
6.3.2	<i>Patrón DTO.....</i>	51
6.3.3	<i>Patrón Factoría</i>	51
6.3.4	<i>Patrón MVC.....</i>	52
6.4	DIAGRAMA JERÁRQUICO	54
7	IMPLEMENTACIÓN	56
7.1	INTRODUCCIÓN	56
7.2	ARQUITECTURA	56
7.2.1	<i>Basic Auth.....</i>	56
7.2.2	<i>Formato peticiones de las operaciones</i>	56
8	DESPLIEGUE	68
9	CONCLUSIONES Y TRABAJO FUTURO.....	72
9.1	CONCLUSIONES	72
9.2	TRABAJO FUTURO	72
10	MANUALES.....	74
10.1	INICIO DE SESIÓN Y MENÚ PRINCIPAL	74
10.2	ACCESOS SIEMPRE DISPONIBLES.....	75
10.3	MANUALES.....	76
10.4	PRODUCTOS.....	76
10.5	MOVIMIENTOS	77
10.6	USUARIOS	78
10.7	CATEGORÍAS	80
10.8	MODIFICACIÓN RÁPIDA.....	81
10.9	FEEDBACK PANTALLAS.....	81
11	BIBLIOGRAFÍA	83

Índice de ilustraciones

FIGURA 1 RELACIÓN DE DEPENDENCIA ENTRE CADA TIPO DE SERVICIO.....	3
FIGURA 2 DIVISIÓN DEL ALCANCE DE CADA MODELO	5
FIGURA 3 EJEMPLO DE DOM REGULAR DE UNA PÁGINA	5
FIGURA 4 BASE ARQUITECTURA ANGULAR	6
FIGURA 5 ESQUEMA FUNCIONAMIENTO REST	7
FIGURA 6 USO LENGUAJES PROGRAMACIÓN EN 2019.....	10
FIGURA 7 PANTALLA PRINCIPAL DE UN PROYECTO EN GITLAB.....	11
FIGURA 8 EJEMPLO DIAGRAMA DE CLASES EN ASTAH.....	11
FIGURA 9 EJEMPLO BASE DE DATOS EN MYSQL WORKBENCH	12
FIGURA 10 INTERFAZ VISUAL STUDIO CODE	13
FIGURA 11 INTERFAZ ECLIPSE CON SPRING TOOLS INSTALADO	13
FIGURA 12 INTERFAZ MICROSOFT WORD EN OFFICE 365	14
FIGURA 13 INTERFAZ POSTMAN	15
FIGURA 14 ESQUEMA SCRUM.....	17
FIGURA 15 ESQUEMA GESTIÓN RIESGOS.....	26
FIGURA 16 MATRIZ DE RIESGOS GENERAL.....	26
FIGURA 17 RESUMEN COSTES PROYECTO	32
FIGURA 18 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	43
FIGURA 19 MODELO CONCEPTUAL INICIAL	49
FIGURA 20 MODELO CONCEPTUAL FINAL.....	50
FIGURA 21 EJEMPLO CLASES DAO	51
FIGURA 22 EJEMPLO DTO	51
FIGURA 23 ESTRUCTURA FACTORÍA	52
FIGURA 24 ESQUEMA MVC.....	53
FIGURA 25 DIAGRAMA JERÁRQUICO DEL PROYECTO.....	54
FIGURA 26 PANEL USUARIO DIGITALOCEAN.....	68
FIGURA 27 LOGS DIGITALOCEAN.....	69
FIGURA 28 PANEL ELASTIC BEANSTALK AWS.....	70
FIGURA 29 PANEL OBJETOS BUCKET AMAZON S3.....	70
FIGURA 30 AMAZON S3 ALOJAMIENTO DE SITIO WEB ESTÁTICO	71
FIGURA 31 AMAZON S3 CONFIGURACIÓN CORS	71
FIGURA 32 PANTALLA INICIO SESIÓN	74
FIGURA 33 PANTALLA REGISTRO EMPRESA.....	74
FIGURA 34 MENÚ PRINCIPAL.....	75
FIGURA 35 PANTALLA VER PERFIL.....	75
FIGURA 36 PANTALLA MANUALES.....	76
FIGURA 37 LISTADO PRODUCTOS.....	76
FIGURA 38 PANTALLA EDICIÓN PRODUCTO.....	77
FIGURA 39 PANTALLA CREAR NUEVO PRODUCTO	77
FIGURA 40 LISTADO MOVIMIENTOS.....	78
FIGURA 41 PANTALLA DETALLE MOVIMIENTO	78
FIGURA 42 LISTADO USUARIOS.....	79
FIGURA 43 REGISTRO NUEVO USUARIO	79
FIGURA 44 DETALLE USUARIO.....	80
FIGURA 45 LISTADO CATEGORÍAS	80
FIGURA 46 PANTALLA MODIFICACIÓN RÁPIDA.....	81
FIGURA 47 EJEMPLO MENSAJE ERROR	81
FIGURA 48 EJEMPLO MENSAJE REGISTRO	82
FIGURA 49 EJEMPLO MENSAJE INFORMATIVO	82
FIGURA 50 MENSAJE REGISTRO CORRECTO.....	82
FIGURA 51 MENSAJE ERROR EN REGISTRO.....	83

Índice de tablas

TABLA 1 APROXIMACIÓN TIEMPO ESTIMADO EN SPRINTS	18
TABLA 2 CALENDARIO DETALLADO	19
TABLA 3 H.U. SPRINT 0.....	20
TABLA 4 H.U. SPRINT 1.....	21
TABLA 5 H.U. SPRINT 2.....	22
TABLA 6 H.U. SPRINT 3.....	23
TABLA 7 H.U. SPRINT 4.....	24
TABLA 8 H.U. SPRINT 5.....	25
TABLA 9 RESUMEN RIESGOS	27
TABLA 10 RIESGO 1.....	28
TABLA 11 RIESGO 2.....	28
TABLA 12 RIESGO 3.....	28
TABLA 13 RIESGO 4.....	29
TABLA 14 RIESGO 5.....	29
TABLA 15 RIESGO 6.....	29
TABLA 16 RIESGO 7.....	30
TABLA 17 RIESGO 8.....	30
TABLA 18 MATRIZ DE RIESGOS DEL PROYECTO.....	30
TABLA 19 COSTE INICIAL PROYECTO.....	32
TABLA 20 HISTORIA DE USUARIO 1- INICIO PROYECTO.	35
TABLA 21 HISTORIA DE USUARIO 2- ESTUDIO TECNOLOGÍAS USADAS.	35
TABLA 22 HISTORIA DE USUARIO 3- DIVISIÓN DEL TIEMPO Y MÉTODOS DE DESARROLLO.....	35
TABLA 23 HISTORIA DE USUARIO 4- VIABILIDAD DEL PROYECTO.....	35
TABLA 24 HISTORIA DE USUARIO 5- REQUISITOS FUNCIONALES.	36
TABLA 25 HISTORIA DE USUARIO 6- REQUISITOS NO FUNCIONALES.	36
TABLA 26 HISTORIA DE USUARIO 7- CREACIÓN MODELO DE DATOS.	36
TABLA 27 HISTORIA DE USUARIO 8- APLICAR MODELO EN BASE DE DATOS.	36
TABLA 28 HISTORIA DE USUARIO 9- CREACIÓN INICIAL DESARROLLO DEL PROYECTO.	37
TABLA 29 HISTORIA DE USUARIO 10- CONEXIÓN BACKEND CON BASE DE DATOS.	37
TABLA 30 HISTORIA DE USUARIO 11- CREACIÓN ENTITIES.	37
TABLA 31 HISTORIA DE USUARIO 12- CREACIÓN REPOSITORIES.....	37
TABLA 32 HISTORIA DE USUARIO 13- CREACIÓN SERVICIOS.....	38
TABLA 33 HISTORIA DE USUARIO 14- CREACIÓN SISTEMA AUTENTIFICACIÓN.....	38
TABLA 34 HISTORIA DE USUARIO 15-COMPROBACIÓN Y PRUEBAS BACKEND.	38
TABLA 35 HISTORIA DE USUARIO 16- SISTEMA INICIO SESIÓN Y PERFIL.	39
TABLA 36 HISTORIA DE USUARIO 17- PANTALLAS PRODUCTOS.	39
TABLA 37 HISTORIA DE USUARIO 18- PANTALLAS MOVIMIENTOS.	39
TABLA 38 HISTORIA DE USUARIO 19- PANTALLAS DE USUARIOS.....	40
TABLA 39 HISTORIA DE USUARIO 20- NAVEGACIÓN Y MENÚS PERSONALES.	40
TABLA 40 HISTORIA DE USUARIO 21- PANTALLAS DE APOYO.....	40
TABLA 41 HISTORIA DE USUARIO 22 - PRUEBAS Y ARREGLOS SISTEMA.....	40
TABLA 42 HISTORIA DE USUARIO 23 -DESPLIEGUE SISTEMA	41
TABLA 43 HISTORIA DE USUARIO 24 - DOCUMENTACIÓN.....	41
TABLA 44 C.U.01 INICIO SESIÓN	44
TABLA 45 CU.02 VER CATEGORÍAS SISTEMA.....	44
TABLA 46 CU.03 CERRAR SESIÓN.....	45
TABLA 47 CU.04 VER PERFIL.....	45
TABLA 48 CU.05 MODIFICAR UNIDADES DEL PRODUCTO.....	45
TABLA 49 CU.06 VER MOVIMIENTOS DEL SISTEMA.	46
TABLA 50 CU.07 VER USUARIOS DEL SISTEMA.	46
TABLA 51 CU.08 VER DETALLE USUARIO.....	47
TABLA 52 BORRAR USUARIO.....	47
TABLA 53 CU.10 VER PRODUCTOS DEL SISTEMA.....	48
TABLA 54 CU.11 BORRAR PRODUCTO.	48

1 INTRODUCCIÓN

1.1 INTRODUCCIÓN

Este trabajo final de estudios se llevará a cabo una aplicación web que permita a las tiendas gestionar de forma sencilla el stock, principalmente centrado para tiendas pequeñas, como pueden ser tiendas de barrio que quieren tener una solución digital y gratuita para tener todo centralizado.

Muchas de las alternativas que tenemos en el mercado se basan en las denominadas ERP [1], que normalmente son sistemas masivos y con una complejidad muy alta, que requieren de gente con experiencia para usarlo, por lo que la viabilidad de uso para pequeñas empresas es casi inexistente.

Por todo ello se busca que la aplicación web implementada sea una aplicación web sencilla, usable y que pueda ser utilizada por usuarios no técnicos o sin formación específica en materia de nuevas tecnologías. Adicionalmente se dispondrá de una opción para registrar una nueva empresa, así como registrar empleados, productos y venta. De este modo se consigue tener todo digitalizado en un mismo lugar.

Por otro lado, la aplicación permitirá, a través de un modo rápido poder escanear productos directamente y dispondrá de un modo de uso *avanzado* que permitirá añadir o quitar productos y usuarios con detalle, al igual que editar parámetros o precios de ciertos productos.

Se podrá realizar una clasificación de los productos y dicha clasificación tendrá el IVA de cada categoría, para poder saber las ganancias a la hora de la venta. Además de poder introducir el coste de compra de cada producto cuando en la tienda se reciba nueva mercancía.

También se dispone de la posibilidad de descargar historiales de movimientos detallados para ver el estado de nuestro negocio y el rendimiento económico del mismo en cualquier periodo de tiempo.

La aplicación web está hecha con tecnologías entre las que destacan Angular [2] y Spring Boot [3] mediante una API REST [4]. Todas estas tecnologías son bastante actuales y con una gran comunidad detrás apoyándolas, por lo que se disponen muchísimas opciones de mantenimiento de cara al futuro, además de que es mucho más sencillo resolver posibles problemas al haber abundante documentación.

1.2 MOTIVACIÓN

La motivación viene principalmente de un ámbito personal y de amigos cercanos de un pasado con experiencia profesional previa vendiendo algunos productos en plataformas en línea. Todos se encuentran con el mismo problema en cuanto se empieza a manejar cierta cantidad de productos, y es que no hay manera rápida y sencilla de tenerlo controlado.

Navegando por la red se pueden encontrar muchas alternativas, pero muchas requieren que se usen sus servicios o bien que se pague un plan de suscripción mensual como por ejemplo la web controldestock.com [5].

En las de primer tipo por ejemplo se destaca el caso de Ebay, tiene un excelente gestor, pero solo para cosas que vayas a vender en su plataforma con comisiones de casi el 12% por venta [6], las del segundo tipo todas son parecidas a la web citada, con una versión muy sencilla para un

usuario, muy pocas opciones y en cuanto quieres alguna opción más la factura mensual empieza a incrementar considerablemente.

En el caso de un ERP grande como los de Microsoft [7] o Oracle [8] los costes son simplemente inasumibles para empresas pequeñas con 1-2 empleados. La motivación de este trabajo es diseñar una alternativa gratuita y con la funcionalidad necesaria para este tipo de negocios.

El caso de los ERP pequeños suelen ser demasiado limitados en funciones, y en cuanto se quiere alguna función añadida suelen ser de pago, o bien si el volumen de productos es alto, limitan su uso. En el caso de este proyecto se ofrecerá el código para que cada empresa pueda disponer un pequeño servidor montado para uso personal, además de tener la opción de usar el servicio de forma pública sin coste.

1.3 OBJETIVOS

Con base en las motivaciones anteriormente descritas, El principal objetivo de este proyecto es desarrollar una aplicación web que permita la gestión de stock para empresas reuniendo las características de una implementación ligera, mantenimiento a coste cero y una curva de aprendizaje muy sencilla, para que cualquier persona incluso que no esté familiarizada con otros sistemas de gestión de stock sepan usarlo perfectamente. Se engloban los siguientes objetivos:

- Poder registrar los datos de la empresa, así como nuevos empleados y los datos de las ventas realizadas.
- Escanear productos a través de un escáner manual, pudiendo modificar atributos y precio, así como poder clasificarlos de acuerdo a las necesidades de la empresa.

Permitir asimismo tener un control y un histórico de los movimientos de la tienda por días y horas según los registros de entrada/salida de los productos.

2 ESTADO DEL ARTE

En este apartado se recopilan las principales tecnologías usadas en este proyecto, con detalles generales de las mismas y la razón de la incorporación de las mismas al proyecto.

2.1 COMPUTACIÓN EN LA NUBE

Esta es la base del proyecto, se le conoce mucho más por su nombre en inglés *Cloud Computing* y es el uso de servidores conectados a internet para poder procesar y almacenar datos, software, redes... Hoy en día todo está poco a poco avanzando hacia ser un mundo conectado, y la computación en la nube es el paradigma que lo engloba todo. [9]

2.1.1 Modelos/implementaciones

La computación en la nube contiene 3 modelos de servicio distintos que tienen como objetivo satisfacer diferentes requisitos de las empresas, dichos modelos se clasifican como SaaS, PaaS e IaaS que veremos en detalle a continuación. [10]

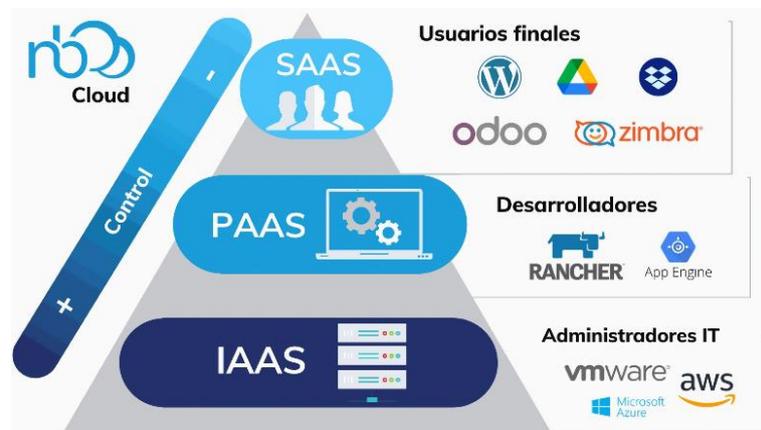


Figura 1 Relación de dependencia entre cada tipo de servicio de nanobytes [59]

Estos tres modelos lo que permiten es tener una mayor abstracción del sistema, de tal forma que parte de los servicios pueden ser gestionados por un tercero o un proveedor, lo que permite a la usar tecnologías comunes y abaratar costes tanto de desarrollo como de evolución si se necesitan futuros cambios.

En la Figura 1 se puede observar el control que tiene cada empresa según el tipo de servicio, la clasificación se suele representar con una estructura piramidal.

2.1.1.1 SaaS – Software as a service

En este tipo de servicio toda la infraestructura la tiene el proveedor. Se puede acceder a los servicios online normalmente a través de una web mediante un sistema de inicio de sesión o bien mediante un api de acceso para poder implementarlo en nuestros proyectos. En este tipo de

servicios el proveedor se encarga de todo y el usuario final solo tiene acceso a la capa final o la interfaz visible, por ejemplo, nuestro correo electrónico.

Actualmente las aplicaciones SaaS son las más populares ya que hacen que el cliente final tenga ya una base sobre la que hacer modificaciones o implementar funcionalidad en caso de necesitarla.

Se puede implementar con suscripciones mensuales, mediante publicidad... Eso depende de cada empresa. En nuestro caso el usuario final o las empresas que usen nuestro Software tendrán este tipo de servicios, donde nuestro proyecto será el proveedor.

2.1.1.2 PaaS – Platform as a service

Este tipo de servicio ofrece plataformas para los desarrolladores que quieren centrarse en el desarrollo e implementación de sus sistemas, pero teniendo cierto control sobre lo que está pasando por debajo.

Son muy útiles cuando necesitamos tener acceso a partes intermedias como por ejemplo bases de datos, sin tener que preocuparnos por cosas más complejas como herramientas de desarrollo, que ya vendrán incluidas por el proveedor. Es el punto medio entre SaaS e IaaS.

2.1.1.3 IaaS – Infrastructure as a service

En este tipo de servicios el proveedor solo ofrece la estructura base para montar nuestro proyecto, que normalmente son los servidores, almacenamiento y redes.

La ventaja que tienen es que tenemos una mayor flexibilidad y control sobre lo que queremos hacer, pudiendo ir implementando sobre esa base servicios diferentes sin necesidad de que sean los mismos que proporciona el proveedor, que además puede suponer un posterior ahorro en mantenimiento.

2.1.1.4 Resumen

La elección de cada tipo de servicio depende del proyecto que se vaya a realizar o las necesidades de cada uno. En este caso se va a usar un IaaS, con terceros como DigitalOcean [11] que será el encargado de proveer toda la infraestructura de la base de datos. De cara a las empresas o usuarios del proyecto todo será un SaaS, no tendrán acceso a la lógica interna.

En la siguiente figura se ve un pequeño resumen del alcance que tiene cada uno de estos servicios. Para cada servicio tenemos multitud de empresas que los ofertan, con algunas de ellas como Microsoft, Oracle o Google ofertando los tres de ellos dependiendo de lo que busquemos. [12]

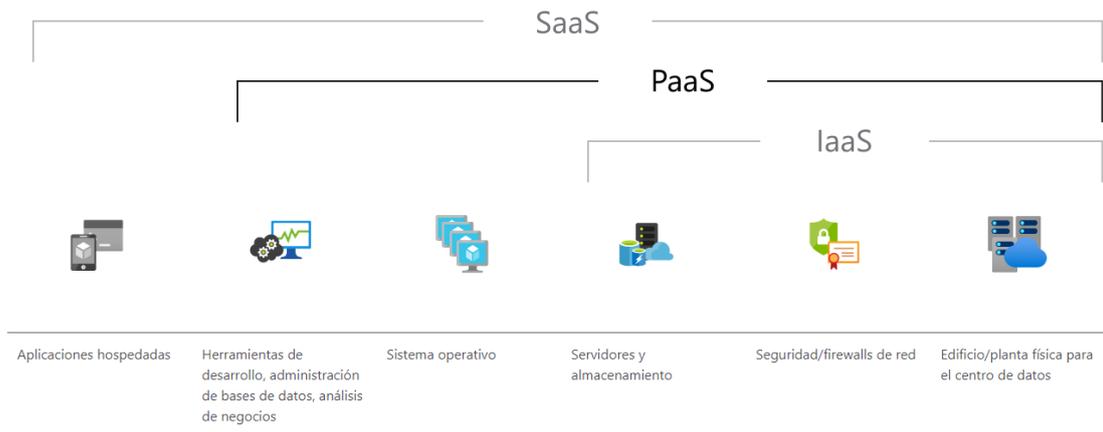


Figura 2 División del alcance de cada modelo [12]

2.2 ANGULAR

Este apartado se va a centrar en tecnologías específicas que se usan en este proyecto. Angular [13] es un framework de código abierto usado para desarrollar aplicaciones web, basado en TypeScript y que actualmente está mantenido por Google.

Su objetivo es realizar aplicaciones webs de una sola página, que estén basadas en el patrón MVC (Modelo-Vista-Controlador, separa cada una de estas partes en capas), tenemos muchas versiones de Angular, en este caso se usa la más moderna, que es Angular 12.

2.2.1 Principales características

Las principales características de Angular y algunas de las razones por las que se ha elegido son las siguientes:

- Angular usa DOM regular, es decir un documento XML o HTML que contiene una estructura de árbol en la que cada nodo es una parte del elemento.

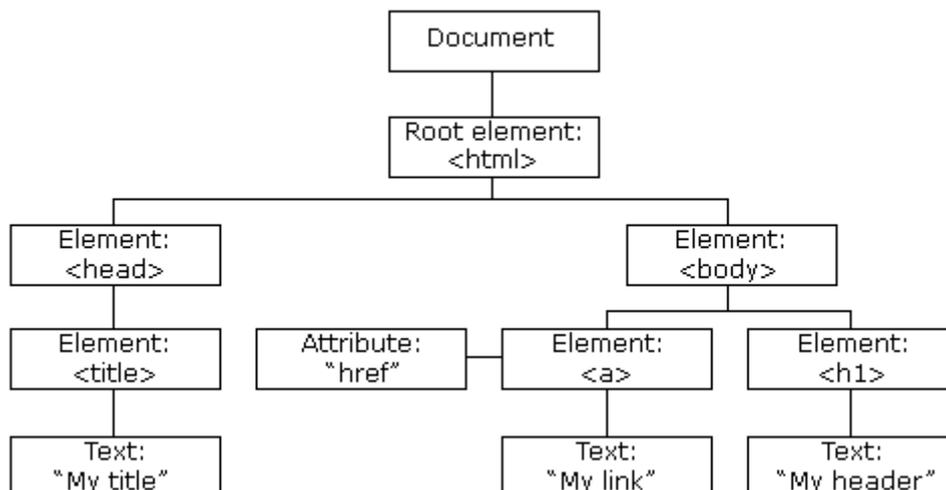


Figura 3 Ejemplo de DOM regular de una página de hiberus blog [14]

- TypeScript, que es un lenguaje de programación que veremos descrito en el apartado 2.9.1 con más detalle.

- Nos proporciona un enlace bidireccional, lo que denomina enlace de datos y permite conectar directamente el modelo de nuestra aplicación con la web, respondiendo de manera dinámica a los cambios que se realizan en la página.
- Angular tiene una arquitectura basada en módulos y componentes, tenemos un módulo raíz sobre el que se monta todo llamado AppModule, que contiene el mecanismo de arranque de la aplicación. Podemos montar luego componentes, que son fragmentos de código que tienen la lógica y datos de una aplicación definiendo una parte de la interfaz de usuario, se pueden importar y usar componentes de terceros.
- Tenemos la posibilidad de usar plantillas, que es una combinación entre Angular y HTML que nos permite modificar los elementos de un HTML antes de que se muestren.
- Se pueden usar servicios que nos permiten compartir información entre diferentes componentes o hacer peticiones http externas, por ejemplo, a APIs de terceros o a bases de datos, los servicios no están asociados a la vista.

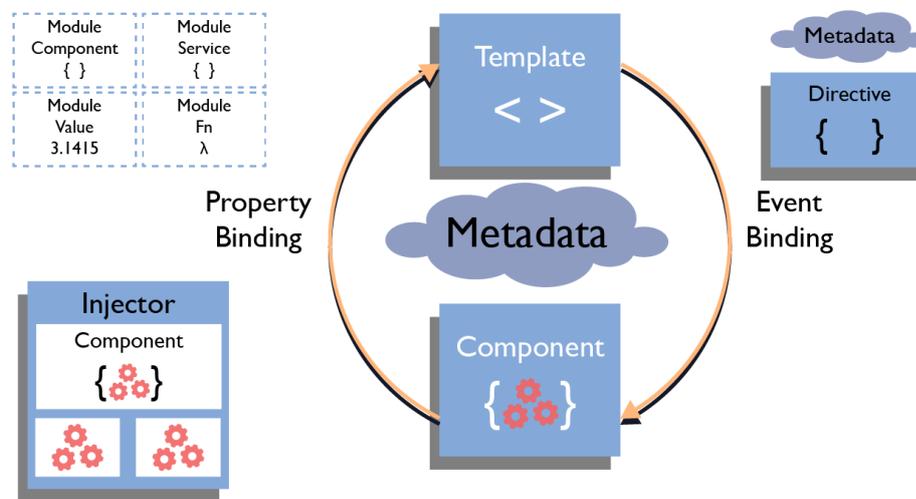


Figura 4 Base arquitectura Angular de Medium [58]

2.3 SPRING BOOT

Es una tecnología que permite crear aplicaciones autocontenidas (esto es, que cada aplicación internamente contiene su propio servidor de aplicaciones), lo que nos permite enfocarnos en el desarrollo de la aplicación y olvidarse de la arquitectura, ya que Spring Boot realiza labores como la configuración de dependencias, el despliegue del servicio o aplicar el proyecto a un servidor de aplicaciones.

Está basado en el framework de Spring, pero es una herramienta del framework que nos permite poder realizar una aplicación y desplegarla con un esfuerzo mínimo, quitándose la necesidad de aprender a usar dicho framework de forma exhaustiva. [15]

Para crear un proyecto y usarlo solo es necesario crear un proyecto Maven con las dependencias que se quieran añadir, desarrollar la aplicación y desplegarla en el servidor, el primer y el último paso se hace casi de manera automática por Spring Boot, solo es necesario centrarse en el desarrollo, esto hace que sea una herramienta muy popular hoy en día.

En el proyecto se va a integrar con una API REST, para poder recuperar la información del sistema en nuestra aplicación.

Hay que conocer tres elementos claves de Spring Boot que se van a mencionar en el futuro para comentar el progreso del proyecto:

- **Entity:** Una clase Java que representa datos que se pueden conservar en la base de datos. Cada entidad representa una tabla de dicha base de datos. Por ejemplo, si tenemos una tabla alumno, deberíamos tener una entidad alumno para poder usarlo con Spring Boot. [16]
- **Repository :** Un repository, conocido en español como repositorio, es un mecanismo para encapsular almacenamiento, recuperación de datos y búsqueda sobre un tipo de entidad definida anteriormente. Tenemos operaciones predefinidas para estos repositorios y podemos definir nuevas a nuestro gusto dependiendo de las necesidades del sistema. [17]
- **Service:** Un service, conocido en español como servicio, contiene la lógica de negocio del sistema. Se crea un servicio para consumir las operaciones de uno o varios repositorios y son el punto de acceso al sistema. [18]

2.4 API REST

Una API REST [19] es una interfaz de programación de aplicaciones que cumple los requisitos definidos por la arquitectura REST y permite la interacción con servicios web de la misma.

2.4.1 Arquitectura RESTful

Una arquitectura REST (Representational State Transfer) [17] contiene unos estándares para permitir la comunicación entre diferentes sistemas que están en la red. El cliente y el servidor en este tipo de arquitectura están separados, además es un paradigma sin estado o “stateless”, es decir, que el servidor no necesita saber en qué estado está el cliente y viceversa, por lo que se pueden entender entre ellos sin saber el historial del otro.

Las peticiones entre el cliente y el servidor se realizan mediante HTTP y hay 4 tipos diferentes depende del objetivo de la misma:

- **GET:** Para recuperar recursos.
- **POST:** Para crear un recurso.
- **PUT:** Para modificar un recurso existente.
- **DELETE:** Para eliminar un recurso

Con estos 4 métodos de petición en combinación con cabeceras HTTP podemos realizar cualquier acción necesaria y gestionar todos los recursos de nuestra aplicación.

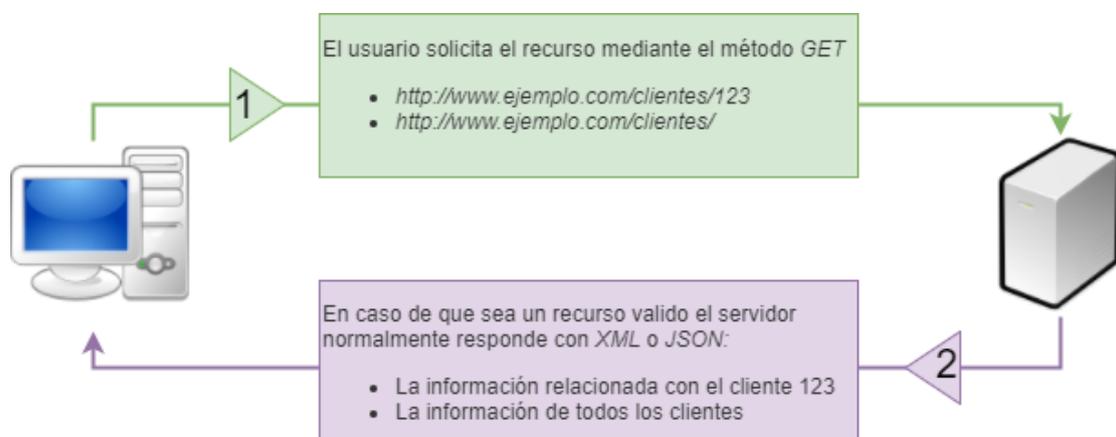


Figura 5 Esquema funcionamiento REST de bigeek [21]

2.5 MYSQL

MySQL [22] es un sistema de gestión de bases de datos relacional, tiene licencia de uso abierto y licencia comercial de Oracle, que fue el desarrollador original del mismo. Tiene muchísimas versiones desde su creación y es uno de los más populares actualmente, se usa en sitios como Google, Facebook...

En el proyecto se usa como el gestor de la base de datos del proyecto, mediante un hosting en DigitalOcean, que permite la creación de bases de datos de forma rápida y sencilla, solo introduciendo los datos de creación de la misma.

2.6 CONTROL DE VERSIONES

El control de versiones es la gestión de los cambios de un proyecto, consiste en ver los cambios que se van realizando y almacenar los mismos para poder seguir el desarrollo del mismo y poder volver a puntos anteriores del mismo en caso de problemas. [23]

También es una forma común de almacenamiento de datos para poder trabajar entre varios usuarios de forma concurrentes.

En el caso de este proyecto se va a usar GitLab, que es una muy popular herramienta basada en Git y que proporciona la propia facultad.

2.6.1 Git

Git es un proyecto de código abierto de control de versiones para software, es el más famoso y más usado en la actualidad debido a su potencial. Tiene una arquitectura distribuida, es decir que los cambios se almacenan en un repositorio remoto sobre el que pueden trabajar varias personas de forma concurrente. [24]

2.7 IMPLICACIÓN EN EL PROYECTO

En este proyecto se van a englobar todas las tecnologías mencionadas, el Frontend se va a realizar en Angular, en este caso en la última versión que es la 13. Para el Backend se va a usar Spring Boot y la comunicación entre ambas se realiza mediante una API REST, esta combinación de elementos es muy popular en la actualidad y da mucho margen a la hora de desarrollar, lo que además es positivo de cara a posibles mejoras futuras, es la principal razón por la que se ha elegido.

Los datos se almacenarán en una base de datos que se accederá desde el Backend. En este caso se usa MySQL por ser uno de los sistemas de gestión más utilizados, además de por ser uno con el que más familiarizado está el desarrollador principal del proyecto por su uso durante el periodo académico y laboral.

Todos los cambios que se vayan realizando durante el proyecto se irán subiendo a GitLab, para su revisión por el tutor y para tener un control de posibles errores y la posibilidad de volver a un código posterior.

Se usará Astah para realizar el modelo de la base de datos al igual que para hacer los diagramas UML necesarios, sobre todos los relativos a los diagramas de actividad.

2.8 LENGUAJES DE PROGRAMACIÓN

Angular y Spring Boot son tecnologías de desarrollo/Frameworks, pero por debajo necesitamos un lenguaje para poder desarrollar un proyecto en ellas, en este caso tenemos TypeScript y Java respectivamente:

2.8.1 TypeScript

Es un lenguaje de programación de licencia libre y código abierto que está desarrollado y mantenido por Microsoft, está basado en JavaScript. [25]

Se utiliza en Angular en nuestro proyecto en conjunto con HTML y CSS principalmente y es el encargado de la lógica de la parte del Frontend, donde normalmente se encontraría el fichero JavaScript.

Todo código JavaScript se puede usar como código TypeScript sin problemas. Las principales diferencias entre ambos lenguajes son las siguientes:

- TypeScript es un lenguaje fuertemente *tipado*, JavaScript no, esto quiere decir que en TypeScript podemos crear tipos e interfaces.
- *Modularización*, en TypeScript se incluye soporte nativo para módulos.
- Soporte para *tuplas* de TypeScript.
- *Orientación a objetos*, esto quiere decir que la sintaxis de Typescript nos permite crear objetos y poder utilizarlos de varias formas y en varios sitios, como se podría hacer en Java o C#.
- Soporte para *decoradores* en TypeScript, que son elementos que nos permiten cambiar dinámicamente el HTML.

Todo esto aporta muchísima mayor versatilidad a la hora de programar. Por otro lado, permite tener elementos como objetos hace que el código sea más legible y fácil de mantener.

2.8.2 Java

Java [23] es uno de los lenguajes más usados y potentes del mercado, en nuestro caso se va a utilizar en la parte del Backend justo con Spring Boot. La sintaxis es bastante sencilla sobre todo si se compara con otros lenguajes potentes como C o C++. Sus principales características son:

- **Orientación a objetos:** Se pueden crear objetos y reutilizarlos en varios lugares y de formas diferentes.
- **Distribuido:** Permite realizar muchas operaciones en red, con conexiones TCP/IP, protocolos HTTP, FTP... Es muy completo en este sentido.
- **Robusto y Seguro:** Busca errores en tiempo de compilación y ejecución, además no permite realizar accesos ilegales a memoria como si hacen por ejemplo C o C++.
- **Arquitectura neural y portabilidad:** Es decir, se compila de forma independiente de la maquina en la que se ejecuta, para poderse ejecutar en cualquier maquina con ese mismo sistema de ejecución, aunque tenga otro hardware.

La información online y la experiencia que se ha ganado durante todos estos años de carrera con este lenguaje hace que sea uno de los más fáciles para trabajar con él, por lo que es el principal lenguaje utilizado en este proyecto.

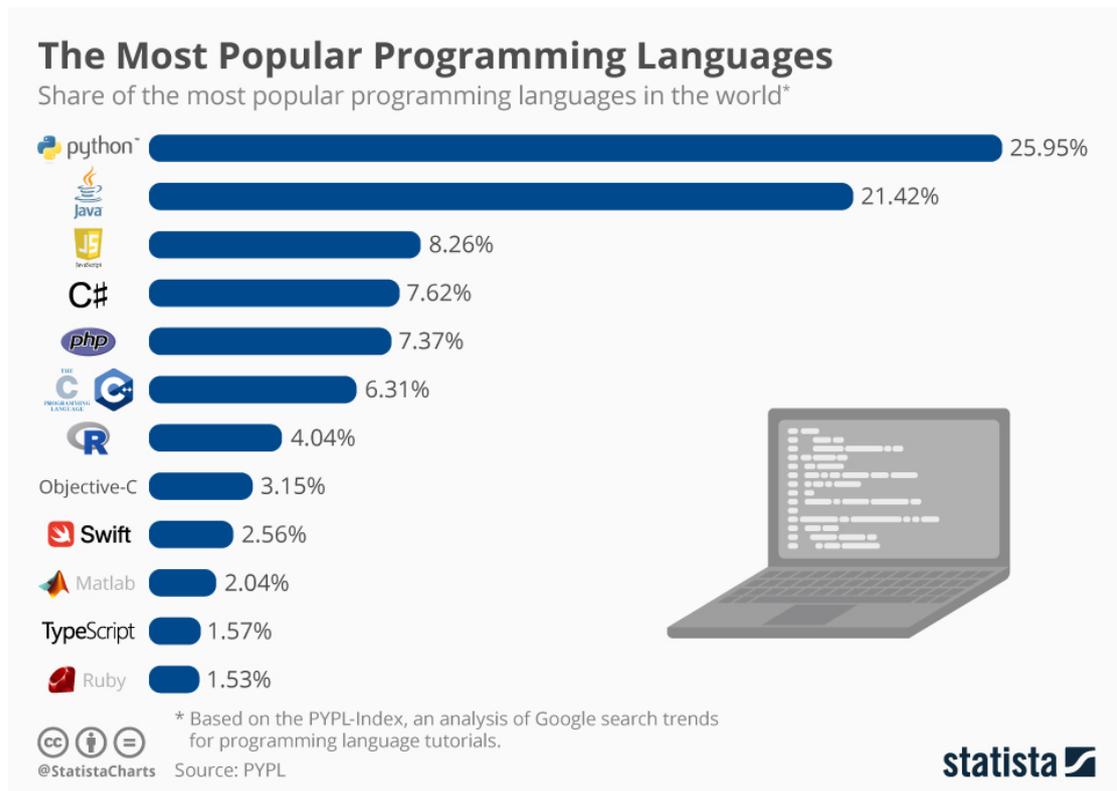


Figura 6 Uso lenguajes programación en 2019 de Statista [27]

3 HERRAMIENTAS UTILIZADAS

Todas las tecnologías usadas descritas en el apartado 2 requieren de unos programas o herramientas que nos permitan trabajar con ellas, en este apartado se desarrolla un pequeño resumen de los programas utilizados en el desarrollo del proyecto:

3.1 GITLAB

Gitlab es una suite que permite usar Git de una forma mucho más sencilla y visual, con una interfaz web muy completa y potente que nos permite visualizar cambios o realizar modificaciones en el repositorio Git de una forma mucho más rápida y sencilla. [20] Se ha elegido porque es la herramienta de control de versiones con la que se está más familiarizado.

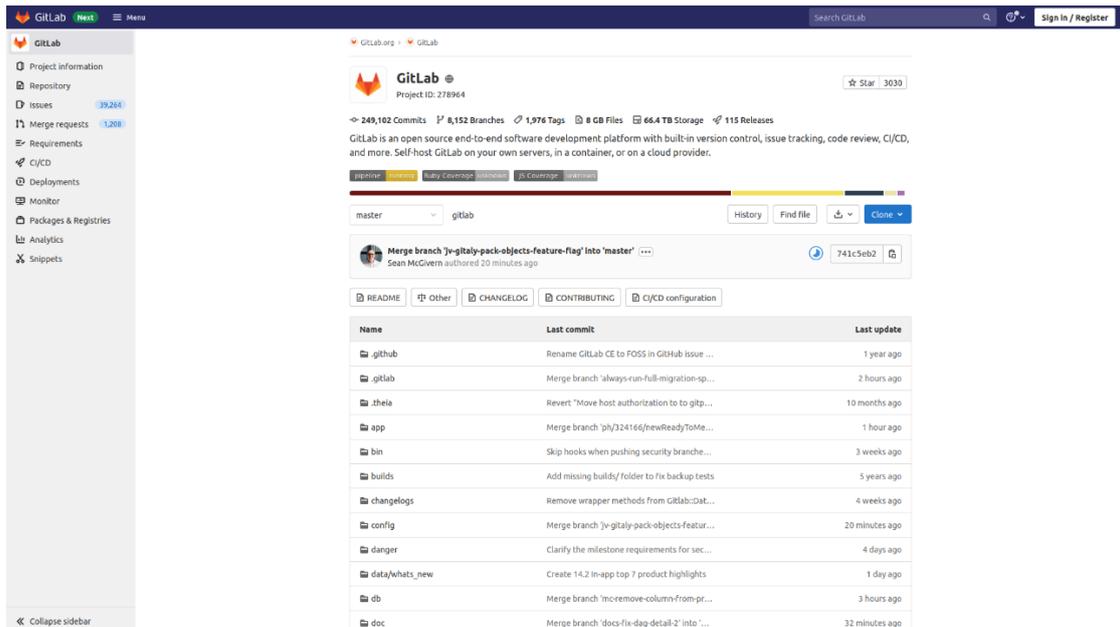


Figura 7 Pantalla principal de un proyecto en GitLab de Wikipedia [28]

3.2 ASTAH

Astah es una herramienta de modelado UML que permite realizar diagramas de requisitos, de flujo de trabajo, diagramas de actividades, de estados, de despliegue, de componentes, de casos de uso, etc. Es una de la más completa del mercado con multitud de opciones para poder modelar un proyecto. Su uso requiere de un pago por la licencia. En este caso se proporciona por la propia facultad y por tanto, se ha escogido por este motivo y por estar ya familiarizado con ella. [26]

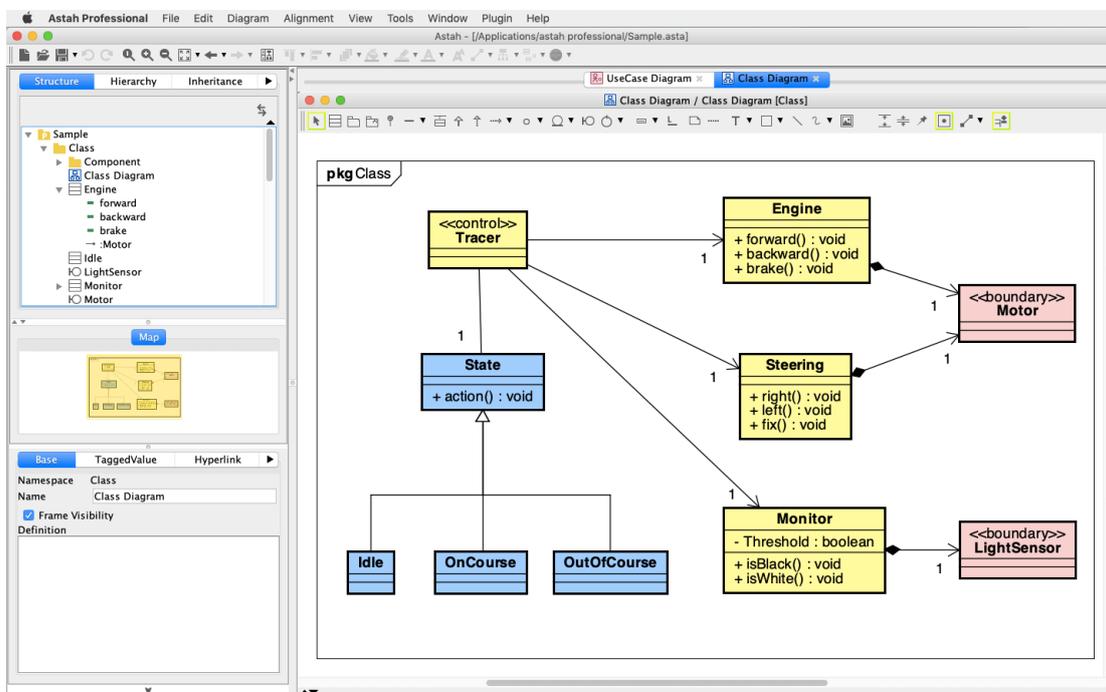


Figura 8 Ejemplo diagrama de clases en Astah de astah.net [30]

3.3 MYSQL WORKBENCH

Es una herramienta de diseño de bases de datos con interfaz visual centrada especialmente en bases de datos que están desarrolladas con el sistema MySQL, tiene muchísimas opciones y permite trabajar con la base de datos únicamente mediante clicks con la interfaz, sin tener que escribir nada de MySQL, lo que facilita su uso para incluso personas con menos formación técnica. Otra característica importante que aporta valor es que se trata de software libre. [28]

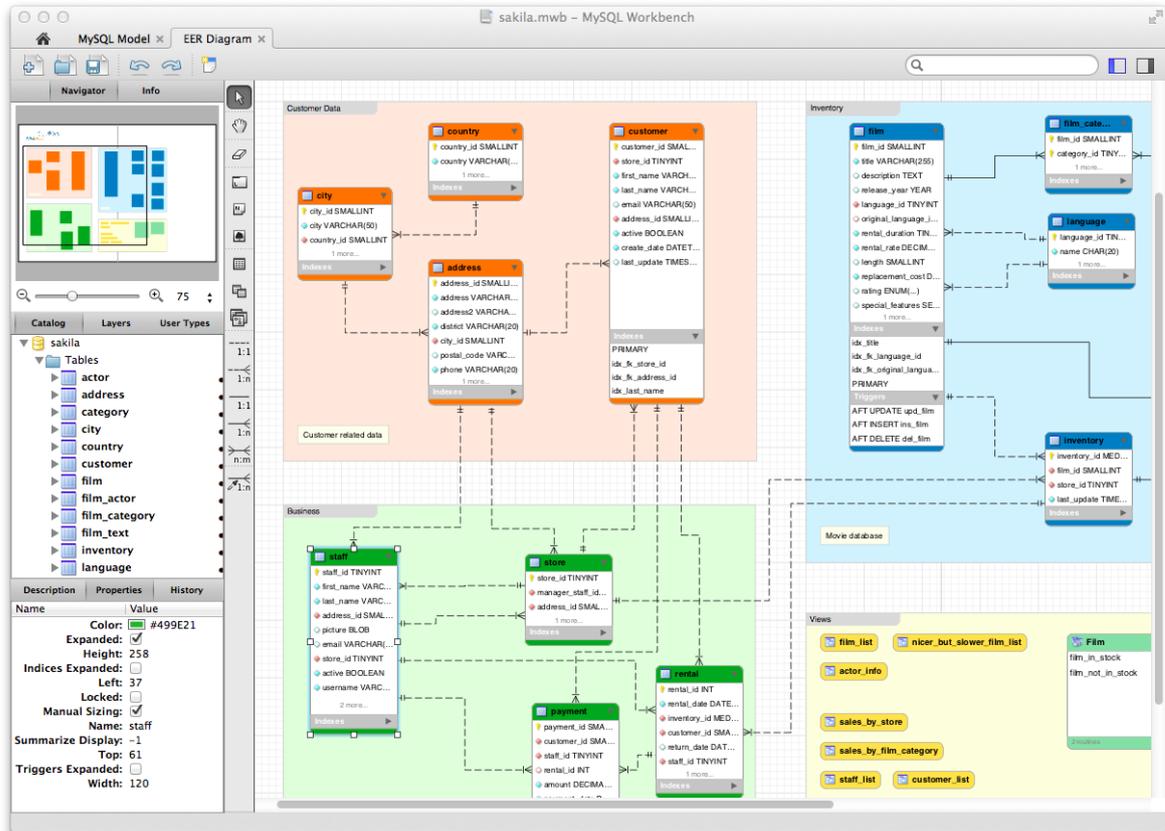


Figura 9 Ejemplo base de datos en MySQL Workbench de MySql.com [32]

3.4 VISUAL STUDIO CODE

Es uno de los editores de código fuente más populares, está desarrollado por Microsoft y es gratuito, aunque no es de código libre y tiene un carácter privativo y características que solo funcionan con cuentas de Microsoft.

Es compatible con control de versiones, resaltado de sintaxis, finalización inteligente de código, fragmentos, refactorización de código, etc.. Asimismo se le pueden añadir extensiones creadas por la comunidad, por lo que las posibilidades son casi infinitas.

En este proyecto se va a usar para la parte de Frontend, con algunas extensiones como la consola dentro del IDE y Prettier para organizar mejor el código. [33]

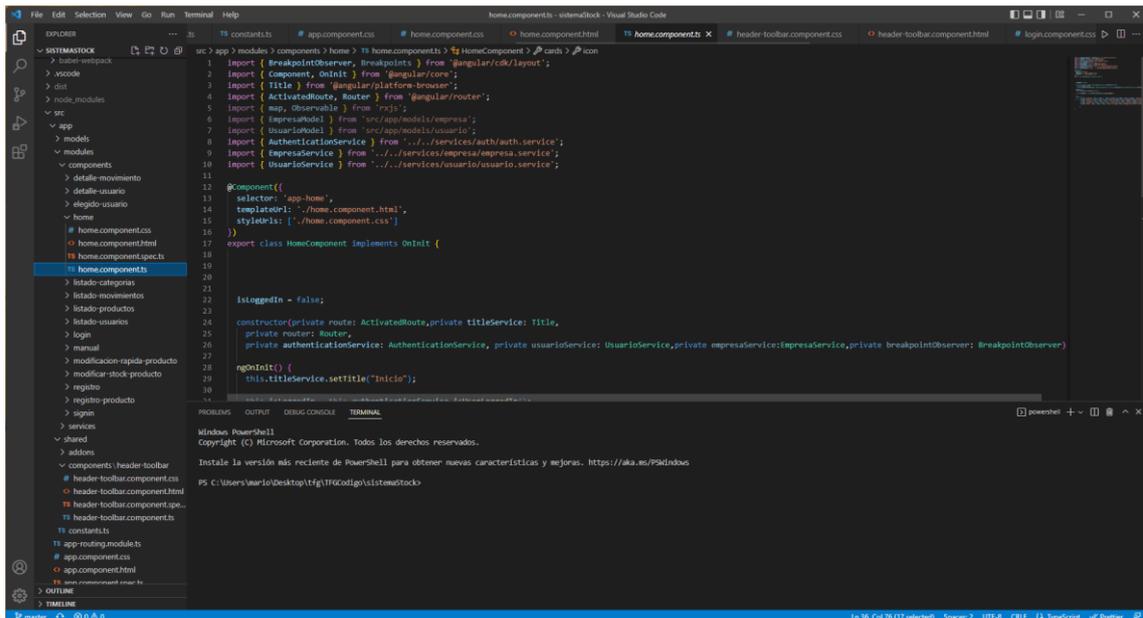


Figura 10 Interfaz Visual Studio Code

3.5 ECLIPSE

Es un entorno de desarrollo integrado sobre el que se pueden montar herramientas de desarrollo para casi cualquier lenguaje de programación. Las características que tiene son similares a Visual Studio Code, incluyendo un Marketplace o tienda de extensiones desde el que se descarga *Spring Tools for Eclipse IDE*, que permite ejecutar parte de Backend con Spring Boot para poder probar el proyecto mientras está en desarrollo.

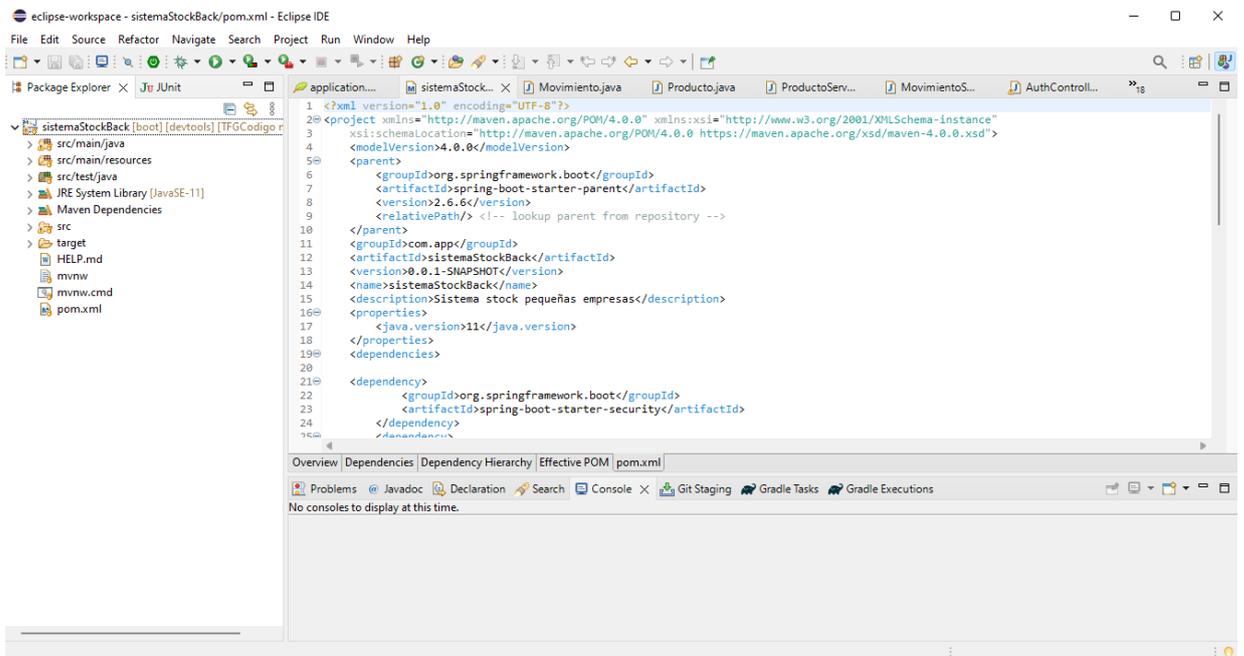


Figura 11 Interfaz Eclipse con Spring Tools instalado

3.6 OFFICE 365

Es una suite de programas de Microsoft, en su variante 365 requiere de una suscripción de pago mensual o anual. Contiene multitud de programas adicionales. Para este proyecto solo se ha usado Microsoft Word, un software de procesamiento de textos muy potente y personalizable con el que se ha redactado esta memoria.

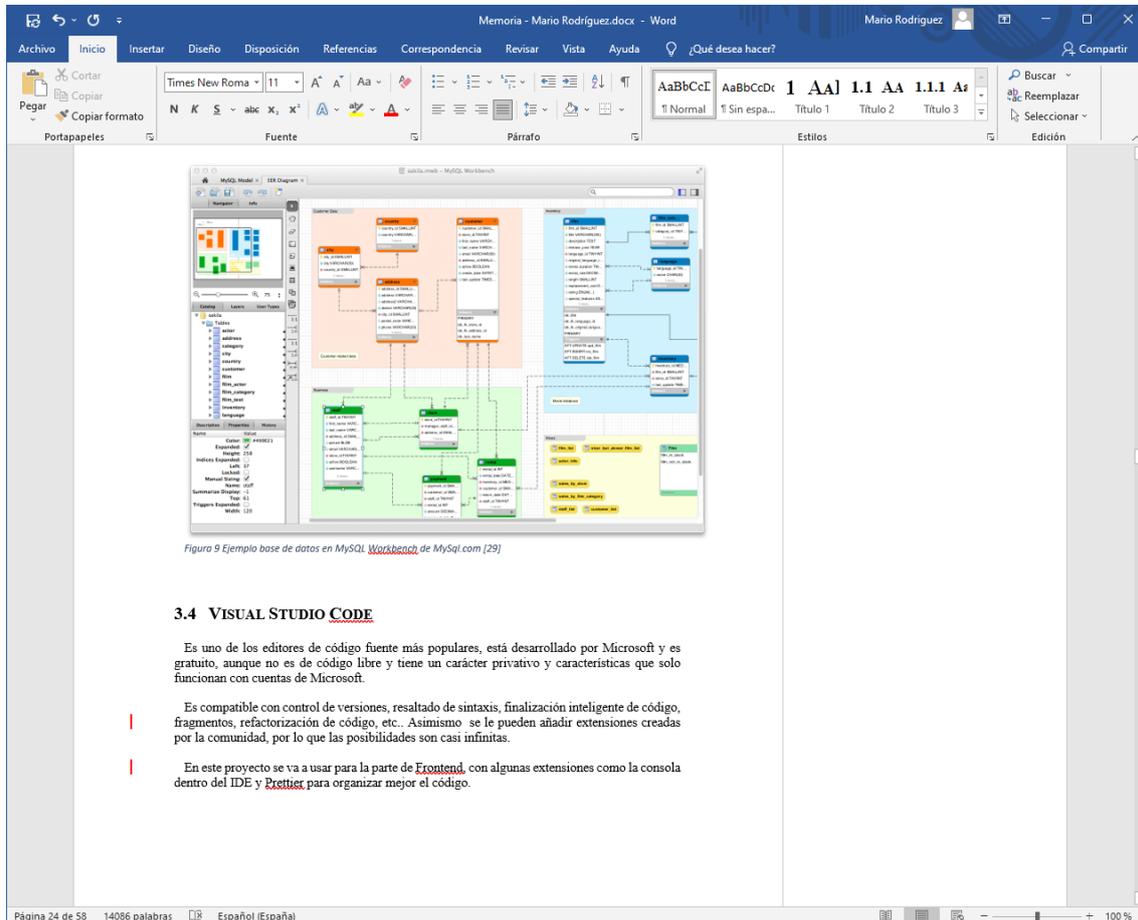


Figura 12 Interfaz Microsoft Word en Office 365

3.4 VISUAL STUDIO CODE

Es uno de los editores de código fuente más populares, está desarrollado por Microsoft y es gratuito, aunque no es de código libre y tiene un carácter privativo y características que solo funcionan con cuentas de Microsoft.

Es compatible con control de versiones, resaltado de sintaxis, finalización inteligente de código, fragmentos, refactorización de código, etc. Asimismo se le pueden añadir extensiones creadas por la comunidad, por lo que las posibilidades son casi infinitas.

En este proyecto se va a usar para la parte de **Frontend**, con algunas extensiones como la consola dentro del IDE y **Prettier** para organizar mejor el código.

3.7 POSTMAN

Postman es una aplicación que permite realizar pruebas con las API. Es un cliente HTTP que da la posibilidad de probar peticiones HTTP a través de una interfaz gráfica de usuario, por medio de la cual se obtienen diferentes tipos de respuesta que posteriormente deberán ser validados. Las posibilidades son casi ilimitadas, desde ver el código de las peticiones que se mandan hasta probar diferentes tipos de autenticación. Permite además guardar consultas e incluso dispone de un historial de peticiones realizadas en caso de necesitar recuperar alguna en el futuro. [34]

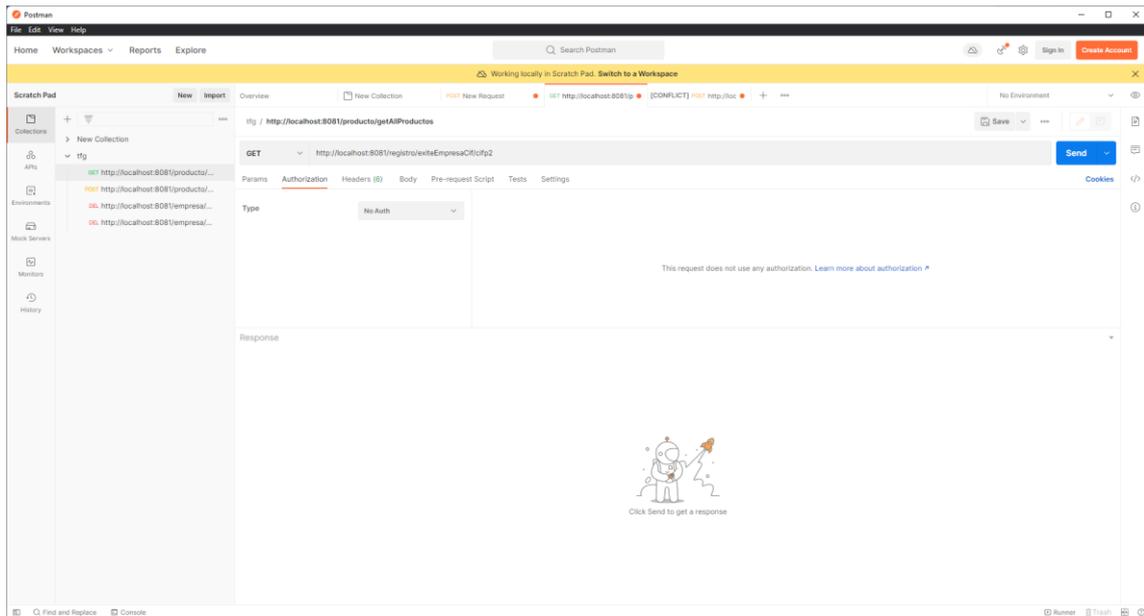


Figura 13 Interfaz Postman

3.8 AWS

Amazon Web Services, comúnmente conocido como AWS es una colección de servicios de computación en la nube de la empresa Amazon.com, es uno de los más populares en la actualidad y uno de los servicios pioneros de este tipo. [35]

En el caso de este proyecto se usará para desplegar el proyecto y que pueda ser accesible por todo el mundo, tiene servidores en todo el mundo, pero para este caso concreto se usará uno de sus servidores en Europa para tener una mejor experiencia. Se va a usar para la parte de Angular y la de Spring Boot.

4 PLAN DE DESARROLLO

4.1 METODOLOGÍA

En primer lugar, se realizará un análisis de los requisitos que debe cumplir el proyecto, las funcionalidades que debe contener el mismo y un pequeño resumen con los objetivos que debe cumplir. Una vez completado el desarrollo del project se deben revisar si realmente se cumplen. La descripción de los mismos se han descrito y tratado en el primer apartado de esta memoria.

Es necesario tener en cuenta que todos los objetivos, requisitos, etc. Se verán limitados por las tecnologías mencionadas anteriormente. Concretamente en este caso, al ser tecnologías muy actuales y versátiles, las limitaciones que pueden imponer son mínimas. Aun así, para identificar el alcance del proyecto, así como sus límites , primero se realizará un estudio de cada una de las tecnologías. Además de una pequeña prueba de diferentes ERP, para ver sus puntos fuertes, sobretodo losrelacionados con la gestión de stock para, por último, ver cómo se pueden mejorar e integrar en el proyecto.

4.2 SCRUM

Scrum [33] [34]es un marco de trabajo ágil en equipo que permite realizar poco a poco entregas parciales para ir viendo el incremento de un proyecto, estos pequeños procesos de avance de una entrega se denominan Sprints. Las tareas o historias de usuario que componen un Sprint están definidas antes del comienzo del mismo.

Al ir avanzando poco a poco en el proyecto se pueden realizar pequeños cambios del mismo sobre la marcha en caso de ser necesarios, tanto para añadir funcionalidad como para resolver posibles problemas que se puedan encontrar en el desarrollo.

En Scrum existen 4 roles principales:

- **Product Owner:** Se trata de la figura que está en contacto con los Stakeholders. Su responsabilidad es realizar la pila de producto, que se describirá más adelante en esta sección.
- **Scrum Master:** Es la figura responsable de que se aplique de forma correcta la metodología ágil . Por otro lado, es el rol “líder” encargado de eliminar posibles problemas del equipo en un Sprint.
- **Equipo de desarrollo:** Son los encargados de realizar las tareas técnicas e ir avanzando el proyecto.
- **Stakeholders:** Son el grupo de interesados en la realización del proyecto y por tanto tienen una implicación directa con el producto.

El Product Owner realiza la pila de producto, que es una recopilación de los requisitos que se van añadiendo a un proyecto. Esta pila es posteriormente revisada por todos los roles, que la dividen en pequeñas tareas y el tiempo o esfuerzo que piensan que se va a tardar cada una de ellas.

En este proyecto la figura del Scrum Master y del equipo de desarrollo es el interesado que es la persona que realiza el proyecto, y la figura del Product Owner es de la tutora, que en este caso es Irene Lavín Perrino.

Estas tareas que poco a poco se van creando luego se ponen en una pila de sprint, que luego se repartirán para cuadrar tiempos en sprints, que son periodos de tiempo en los que se realizan ciertas tareas y que cuando acaba el periodo resultan en un software funcional. Lo ideal es que cada sprint dure entre 1 y 4 semanas

Dentro de estos Sprints se tienen reuniones diarias, dailys, para ver que tal va todo respecto a lo estimado, para ver si hay problemas... Al igual que al final de cada sprint hay dos reuniones, una para ver el entregable y otra para analizar el trabajo en equipo y la metodología aplicada, esta última se denomina reunión de retrospectiva. Se puede tener un denominado Sprint 0, que será algo más corto y servirá para organizar como se va organizar todo, no aporta demasiado valor al negocio, pero se tratan temas relevantes como cuál es el objetivo del proyecto, además de crear una estructura básica en la forma de trabajo.

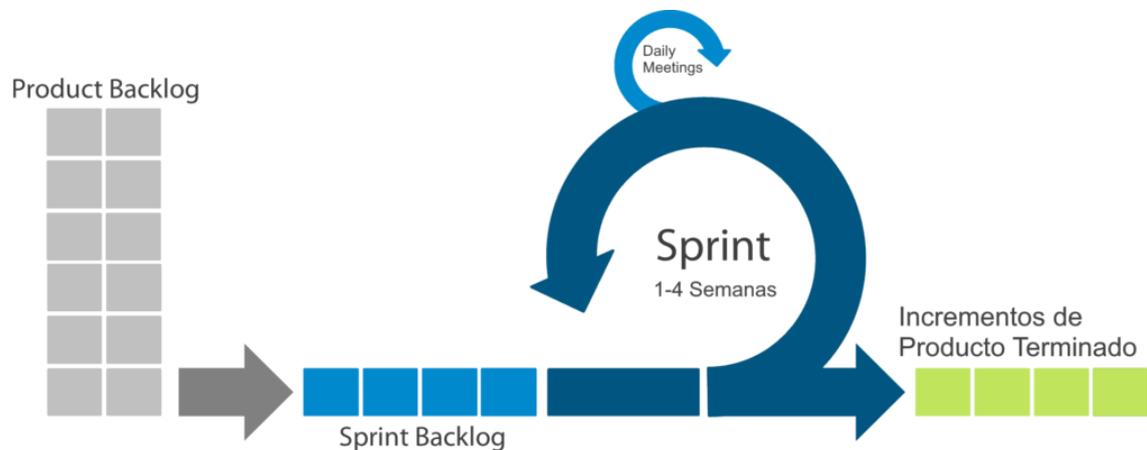


Figura 14 Esquema SCRUM Obtenido de IEBS [38]

En este caso se ha aplicado un Scrum *adaptado*, ya que el proyecto se realiza solo por una persona que tiene que realizar dos roles y el tutor tal y como se ha descrito anteriormente. Esto implica que las reuniones serán más sencillas aunque no se aplica al cien por cien el trabajo en equipo como se podría llevar a cabo en una empresa. Adaptando de este modo la metodología, se mantiene el trasfondo ágil de Scrum y la versatilidad, que son dos de sus claves.

El proyecto se dividirá en Sprints, que inicialmente tendrán entre 1-2 semanas todos ellos. Se adaptarán y dividirán entre la duración del proyecto, en casos de imprevistos o incumplimiento de los mismos se realizarán pequeñas reformulaciones en los tiempos y/o requisitos para cumplir el plazo final de entrega.

4.3 PLANIFICACIÓN

En el apartado anterior se ha descrito la metodología de trabajo que se va a seguir, . Ahora es necesario estructurar los tiempos del proyecto, para poder tener una buena organización y poder cumplir el plazo de entrega del mismo.

El proyecto se va a comenzar el día 12 de marzo de 2022 y la entrega del mismo se plantea para el 24 de junio 2022,.

En este caso para la estimación del tiempo se tiene en cuenta que solo hay un desarrollador trabajando en el proyecto, que le dedicará una media de 3h diarias de lunes a jueves, con 4h los viernes y sábados, ya que por las mañanas y parte de la tarde los días de diario (salvo el viernes) no puede dedicarse al proyecto por motivos laborales. Esto da un total de 300 horas estimadas ideales para realizar el proyecto entero, ya que la idea también es trabajar festivos o días no laborables.

A continuación, se va a dividir el proyecto en historias de usuarios (conjuntos de tareas) de carácter general y las fechas estimadas de las mismas:

	<i>Duración estimada</i>	<i>Rango fechas</i>	<i>Objetivo del Sprint</i>
<i>Sprint 0</i>	24h	12/03-20/03	Recopilar información sobre las tecnologías usadas en el proyecto y puesta a punto inicial.
<i>Sprint 1</i>	29h	21/03-30/03	Recogida requisitos del sistema, creación del modelo base de datos.
<i>Sprint 2</i>	23h	31/03-07/04	Creación base de datos y aplicar modelo. Creación inicial proyecto Angular/Spring Boot y conexión a la base de datos.
<i>Sprint 3</i>	97h	08/04-11/05	Desarrollo Backend cumpliendo los requisitos del sistema.
<i>Sprint 4</i>	91h	12/05-12/06	Desarrollo Frontend cumpliendo los requisitos del sistema.
<i>Sprint 5</i>	36h	13/06-24/06	Documentar, pruebas y despliegue del sistema y ajustes finales
<i>Total</i>	300h	12/03-24/06	

Tabla 1 Aproximación tiempo estimado en Sprints

Todo esto son estimaciones aproximadas realizadas antes de comenzar el proyecto. Los tiempos se verán afectados por posibles imprevistos a la hora de la ejecución del proyecto, y los tiempos se irán rectificando en las retrospectivas al final de cada sprint. La descripción de estos ajustes y la estimación final del proyecto se puede encontrar en la sección 4.3.1.7.

4.3.1 Calendario detallado

	<i>L</i>	<i>M</i>	<i>M</i>	<i>J</i>	<i>V</i>	<i>S</i>	<i>D</i>
<i>Mar.</i> 2022	7	8	9	10	11	12 Sprint 0	13
	14 Sprint 0	15 Sprint 0	16 Sprint 0	17 Sprint 0	18 Sprint 0	19 Sprint 0	20
	21 Sprint 1	22 Sprint 1	23 Sprint 1	24 Sprint 1	25 Sprint 1	26 Sprint 1	27
	28 Sprint 1	29 Sprint 1	30 Sprint 1	31 Sprint 2	1 Sprint 2	2 Sprint 2	3
<i>Abr.</i> 2022	4 Sprint 2	5 Sprint 2	6 Sprint 2	7 Sprint 2	8 Sprint 3	9 Sprint 3	10
	11 Sprint 3	12 Sprint 3	13 Sprint 3	14 Sprint 3	15 Sprint 3	16 Sprint 3	17
	18 Sprint 3	19 Sprint 3	20 Sprint 3	21 Sprint 3	22 Sprint 3	23 Sprint 3	24
	25 Sprint 3	26 Sprint 3	27 Sprint 3	28 Sprint 3	29 Sprint 3	30 Sprint 3	1
<i>May.</i> 2022	2 Sprint 3	3 Sprint 3	4 Sprint 3	5 Sprint 3	6 Sprint 3	7 Sprint 3	8
	9 Sprint 3	10 Sprint 3	11 Sprint 3	12 Sprint 4	13 Sprint 4	14 Sprint 4	15
	16 Sprint 4	17 Sprint 4	18 Sprint 4	19 Sprint 4	20 Sprint 4	21 Sprint 4	22
	23 Sprint 4	24 Sprint 4	25 Sprint 4	26 Sprint 4	27 Sprint 4	28 Sprint 4	29
	30 Sprint 4	31 Sprint 4	1 Sprint 4	2 Sprint 4	3 Sprint 4	4 Sprint 4	5
<i>Jun.</i> 2022	6 Sprint 4	7 Sprint 4	8 Sprint 4	9 Sprint 4	10 Sprint 4	11 Sprint 4	12
	13 Sprint 5	14 Sprint 5	15 Sprint 5	16 Sprint 5	17 Sprint 5	18 Sprint 5	19
	20 Sprint 5	21 Sprint 5	22 Sprint 5	23 Sprint 5	24 Sprint 5	25	26

Tabla 2 Calendario detallado

En este calendario se puede ver en detalle la carga de trabajo por días, como se puede ver el Sprint 3 es el más costoso y largo por diferencia, sobrepasando incluso el máximo que dicta Scrum de 6 semanas por Sprint. Una vez se tengan los requisitos bien definidos y se cree la pila de producto entonces este Sprint se podrá probablemente dividir en varios más pequeños, aunque en la memoria se trate como uno solo, dentro tendremos probablemente dos Sprint bien definidos que se describe más adelante.

Una de las mejores características a la hora de aplicar una metodología ágil [36] es que se pueden seguir sus principios modificándolos un poco para acoplarlos al proyecto, como es por ejemplo en este caso crear un *Sprint 3* más extenso de lo normal.

A continuación, se muestran los rendimientos que se han conseguidos en cada sprint, estos resultados se sacan una vez se ha acabado el mismo, se ve si se ha cumplido la estimación y posibles problemas o cambios que se deben hacer al proyecto.

4.3.1.1 Resumen Sprint 0

En este sprint se empieza realizando un estudio de que tecnologías son las ideales para realizar el proyecto y a describirlas brevemente, también se divide de forma inicial el tiempo disponible para el proyecto en Sprints y se analizan las características principales del proyecto, como costes o riesgos.

A continuación, se ve un resumen de las historias de usuario realizadas en este sprint que va del 14/03-20/03, el tiempo estimado son las horas que debería llevarnos realizar esa tarea y la desviación nos dice la diferencia entre las horas reales utilizadas y las estimadas para la realización de la misma (si por ejemplo es -3h, es que nos han sobrado 3h al realizar dicha historia de usuario).

<i>Sprint 0</i>	<i>Periodo estimado</i>	<i>Tiempo estimado</i>	<i>Desviación Tiempo</i>
<i>HU.1- Inicio proyecto</i>	12/03-14/03	7h	Ninguna
<i>HU.2- Estudio tecnologías que se van a usar</i>	15/03-16/03	6h	Ninguna
<i>HU.3- División del tiempo y métodos de desarrollo.</i>	17/03-18/03	7h	Ninguna
<i>HU.4-Gestión de riesgos y costes. Viabilidad proyecto</i>	19/03-20/03	4h	Ninguna

Tabla 3 H.U. Sprint 0

- **HU.1:** Es la historia de usuario más importante del proyecto, en ella se deja más clara la idea de lo que queremos hacer en el proyecto, las razones que nos llevan a la idea del mismo y una pequeña concreción de lo que buscamos conseguir una vez está acabado.
- **HU.2:** Con la idea de lo que se quiere en mente, se hace un pequeño estudio de las tecnologías disponibles para cumplir lo que se quiere realizar y se seleccionan las que más se adecuen al proyecto.
- **HU.3:** Se seleccionan los métodos de desarrollo que podemos adaptar al proyecto, se elige uno y se distribuye la carga de trabajo en el tiempo disponible con el método seleccionado

- **HU.4:** Se realiza un pequeño resumen de posibles riesgos e impactos que puede haber durante el desarrollo del proyecto, al igual que un pequeño apartado de costes del mismo, aunque sea solo de forma orientativa.

4.3.1.1.1 *Retrospectiva Sprint 0*

El Sprint se ha desarrollado cumpliendo los tiempos estimados, pero no los plazos, un par de días no se pudo avanzar en el proyecto por COVID y el periodo estimado en vez de acabar del día 20/03 para el sprint finalizó el día 22/03, pero se irá recuperando este tiempo que es bastante corto en los siguientes Sprint sin problemas. Además, se ha realizado toda la documentación correspondiente, que ha sido algo más extensa de lo esperado sin grandes dificultades.

4.3.1.2 *Resumen Sprint 1*

En este Sprint se reúnen los requisitos funcionales y no funcionales que debe cumplir nuestro proyecto para poder tenerlos en cuenta a la hora del desarrollo y de la creación de las historias de usuario.

Además, se realiza un modelo para la base de datos en Astah [30] que es la versión inicial sobre la que se basará el modelo final de la base de datos. A continuación, un resumen de las historias de usuario del sprint:

<i>Sprint 1</i>	<i>Periodo estimado</i>	<i>Tiempo estimado</i>	<i>Desviación Tiempo</i>
HU.5- Requisitos funcionales	21/03-23/03	9h	Ninguna
HU.6- Requisitos no funcionales	24/03-25/03	7h	Ninguna
HU.7- Creación modelo de datos	26/03-30/03	13h	-3h

Tabla 4 H.U. Sprint 1

- **HU.5:** En esta historia recopilamos todos los requisitos funcionales (y de información) que debe tener por lo menos nuestro proyecto, se pueden ver en el apartado 5.1.1.
- **HU.6:** En esta historia recopilamos todos los requisitos funcionales (y de información) que debe tener por lo menos nuestro proyecto, se pueden ver en el apartado 5.1.2.
- **HU.7:** Se crea un modelo inicial para realizar la base de datos en el que se tienen en cuenta los requisitos anteriores y la funcionalidad que queremos que tenga el sistema, este modelo cumple los campos mínimos y se podrá ampliar en el pasado sin realizar grandes cambios en el mismo.

4.3.1.2.1 *Retrospectiva Sprint 1*

El Sprint se ha desarrollado de forma satisfactoria, ha sobrado tiempo en la creación del modelo de datos de la última historia de usuario que se ha compensado con los días de retraso que teníamos en el Sprint 0 para poder tener el proyecto al día y siguiendo el calendario establecido.

4.3.1.3 *Resumen Sprint 2*

En este Sprint se aplica el modelo desarrollado en el anterior a la base de datos, creando un fichero SQL que se puede cargar en nuestra base de datos MySQL y posteriormente se crea la estructura inicial del proyecto, conectando el Backend del mismo con dicha base de datos.

A continuación, se muestra un pequeño resumen de las historias de usuario realizadas en este Sprint:

<i>Sprint 2</i>	<i>Periodo estimado</i>	<i>Tiempo estimado</i>	<i>Desviación Tiempo</i>
<i>HU.8- Aplicar modelo en base de datos</i>	31/03-01/04	7h	Ninguna
<i>HU.9- Creación inicial desarrollo del proyecto</i>	02/04-04/04	7h	Ninguna
<i>HU.10- Conexión Backend con base de datos</i>	05/04-07/04	9h	Ninguna

Tabla 5 H.U. Sprint 2

- **HU.8:** En esta historia se crea un fichero sql para meter en la base de datos siguiendo el modelo creado anteriormente.
- **HU.9:** En esta historia realizamos la configuración inicial de un proyecto angular y Spring Boot.
- **HU.10:** En esta historia se parte de la configuración inicial de Spring Boot y se realiza la conexión con nuestra base de datos, se puebla la base de datos y se comprueba que la conexión se realiza de forma correcta.

4.3.1.3.1 Retrospectiva Sprint 2

El Sprint se ha desarrollado de forma satisfactoria en cuanto a tiempo, las dos primeras H.U. del mismo han ido sin problemas y la tercera se ha complicado algo por ser la primera vez que se hacía, por suerte en la propia web de Spring hay documentación [40] para hacer la conexión paso a paso que ha facilitado mucho el trabajo y ha permitido acabar todo en el tiempo estimado.

4.3.1.4 **Resumen Sprint 3**

En este sprint se realiza el Backend del sistema que nos permitirá obtener la información de la base de datos, para ello nos basamos en el modelo inicial (6.2.1) y se ve si es necesario realizar cambios sobre el mismo para poder obtener todos los datos requeridos.

A continuación, vemos un resumen de las historias de usuario de este Sprint:

<i>Sprint 3</i>		<i>Periodo estimado</i>	<i>Tiempo estimado</i>	<i>Desviación Tiempo</i>
<i>HU.11- entities</i>	<i>Creación</i>	08/04-14/04	20h	-5h
<i>HU.12- repositories</i>	<i>Creación</i>	15/04-21/04	20h	-5h
<i>HU.13- servicios</i>	<i>Creación</i>	22/04-28/04	20h	-4h
<i>HU.14- sistema autenticación</i>	<i>Creación</i>	29/04-06/05	24h	Ninguna
<i>HU.15- Comprobación y pruebas backend</i>		07/05-11/05	13h	Ninguna

Tabla 6 H.U. Sprint 3

- ***HU.11:*** En esta historia se crean las *entities* que va a tener el sistema basándose en el modelo de tablas de la base de datos.
- ***HU.12:*** En esta historia se crean los repositorios necesarios donde se definen operaciones que necesitaremos utilizar en los servicios sobre las *entities* del paso anterior.
- ***HU.13:*** En esta historia se crean los servicios necesarios para el sistema, en este caso se crea un servicio por cada tabla del sistema.
- ***HU.14:*** En esta historia se crea un sistema de autenticación en el sistema, para que solo se tenga acceso a los datos teniendo las claves de acceso del mismo.
- ***HU.15:*** En esta historia se revisa que toda la parte del *Backend* funcione correctamente y cumpla los requisitos para poder usarse en el siguiente sprint. Se utiliza Postman (3.7) para realizar dichas comprobaciones.

4.3.1.4.1 Retrospectiva Sprint 3

El Sprint se ha desarrollado de forma óptima, incluso con historias en la que nos ha sobrado más tiempo, en total de forma conjunta han sobrado 14 horas, lo que ha hecho que el Sprint 4 se pueda comenzar el día 7 de mayo.

La parte más compleja de este sprint ha sido la relativa a la HU.14, ya que hay muchas formas de implementar la autenticación y al final se ha optado por un sistema sencillo que genera un token según los datos de inicio de sesión cada vez que se manda una petición para comprobar si se puede tener acceso a ese elemento, es la denominada coloquialmente como *Basic Auth*.

El resto de historias no han supuesto problema alguno en su realización al haber hecho proyectos similares previamente. En la HU.15 se detectó algún error de datos mal devueltos o errores del sistema por fallos de escritura que se corrigieron de forma satisfactoria.

4.3.1.5 **Resumen Sprint 4**

En este sprint se utiliza el Backend que se ha realizado en el sprint anterior y se enlaza con el Frontend, mostrando así los datos deseados del usuario. Para ello se realizan diferentes vistas y servicios que cumplan los datos requeridos.

A continuación, vemos un resumen de las historias de usuario de este Sprint:

<i>Sprint 4</i>	<i>Periodo estimado</i>	<i>Tiempo estimado</i>	<i>Desviación Tiempo</i>
HU.16- Sistema inicio sesión y perfil	12/05-16/05	14h	-3h
HU.17- Pantallas productos	17/05-23/05	20h	-3h
HU.18- Pantallas movimientos	24/05-27/05	13h	-2h
HU.19- Pantallas usuarios	28/05-04/06	20h	-4h
HU.20- Navegación y menús personales	06/06-09/06	12h	Ninguna
HU.21- Pantallas de apoyo	10/06-11/06	8h	Ninguna

Tabla 7 H.U. Sprint 4

- **HU.16:** En esta historia se crean todas las pantallas y servicios necesarios para la autenticación de un usuario en el sistema.
- **HU.17:** En esta historia se crean las pantallas y servicios relacionados con los productos, tanto para crear un producto, ver los productos del sistema... Las necesarias para cumplir los requisitos.
- **HU.18:** En esta historia se crean las pantallas y servicios relacionados con los movimientos de los productos del sistema, tantas como sean necesarias para cumplir los requisitos.
- **HU.19:** En esta historia se crean las pantallas y servicios relacionados con los usuarios del sistema, tantas como sean necesarias para cumplir los requisitos.
- **HU.20:** En esta historia se realiza un sistema de navegación y menús a través de las pantallas, para que se pueda tener toda la información siempre a mano.
- **HU.21:** En esta historia se revisan las pantallas ya creadas y se realizan pequeños cambios y otras pantallas de apoyo en caso de verlas necesarias para completar la funcionalidad del sistema.

4.3.1.5.1 Retrospectiva Sprint 4

El Sprint se ha desarrollado mejor de lo esperado, además se ha intentado ir algo más deprisa para poder acabar antes de fecha y tener más margen para revisiones.

En las pantallas de apoyo se han realizado más de las esperadas al ver que podían aportar información relevante al usuario, entre ellas un listado de categorías de producto y un pequeño manual general con las descripciones de cada pantalla.

Para tener información más relevante sobre los movimientos se han realizado cambios en el modelo de datos (6.2.2), además de añadir una descripción a las categorías de los productos.

Al final se ha acabado antes de la fecha, eso junto con el tiempo sobrante del sprint anterior hace que el 02/06 haya acabado el Sprint por completo.

4.3.1.6 **Resumen Sprint 5**

En este sprint ya tenemos el proyecto creado, falta revisarlo, hacer pruebas de que todo funciona como debe, desplegarlo a la nube y por último documentar todos los pasos en este documento.

A continuación, vemos un resumen de las historias de usuario de este Sprint:

<i>Sprint 5</i>	<i>Periodo estimado</i>	<i>Tiempo estimado</i>	<i>Desviación Tiempo</i>
HU.22- Pruebas y arreglos sistema	13/06-16/06	12h	Ninguna
HU.23- Despliegue sistema	17/06-20/06	11h	Ninguna
HU.24- Documentación	21/06-24/06	13h	Ninguna

Tabla 8 H.U. Sprint 5

- **HU.22:** En esta historia se verifica a fondo toda la funcionalidad del sistema y en caso de ver algún error pequeño se corrige para tener el sistema listo por completo.
- **HU.23:** En esta historia se despliega el proyecto en la nube para que sea accesible de forma pública. Se usa AWS como se ha mencionado anteriormente.
- **HU.24:** En esta historia se documenta todo lo restante que se ha dejado pendiente a lo largo del proyecto, ya que se tiene el sistema acabado y se sabe cómo funciona por completo.

4.3.1.6.1 Retrospectiva Sprint 5

El Sprint se ha desarrollado de forma óptima, no se han encontrado problemas durante su desarrollo y el haber empezado el 02/06 con él y no tener problemas ha resultado en que el día 15/06 estuviese todo completado y listo para su presentación al cliente final.

4.4 RIESGOS Y SU GESTIÓN

Un riesgo en el desarrollo software es un evento o condición incierta que en caso de que suceda tiene un impacto en al menos uno de los objetivos del proyecto. [41]

En este apartado vamos a tratar posibles riesgos que se pueden producir a lo largo del proyecto, se analizan al principio del mismo junto con una descripción breve, la probabilidad y el impacto que pueden suponer en el desarrollo del proyecto. Hay que tenerlos en cuenta y gestionarlos tratando de impedirlos, transferirlos o aceptar los riesgos a través de un plan de contingencia o de mitigación en la medida de lo posible para tener un buen desarrollo del proyecto, normalmente la gestión de riesgos se divide en varios pasos: planificación de riesgos, identificación de los mismos, analizar el impacto de los mismos en cuanto a calidad y cantidad en el proyecto, planificar una posible respuesta y por último monitorearlos constantemente. [38]

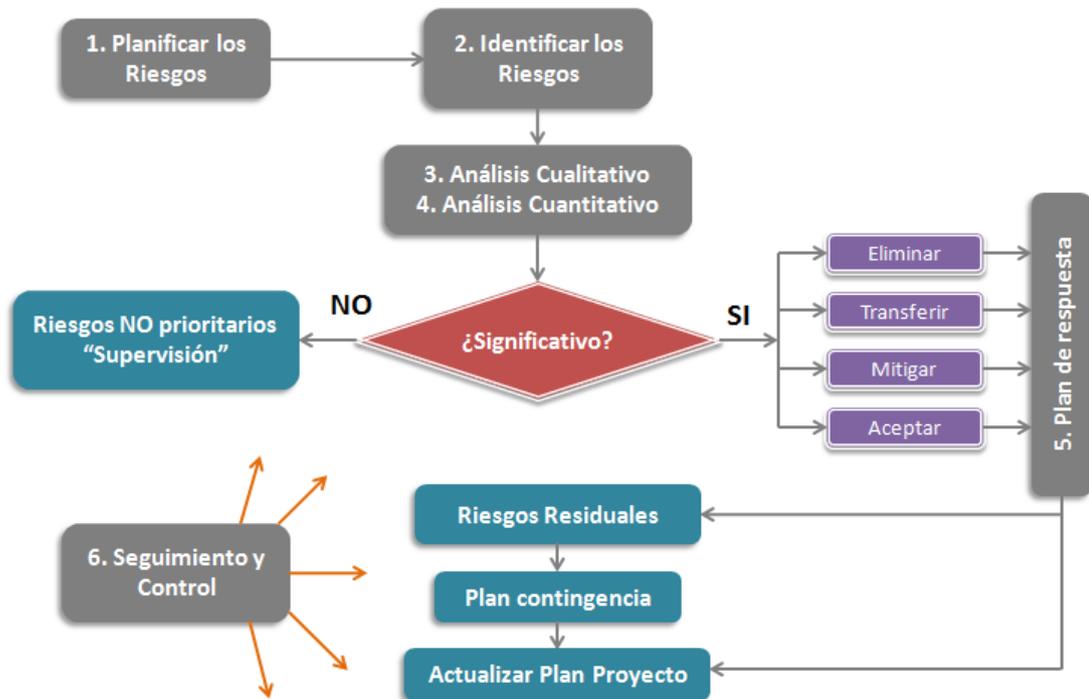


Figura 15 Esquema gestión riesgos de Gladys Gbenedji [43]

Dependiendo de lo probable que pueda suceder un riesgo y del impacto del mismo, se dividen en lo que denomina una matriz de riesgos:

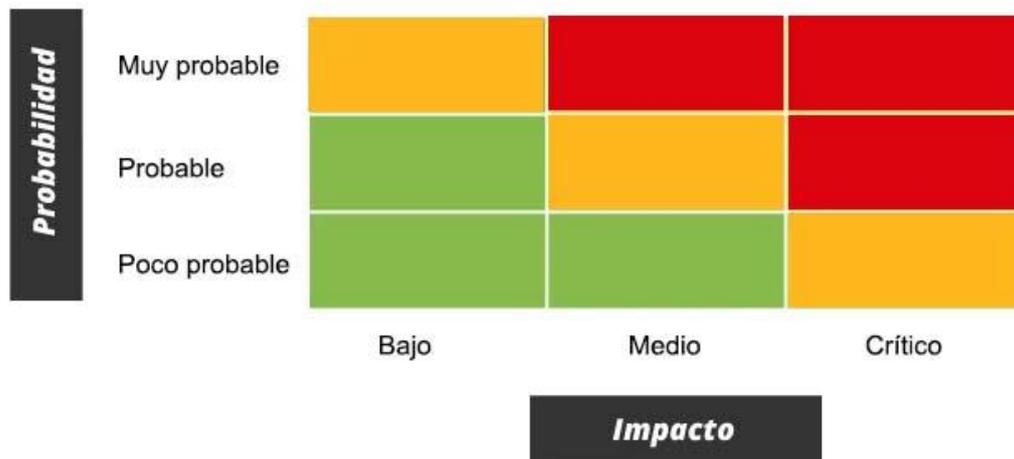


Figura 16 Matriz de riesgos general de HubSpot [44]

Esta matriz define en que riesgos es necesario centrarse más, los que siempre se tienen que tener en cuenta y monitorear para intentar que no sucedan. En este caso primero se tendrían que monitorear los rojos, luego los amarillos y por ultimo los verdes.

- **Rojo:** Riesgo muy elevado. Se deben aplicar medidas preventivas cuanto antes. Son los que pueden afectar y comprometer gravemente al proyecto, por lo que deben ser tratados antes del inicio del mismo.
- **Amarillo:** Riesgo medio. Se deben aplicar medidas preventivas. Por otro lado, es necesario una vigilancia constante durante el desarrollo del proyecto.

- **Verde:** Riesgo bajo. Se sugiere valorar y aplicar medidas preventivas para evitar que aumente el riesgo. Asimismo, se debe mantener el control sobre los mismos durante el desarrollo del proyecto.

Con cada celda representando un riesgo diferente, el plan de contingencia irá asociado dependiendo del color que es lo que da la importancia en el proyecto a los riesgos, en el punto 4.4.2 se adjunta la matriz aplicada a este proyecto junto con los planes de mitigación y contingencia.

4.4.1 Tipos de riesgos

Tenemos de forma general dos tipos de riesgos:

- **Riesgos conocidos:** Estos están identificados a continuación y se puede tener en cuenta que respuesta se va a tener en caso de que se produzcan.
- **Riesgos desconocidos específicos:** No se puede saber de antemano que riesgos son, por lo que lo ideal es tener un plan de mitigación/contingencia genérica para riesgos de este tipo.

De forma más específica los riesgos también se pueden dividir en:

- **Riesgos de proyecto:** Afectan directamente al plan del proyecto en sí.
- **Riesgos técnicos:** Los que afectan a la calidad del proyecto.
- **Riesgos de negocio:** Afectan a la viabilidad del proyecto.

4.4.2 Riesgos conocidos

<i>Identificador</i>	<i>Descripción</i>	<i>Probabilidad/ Impacto</i>	<i>Riesgo resultante</i>
<i>R01</i>	Caída servidores base de datos.	Muy baja/Alto.	Bajo
<i>R02</i>	Rotura equipos de trabajo.	Muy baja/Alto.	Bajo
<i>R03</i>	Enfermedad del desarrollador.	Medio/Muy alto.	Alto
<i>R04</i>	Perdida progreso proyecto.	Medio/Medio.	Medio
<i>R05</i>	Cambio o modificación de componentes usados en el proyecto.	Muy baja/Muy alto.	Medio
<i>R06</i>	Estimación errónea de tiempos del proyecto.	Medio/Muy alto.	Alto
<i>R07</i>	Caída servidores hosting proyecto.	Muy baja/Medio.	Bajo
<i>R08</i>	Se producen dos movimientos a la vez sobre el mismo producto.	Medio/Bajo.	Bajo

Tabla 9 Resumen riesgos

A continuación, se van a desglosar en detalle los posibles riesgos que podrían ocurrir a la hora de realizar el proyecto.

<i>Identificador</i>	<i>R01</i>
<i>Descripción breve</i>	Caída servidores base de datos.
<i>Causa</i>	Los servidores de DigitalOcean dejan de funcionar, ya sea por un problema de mi base de datos o de la compañía, perdiendo los datos de la misma.
<i>Probabilidad / Impacto</i>	Muy baja/Alto.
<i>Plan contingencia</i>	Crear una nueva base de datos en otra plataforma y poblarla de nuevo.
<i>Plan mitigación</i>	Realizar copias de datos periódicas de los datos y estructura de la misma.
<i>Tipo</i>	Riesgo técnico.

Tabla 10 Riesgo 1

<i>Identificador</i>	<i>R02</i>
<i>Descripción breve</i>	Rotura equipos de trabajo.
<i>Causa</i>	El/los ordenadores usados para desarrollar el proyecto dejan de funcionar o tienen algún problema, ya sea de hardware o software.
<i>Probabilidad / Impacto</i>	Muy baja/Alto.
<i>Plan contingencia</i>	Usar un nuevo equipo suplementario o comprar uno nuevo.
<i>Plan mitigación</i>	Intentar cuidar los equipos que se usan y no hacer nada extraño con ellos.
<i>Tipo</i>	Riesgo del proyecto.

Tabla 11 Riesgo 2

<i>Identificador</i>	<i>R03</i>
<i>Descripción breve</i>	Enfermedad del desarrollador.
<i>Causa</i>	El principal desarrollador del proyecto cae enfermo y no le es posible trabajar o lo hace de forma más lenta.
<i>Probabilidad / Impacto</i>	Medio/Muy alto.
<i>Plan contingencia</i>	Recuperar las horas perdidas posteriormente cuando esté de nuevo bien.
<i>Plan mitigación</i>	Intentar llevar un modo de vida saludable.
<i>Tipo</i>	Riesgo del proyecto.

Tabla 12 Riesgo 3

Identificador R04

<i>Descripción breve</i>	Perdida progreso proyecto.
<i>Causa</i>	Se pierde parte del progreso que ha realizado el desarrollador, ya sea por un error humano o de la maquina en la que se está trabajando.
<i>Probabilidad / Impacto</i>	Medio/Medio.
<i>Plan contingencia</i>	Recuperar todo el trabajo posible del repositorio.
<i>Plan mitigación</i>	Intentar tener el repositorio lo más actualizado posible, con commits continuados.
<i>Tipo</i>	Riesgo técnico.

Tabla 13 Riesgo 4

Identificador R05

<i>Descripción breve</i>	Cambio o modificación de componentes usados en el proyecto.
<i>Causa</i>	Alguna de las tecnologías usadas en el proyecto sufre una modificación que hace que el proyecto deje de funcionar total o parcialmente.
<i>Probabilidad / Impacto</i>	Muy baja/Muy alto.
<i>Plan contingencia</i>	Ver el cambio de la tecnología e intentar adaptar el proyecto a la misma.
<i>Plan mitigación</i>	Intentar tener las últimas versiones de todo a la hora del comienzo del proyecto.
<i>Tipo</i>	Riesgo técnico.

Tabla 14 Riesgo 5

Identificador R06

<i>Descripción breve</i>	Estimación errónea de tiempos del proyecto.
<i>Causa</i>	Las estimaciones iniciales están mal hechas y una vez se trabaja con ellas se ve que va a llevar mucho más tiempo del que se tenía previsto.
<i>Probabilidad / Impacto</i>	Medio/Muy alto.
<i>Plan contingencia</i>	Intentar dedicarle más horas al proyecto o recortar el alcance del mismo.
<i>Plan mitigación</i>	Intentar tener las últimas versiones de todo a la hora del comienzo del proyecto.
<i>Impacto</i>	Riesgo del proyecto.

Tabla 15 Riesgo 6

<i>Identificador</i>	<i>R07</i>
<i>Descripción breve</i>	Caída servidores hosting proyecto.
<i>Causa</i>	Los servidores de AWS dejan de funcionar, ya sea por un problema de mi proyecto o de la compañía, teniendo que volver a desplegar la aplicación.
<i>Probabilidad / Impacto</i>	Muy baja/Medio.
<i>Plan contingencia</i>	Desplegar de nuevo la aplicación en un nuevo contenedor de AWS
<i>Plan mitigación</i>	Realizar subidas periódicas de los cambios que se van haciendo y probarlo antes de casa subida.
<i>Tipo</i>	Riesgo técnico.

Tabla 16 Riesgo 7

<i>Identificador</i>	<i>R08</i>
<i>Descripción breve</i>	Se producen dos movimientos a la vez sobre el mismo producto.
<i>Causa</i>	Se realizan dos movimientos sobre el mismo producto al mismo tiempo.
<i>Probabilidad / Impacto</i>	Medio/Bajo.
<i>Plan contingencia</i>	Guardar el ultimo cambio realizado en el producto y registrar ambos movimientos.
<i>Plan mitigación</i>	Tener el proyecto preparado para poder guardar en la base de datos de forma concurrente.
<i>Tipo</i>	Riesgo técnico.

Tabla 17 Riesgo 8

Con estas estimaciones en cuenta la matriz de riesgos quedaría así:

Probabilidad	<i>Muy alto</i>						
	<i>Alto</i>						
	<i>Medio</i>	R08		R04	R03	R06	
	<i>Bajo</i>				R02	R01	R05
	<i>Muy bajo</i>	R07					
	<i>Muy bajo</i>	<i>Bajo</i>	<i>Medio</i>	<i>Alto</i>	<i>Muy alto</i>		
	Impacto						

Tabla 18 Matriz de riesgos del proyecto

De esta matriz se concluye que hay que centrarse principalmente en los riesgos R06-R03-R04-R05 de proyecto, sobre todo cuando se trata durante el desarrollo cosas relacionadas con la temática de estos riesgos, aunque con especial atención a los riesgos R03-R06 ya que son los únicos que se encuentran en una zona roja de la matriz.

4.5 IMPREVISTOS EN PLANIFICACIÓN

Los imprevistos que han surgido durante el desarrollo de las historias de usuario han sido mínimos, no han supuesto ningún retraso en la finalización del proyecto ni un aumento de costes del mismo.

Se puede ver en el apartado 4.3.1 los tiempos de desarrollo estimados y finales de cada historia de usuario, solo hubo un problema en el Sprint 0 al contraer COVID, era un riesgo ya contemplado en el apartado 4.4.2 como el riesgo R03 (*Enfermedad del desarrollador*), por suerte solo fue un par de días y se pudo recuperar el avance sin dificultades en el siguiente Sprint para estar al día de nuevo.

4.6 ANÁLISIS DE COSTES

Para todo proyecto es importante describir este apartado. En él se desglosan los costes que se estiman va a tener el proyecto y si se desarrolla todo como está previsto. Esto permite poder calcular la rentabilidad del mismo y el margen de maniobra disponible.

Se va a dividir este apartado en 3 tipos de recursos diferentes: Hardware utilizado, Software y Personal.

4.6.1 Hardware

En este apartado se desglosan los equipos utilizados para trabajar en el proyecto. En este caso, se trata de un equipo con un i5 12600k, 48gb de ram y una rx 6600 xt junto con teclado, ratón y un monitor ultrawide. También es necesaria la compra de un escáner inalámbrico para poder probar el funcionamiento del proyecto. El coste aproximado de todo el conjunto es alrededor de los 1520 euros en este apartado . No se requiere de más hardware.

4.6.2 Software

Se va a usar licencia profesional de Astah, una licencia educativa de Windows, una licencia de Office 365 para redactar este documento y varios IDE como Eclipse o Visual Studio Code para realizar el proyecto.

La licencia de Astah la proporciona la facultad, pero en caso de no hacerlo tendría un coste de 119 dólares anuales, y en el caso de Office 365 que se cuenta con una cuenta personal que tiene un coste de 69 euros anuales.

Para el hosting se va a usar Digital Ocean para la base de datos MySQL por tener licencia educativa, en caso de no tenerla el coste serían de 15 dólares mensuales para 1GB de RAM, con 10GB de memoria y una CPU.

Para el hosting de la aplicación tanto Angular como Spring Boot el precio de AWS es de 0,0245 dólares por cada GB de uso, además de 0,0054 dólares por cada 1000 peticiones PUT, COPY, POST LIST y 0,00043 dólares por cada 100 peticiones de tipo GET, SELECT. El inicio del servicio es gratuito, solo se paga por uso de forma mensual.

4.6.3 Personal

En este caso es un proyecto que se lleva a cabo por una sola persona, por lo que se va a extrapolar lo que esta persona cobra de media en su puesto laboral y ajustar a las horas estimadas del proyecto, para tener una idea del coste medio que podría suponer en este apartado.

El coste medio que sale es de 12 euros la hora, que es lo que cobra dicho empleado como salario en su empresa en la actualidad. En este caso y tal y como se describió en el apartado 3.3 se estiman unas 300 horas de trabajo, lo que nos lleva a un coste de $12 \cdot 267 = 3204\text{€}$ en este apartado.

4.6.4 Coste inicial proyecto

Los costes iniciales de un proyecto son los que se producen durante el proceso de desarrollo, en el caso de este proyecto son la suma de los apartados anteriores.

Elemento	Coste
Software	69 euros
Hardware	1520 euros
Personal	3204 euros
Total	4793 euros

Tabla 19 Coste inicial proyecto

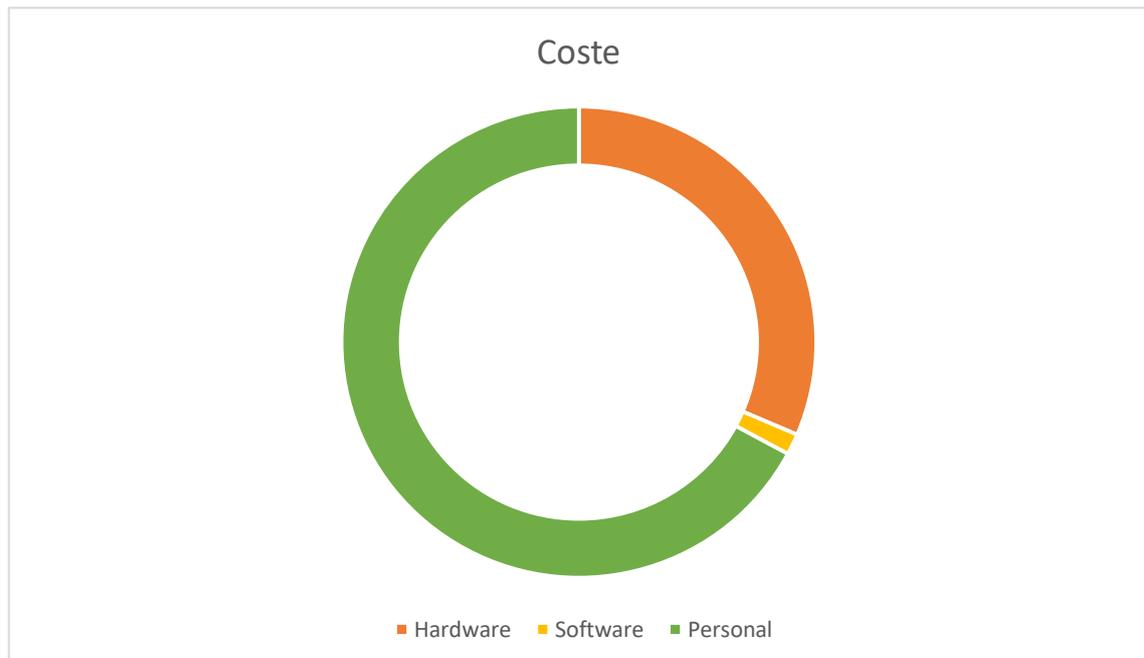


Figura 17 Resumen costes proyecto

5 ANÁLISIS DEL PROYECTO

En este capítulo, con el conocimiento de las tecnologías que se van a utilizar en el proyecto, metodología y planificación... se puede empezar a concretar la división interna de estos Sprints vistos en el apartado 4.3. Para ello primero se va a describir los requisitos que tiene que cumplir el proyecto para posteriormente hacer una división en historias de usuario.

5.1 REQUISITOS

Los requisitos son condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación [45], los podemos clasificar en dos tipos, funcionales y no funcionales.

5.1.1 Requisitos funcionales

Definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular cómo debe comportarse ante situaciones particulares. Dentro de este tipo de requisitos se pueden clasificar como “requisitos de información” si indican el tipo de información que guarda el sistema.

En este proyecto los requisitos funcionales son los siguientes:

- **RF.01:** El usuario podrá iniciar sesión en su cuenta con su DNI y contraseña.
- **RF.02:** Hay dos tipos de cuentas, autorizado y no autorizado, el autorizado puede realizar más acciones en el sistema, a la hora del registro se elige el tipo de usuario.
- **RF.03:** Todo usuario pertenece a una empresa y una cuenta autorizada de la misma podrá registrar nuevos usuarios.
- **RF.04:** El sistema permitirá a los usuarios registrar nuevos productos en el mismo.
- **RF.05:** El sistema permitirá cambiar las unidades de stock de un producto.
- **RF.06:** El sistema dejará ver al usuario autorizado todos los movimientos y cambios de stock del sistema y filtrar por parámetros.
- **RF.07:** El sistema permitirá a los usuarios editar las características de un producto ya existente.
- **RF.08:** El sistema permitirá ver los usuarios de una empresa a las cuentas autorizadas y borrar o modificar los mismos.
- **RF.09:** El sistema tendrá una opción para poder aumentar el stock de productos que ya se encuentran en el sistema.
- **RF.10:** El sistema dará la posibilidad de registrar nuevos usuarios a las cuentas autorizadas.
- **RF.11:** El sistema dejará descargar a todas las cuentas un documento en PDF que tenga los movimientos del sistema con la fecha y usuario que lo realizó.
- **RF.12:** El sistema dará la posibilidad de borrar usuarios registrados y movimientos asociados a las cuentas autorizadas.
- **RF.13:** El sistema dará la posibilidad de borrar productos y movimientos asociados a las cuentas autorizadas.
- **RF.14:** El sistema dejará descargar a todas las cuentas un documento en PDF que tenga los usuarios y los datos de los mismos.
- **RF.15:** El sistema dejará descargar a todas las cuentas un documento en PDF que tenga los productos del sistema con las características básicas de los mismos.
- **RF.16:** El sistema dejará registrar una nueva empresa en el sistema y registrar una cuenta autorizada asignada a la misma empresa.

Y los requisitos de información:

- **RFI.01:** El sistema deberá almacenar el nombre, DNI, fecha de incorporación, si tiene autorización para crear nuevos usuarios, si tiene autorización para ver todos los movimientos, la empresa a la que pertenece y puesto de cada empleado.
- **RFI.02:** El sistema deberá almacenar todos los movimientos que haya en el mismo que modifiquen la base de datos, con la fecha, el movimiento y el usuario que lo ha realizado.
- **RFI.03:** El sistema deberá almacenar de un producto su EAN, categoría, unidades, título y descripción, y como opcionales se podrán almacenar el precio y/o la talla.
- **RFI.04:** El sistema deberá almacenar de una empresa su CIF, nombre y fecha de incorporación al sistema.
- **RFI.05:** El sistema deberá almacenar de una categoría el porcentaje de IVA asociado, el nombre y la descripción de la misma.

5.1.2 Requisitos no funcionales.

Los requisitos no funcionales son restricciones que afectan a los servicios o funciones del sistema, tales como restricción de tiempo, sobre el proceso de desarrollo, estándares, etc.

En el caso de este proyecto los requisitos no funcionales son:

- **RNF.01:** La base de datos debe usar MySQL
- **RNF.02:** El sistema se implementará con un combo de Angular y Spring Boot conectados a través de una API REST .
- **RNF.03:** El sistema deberá ser accesible de forma sencilla desde cualquier navegador actual.
- **RNF.04:** El sistema estará por completo en español.
- **RNF.05:** El sistema será escalable, con la posibilidad de añadir nuevas empresas al mismo con nuevos usuarios.
- **RNF.05:** El sistema será escalable, con la posibilidad de añadir nuevas empresas al mismo con nuevos usuarios.
- **RNF.06:** El sistema estará desplegado en AWS la parte de Spring Bott y Angular y en Digital Ocean la base de datos.
- **RNF.07:** El sistema solo dejará hacer cambios o consultas a usuarios registrados en el sistema previamente.

5.2 HISTORIAS DE USUARIO

Una historia de usuario, comúnmente conocidas como HU, es la unidad de trabajo más pequeña en un proyecto Scrum, es la explicación general e informal de una función software desarrollada desde la perspectiva del usuario final o cliente. [46]

A continuación, están las historias de usuario del proyecto:

HU.1	Inicio proyecto.
Descripción	<i>Es la historia de usuario más importante del proyecto, en ella se deja más clara la idea de lo que se quiere hacer en el proyecto, las razones que llevan a la idea del mismo y una pequeña concreción de lo que se busca conseguir con el mismo.</i>
Tiempo estimado	<i>3 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero una idea clara de que objetivos tiene que cumplir el proyecto para poder empezar el desarrollo.</i>
Prioridad	<i>Alta</i>

Tabla 20 Historia de usuario 1- Inicio proyecto.

HU.2	Estudio tecnologías usadas.
Descripción	<i>Con la idea de lo que se quiere en mente, se hace un pequeño estudio de las tecnologías disponibles para cumplir lo que se quiere realizar y se seleccionan las que más se adecuen al proyecto.</i>
Tiempo estimado	<i>6 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener claras las tecnologías que se van a aplicar en el desarrollo del proyecto para poder saber el funcionamiento de cada una de ellas.</i>
Prioridad	<i>Media</i>

Tabla 21 Historia de usuario 2- Estudio tecnologías usadas.

HU.3	División del tiempo y métodos de desarrollo.
Descripción	<i>Se seleccionan los métodos de desarrollo que podemos adaptar al proyecto, se elige uno y se distribuye la carga de trabajo en el tiempo disponible con el método seleccionado.</i>
Tiempo estimado	<i>6 horas</i>
Requisitos validación	<i>Como Scrum Master quiero tener el tiempo del proyecto dividido según la carga de trabajo para poder comenzar el reparto de historias de usuario de cada Sprint.</i>
Prioridad	<i>Alta</i>

Tabla 22 Historia de usuario 3- División del tiempo y métodos de desarrollo.

HU.4	Viabilidad del proyecto.
Descripción	<i>Se realiza un pequeño resumen de posibles riesgos e impactos que puede haber durante el desarrollo del proyecto, al igual que un pequeño apartado de costes del mismo, aunque sea solo de forma orientativa.</i>
Tiempo estimado	<i>3 horas</i>
Requisitos validación	<i>Como Product Owner quiero tener un estimado de costes del proyecto y riesgos que se pueden encontrar durante el desarrollo para ver la viabilidad del mismo.</i>
Prioridad	<i>Alto</i>

Tabla 23 Historia de usuario 4- Viabilidad del proyecto.

HU.5	Requisitos funcionales.
Descripción	<i>Se recopilan todos los requisitos funcionales y de información del sistema.</i>
Tiempo estimado	<i>9 horas</i>
Requisitos validación	<i>Como Scrum Master quiero tener un listado de requisitos funcionales y de información que permitan cumplir toda la funcionalidad que queremos que tenga el proyecto para poder empezar el desarrollo de mismo.</i>
Prioridad	<i>Alta</i>

Tabla 24 Historia de usuario 5- Requisitos funcionales.

HU.6	Requisitos no funcionales.
Descripción	<i>Se recopilan todos los requisitos no funcionales del sistema.</i>
Tiempo estimado	<i>7 horas</i>
Requisitos validación	<i>Como Scrum Master quiero tener un listado de requisitos no funcionales y de información que permitan cumplir toda la funcionalidad que queremos que tenga el proyecto para poder empezar el desarrollo de mismo.</i>
Prioridad	<i>Alta</i>

Tabla 25 Historia de usuario 6- Requisitos no funcionales.

HU.7	Creación modelo de datos.
Descripción	<i>Se realiza una versión inicial del modelo que se va a aplicar en la base de datos, dicho modelo tiene que estar pensado para sufrir pocas variaciones a lo largo del desarrollo de todo el proyecto.</i>
Tiempo estimado	<i>13 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener un modelo inicial para la base de datos que cumpla todos los requisitos definidos para el proyecto para poder empezar a trabajar sobre la misma.</i>
Prioridad	<i>Medio</i>

Tabla 26 Historia de usuario 7- Creación modelo de datos.

HU.8	Aplicar modelo en base de datos.
Descripción	<i>Se toma como referencia el modelo desarrollado en la historia de usuario anterior y se crea un fichero SQL con él, posteriormente se carga dicho fichero en nuestra base de datos.</i>
Tiempo estimado	<i>7 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener la base de datos ya funcionando con las tablas definidas en el modelo para poder conectar la misma con nuestro proyecto.</i>
Prioridad	<i>Medio</i>

Tabla 27 Historia de usuario 8- Aplicar modelo en base de datos.

HU.9 Creación inicial desarrollo del proyecto.

Descripción	<i>Se crea un nuevo proyecto en Angular y Spring Boot desde cero, se crea también un repositorio en GitLab y se sube el proyecto a dicho repositorio.</i>
Tiempo estimado	<i>7 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener el código básico inicial del proyecto en GitLab, tanto frontend como backend para poder ir viendo los cambios del proyecto.</i>
Prioridad	<i>Medio</i>

Tabla 28 Historia de usuario 9- Creación inicial desarrollo del proyecto.

HU.10 Conexión backend con base de datos.

Descripción	<i>Se conecta el backend del proyecto a nuestra base de datos y se comprueba que la conexión es correcta con pequeños test para lo que se necesita poblar parcialmente la base de datos.</i>
Tiempo estimado	<i>9 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener el backend de nuestro proyecto conectado con la base de datos del mismo para comprobar que se pueden hacer operaciones en la base de datos desde el backend mediante un API REST.</i>
Prioridad	<i>Alto</i>

Tabla 29 Historia de usuario 10- Conexión backend con base de datos.

HU.11 Creación entities.

Descripción	<i>Se crea una entity por cada tabla que tengamos en la base de datos, con los mismos campos que tengan estas y conectadas de una forma óptima para poder usarlas en el futuro.</i>
Tiempo estimado	<i>20 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener una entity por cada tabla y que cada una de ellas tenga todos los campos de dicha tabla para poder realizar los repositorios de las mismas.</i>
Prioridad	<i>Medio</i>

Tabla 30 Historia de usuario 11- Creación entities.

HU.12 Creación repositories.

Descripción	<i>Se crea un repositorio para cada entity y se añaden las operaciones que nos harán falta usar sobre dichas entidades para cumplir los requisitos del sistema.</i>
Tiempo estimado	<i>20 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener un repositorio para cada entity y que cada uno de ellos tenga las operaciones necesarias para poder realizar los servicios del proyecto</i>
Prioridad	<i>Medio</i>

Tabla 31 Historia de usuario 12- Creación repositories.

HU.13 Creación servicios.

Descripción	<i>Se crean los servicios que consumen los métodos de los repositorios y que nos sirven como puerta de acceso a los datos de la base de datos. Se crea un servicio por cada tabla.</i>
Tiempo estimado	<i>20 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener servicios que cubran todas las operaciones que necesitaremos para obtener, modificar y eliminar los datos desde el frontend.</i>
Prioridad	<i>Medio</i>

Tabla 32 Historia de usuario 13- Creación servicios.

HU.14 Creación sistema autenticación.

Descripción	<i>Se crea un sistema de autenticación en el que solo nos deja acceder a ciertos elementos del backend si tenemos las credenciales de usuario almacenadas en una de las tablas.</i>
Tiempo estimado	<i>24 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener acceso a ciertos elementos si tenemos las claves para en caso de no tenerlas o ser incorrectas que nos devuelva un error 401 de usuario no autorizado.</i>
Prioridad	<i>Medio</i>

Tabla 33 Historia de usuario 14- Creación sistema autenticación.

HU.15 Comprobación y pruebas backend.

Descripción	<i>Se realizan las pruebas necesarias sobre el backend, realizando pequeños cambios en caso de ver errores o datos mal devueltos.</i>
Tiempo estimado	<i>13 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero comprobar que todos los servicios funcionan como deben y que solo se puede acceder a ciertos servicios si tenemos los datos de usuario que correspondan con los datos de la tabla para poder confirmar el correcto funcionamiento del backend</i>
Prioridad	<i>Baja</i>

Tabla 34 Historia de usuario 15-Comprobación y pruebas backend.

HU.16	Sistema inicio sesión y perfil.
Descripción	<i>Se realiza el frontend relacionado a la autenticación del usuario, de tal forma que solo pueda mandar peticiones al back si está autenticado. Además, se muestran los datos del usuario que ha iniciado sesión en otra pantalla.</i>
Tiempo estimado	<i>14 horas</i>
Requisitos validación	<i>Como equipo de desarrollo quiero tener un inicio de sesión en el sistema, de tal forma que solo se pueda acceder a ciertos recursos predefinidos si ha introducido las claves previamente para poder ver qué tipo de usuario son y poder mostrar la información relacionada.</i> <i>Como usuario quiero poder iniciar sesión en el sistema para poder tener acceso a la información del mismo.</i>
Prioridad	<i>Medio</i>

Tabla 35 Historia de usuario 16- Sistema inicio sesión y perfil.

HU.17	Pantallas productos.
Descripción	<i>Se realizan todas las pantallas que requieran tratamiento de información de los productos, tanto listados, modificación de productos, borrado, registros de nuevos productos y descarga de datos. Tiene que permitirse cargar un producto mediante un escáner.</i>
Tiempo estimado	<i>20 horas</i>
Requisitos validación	<i>Como cuenta del sistema quiero tener un listado completo de productos que deje filtrar, borrar, ordenar y descargar los datos, creación y edición de productos. Como usuario final quiero la posibilidad de ver y editar las unidades de un producto con un escáner de códigos</i> <i>En ambos casos lo quiero para poder realizar cambios y consultar los productos de la empresa.</i>
Prioridad	<i>Medio</i>

Tabla 36 Historia de usuario 17- Pantallas productos.

HU.18	Pantallas movimientos.
Descripción	<i>Se realizan todas las pantallas que requieran tratamiento de información de los movimientos de los productos, que nos permitan encontrar movimientos de forma sencilla y ver el cambio que supuso dicho movimiento en el producto.</i>
Tiempo estimado	<i>13 horas</i>
Requisitos validación	<i>Como cuenta autorizada quiero tener un listado completo de movimientos que deje filtrar, ordenar y descargar los datos, además de poder ver el detalle de cada movimiento para poder ver todos los cambios del sistema.</i>
Prioridad	<i>Medio</i>

Tabla 37 Historia de usuario 18- Pantallas movimientos.

HU.19

Pantallas de usuarios.

Descripción *Se realizan todas las pantallas que requieran tratamiento de información de usuarios del sistema, que nos permitan ver los detalles de los mismos, modificar su contraseña y borrarlos.*

Tiempo estimado *20 horas*

Requisitos validación *Como cuenta del sistema quiero tener un listado completo de usuarios que deje filtrar, ordenar y descargar los datos, además de poder ver el detalle de cada usuario y borrar el usuario y todos sus movimientos asociados. Se tendrá en cuenta si un usuario es autorizado para dar la opción del borrado de un usuario existente, la edición de contraseña de otro usuario, la creación de un usuario nuevo y ver los DNI de todos los usuarios del sistema.*

Prioridad *Medio*

Tabla 38 Historia de usuario 19- Pantallas de usuarios.

HU.20

Navegación y menús personales.

Descripción *Se realizan menús, pantallas y botones que nos permitan interconectar todas las pantallas creadas previamente.*

Tiempo estimado *12 horas*

Requisitos validación *Como equipo de desarrollo quiero tener todas las pantallas del sistema conectadas y accesibles en cualquier momento, con los enlaces de mayor uso siempre en puntos visibles de la interfaz para que las cuentas del sistema puedan navegar por todas las pantallas.*

Prioridad *Medio*

Tabla 39 Historia de usuario 20- Navegación y menús personales.

HU.21

Pantallas de apoyo.

Descripción *Se realizan pantallas para mostrar la información restante del sistema que no se ha realizado en el resto de HU del sprint.*

Tiempo estimado *8 horas*

Requisitos validación *Como cuenta autorizada quiero tener acceso a todos los datos almacenados relacionados con la empresa a la que pertenece mediante diferentes pantallas para poder consultar los datos cuando se requiera.*

Prioridad *Bajo*

Tabla 40 Historia de usuario 21- Pantallas de apoyo.

HU.22

Pruebas y arreglos sistema

Descripción *En esta historia se verifica a fondo toda la funcionalidad del sistema y en caso de ver algún error pequeño se corrige para tener el sistema listo por completo.*

Tiempo estimado *12 horas*

Requisitos validación *Como cuenta del sistema quiero tener una experiencia sin fallos ni bugs para poder utilizar el sistema de forma óptima.*

Prioridad *Bajo*

Tabla 41 Historia de usuario 22 - Pruebas y arreglos sistema

HU.23

Despliegue sistema

Descripción	<i>En esta historia se despliega el proyecto en la nube para que sea accesible de forma pública. Se usa AWS como se ha mencionado anteriormente.</i>
Tiempo estimado	<i>11 horas</i>
Requisitos validación	<i>Como cuenta del sistema quiero tener acceso a la aplicación de forma pública en la nube poder entrar en la misma cuando se quiera.</i>
Prioridad	<i>Medio</i>

Tabla 42 Historia de usuario 23 -Despliegue sistema

HU.24

Documentación

Descripción	<i>En esta historia se documenta todo lo restante que se ha dejado pendiente a lo largo del proyecto, ya que se tiene el sistema acabado y se sabe cómo funciona por completo.</i>
Tiempo estimado	<i>13 horas</i>
Requisitos validación	<i>Como cliente quiero tener acceso a todos los procesos que se han realizado en el desarrollo del sistema para poder consultar los costes y rendimiento del dinero proporcionado en el desarrollo.</i>
Prioridad	<i>Medio</i>

Tabla 43 Historia de usuario 24 - Documentación

6 DISEÑO

6.1 CASOS DE USO

6.1.1 Diagrama casos de uso

Primero se muestra un diagrama de casos de uso del sistema. Este diagrama representa gráficamente las interacciones que tiene el usuario con el sistema, con las relaciones entre los diferentes tipos de usuario y las acciones que puede realizar cada uno de ellos. Dichas acciones están representadas por círculos y los actores son pequeñas figuras de palo [47].

Cada caso de uso es una secuencia de acciones que da lugar a un resultado de valor observable, además cada caso nos permite comprobar y recopilar los requisitos funcionales del sistema que se tienen que cumplir en el proyecto.

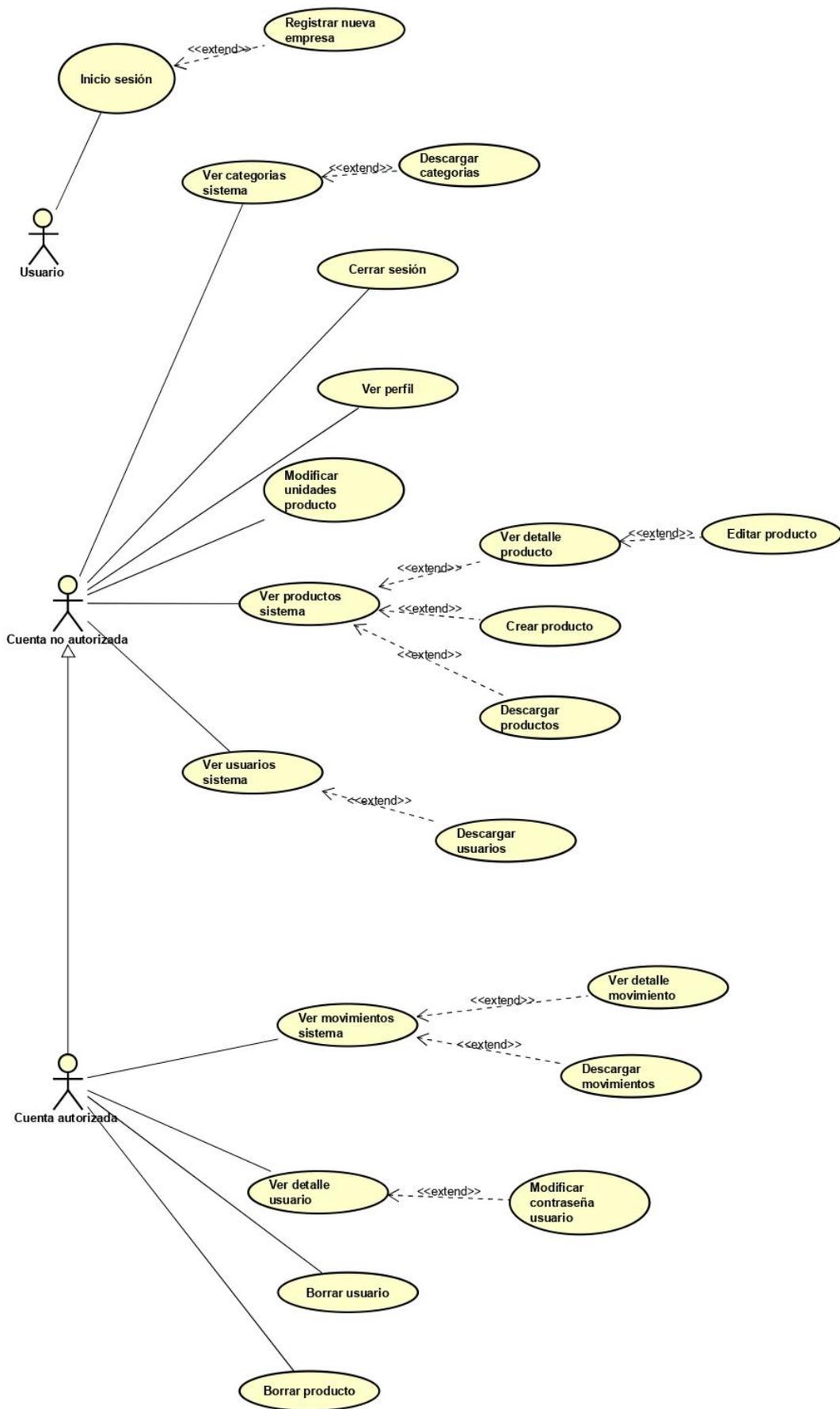


Figura 18 Diagrama de casos de uso del sistema.

6.1.2 Detalle casos de uso

En esta sección se muestran los detalles de cada uno de los casos de uso del diagrama de forma resumida con las acciones principales.

CU.01 Inicio sesión

Descripción	Un usuario inicia sesión en el sistema a través de la aplicación web.
Actor	Usuario
Precondiciones	Se requiere conexión a internet
Postcondiciones	El usuario está identificado en el sistema y accede con una cuenta al mismo.
Flujo normal	<ol style="list-style-type: none">1. El usuario introduce su DNI y su contraseña en el sistema.2. El sistema comprueba los datos y redirecciona a la página principal del sistema, siendo ahora el usuario una cuenta autorizada/no autorizada dependiendo de los datos introducidos.
Flujo alternativo	2.A El usuario quiere registrar una nueva empresa y un nuevo usuario introduciendo un CIF, nombre de empresa, nombre de usuario, DNI, cargo y contraseña de acceso.
Excepciones	2.B Si los datos de accesos son incorrectos el sistema informará de ello.

Tabla 44 C.U.01 Inicio sesión

CU.02 Ver Categorías sistema

Descripción	Una cuenta consulta todas las categorías disponibles en el sistema y filtra dicha información.
Actor	Cuenta no autorizada
Precondiciones	Se requiere haber iniciado sesión en el sistema
Postcondiciones	El usuario puede consultar el listado de categorías del sistema
Flujo normal	<ol style="list-style-type: none">1. El usuario solicita ver las categorías del sistema a través de la opción "Categorías".2. El sistema muestra todas las categorías del sistema.
Flujo alternativo	2.A El usuario puede filtrar por palabra u ordenar las categorías que muestra el sistema. 2.B El usuario puede descargar en PDF el listado con los filtros que ha realizado.
Excepciones	2.C Si hay un problema con la carga de categorías la página se quedará cargando. 2.D Si no hay resultados disponibles para el filtrado no se muestran categorías para esa búsqueda.

Tabla 45 CU.02 Ver categorías sistema

CU.03 Cerrar sesión

Descripción Una cuenta cierra sesión en el sistema web.

Actor Cuenta no autorizada.

Precondiciones

Postcondiciones El usuario no tiene sesión iniciada en el sistema.

Flujo normal

1. El usuario solicita cerrar sesión en el sistema mediante la opción “Cerrar sesión” de la parte superior.
2. El sistema cierra la sesión y lo redirige a la pantalla de inicio de sesión.

Flujo

alternativo

Excepciones

Tabla 46 CU.03 Cerrar sesión

CU.04 Ver perfil

Descripción Una cuenta consulta los datos de su cuenta y de la empresa a la que pertenece.

Actor Cuenta no autorizada.

Precondiciones

Postcondiciones

Flujo normal

1. El usuario solicita ver los datos de su perfil en el apartado “Ver Perfil”.
2. El sistema muestra los datos del usuario del sistema y la información de la empresa a la que pertenece dicho perfil.

Flujo alternativo

2.A- Si el actor es una cuenta autorizada el sistema da la opción de cambiar el nombre de la empresa, si se realiza correctamente el sistema muestra un mensaje confirmándolo.

Excepciones

Tabla 47 CU.04 Ver perfil

CU.05 Modificar unidades del producto

Descripción Una cuenta modifica las unidades de un producto de su empresa.

Actor Cuenta no autorizada.

Precondiciones

Postcondiciones El producto tiene el número de unidades actualizadas que ha solicitado el usuario.

Flujo normal

1. El usuario solicita modificar el número de unidades de un producto a través de la opción “Edición Stock”.
2. El sistema solicita el EAN del producto.
3. El usuario introduce el EAN del producto.
4. El sistema comprueba el EAN y muestra el producto relativo a dicho EAN y ofrece la posibilidad de modificar las unidades del mismo.
5. El usuario introduce el número de unidades a modificar.
6. El sistema actualiza las unidades del producto y muestra un mensaje confirmando el número actual de unidades del mismo.

Flujo

alternativo

Excepciones

4.A- Si el EAN no está registrado en la empresa del usuario muestra un mensaje de error.

Tabla 48 CU.05 Modificar unidades del producto.

CU.06 Ver movimientos del sistema	
Descripción	Una cuenta consulta todos los movimientos relacionados con los productos de su empresa y filtra dicha información
Actor	Cuenta autorizada.
Precondiciones	
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario solicita ver los movimientos del sistema a través de la opción “Movimientos”. 2. El sistema muestra todos los movimientos del sistema.
Flujo alternativo	<p>2.A El usuario puede filtrar por palabra u ordenar los movimientos que se muestren.</p> <p>2.B El usuario puede descargar en PDF el listado con los filtros que ha realizado.</p> <p>2.C El usuario puede pedir consultar los detalles de cada movimiento del listado y el sistema se lo mostrará con toda la información relacionada al mismo.</p>
Excepciones	<p>2.D Si un usuario que no está autorizado intenta entrar el sistema le informa y le redirige a la página principal.</p> <p>2.E Si hay un problema con la carga de movimientos la página se quedará cargando.</p> <p>2.F Si no hay resultados disponibles para el filtrado no se muestran movimientos para esa búsqueda.</p>

Tabla 49 CU.06 Ver movimientos del sistema.

CU.07 Ver usuarios del sistema	
Descripción	Una cuenta consulta todas las cuentas de su empresa registrados en el sistema.
Actor	Cuenta no autorizada.
Precondiciones	
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario solicita ver los usuarios de su empresa del sistema. 2. El sistema muestra todos los usuarios que cumplen dichas condiciones. La cantidad de información mostrada depende de si la cuenta es autorizada o no.
Flujo alternativo	<p>2.A El usuario puede filtrar por palabra u ordenar los usuarios que se muestren.</p> <p>2.B El usuario puede descargar en PDF el listado con los filtros que ha realizado.</p>
Excepciones	<p>2.E Si hay un problema con la carga de usuarios la página se quedará cargando.</p>

Tabla 50 CU.07 Ver usuarios del sistema.

CU.08 Ver detalle usuario

Descripción	Una cuenta autorizada consulta los detalles de una cuenta específica del sistema.
Actor	Cuenta autorizada.
Precondiciones	
Postcondiciones	
Flujo normal	<ol style="list-style-type: none">1. El usuario solicita ver los detalles de una cuenta en específico desde el listado de usuarios.2. El sistema muestra los datos asociados a la cuenta y da la posibilidad de editar su contraseña.
Flujo alternativo	2.A El usuario cambia la contraseña de la cuenta y se muestra un aviso indicando que se ha realizado correctamente.
Excepciones	2.B Si hay un problema con la carga de la cuenta, la página se quedará cargando. 2. C Si hay problemas en el cambio de contraseña se notificará al usuario.

Tabla 51 CU.08 Ver detalle usuario

CU.09 Borrar usuario

Descripción	Una cuenta autorizada solicita el borrado de otra cuenta de la empresa.
Actor	Cuenta autorizada.
Precondiciones	
Postcondiciones	
Flujo normal	<ol style="list-style-type: none">1. El usuario solicita el borrado de una cuenta en específico desde el listado de usuarios.2. El sistema muestra un mensaje advirtiendo del borrado y solicitando confirmación.3. El usuario confirma el borrado.4. El sistema borra el usuario y todos los movimientos asociados.
Flujo alternativo	2.A El usuario cancela el borrado y vuelve al listado.
Excepciones	2.B Si hay un problema con la eliminación de la cuenta, el sistema informa de ello.

Tabla 52 Borrar usuario

CU.10 Ver productos del sistema

Descripción	Una cuenta consulta todos los productos de su empresa registrados en el sistema.
Actor	Cuenta no autorizada.
Precondiciones	Se requiere haber iniciado sesión en el sistema
Postcondiciones	
Flujo normal	<ol style="list-style-type: none">1. El usuario solicita ver los productos de su empresa del sistema.2. El sistema muestra todos los productos que cumplen dichas condiciones.
Flujo alternativo	<p>2.A El usuario puede filtrar por palabra u ordenar los productos que se muestren.</p> <p>2.B El usuario puede descargar en PDF el listado con los filtros que ha realizado.</p> <p>2.C El usuario puede pedir consultar los detalles de cada producto del listado y el sistema se lo mostrará con toda la información relacionada al mismo, además de dejarle editar el título, descripción, unidades y precio del mismo. En caso de editarlo el sistema notificará de la edición y creará el movimiento asociado a dicha edición.</p> <p>2.E El usuario puede registrar un nuevo producto en el sistema introduciendo EAN, título, descripción, talla, unidades, precio y seleccionando una categoría, el sistema mostrará un mensaje confirmando el registro del producto.</p>
Excepciones	<p>2.E Si hay un problema con la carga de productos la página se quedará cargando.</p> <p>2.F Si hay problemas en la edición el sistema notificará al usuario.</p> <p>2.G Si hay problemas en el registro de un nuevo producto o alguno de los datos introducidos no son válidos el sistema notificará al usuario.</p>

Tabla 53 CU.10 Ver productos del sistema.

CU.11 Borrar producto

Descripción	Una cuenta solicita el borrado de un producto de la empresa.
Actor	Cuenta no autorizada.
Precondiciones	
Postcondiciones	
Flujo normal	<ol style="list-style-type: none">1. El usuario solicita el borrado de un producto en específico desde el listado de productos.2. El sistema muestra un mensaje advirtiendo del borrado y solicitando confirmación.3. El usuario confirma el borrado.4. El sistema borra el producto y todos los movimientos asociados.
Flujo alternativo	2.A El usuario cancela el borrado y vuelve al listado.
Excepciones	2.B Si hay un problema con el borrado del producto, el sistema informa de ello.

Tabla 54 CU.11 Borrar producto.

6.2 MODELO CONCEPTUAL

6.2.1 Modelo inicial

A continuación, se muestra la primera iteración creada del modelo de la base de datos, que es con el que se comenzó la realización del proyecto y con los campos que se creían necesarios para completar toda la funcionalidad.

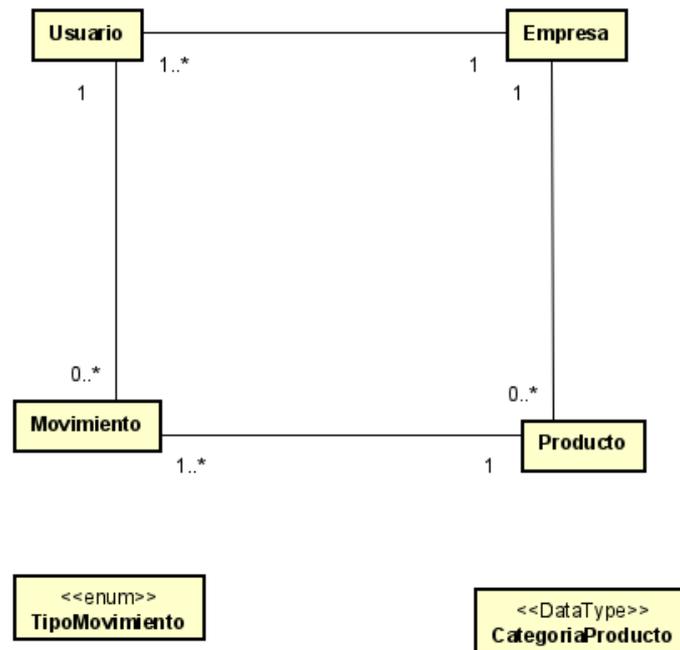


Figura 19 Modelo conceptual inicial

6.2.2 Modelo final

A continuación, se muestra el modelo final una vez se ha acabado el proyecto, con todos los cambios sobre la base de datos realizados.

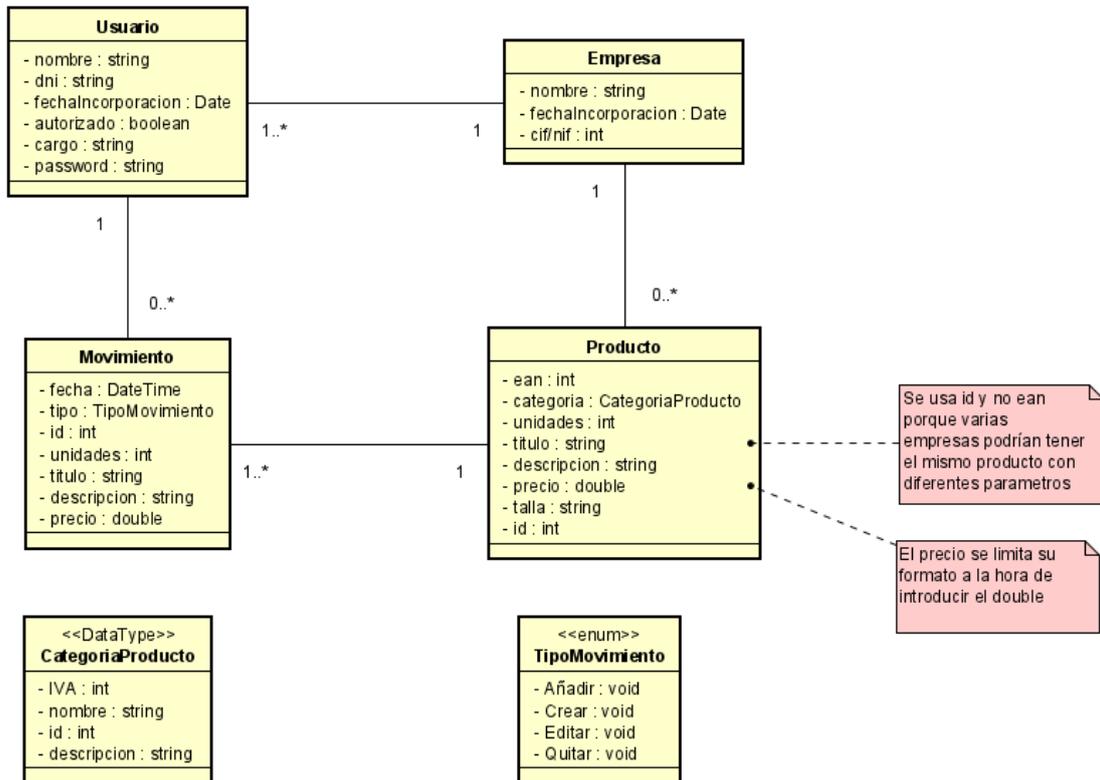


Figura 20 Modelo conceptual final.

6.3 PATRONES DE DISEÑO

Los patrones de diseño, conocidos popularmente en inglés como *design patterns* son soluciones generales, reutilizables y aplicables a diferentes problemas de diseño software. Son plantillas que identifican problemas de carácter general y proponen soluciones apropiadas para estos.

A continuación, se muestran algunos de los patrones de diseño usados en el proyecto, muchos de ellos se usan debido a las tecnologías usadas.

6.3.1 Patrón DAO

El patrón DAO o *Data Access Object* nos permite independizar la forma de acceder la base de datos desde nuestra aplicación. Para ello lo que hace es centrar el código que accede a las bases de datos o repositorios de datos en las llamadas clases DAO, fuera de estas clases no debe haber ningún acceso a la base de datos para que se cumpla este patrón.

La mayor ventaja de tener centralizado todo el código de acceso en estas clases es que para cambiar todo el código relativo al acceso de datos solo tenemos que cambiar estas clases, además de ser mucho más sencillo implementar políticas de seguridad debido a esta misma razón. [48]

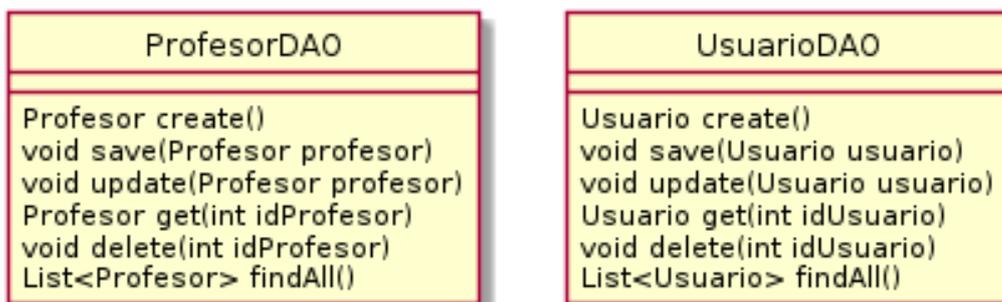


Figura 21 Ejemplo clases DAO

6.3.2 Patrón DTO

Los DTO o *Data Transfer Objects* son objetos que transportan datos entre procesos para poder reducir el número de llamadas. Nos sirven para poder generar un mayor aislamiento entre los objetos de negocio y las aplicaciones sencillas.

A continuación, se muestra un ejemplo lo que es un DTO, son estructuras de datos planas sin lógica de negocio, solo contienen las variables del objeto y los accesos a las mismas. Estos datos se relacionan con los DTO correspondientes para poder realizar operaciones con los objetos. [49]



Figura 22 Ejemplo DTO

6.3.3 Patrón Factoría

El patrón factoría o *Factory Method* es un patrón creacional que permite asegurar que una clase tenga una única instancia, además de proporcionar un punto de acceso a dicha instancia desde toda la aplicación. [50]

Sirve para simplificar el código y asegura que la clase solo se inicializa tras el primer uso, teniendo los demás usos un acceso a la clase ya inicializada.

La estructura de una clase que aplica este patrón quedaría como en el siguiente diagrama.

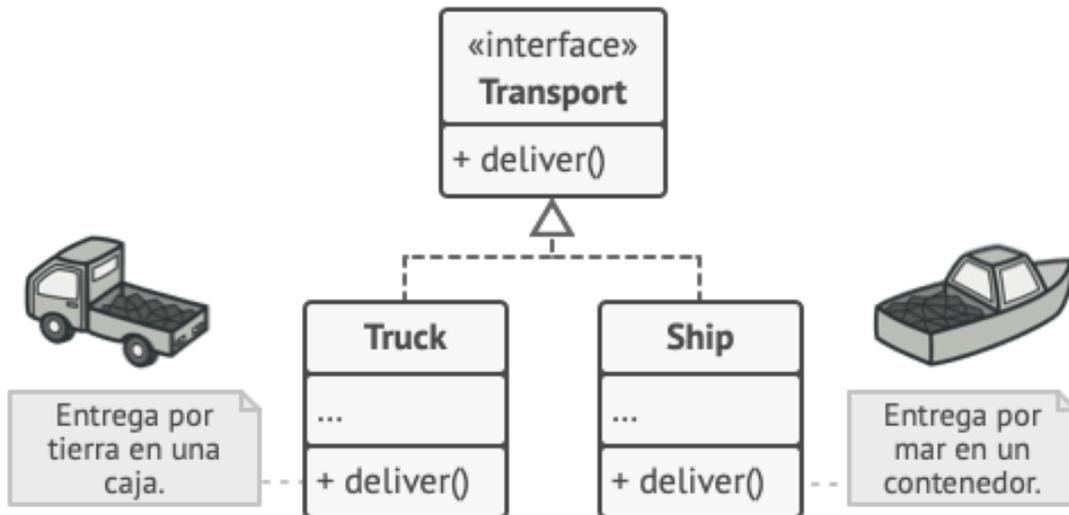


Figura 23 Estructura Factoría

6.3.4 Patrón MVC

El patrón MVC, más conocido como *Modelo-Vista-Controlador* se utiliza para implementar interfaces de usuario, datos y lógica de control. Realiza una separación entre la capa de la lógica de negocios y su visualización. Esta separación facilita la mantenibilidad de la aplicación y realiza una mejor división del trabajo. [51]

A continuación, se muestra la descripción de cada una de las partes:

- **Modelo:** Es el encargado de los datos y la lógica de negocio, define qué datos debe contener la aplicación, si estos datos cambian el modelo será el responsable de notificar a la vista y al controlador en caso de ser necesario.
- **Vista:** Es en encargado de la presentación, la parte que define como se deben mostrar los datos de la aplicación.
- **Controlador:** Es el encargado de la conexión de la vista y el modelo, contiene la capa lógica que actualiza el modelo y la vista en respuesta a las peticiones del usuario.

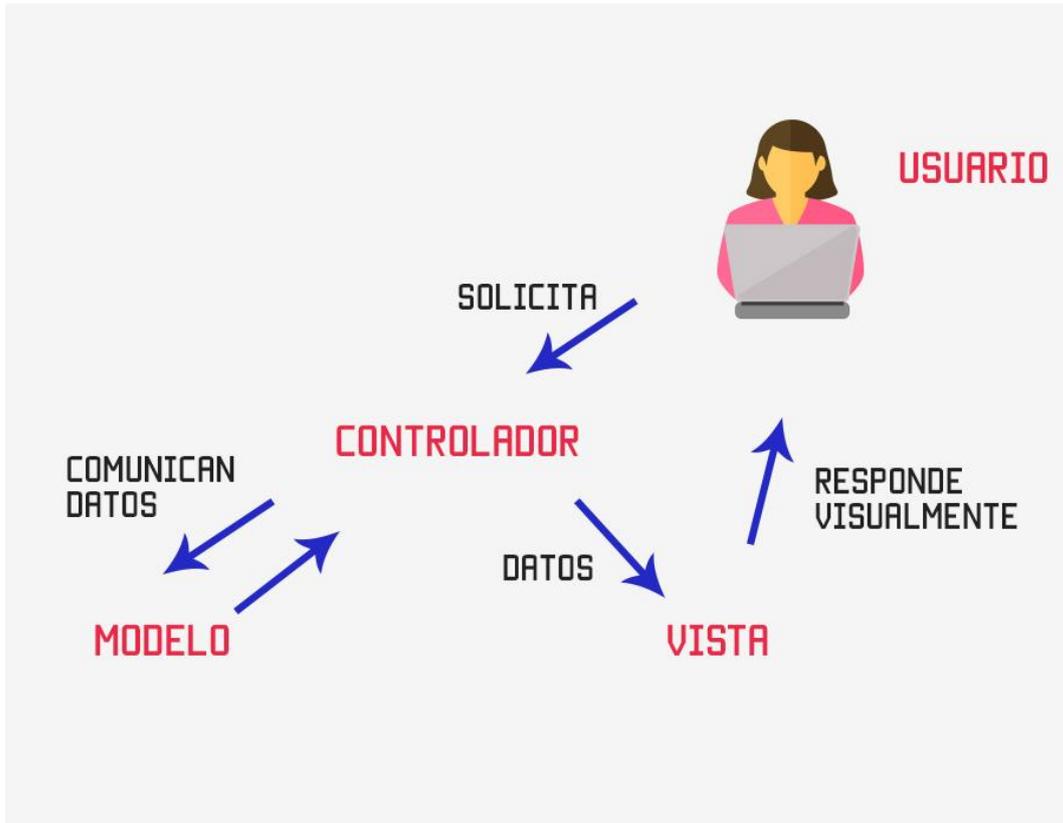


Figura 24 Esquema MVC de códigofacilito [52].

6.4 DIAGRAMA JERÁRQUICO

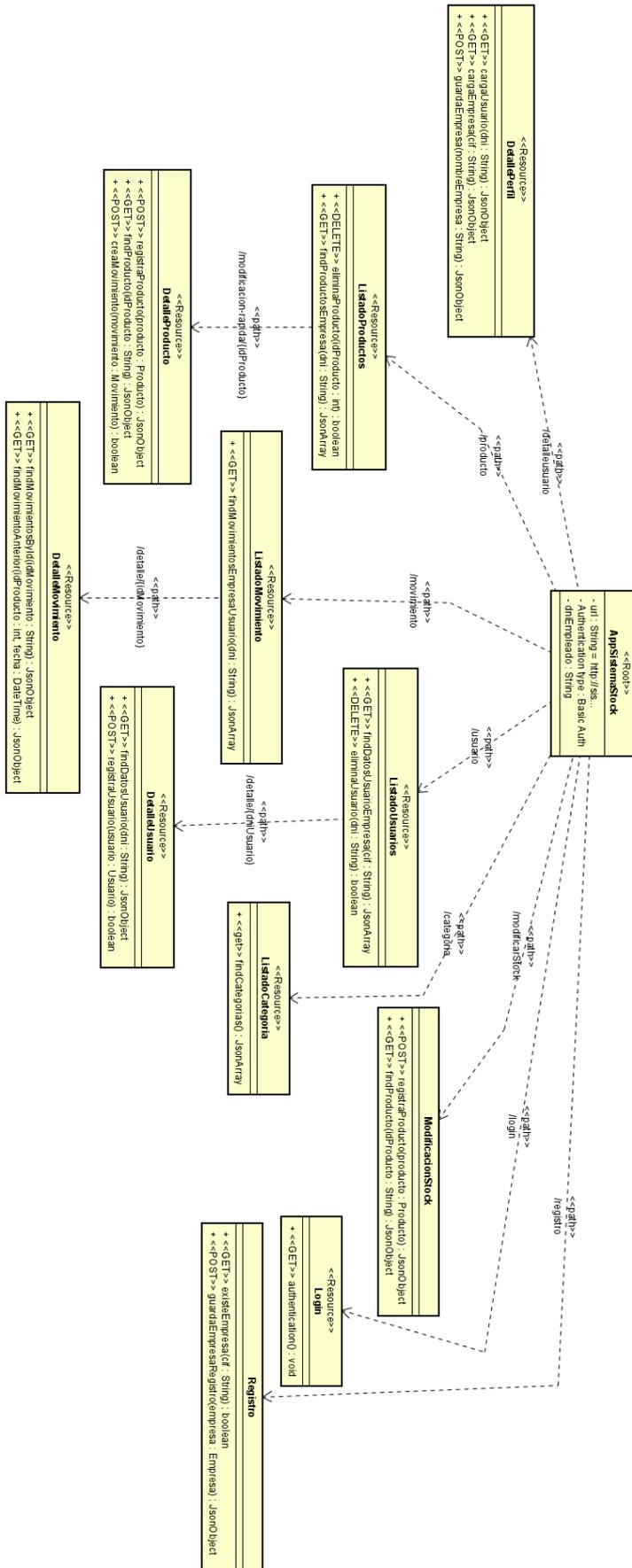


Figura 25 Diagrama jerárquico del proyecto

Un diagrama de jerarquía nos muestra una visión global sobre las operaciones de API REST que se realizan en el proyecto y como se puede acceder a ellas a través de las diferentes URLs, en el apartado 7.2 se ve con detalle que hace cada una de las operaciones descritas en el diagrama.

En el caso de que el usuario no haya iniciado sesión e intente entrar en una ruta que no pueda se le redirige a la pantalla de inicio de sesión. Lo mismo pasa si ya ha iniciado sesión e intenta entrar en la pantalla de inicio de sesión, aunque en este caso le llevará al menú principal de la aplicación.

7 IMPLEMENTACIÓN

7.1 INTRODUCCIÓN

En este apartado se va a describir el diseño que se ha realizado en el proyecto de forma detallada y la arquitectura del mismo, sobre todo se centra en el Backend y la API usada para la conexión con la base de datos.

7.2 ARQUITECTURA

Para poder acceder a los datos del proyecto usamos llamadas a métodos HTTP hacía el Backend realizado en Spring Boot, que a su vez conecta con la base de datos para obtener los datos. Las operaciones que se pueden realizar ya están predefinidas y se consumen en el Frontend realizado en Angular mediante servicios que hacen dichas llamadas. A continuación, se describen las operaciones que se pueden realizar.

7.2.1 Basic Auth

La autenticación de acceso básica, comúnmente conocida como Basic Auth es un método que permite proveer credenciales en el navegador para poder acceder a datos cuando se solicitan en el servidor. [53]

En el caso del proyecto se realiza para bloquear el acceso a las rutas que no queramos que accedan usuarios no registrados, para ello en cada petición que se manda desde el Frontend al Backend que requiera dicha autorización hay que pasar los parámetros de la misma. Estos parámetros se almacenan en la memoria del navegador al iniciar sesión como un token cifrado que se manda en cada petición mediante un servicio de interceptación realizado que detecta si es necesario.

Las únicas rutas de la API accesibles sin necesidad de autorización son las relacionadas con “/registro/**” que son las encargadas del nuevo registro en el sistema de una empresa y una nueva cuenta.

7.2.2 Formato peticiones de las operaciones

Todas las operaciones tienen el formato estándar de métodos HTTP para GET, POST y DELETE en este proyecto, y los resultados son en JSON que se interpreta desde el frontend para poder hacer un manejo correcto de los datos.

Por ejemplo, si se quiere obtener todos los productos de la empresa a la que pertenece el DNI de un usuario la petición mandada sería:

```

fetch("http://sistemastock-env.eba-im24g8nr.eu-central-1.elasticbeanstalk.com/producto/getByDni/1n", {
  "headers": {
    "accept": "application/json, text/plain, */*",
    "accept-language": "es-ES,es;q=0.9,zh-CN;q=0.8,zh;q=0.7,en;q=0.6",
    "authorization": "Basic MW46cGFz",
    "cache-control": "no-cache",
    "content-type": "application/json",
    "pragma": "no-cache"
  },
  "referrer": "http://sistemastock.s3-website.eu-central-1.amazonaws.com/",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body": null,
  "method": "GET",
  "mode": "cors",
  "credentials": "include"
});

```

En este caso es una petición de tipo GET, que como se ve en las cabeceras necesita autenticación para verse, por lo que se le pasa el parámetro “authorization” con el token y la URL de solicitud es *URLDespliegue/producto/getByDni/{dni del usuario que quiero consultar}*, si la operación se realiza satisfactoriamente nos devuelve un JSONArray con los datos solicitados, que en este caso serían dos productos:

```

[
  {
    "id": 3,
    "ean": "1234567890123",
    "unidades": 9,
    "titulo": "Pantalones chandal Mujer",
    "descripcion": "Ejemplo composición de los pantalones",
    "precio": 19.0,
    "talla": "M",
    "categoria": {
      "id": 2,
      "nombre": "Ropa y accesorios\t",
      "descripcion": "Complementos, chaquetas, sujetadores, vestidos, sombreros, ropa de abrigo, pantalones, camisas, pantalones cortos, faldas, pijamas, calcetines, medias, trajes, jerséis, bañadores, ropa interior\t",
      "iva": 21
    },
    "cif": "cifp"
  },
  {
    "id": 4,
    "ean": "1231231231231",
    "unidades": 1,
    "titulo": "Ropa estandar",
    "descripcion": "Ropa estandar",
    "precio": 2.33,
    "talla": "Única",
    "categoria": {
      "id": 2,
      "nombre": "Ropa y accesorios\t",
      "descripcion": "Complementos, chaquetas, sujetadores, vestidos, sombreros, ropa de abrigo, pantalones, camisas, pantalones cortos, faldas, pijamas, calcetines, medias, trajes, jerséis, bañadores, ropa interior\t",
      "iva": 21
    },
    "cif": "cifp"
  }
]

```

7.2.2.1 GET – Inicio de sesión

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

URLDespliegue/basicauth

Dentro tiene que tener como parámetro en las cabeceras el token relacionado con los datos que se han proporcionado de inicio de sesión, devuelve 200 OK en caso de ser datos correctos y 401 *Unauthorized* en caso de ser incorrectos.

7.2.2.2 GET – Mostrar producto por idProducto

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

URLDespliegue /producto/getProducto/{idProducto}

Donde idProducto es el identificador único del producto en el sistema. En caso de no existir un producto con dicho identificador devuelve NULL, y en caso de haberlo devuelve un JSONObject, este sería un ejemplo de respuesta:

```
{
  "id": 4,
  "ean": "1231231231231",
  "unidades": 1,
  "titulo": "Ropa estandar",
  "descripcion": "Ropa estandar",
  "precio": 2.33,
  "talla": "Única",
  "categoria": {
    "id": 2,
    "nombre": "Ropa y accesorios\t",
    "descripcion": "Complementos, chaquetas, sujetadores, vestidos, sombreros, ropa de abrigo, pantalones, camisas, pantalones cortos, faldas, pijamas, calcetines, medias, trajes, jerséis, bañadores, ropa interior\t",
    "iva": 21
  },
  "cif": "cifp"
}
```

7.2.2.3 GET – Mostrar producto por EAN de una empresa

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

URLDespliegue /producto/getProductoCif/{ean}/{cif}

Donde cif es el identificador único de la empresa en el sistema y EAN es uno de los parámetros del producto, en el sistema no se puede registrar en una misma empresa un producto con dos EAN iguales. En caso de no existir un producto con dicho identificador devuelve NULL, y en caso de haberlo devuelve un JSONObject, este sería un ejemplo de respuesta:

```

{
  "id": 10,
  "ean": "5038061139525",
  "unidades": 13,
  "titulo": "Remington Afeitadora Rotativa Virtually Indestructible",
  "descripcion": "Remington Afeitadora Rotativa Virtually Indestructible - A
  rmazón de Policarbonato Prácticamente Indestructible, Resistente al Agua, Cabe
  zal Pivotante, Autonomía 60 min, Gris y Amarillo - PR1855\n",
  "precio": 49.99,
  "talla": "Única",
  "categoria": {
    "id": 7,
    "nombre": "Electrónica",
    "descripcion": "Televisores, reproductores de CD, equipos de sonido pa
    ra vehículos, GPS, teléfonos, cámaras\t,Fundas, cables, cargadores, soportes (
    o soportes de pared), auriculares",
    "iva": 21
  },
  "cif": "cifp"
}

```

7.2.2.4 POST – Registrar producto

En este caso es una llamada de tipo POST a la siguiente dirección:

URLDespliegue /producto/guardarProducto

Los parámetros del producto que queremos registrar se introducen directamente en el cuerpo de la petición, este por ejemplo sería un registro de un producto nuevo:

```

fetch("http://sistemastock-env.eba-im24g8nr.eu-central-
1.elasticbeanstalk.com/producto/guardarProducto", {
  "headers": {
    "accept": "application/json, text/plain, */*",
    "accept-language": "es-ES,es;q=0.9,zh-
CN;q=0.8,zh;q=0.7,en;q=0.6",
    "authorization": "Basic MW46cGFz",
    "cache-control": "no-cache",
    "content-type": "application/json",
    "pragma": "no-cache"
  },
  "referrer": "http://sistemastock.s3-website.eu-central-
1.amazonaws.com/",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body":
  "{\"id\":0,\"ean\": \"1234567888888\", \"unidades\":0, \"titulo\": \"Ejemplo\", \"
  descripcion\": \"Ejemplo\", \"precio\":0, \"talla\": \"Única\", \"categoria\": {\"i
  d\":27, \"nombre\": \"Alimentación básica\t\", \"descripcion\": \" pan común,
  harinas panificables, leche, queso, huevos, frutas, verduras, hortalizas,
  legumbres, tubérculos y
  cereales\", \"iva\":4}, \"cif\": \"cifp\", \"movimientos\":[]}",
  "method": "POST",
  "mode": "cors",
  "credentials": "include"
});

```

Si el registro se realiza de forma correcta se devuelve el propio objeto recién creado desde el servidor, con la misma estructura que se ha visto en la respuesta del apartado 7.2.2.3.

7.2.2.5 **POST – Modificar stock de un producto**

En este caso es una llamada de tipo POST a la siguiente dirección:

```
URLDespliegue /producto/modificarStock
```

Los parámetros que se pasan y la forma de mandar la petición son los mismos que en el apartado 7.2.2.4 lo único que en este caso particular solo se tienen en cuenta el campo idProducto y el campo unidades para modificarlos en la base de datos. La respuesta es “true” si se modifica correctamente y “false” si da algún tipo de error.

7.2.2.6 **DELETE – Borrar un producto**

En este caso es una llamada de tipo DELETE a la siguiente dirección:

```
URLDespliegue/producto/eliminarProducto/{idProducto}
```

La petición es muy parecida a la del 7.2.2, solo que en este cambio cambia la url y el método, el resto se envía de la misma forma. La respuesta es “true” si se borra correctamente y “false” si da algún tipo de error. Hay que tener en cuenta que se borran todos los movimientos asociados a dicho producto.

7.2.2.7 **GET – Listado productos de empresa por usuario**

En este caso es una llamada de tipo GET a la siguiente dirección:

```
URLDespliegue/producto/getByDni/{dni}
```

Tanto la petición como la respuesta son las que se usan como ejemplo en el apartado 7.2.2

7.2.2.8 **GET – Empresa a la que pertenece un usuario.**

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

```
URLDespliegue/empresa/getByDni/{dni}
```

En caso de no existir dicha empresa nos devuelve vacío, y si existe nos devuelve un JSONObject con los datos de la empresa:

```
{  
  "cif": "cifp",  
  "nombre": "Nombre empresa",  
  "fechaincorporacion": "2022-01-11"  
}
```

7.2.2.9 **POST – Guardar una empresa.**

En este caso es una llamada de tipo POST similar a la del apartado 7.2.2.4 a la siguiente dirección:

`URLDespliegue/registro/guardarEmpresa`

En este caso en el apartado de body tendremos los datos de la empresa y al formar parte de /registro/* no nos hace falta el token de autorización en los headers.

Si el registro se realiza de forma correcta se devuelve el propio objeto recién creado desde el servidor, con la misma estructura que se ha visto en la respuesta del apartado 7.2.2.8.

7.2.2.10 GET – *Confirmar existencia empresa por CIF*

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

`URLDespliegue/registro/exiteEmpresaCif/{cif}`

En este caso al formar parte de /registro/* no nos hace falta el token de autorización en los headers. En caso de no existir dicha empresa nos devuelve false y si existe devuelve true.

7.2.2.11 POST – *Guardar usuario*

En este caso es una llamada de tipo POST a la siguiente dirección:

`URLDespliegue/registro/guardarUsuarios`

Esta es una petición de ejemplo donde se ve el formato:

```
fetch("http://sistemastock-env.eba-im24g8nr.eu-central-1.elasticbeanstalk.com /registro/guardarUsuarios", {
  "headers": {
    "accept": "application/json, text/plain, */*",
    "accept-language": "es-ES,es;q=0.9,zh-CN;q=0.8,zh;q=0.7,en;q=0.6",
    "cache-control": "no-cache",
    "content-type": "application/json",
    "pragma": "no-cache",
    "sec-ch-ua": "\" Not A;Brand\";v=\"99\", \"Chromium\";v=\"102\", \"Google Chrome\";v=\"102\"",
    "sec-ch-ua-mobile": "?0",
    "sec-ch-ua-platform": "\"Windows\"",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-site"
  },
  "referrer": " http://sistemastock.s3-website.eu-central-1.amazonaws.com/",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body":
  "{\n\"dni\": \"123456127\", \"cif\": \"Q2826000A\", \"nombre\": \"Luis garcía\", \"fechaincorporacion\": \"2022-06-07T22:00:00.000Z\", \"autorizado\": true, \"cargo\": \"jefe\", \"password\": \"password\", \"movimientos\": []}",
  "method": "POST",
  "mode": "cors",
  "credentials": "omit"
});
```

Si el usuario se guarda correctamente nos devuelve true, pero si hay algún tipo de problema nos devuelve false. En caso de intentar guardar un usuario con un dni que ya existe en el sistema devuelve false.

7.2.2.12 GET – Usuarios empresa

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

```
URLDespliegue/usuario/getAllUsuariosEmpresa/{cif}
```

Como respuesta nos devuelve un JSONArray con los usuarios de dicha empresa, siempre tiene que haber al menos un usuario, por ejemplo, esta sería la respuesta para una empresa con un único usuario:

```
[
  {
    "dni": "3N",
    "nombre": "Luis Garcia",
    "fechaincorporacion": "2015-01-01",
    "autorizado": true,
    "cargo": "currito",
    "cif": "cifp2",
    "password": "pas",
    "movimientos": [
      {
        "id": 57,
        "fecha": "2022-05-27T18:15:45",
        "dni": "3N",
        "titulo": "asdasda",
        "descripcion": "asdasdasda",
        "precio": 0.0,
        "unidades": 0,
        "idproducto": {
          "id": 15,
          "ean": "1234567890123",
          "unidades": 0,
          "titulo": "asdasda",
          "descripcion": "asdasdasda",
          "precio": 0.0,
          "talla": "Única",
          "categoria": {
            "id": 17,
            "nombre": "Bricolaje y herramientas",
            "descripcion": "\tHerramientas manuales, elementos de montaje, ropa de trabajo, equipos de protección, alarmas de incendios y sistemas de seguridad, herramientas eléctricas, accesorios de herramientas eléctricas, productos eléctricos, decoración, herramientas de jardín, materiales de construcción, fontanería y baño, componentes",
            "iva": 21
          },
          "cif": "cifp2"
        },
        "idtipo": {
          "idTipo": 4,
          "nombreTipo": "Creación producto"
        }
      }
    ]
  }
]
```

7.2.2.13 GET – Mostrar usuario por DNI

En este caso es una llamada de tipo GET a la siguiente dirección:

`URLDespliegue/usuario/getByDni/{dni}`

La petición es muy parecida a la del 7.2.2, solo que en este cambio cambia la url. La respuesta es vacía si no existe ese usuario o un JSONObject con los datos del usuario si existe. Por ejemplo:

```
{
  "dni": "3N",
  "nombre": "Luis García",
  "fechaincorporacion": "2015-01-01",
  "autorizado": true,
  "cargo": "currito",
  "cif": "cifp2",
  "password": "pas",
  "movimientos": [
    {
      "id": 57,
      "fecha": "2022-05-27T18:15:45",
      "dni": "3N",
      "titulo": "asdasda",
      "descripcion": "asdasdasda",
      "precio": 0.0,
      "unidades": 0,
      "idproducto": {
        "id": 15,
        "ean": "1234567890123",
        "unidades": 0,
        "titulo": "asdasda",
        "descripcion": "asdasdasda",
        "precio": 0.0,
        "talla": "Única",
        "categoria": {
          "id": 17,
          "nombre": "Bricolaje y herramientas",
          "descripcion": "\tHerramientas manuales, elementos de montaje, ropa de trabajo, equipos de protección, alarmas de incendios y sistemas de seguridad, herramientas eléctricas, accesorios de herramientas eléctricas, productos eléctricos, decoración, herramientas de jardín, materiales de construcción, fontanería y baño, componentes",
          "iva": 21
        }
      },
      "cif": "cifp2"
    },
    {
      "idtipo": {
        "idTipo": 4,
        "nombreTipo": "Creación producto"
      }
    }
  ]
}
```

7.2.2.14 DELETE – Borrar usuario

En este caso es una llamada de tipo DELETE a la siguiente dirección:

```
URLDespliegue/usuario/eliminarUsuario/{dni}
```

La petición es muy parecida a la del 7.2.2, solo que en este cambio cambia la url y el método, el resto se envía de la misma forma. La respuesta es “true” si se borra correctamente y “false” si da algún tipo de error. Hay que tener en cuenta que se borran todos los movimientos asociados a dicho usuario.

7.2.2.15 GET – Confirmar la existencia un usuario con un DNI

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

```
URLDespliegue/usuario/existsDni/{dni}
```

La petición es muy parecida a la del 7.2.2, solo que en este cambio cambia la url. En caso de no existir dicho usuario nos devuelve false y si existe devuelve true.

7.2.2.16 GET- Listado de categorías del sistema

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

```
URLDespliegue/categoria/getAllCategorias
```

La respuesta es un JSONArray con todas las categorías del sistema, por ejemplo:

```
[
  {
    "id": 1,
    "nombre": "Alimentación y bebidas\t",
    "descripcion": "Bebidas, aperitivos y postres, condimentos, productos
de consumo para el hogar, comida para bebés\t",
    "iva": 10
  },
  {
    "id": 2,
    "nombre": "Ropa y accesorios\t",
    "descripcion": "Complementos, chaquetas, sujetadores, vestidos, sombre
ros, ropa de abrigo, pantalones, camisas, pantalones cortos, faldas, pijamas,
calcetines, medias, trajes, jerséis, bañadores, ropa interior\t",
    "iva": 21
  },
  {
    "id": 3,
    "nombre": "Libros, Música, Vídeo y DVD",
    "descripcion": "Música, libros, vídeo, DVD, Blu-ray",
    "iva": 10
  }
]
```

7.2.2.17 GET – Listado de movimientos de una empresa

En este caso es una llamada de tipo GET como la del ejemplo del apartado 7.2.2 a la siguiente dirección:

```
URLDespliegue/movimiento/getAllMovimientosEmpresa/{dni}
```

Se pasa como parámetro en la url el DNI de uno de los empleados en la empresa, como respuesta se devuelve vacío si no hay movimientos y un JSONArray de movimientos en caso de haberlos, por ejemplo:

```
[
  {
    "id": 38,
    "fecha": "2022-05-16T19:38:58",
    "dni": "1N",
    "titulo": "Pantalones chandal Mujer",
    "descripcion": "Ejemplo composición de los pantalones",
    "precio": 19.99,
    "unidades": 10,
    "idproducto": {
      "id": 3,
      "ean": "1234567890123",
      "unidades": 9,
      "titulo": "Pantalones chandal Mujer",
      "descripcion": "Ejemplo composición de los pantalones",
      "precio": 19.0,
      "talla": "M",
      "categoria": {
        "id": 2,
        "nombre": "Ropa y accesorios\t",
        "descripcion": "Complementos, chaquetas, sujetadores, vestidos
, sombreros, ropa de abrigo, pantalones, camisas, pantalones cortos, faldas, p
ijamas, calcetines, medias, trajes, jerséis, bañadores, ropa interior\t",
        "iva": 21
      },
      "cif": "cifp"
    },
    "idtipo": {
      "idTipo": 4,
      "nombreTipo": "Creación producto"
    }
  },
  {
    "id": 42,
    "fecha": "2022-05-18T19:05:31",
    "dni": "1N",
    "titulo": "Pantalones chandal Mujer",
    "descripcion": "Ejemplo composición de los pantalones",
    "precio": 19.0,
    "unidades": 10,
    "idproducto": {
      "id": 3,
      "ean": "1234567890123",
      "unidades": 9,
      "titulo": "Pantalones chandal Mujer",
      "descripcion": "Ejemplo composición de los pantalones",
      "precio": 19.0,
      "talla": "M",
      "categoria": {
        "id": 2,
        "nombre": "Ropa y accesorios\t",
```

7.2.2.18 GET – Mostrar movimiento por id

En este caso es una llamada de tipo GET a la siguiente dirección:

`URLDespliegue/movimiento/getMovimientoById/{id}`

La petición es muy parecida a la del 7.2.2, solo que en este cambio cambia la URL, la id de la URL es el identificador único del movimiento en el sistema. La respuesta es vacía si no existe ese usuario o un JSONObject con los datos del usuario si existe. Por ejemplo:

```
{
  "id": 47,
  "fecha": "2022-05-24T18:20:14",
  "dni": "1N",
  "titulo": "Beurer mosquitos",
  "descripcion": "Producto salud",
  "precio": 200.0,
  "unidades": 1,
  "idproducto": {
    "id": 8,
    "ean": "4211125603182",
    "unidades": 1,
    "titulo": "Beurer mosquitos",
    "descripcion": "Producto salud",
    "precio": 200.0,
    "talla": "Única",
    "categoria": {
      "id": 21,
      "nombre": "Salud y cuidado personal",
      "descripcion": "Cuidado del cabello, cuidado bucal, cepillos y tratamientos corporales, afeitado y cabello, suministros y equipo sanitarios, masaje y relax, lámparas de bronceado, básculas de baño, ayudas a la movilidad y productos para la vida diaria.\t",
      "iva": 10
    },
    "cif": "cifp"
  },
  "idtipo": {
    "idTipo": 4,
    "nombreTipo": "Creación producto"
  }
}
```

7.2.2.19 GET – Mostrar movimiento anterior a una fecha por producto

En este caso es una llamada de tipo GET a la siguiente dirección:

`URLDespliegue/movimiento/getMovimientoAnterior/{idProducto}/{fecha}`

La petición es muy parecida a la del 7.2.2, solo que en este cambio cambia la URL, la id de la URL es el identificador único del producto en el sistema. La respuesta es vacía si no existe ese movimiento o un JSONObject con los datos del movimiento si existe. Se busca el movimiento anterior a la fecha introducida para dicho producto, la fecha tiene formato 'AAAA-MM-DDTHH:MM:SS'. Un ejemplo de respuesta sería el del apartado 7.2.2.18.

7.2.2.20 POST – Guardar un movimiento

En este caso es una llamada de tipo POST a la siguiente dirección:

URLDespliegue/movimiento/guardarMovimiento

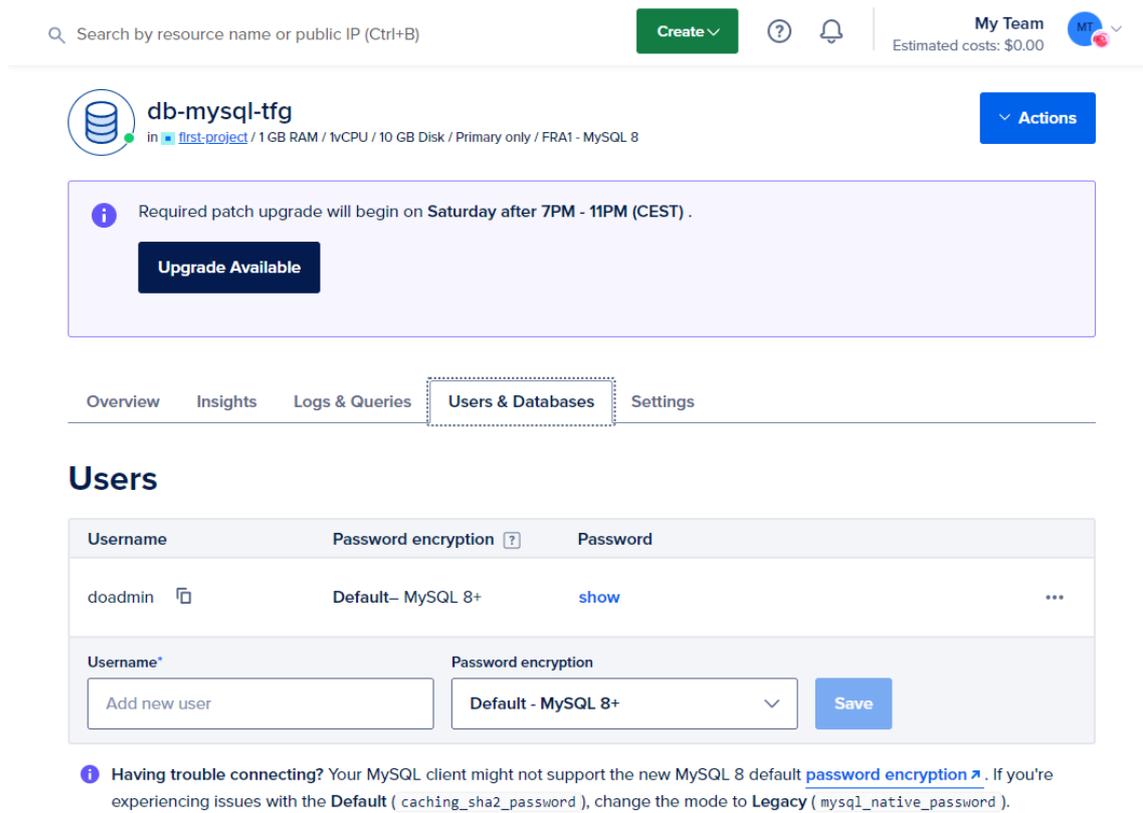
La petición tiene en el cuerpo los datos del movimiento que queremos guardar. En caso de que se guarde correctamente la respuesta será true, en caso contrario será false. Un ejemplo de petición de guardado sería:

```
fetch("http://sistemastock-env.eba-im24g8nr.eu-central-1.elasticbeanstalk.com/movimiento/guardarMovimiento", {
  "headers": {
    "accept": "application/json, text/plain, */*",
    "accept-language": "es-ES,es;q=0.9,zh-CN;q=0.8,zh;q=0.7,en;q=0.6",
    "authorization": "Basic MW46cGFzc2E=",
    "cache-control": "no-cache",
    "content-type": "application/json",
    "pragma": "no-cache",
    "sec-ch-ua": "\" Not A;Brand\";v=\"99\"",
    "\"Chromium\";v=\"102\"", "\"Google Chrome\";v=\"102\"",
    "sec-ch-ua-mobile": "?0",
    "sec-ch-ua-platform": "\"Windows\"",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-site"
  },
  "referrer": "http://sistemastock.s3-website.eu-central-1.amazonaws.com/",
  "referrerPolicy": "strict-origin-when-cross-origin",
  "body": "{\"id\":null,\"fecha\":\"2022-06-08T21:54:57.201Z\", \"idproducto\":10, \"idtipo\":1, \"dni\":\"1N\", \"titulo\": \"Remington Afeitadora Rotativa Virtually Indestructible\", \"descripcion\": \"Remington Afeitadora Rotativa Virtually Indestructible - Armazón de Policarbonato Prácticamente Indestructible, Resistente al Agua, Cabezal Pivotante, Autonomía 60 min, Gris y Amarillo - PR1855\\n\", \"precio\":49.99, \"unidades\":16}",
  "method": "POST",
  "mode": "cors",
  "credentials": "include"
});
```

8 DESPLIEGUE

En este caso el despliegue se ha realizado mediante dos plataformas distintas, en primera instancia la base de datos MySQL se ha realizado en la nube de DigitalOcean, que se ha elegido principalmente por ser gratuita para el despliegue y la puesta en funcionamiento del proyecto al tener una cuenta educativa universitaria.

La configuración de la base de datos que se realiza es muy sencilla, solo tiene un usuario que tiene todos los permisos de administrador que es al que se conecta a través de la aplicación Spring Boot para poder realizar las operaciones.



Search by resource name or public IP (Ctrl+B) Create ? ? My Team Estimated costs: \$0.00

db-mysql-tfg in first-project / 1 GB RAM / 1vCPU / 10 GB Disk / Primary only / FRA1 - MySQL 8 Actions

Required patch upgrade will begin on Saturday after 7PM - 11PM (CEST). Upgrade Available

Overview Insights Logs & Queries Users & Databases Settings

Users

Username	Password encryption ?	Password
doadmin	Default - MySQL 8+	show

Username* Password encryption

Add new user Default - MySQL 8+ Save

Having trouble connecting? Your MySQL client might not support the new MySQL 8 default [password encryption](#). If you're experiencing issues with the **Default** (`caching_sha2_password`), change the mode to **Legacy** (`mysql_native_password`).

Figura 26 Panel usuario DigitalOcean

Destacar que desde el panel se puede controlar absolutamente toda la base de datos, no solo la configuración de la misma si no que contiene hasta historiales de consultas y la IP desde donde se han realizado.

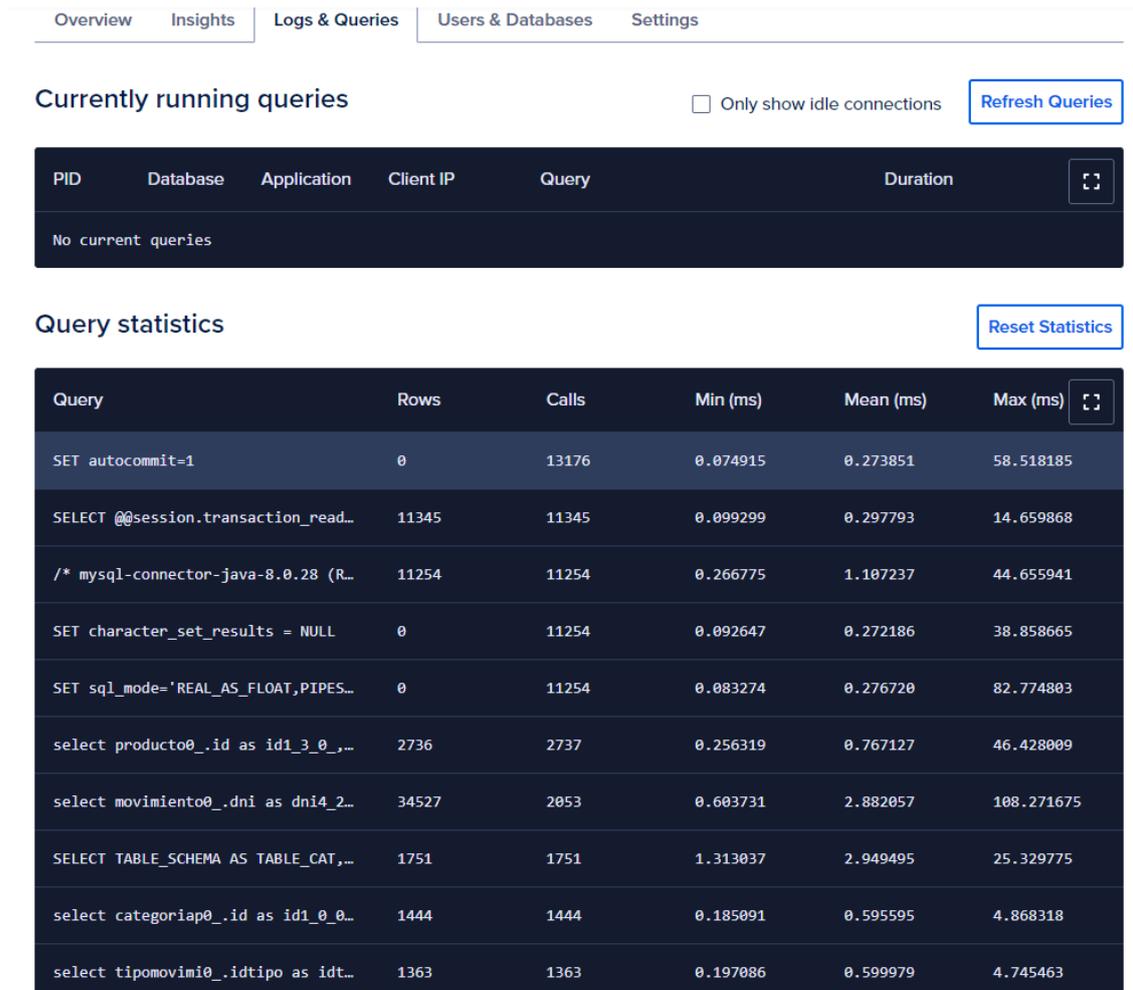


Figura 27 Logs DigitalOcean

Para la parte del proyecto de Spring Boot y Angular se hace uso de AWS para y de su *Free Tier* para el despliegue y pruebas del proyecto. Esto hace posible tener una subida inicial del código a coste cero. Además tiene otra ventaja ya que es de los más baratos de mantener a largo plazo.

Para la parte de Spring Boot se usa el servicio que denominan *Elastic Beanstalk*, que permite directamente subir el proyecto Maven creado y listo para funcionar, es tan sencillo como elegir el .jar creado por Maven al hacer un *install* de nuestro proyecto y ya la plataforma se encarga del despliegue y nos proporciona una URL de acceso.

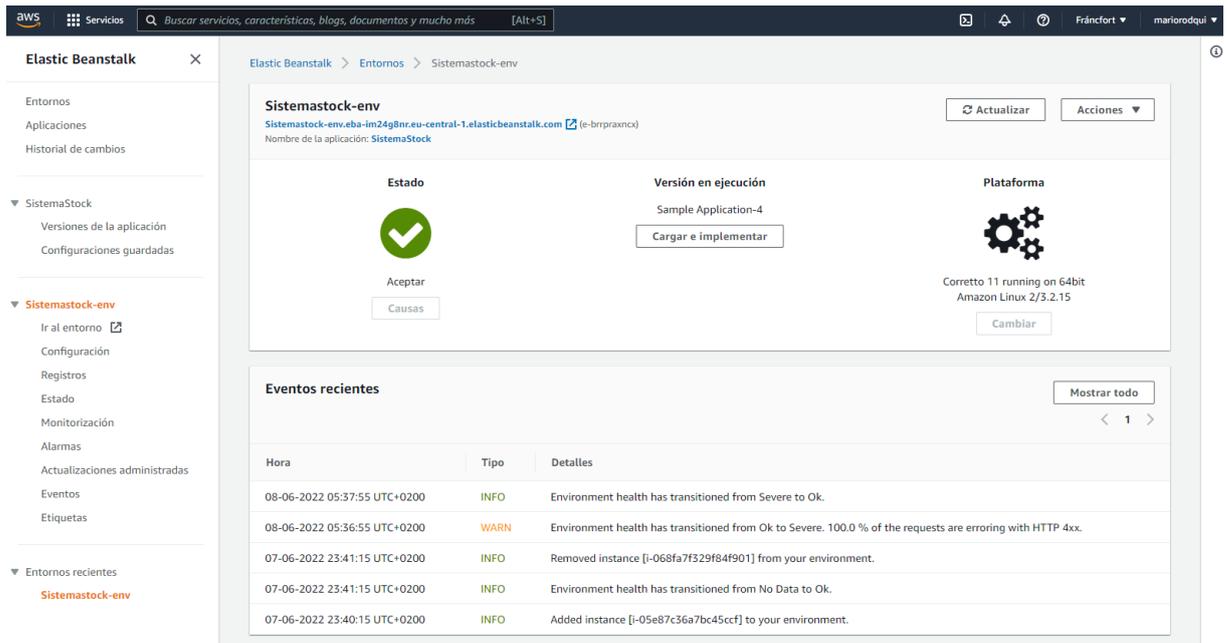


Figura 28 Panel Elastic Beanstalk AWS

La parte del despliegue del proyecto Angular es algo más compleja, primero se tiene que apuntar que las peticiones del proyecto vayan a la dirección generada para Spring Boot. Una vez realizado, se configura un “ng build” que nos crea los ficheros necesarios para el despliegue en la nube del proyecto.

El siguiente paso es ir a la parte de *Amazon S3* para crear un nuevo *bucket* que contenga todos los ficheros creados para el sistema.

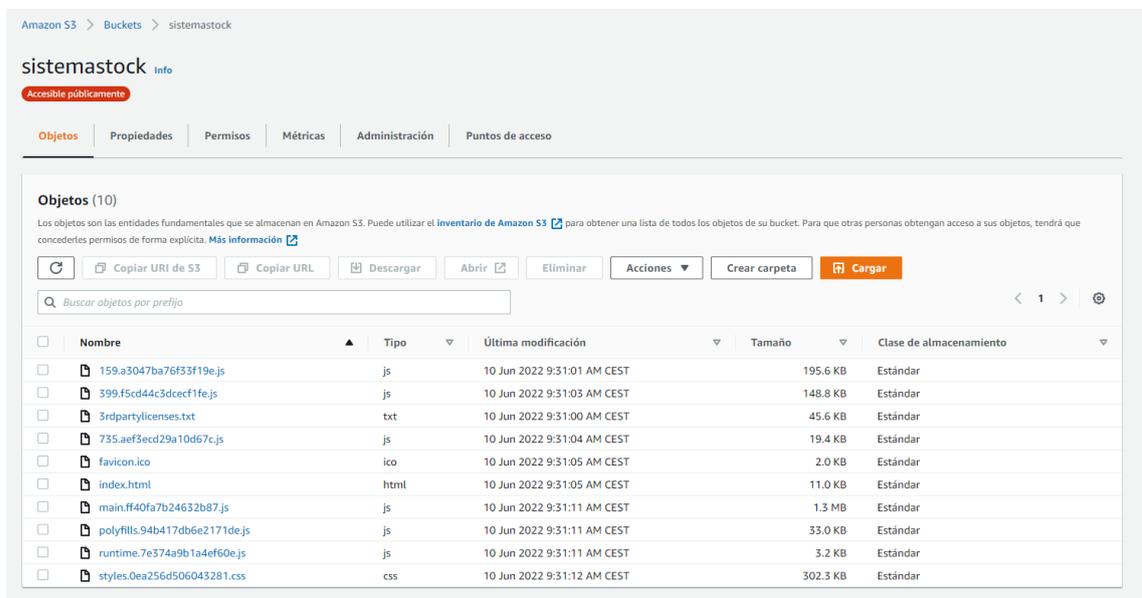


Figura 29 Panel objetos bucket Amazon S3

A continuación, hay que configurar el acceso al *bucket* como sitio web, para ello se navega a “Alojamiento de sitios web estáticos” y lo se habilita seleccionando el *bucket*.

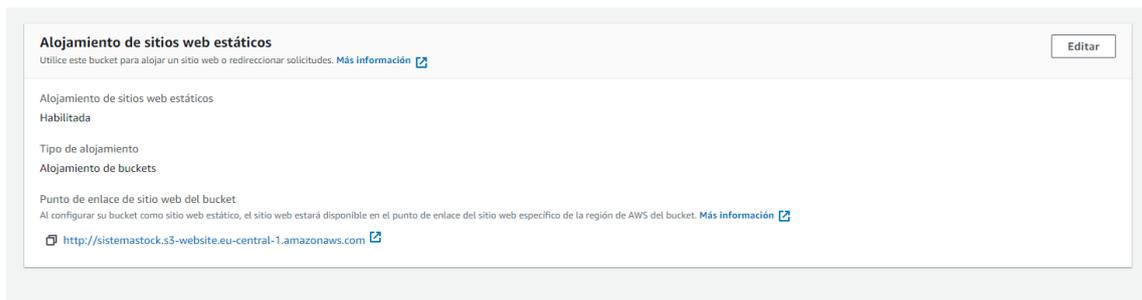


Figura 30 Amazon S3 alojamiento de sitio web estático

Es necesario configurar el *bucket* para poder tener acceso público y permitirle operaciones a la aplicación. En permisos se deshabilita el bloqueo de acceso público, luego se configura la política del *bucket* para permitir conexiones con el Backend y por último se activa la política CORS entre el Backend y el Frontend.



Figura 31 Amazon S3 configuración CORS

Con todo esto, ya está el proyecto desplegado por completo, ahora solo hay que ir a la dirección del último *bucket* creado y está todo funcional:

<http://sistemastock.s3-website.eu-central-1.amazonaws.com/>

Si se quiere poner un dominio concreto para el proyecto simplemente se puede enrutar desde el propio AWS desde el apartado *Route 53*, aunque no se ha hecho para el despliegue inicial del proyecto para reducir costes.

9 CONCLUSIONES Y TRABAJO FUTURO

9.1 CONCLUSIONES

Durante la realización del proyecto he aplicado diferentes conocimientos que he aprendido a lo largo del todo grado, especialmente los relacionados con la asignatura *Desarrollo Basado en Componentes y Servicios*, además de ello me ha servido para familiarizarme más con *AWS* y con *DigitalOcean* a la hora de crear proyectos.

He desarrollado también nuevos conocimientos tanto en *Angular* como en *Spring Boot* para solucionar cuestiones de autenticación que no había visto hasta este proyecto, además de pequeños problemas que me han surgido durante la realización y que me han llevado a buscar los métodos más óptimos de mostrar la información en algunos de los supuestos que contiene el proyecto.

Es además la primera vez que realizo solo un proyecto de tal calibre y me he tenido que planificar y seguir las técnicas de la metodología ágil aplicada a un proyecto real, incluso teniendo algún imprevisto en la realización del mismo, me he dado cuenta que en el día a día al final siempre surgen imprevistos que te desvían un poco de los plazos marcados.

Los conocimientos adquiridos en el apartado de Amazon Web Services me vienen además genial para mi entorno laboral, puesto que trabajamos a diario con ellos y la formación inicial que tuve sobre el tema fue muy vaga al ser la empresa nueva en este campo, así que el haber aplicado diversas horas en el despliegue es algo que se agradece y que aplicaré en el futuro en mi entorno laboral.

Sobre bases de datos me he dado cuenta de la importancia de tener varios entornos de pruebas a la hora de desarrollar o bien copias periódicas, por si haces cambios indebidos en uno poder sacar del otro una copia de seguridad reciente y no tener que perder tiempo poblando todo de nuevo, me ha pasado a lo largo del proyecto de modificar campos de forma indebida directamente con consultas y luego que la copia de seguridad fuera antigua y tener que poblar de nuevo todo con casos de prueba, con la consiguiente pérdida de tiempo (aunque por suerte no aumentó el tiempo de horas de esa historia de usuario).

Como conclusión general me parece que un proyecto al final del periodo académico en el que se aplican los diferentes conocimientos de la carrera con cierto carácter laboral es muy útil para poder entender los mecanismos usados de forma práctica y no solo teórica como había visto en mi caso.

9.2 TRABAJO FUTURO

En el futuro se centraría el proyecto en certificar la web con SSL y hacer más seguros el acceso a los datos de la misma con una autenticación más compleja, además de tener seguridad en los formularios de registro con un captcha o similar para que no haya problemas de accesos no deseados por terceros.

Podría también mejorarse el registro de empresa una comprobación real de NIF, para ver si los datos introducidos corresponden a una empresa real, y en caso contrario no dejar el acceso al sistema.

Se añadiría un banner lateral o inferior con publicidad para poder hacer frente de los costes si el uso supera las expectativas de los planes gratuitos de hosting que se tienen, además de que se podría dar un pequeño lavado de cara para hacer la página más atractiva visualmente.

Se puede adaptar también de forma sencilla a otros idiomas simplemente editando las palabras clave de los ficheros HTML, sin necesidad de tocar nada de la parte lógica.

Otro punto interesante sería tener una integración continua del desarrollo del software futuro, para poder tener todo organizado y facilitar el trabajo a futuros desarrolladores que se puedan incorporar al proyecto. Esto significa que se crean prototipos, se validan y se ve si hay errores antes de lanzar nada. Además, apostaría por un proceso continuo de entrega para poder realizar los despliegues de nueva funcionalidad de forma más sencilla.

10 MANUALES

En este apartado se hace un pequeño resumen de las partes más importantes de la aplicación y la lógica de funcionamiento de la misma, para poder entender todos los apartados que tiene de forma sencilla.

10.1 INICIO DE SESIÓN Y MENÚ

Al entrar en la aplicación por defecto redirige a la página de inicio de sesión si no se tienen las credenciales de acceso introducidas previamente, en caso de tenerlas redirige al menú principal.

Desde la pantalla de inicio de sesión se puede entrar en el sistema o bien registrar una nueva empresa, a la hora de registrar una nueva empresa un requisito es introducir un usuario para la misma que es el que se usa para acceder por primera vez.



Inicio Sesión

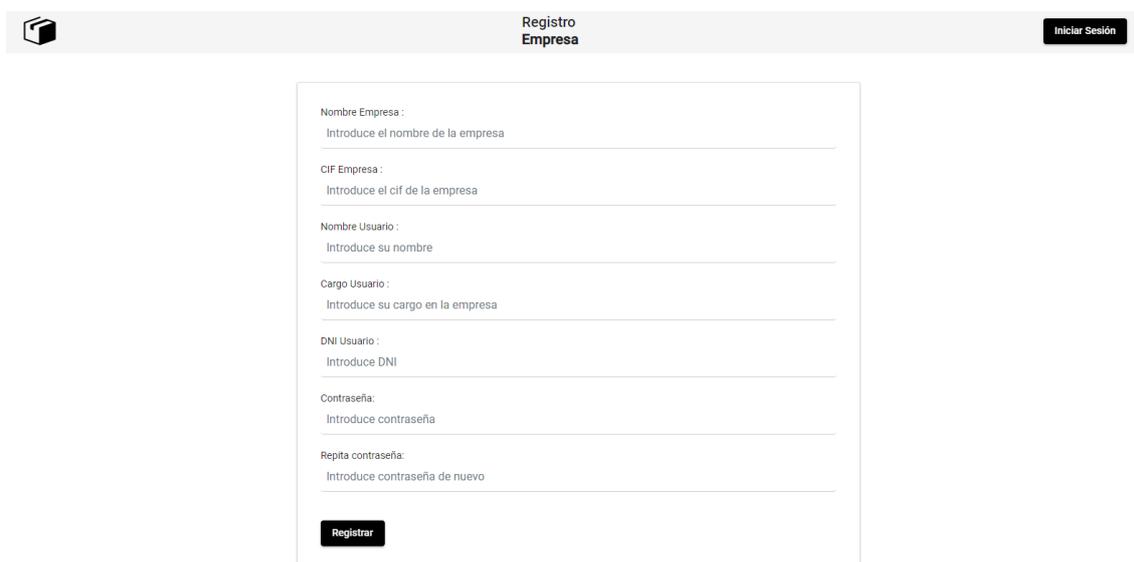
Registrar nueva empresa

Usuario:
Introduce DNI

Contraseña:
Introduce contraseña

Entrar

Figura 32 Pantalla inicio sesión



Registro Empresa

Iniciar Sesión

Nombre Empresa:
Introduce el nombre de la empresa

CIF Empresa:
Introduce el cif de la empresa

Nombre Usuario:
Introduce su nombre

Cargo Usuario:
Introduce su cargo en la empresa

DNI Usuario:
Introduce DNI

Contraseña:
Introduce contraseña

Repita contraseña:
Introduce contraseña de nuevo

Registrar

Figura 33 Pantalla registro empresa

Nada más se entra en el sistema se muestra el menú principal con opciones, además de tener siempre disponible un menú lateral que se puede minimizar para una navegación más rápida.

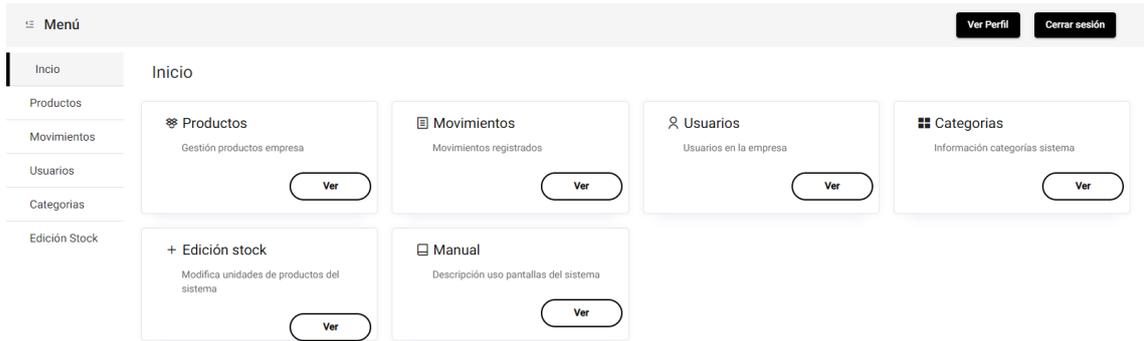


Figura 34 Menú principal

10.2 ACCESOS SIEMPRE DISPONIBLES

En pantalla se tiene siempre disponible el menú lateral ya mencionado, además de tener los botones de Ver Perfil y Cerrar sesión, este último nos saca del sistema y nos lleva a la pantalla de inicio de nuevo. Ver Perfil muestra los datos de la cuenta actual, además de los de la empresa de dicho usuario, dando la posibilidad de editar el nombre de la empresa si es una cuenta autorizada.

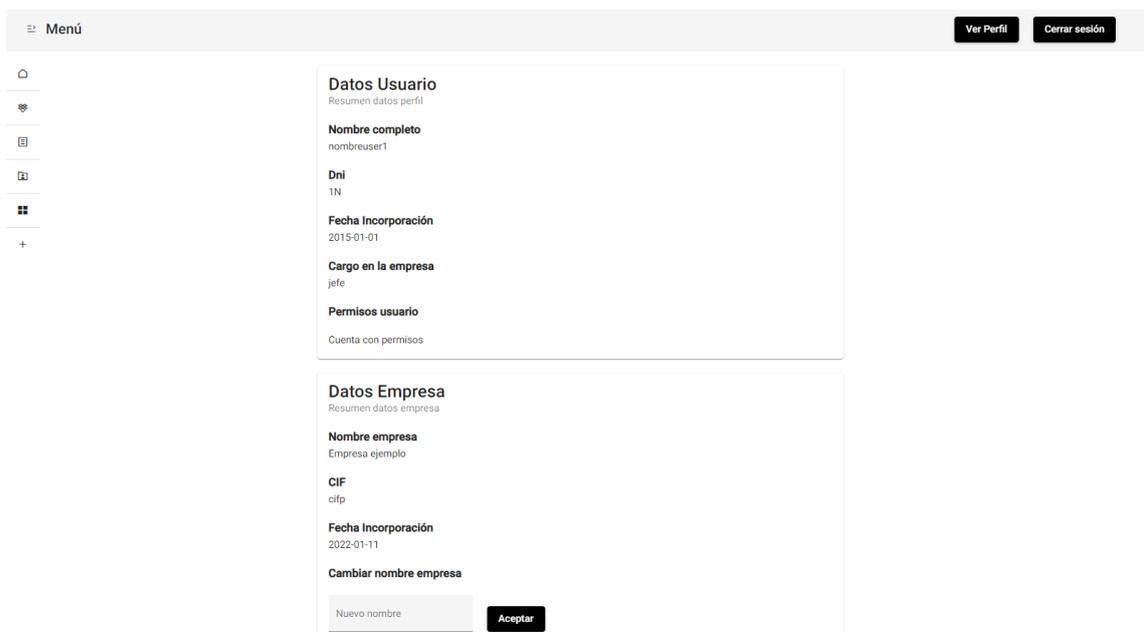


Figura 35 Pantalla Ver Perfil

10.3 MANUALES

Únicamente desde la pantalla principal se tiene acceso a un apartado de manuales, donde se describe brevemente que hace cada una de las pantallas y donde están las diferentes funcionalidades del sistema.

Pantalla "Productos"

Desde esta pantalla se pueden ver todos los productos en forma de listado. Además podemos registrar productos nuevos en el sistema. Los listados tienen la opción de descargarse, que lo hace en formato .pdf con los datos visibles en pantalla en ese momento. Son interactivos, se pueden ordenar por columnas haciendo click en el título y buscar por palabra en el buscador superior. En acciones tenemos la posibilidad de ver los detalles del producto y editarlos además de poder borrar el producto por completo si estamos autorizados.

Pantalla "Movimientos"

Pantalla solo para usuarios autorizados. Desde esta pantalla se pueden ver todos los movimientos asociados a los productos del sistema en forma de listado. Los listados tienen la opción de descargarse, que lo hace en formato .pdf con los datos visibles en pantalla en ese momento. Son interactivos, se pueden ordenar por columnas haciendo click en el título y buscar por palabra en el buscador superior. En acciones tenemos la posibilidad de ver los detalles del movimiento, que puede ser una modificación o una creación y los datos antes y después de dicho movimiento.

Pantalla "Usuarios"

Desde esta pantalla se pueden ver todos los usuarios en forma de listado. Además podemos registrar usuarios nuevos en el sistema si estamos autorizados. Los listados tienen la opción de descargarse, que lo hace en formato .pdf con los datos visibles en pantalla en ese momento. Son interactivos, se pueden ordenar por columnas haciendo click en el título y buscar por palabra en el buscador superior. En acciones tenemos la posibilidad de ver los detalles del usuario, que nos permite ver su contraseña si estamos autorizados, sus movimientos y se nos permite borrar el usuario del sistema si estamos autorizados.

Pantalla "Categorías"

Desde esta pantalla se pueden ver todas las categorías de producto en forma de listado. Los listados tienen la opción de descargarse, que lo hace en formato .pdf con los datos visibles en pantalla en ese momento. Son interactivos, se pueden ordenar por columnas haciendo click en el título y buscar por palabra en el buscador superior.

Figura 36 Pantalla Manuales

10.4 PRODUCTOS

Nada más se entra se carga un listado interactivo completo con todos los productos de la empresa. La barra de búsqueda superior permite filtrarlo por todos los elementos de un producto, incluso por palabras de la descripción del producto, aunque no estén visibles en el listado. Se puede ordenar tocando el nombre de la columna en cada una de las columnas.

Título ↑	EAN	Unidades	Precio	Talla	Categoría	Acciones
Adidas FWD-02 Sport Auriculares	7340055375429	2	169€	Única	Electrónica	
Belkin cargador de pared BoostCharge de 25 W	745883825066	11	12.99€	Única	Electrónica	
Beurer mosquitos	4211125603182	1	200€	Única	Salud y cuidado personal	
Ejemplo	1234567888888	0	0€	Única	Alimentación básica	
Irigador oral sin cable de agua y aire Philips	8710103991434	4	105.45€	Única	Salud y cuidado personal	
JBL TUNE 230NC TWS	6925281993176	4	47.54€	Única	Electrónica	
Jabra BlueParrot S650-XT	5707055054945	1	224.99€	Única	Electrónica	
Mentos pack 6	8003440979600	2	7.99€	Única	Alimentación y bebidas	
Pantalones chandal Mujer	1234567890123	9	19€	M	Ropa y accesorios	
Ram Kingston FURY	740617326147	2	299.99€	Única	Electrónica	

Files por página: 10 | 1 - 10 de 14

Figura 37 Listado productos.

En cada una de las líneas del producto hay un icono para ver los detalles de cada producto y poder editarlos o para borrar dicho producto y todos los movimientos asociados (se pide confirmación para el borrado y solo es accesible para cuentas autorizadas).

Menú

Inicio

Productos

Movimientos

Usuarios

Categorías

Edición Stock

Datos producto

EAN
5038061139525

Título
Remington Afetadora Rotativa Virtualmente Indestructible

Descripción
Remington Afetadora Rotativa Virtualmente Indestructible - Armazón de Policarbonato Prácticamente Indestructible, Resistente al Agua, Cabezal Pivotante, Autonomía 60 min, Gris y Amarillo - PR1855

Unidades
16

Precio
49.99

Talla
Única

Categoría
Electrónica

IVA Categoría
21%

Cerrar edición

Edición producto

Título: Título *
Remington Afetadora Rotativa Virtualmente Indestructible
Máximo 100 caracteres

Descripción: Descripción *
Remington Afetadora Rotativa Virtualmente Indestructible - Armazón de Policarbonato Prácticamente Indestructible, Resistente al Agua, Cabezal Pivotante, Autonomía 60 min, Gris y Amarillo - PR1855
Máximo 1000 caracteres

Unidades: Unidades *
16
Número de unidades del producto.

Precio: Precio *
49.99
Precio del producto en euros.Introducir con formato XXX

Registrar cambios

Figura 38 Pantalla edición producto.

Al lado de la barra de búsqueda del listado se tiene la opción de crear un nuevo producto y de descargar, al descargar se pasa a pdf todos los elementos visibles en el listado en dicho momento y en el orden en el que los tenga clasificados el usuario.

Menú

Inicio

Productos

Movimientos

Usuarios

Categorías

Edición Stock

Registro Producto

EAN: EAN *
Identificador único del producto. 13 caracteres

Título: Título *
Máximo 100 caracteres

Descripción: Descripción *
Máximo 1000 caracteres

Talla: Talla *
Única
Máximo 20 caracteres, tala única por defecto.

Unidades: Unidades *
0
Número de unidades del producto.

Precio: Precio *
0
Precio del producto en euros.Introducir con formato XXX

Categoría: Categoría del producto *

Crear nuevo producto

Figura 39 Pantalla crear nuevo producto

10.5 MOVIMIENTOS

Nada más se entra se carga un listado interactivo con todos los movimientos asociados a los productos de una empresa, esta pantalla solo es accesible por cuentas autorizadas del sistema.

El listado se comporta igual que el de productos, con búsqueda, ordenación y descarga. Tenemos también la opción de ver detalles de cada movimiento.

Menú Ver Perfil Cerrar sesión

Listado completo movimientos

Buscar por datos del movimiento Info Descargar

Fecha	DNI	Acción	Nombre Producto	EAN Producto	Acciones
2022-05-16 19:38:58	1N	Creación producto	Pantalones chandal Mujer	1234567890123	
2022-05-18 19:05:31	1N	Editar producto	Pantalones chandal Mujer	1234567890123	
2022-06-07 00:44:33	1N	Quitar unidades	Pantalones chandal Mujer	1234567890123	
2022-05-18 18:57:51	1N	Creación producto	Ropa estandar	1231231231231	
2022-06-08 20:54:22	1N	Añade unidades	Ropa estandar	1231231231231	
2022-05-24 18:19:01	1N	Creación producto	Ram Kingston FURY	740617326147	
2022-05-24 19:44:39	1N	Editar producto	Ram Kingston FURY	740617326147	
2022-05-24 19:44:39	1N	Añade unidades	Ram Kingston FURY	740617326147	
2022-06-10 09:03:23	1N	Editar producto	Ram Kingston FURY	740617326147	
2022-05-24 18:20:14	1N	Creación producto	Beurer mosquitos	4211125603182	

Filas por página: 10 1 - 10 de 29

Figura 40 Listado movimientos

La pantalla de detalles de movimientos es diferente dependiendo del tipo, si es de creación se muestran los datos de creación del producto y los de carácter general, si es de edición o cambio de unidades se muestran los detalles después de dicho movimiento y antes del mismo.

Menú Ver Perfil Cerrar sesión

Detalle Movimiento

Fecha
2022-05-24 19:44:39

Dni usuario
1N

EAN Producto
740617326147

Tipo Movimiento
Editar producto

Datos antiguos del producto

Título
Ram kingston

Descripcion
32gb ddr5

Precio
300

Unidades
1

Datos nuevos del producto

Título
Ram kingston

Descripcion
Kingston FURY Impact 32GB (2x16GB) 4800MT/s DDR5 CL38 SODIMM Memoria Gamer para Portátil Kit de 2 - KF548S38IBK2-32

Precio
299.99

Unidades
2

Figura 41 Pantalla detalle movimiento

10.6 USUARIOS

Nada más entrar muestra un listado similar a las pantallas anteriores con las mismas opciones, el número de columnas a mostrar depende si la cuenta que accede es autorizada o no, columnas de datos personales como el dni solo están disponibles a usuarios autorizados. Asimismo, solo un

usuario autorizado tiene la posibilidad de crear un nuevo usuario en dicha empresa, ver los detalles de un usuario en concreto o borrar dicho usuario y todos sus movimientos asociados.

Dni	Nombre	Fecha Incorporación	Cargo	Autorizado	Acciones
1N	nombreuser1	2015-01-01	jefe	<input checked="" type="checkbox"/>	✎ ✕
71181111N	Mario Rodríguez	2022-05-22	Tester	<input checked="" type="checkbox"/>	✎ ✕
71181128S	Mario no auto	2022-06-11	No autorizado	<input type="checkbox"/>	✎ ✕

Figura 42 Listado usuarios

Registro Usuario

DNI: DNI *
Longitud 9 caracteres

Nombre completo: Nombre *
Mínimo 5 caracteres

Cargo: Cargo *
Mínimo 3 caracteres

Contraseña: Contraseña *
Mínimo 5 caracteres

Autorizado: Permisos usuario * ▾

Registrar

Figura 43 Registro nuevo usuario

La pantalla de detalles del usuario muestra los movimientos del usuario de forma genérica, los detalles de dicho usuario y su contraseña al dar click, además de dar la posibilidad de cambiar dicha contraseña si el usuario la olvida.

- ☰
- 🏠
- 📄
- 📊
- ⚙️
- +

Datos Usuario

Resumen datos perfil

Nombre completo
nombreuser1

Dni
1N

Fecha Incorporación
2015-01-01

Cargo en la empresa
jefe

Permisos usuario
Cuenta con permisos

Número movimientos
29
[Se puede ver el detalle completo de los movimientos de un usuario buscando aquí](#)

Cambiar contraseña usuario

Nueva contraseña
password

Aceptar

Mínimo 5 caracteres. Se muestra la contraseña actual al dar click.

Movimientos Usuario

Fecha	Acción	Nombre Producto	EAN Producto
2022-05-16 19:38:58	Creación producto	Pantalones chandal Mujer	1234567890123
2022-05-18 18:57:51	Creación producto	Ropa estandar	1231231231231
2022-05-18 19:05:31	Editar producto	Pantalones chandal Mujer	1234567890123
2022-05-24 18:19:01	Creación producto	Ram Kingston FURY	740617326147

Figura 44 Detalle usuario

10.7 CATEGORÍAS

Se muestra un listado interactivo como los anteriores, pero con las categorías registradas en el sistema en las que se pueden filtrar los productos.

☰ Menú
Ver Perfil
Cerrar sesión

🏠 Listado completo categorías

🔍 Buscar por datos de la categoría Info ⓘ Descargar ⬇

IVA	Nombre	Descripción productos
10%	Alimentación y bebidas	Bebidas, aperitivos y postres, condimentos, productos de consumo para el hogar, comida para bebés
21%	Ropa y accesorios	Complementos, chaquetas, sujetadores, vestidos, sombreros, ropa de abrigo, pantalones, camisas, pantalones cortos, faldas, pijamas, calcetines, medias, trajes, jerséis, bañadores, ropa interior
10%	Libros, Música, Video y DVD	Música, libros, video, DVD, Blu-ray
4%	Periódicos y revistas	
21%	Consolas y videojuegos	
21%	Informática	Ordenadores de sobremesa, ordenadores portátiles, discos, almacenamiento
21%	Electrónica	Televisores, reproductores de CD, equipos de sonido para vehículos, GPS, teléfonos, cámaras, Fundas, cables, cargadores, soportes (o soportes de pared), auriculares
21%	Hogar y cocina	
21%	Electrodomésticos	Neveras, frigorífico, lavavajillas, lavadoras, microondas
21%	Software y videojuegos de PC	Software, video, videojuegos de PC

Filas por página: 10 1 - 10 de 26 < >

Figura 45 Listado categorías

10.8 MODIFICACIÓN RÁPIDA

Esta pantalla tiene compatibilidad con los escáneres de código de barras más habituales, aunque se puede usar a mano en caso de no tener uno disponible sin problemas. Al escanear un producto el sistema carga la información del mismo y da la posibilidad de editar de forma rápida el número de unidades de dicho producto, tanto para añadir como para quitar unidades del producto en el sistema.

The screenshot shows a web interface for 'Modificación rápida'. At the top left is a 'Menú' button. At the top right are 'Ver Perfil' and 'Cerrar sesión' buttons. The main content area is divided into three sections:

- Modificación rápida:** A section with a sub-header and a description: 'Nos permite cambiar el número de unidades de forma rápida introduciendo los datos del producto. Solo se muestra información si el producto está ya registrado previamente en el sistema.' Below this is an input field for 'EAN Producto' with a search icon.
- Cambio unidades:** A section with two circular arrows (down and up) and an input field for 'Diferencia unidades' containing the number '1'. A green checkmark is visible to the right.
- Busqueda EAN:** A section displaying product details for EAN '740617326147':
 - Título:** Ram Kingston FURY
 - Descripción:** Kingston FURY Impact 32GB (2x16GB) 4800MT/s DDR5 CL38 SODIMM Memoria Gamer para Portátil Kit de 2 - KF548S38IBK2-32
 - Unidades:** 2
 - Precio:** 299.99
 - Precio:** 299.99

Figura 46 Pantalla modificación rápida

10.9 FEEDBACK PANTALLAS

Todas las pantallas en las que se realiza una acción que requiere una modificación sobre la base de datos tienen un *popup* que muestra el estado de dicha operación, tanto si hay algún error como si se ha realizado correctamente.

The screenshot shows a modal error message. The popup is a white box with a red 'X' icon and the text 'Datos introducidos invalidos'. The background is a blurred view of a form with the following fields:

- EAN:** Input field containing 'dasd'. Below it, the text 'Identificador unico del producto. 13 e' is visible.
- Título:** Input field containing 'sadsad'. Below it, the text 'Máximo 100 caracteres' is visible.
- Descripción:** Input field containing 'sadasd'.

Figura 47 Ejemplo mensaje error

<div style="text-align: center;">  Producto registrado </div>					
EAN	Unidades	Precio	Talla	Categoría	
1234567890123	9	19€	M	Ropa y accesorios	
1231231231231	4	2.33€	Única	Ropa y accesorios	
740617326147	2	299.99€	Única	Electrónica	

Figura 48 Ejemplo mensaje registro

Estas ventanas solo duran un par de segundos activas y se cierran automáticamente. En la pantalla movimientos también tenemos un apartado informativo que utiliza este tipo de ventanas, pero con un cierre manual.

DNI	Acción	<div style="text-align: center;">  Información movimientos </div>		EAN Producto	
IN	Creación producto	<p>Se muestran todos los movimientos asociados a la empresa, cada vez que se produce una acción sobre un producto se registra el tipo, el usuario y la fecha en la que se ha realizado dicha acción. Se puede buscar y ordenar por columnas.</p> <div style="text-align: center;"> <input type="button" value="Entendido"/> </div>		1234567890123	
IN	Editar producto			1234567890123	
IN	Quitar unidades			1234567890123	
IN	Creación producto			1231231231231	
IN	Añade unidades			1231231231231	
IN	Creación producto			740617326147	
IN	Editar producto			740617326147	
IN	Añade unidades			Ram Kingston FURY	740617326147

Figura 49 Ejemplo mensaje informativo

La única pantalla que no tiene dicho *popup* cuando se realiza un cambio en la base de datos es el registro de empresa, que cuenta con un pequeño mensaje en la parte superior confirmando el registro o el error.

Registro Empresa

Registro correcto, puede [iniciar sesión aquí](#)

Nombre Empresa :

Empresa a borrar

CIF Empresa :

Q2826010H

Nombre Usuario :

Mario ejemplo

Figura 50 Mensaje registro correcto

Empresa ya registrada en el sistema

Nombre Empresa :
Empresa a borrar

CIF Empresa :
Q2826010H

Nombre Usuario :
Mario ejemplo

Figura 51 Mensaje error en registro

11 BIBLIOGRAFÍA

- [«Sistema de planificación de recursos empresariales,» [En línea]. Available:
1] https://es.wikipedia.org/wiki/Sistema_de_planificaci%C3%B3n_de_recursos_empresariales. [Último acceso: 2022 03 01].
- [«Angular,» [En línea]. Available: <https://angular.io/>.
2]
- [«Spring Boot,» [En línea]. Available: <https://spring.io/projects/spring-boot>.
3]
- [«¿Qué es una API REST?,» [En línea]. Available:
4] <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.
- [«controldestock.com,» [En línea]. Available: <https://www.controldestock.com/es/>.
5]
- [«Ebay tarifas,» [En línea]. Available: <https://www.ebay.es/help/selling/fees-credits-invoices/servicios-de-pago-comisiones-y-tarifas-para-vendedores-particulares?id=4822>.
6]
- [«Dynamics 365,» [En línea]. Available: <https://dynamics.microsoft.com/en-us/pricing/#CustomerDataPlatform>.
7]
- [«Oracle Netsuite,» [En línea]. Available: <https://www.netsuite.com/portal/es/home.shtml>.
8]
- [«Computación en la nube,» [En línea]. Available:
9] https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube.
- [«IaaS, SaaS, PaaS,» [En línea]. Available: <https://nanobytes.es/blog/blog-nanobytes-1/cloud-computing-diferencia-entre-iaas-paas-y-saas-7>.
10]
- [«Hosting DigitalOcean,» [En línea]. Available: <https://www.digitalocean.com/>.
11]
- [«Azure diccionario,» [En línea]. Available: <https://azure.microsoft.com/es-es/overview/cloud-computing-dictionary/>.
12]
- [«Datos Angular,» [En línea]. Available:
13] [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework)).
]

[«¿Qué es Angular y para qué sirve?,» [En línea]. Available:
14 <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>.
]

[«Spring framework vs boot,» [En línea]. Available:
15 https://www.ibm.com/cloud/learn/java-spring-boot#toc-spring-boo-HII6V_zV.
]

[«Entity JPA,» [En línea]. Available: <https://www.baeldung.com/jpa-entities>.
16
]

[«Repositorio Spring Boot.,» [En línea]. Available:
17 <https://zetcode.com/springboot/repository/#:~:text=A%20repository%20is%20a%20mechanism,be%20autodetected%20through%20classpath%20scanning..>
]

[«Servicio Spring Boot,» [En línea]. Available:
18 https://www.tutorialspoint.com/spring_boot/spring_boot_service_components.htm#:~:text=Advertisements,separated%20from%20%40RestController%20class%20file..
]

[«What is a rest api?,» [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.
19
]

[«What is REST?,» [En línea]. Available: <https://www.codecademy.com/article/what-is-rest>.
20
]

[«Servicios Web: RESTful,» [En línea]. Available: <https://blog.bi-geek.com/servicios-web-restful/>.
21
]

[«MySQL,» [En línea]. Available: <https://es.wikipedia.org/wiki/MySQL>.
22
]

[«Control de versiones,» [En línea]. Available:
23 https://es.wikipedia.org/wiki/Control_de_versiones.
]

[«What is git?,» [En línea]. Available: <https://www.atlassian.com/es/git/tutorials/what-is-git>.
24
]

[«TypeScript,» [En línea]. Available: <https://www.typescriptlang.org/>.
25
]

[«Java,» [En línea]. Available:
26 [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)).
]

[«The Most Popular Programming Languages,» 2019. [En línea]. Available:
27 <https://www.statista.com/chart/16567/popular-programming-languages/>.
]

[«GitLab,» [En línea]. Available: <https://en.wikipedia.org/wiki/GitLab>.
28
]

[«Astah,» [En línea]. Available: https://en.wikipedia.org/wiki/Astah*.
29
]

[«Astah,» [En línea]. Available: <https://astah.net/>.
30
]

- [«MySQL Workbench,» [En línea]. Available:
31 https://es.wikipedia.org/wiki/MySQL_Workbench.
]
- [«MySQL Products: WorkBench,» [En línea]. Available:
32 <https://www.mysql.com/products/workbench/>.
]
- [«Visual Studio Code,» [En línea]. Available:
33 https://es.wikipedia.org/wiki/Visual_Studio_Code.
]
- [«Postman Chrome Web Store,» [En línea]. Available:
34 <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=es>.
]
- [«Amazon Web Services,» [En línea]. Available:
35 https://es.wikipedia.org/wiki/Amazon_Web_Services.
]
- [«¿Qué es Scrum y cómo funciona?,» [En línea]. Available:
36 <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>.
]
- [«¿Qué es Scrum?,» [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>.
37
]
- [«Metodología Scrum,» [En línea]. Available:
38 <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>.
]
- [«Qué son metodologías ágiles,» [En línea]. Available:
39 <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>.
]
- [«Accessing data with MySQL,» [En línea]. Available:
40 <https://spring.io/guides/gs/accessing-data-mysql/>.
]
- [«Northware,» [En línea]. Available: <https://www.northware.mx/blog/7-errores-comunes-en-proyectos-de-desarrollo-de-software/>.
41
]
- [«Gestión de los riesgos de un proyecto,» [En línea]. Available:
42 <https://www.gladysgbegnedji.com/gestion-de-los-riesgos-del-proyecto/>.
]
- [«GESTIÓN DE LOS RIESGOS DEL PROYECTO,» [En línea]. Available:
43 <https://www.gladysgbegnedji.com/gestion-de-los-riesgos-del-proyecto/>.
]
- [«Desarrolla tu mapa de gestión de riesgos en 5 pasos,» [En línea]. Available:
44 <https://blog.hubspot.es/sales/mapa-riesgos-empresa>.
]
- [E. M. Félix Prieto, «Requisitos - FIA,» 2016, p. 45.
45
]
- [«Historia de usuario,» [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories#:~:text=usuario%20del%20software,-Una%20historia%20de%20usuario%20es%20una%20explicaci%C3%B3n%20general%20e%20informal,un%20valor%20particular%20al%20cliente..>
46
]

[«Caso de Uso,» [En línea]. Available: https://es.wikipedia.org/wiki/Caso_de_uso.
47]

[«DAO,» [En línea]. Available:
48 <http://www.cursoshibernate.es/doku.php?id=patrones:dao#:~:text=EI%20patr%C3%B3n%20Data%20Access%20Object,en%20las%20clases%20llamadas%20DAO..>.
] 0Data%20Access%20Object,en%20las%20clases%20llamadas%20DAO..

[«Entity To DTO Conversion for a Spring REST API,» [En línea]. Available:
49 <https://www.baeldung.com/entity-to-and-from-dto-for-a-java-spring-application>.
]

[«Factory method,» [En línea]. Available: [https://refactoring.guru/es/design-](https://refactoring.guru/es/design-patterns/factory-method)
50 [patterns/factory-method](https://refactoring.guru/es/design-patterns/factory-method).
]

[«<https://developer.mozilla.org/es/docs/Glossary/MVC>,» [En línea]. Available: MVC -
51 Glosario.
]

[«MVC,» [En línea]. Available: [https://codigofacilito.com/articulos/mvc-model-view-](https://codigofacilito.com/articulos/mvc-model-view-controller-explicado)
52 [controller-explicado](https://codigofacilito.com/articulos/mvc-model-view-controller-explicado).
]

[«Autenticación de acceso básica,» [En línea]. Available:
53 https://es.wikipedia.org/wiki/Autenticaci%C3%B3n_de_acceso_b%C3%A1sica.
]

[«¿Qué es GitLab?,» [En línea]. Available: <https://geekflare.com/es/gitlab-hosting/>.
54]

[«Github Education,» [En línea]. Available: <https://education.github.com/pack>.
55]

[«Descubrir la interfaz de Word,» [En línea]. Available: [https://www.ediciones-](https://www.ediciones-eni.com/open/mediabook.aspx?idR=1f30bc8f3ff51b8affc43d7a5fb53629)
56 [eni.com/open/mediabook.aspx?idR=1f30bc8f3ff51b8affc43d7a5fb53629](https://www.ediciones-eni.com/open/mediabook.aspx?idR=1f30bc8f3ff51b8affc43d7a5fb53629).
]

[«Spring Tools 4,» [En línea]. Available: <https://spring.io/tools>.
57]

[«Arquitectura en Angular,» [En línea]. Available:
58 [https://medium.com/notasdeangular/arquitectura-de-nuestras-aplicaciones-en-angular-](https://medium.com/notasdeangular/arquitectura-de-nuestras-aplicaciones-en-angular-84df61691b57)
] [84df61691b57](https://medium.com/notasdeangular/arquitectura-de-nuestras-aplicaciones-en-angular-84df61691b57).

[«Cloud Computing: diferencia entre IaaS, PaaS y SaaS,» [En línea]. Available:
59 [https://nanobytes.es/blog/blog-nanobytes-1/cloud-computing-diferencia-entre-iaas-paas-y-](https://nanobytes.es/blog/blog-nanobytes-1/cloud-computing-diferencia-entre-iaas-paas-y-saas-7)
] [saas-7](https://nanobytes.es/blog/blog-nanobytes-1/cloud-computing-diferencia-entre-iaas-paas-y-saas-7).