

GESTIÓN DEL ALMACENADO PARA ECOSISTEMAS EMPRESARIALES

Trabajo de Fin de Grado
Grado en Ingeniería Informática

MEMORIA



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Enero de 2023

Autor

Daniel González Martín

Tutores

Juan Manuel Corchado Rodríguez

Raúl López Blanco

Guillermo Hernández González

Dr. Juan Manuel Corchado Rodríguez, D. Raúl López Blanco y Dr. Guillermo Hernández González, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICAN:

Que el trabajo titulado “Gestión del almacenado para ecosistemas empresariales” ha sido realizado por D. Daniel González Martín con DNI 70908147K y constituye la memoria del trabajo realizado por la superación de la asignatura Trabajo de Fin de Grado de la titulación Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 12 de enero de 2023.

Dr. Juan Manuel Corchado
Rodríguez

Dpto. Informática y
automática

Universidad de Salamanca

D. Raúl López Blanco

Dpto. Informática y
automática.

Universidad de Salamanca

Dr. Guillermo Hernández
González

Dpto. Informática y
automática.

Universidad de Salamanca

Resumen

En este trabajo de Fin de Grado se encuentra el desarrollo de una aplicación que permite realizar un control de inventario de un almacén en entornos empresariales.

Dada la gran importancia que tiene el sector hostelero en la industria mundial y en particular en España, realizar una correcta administración de los recursos disponibles supondrá una actividad positiva del negocio que permitirá evitar pérdidas e impulsará las ganancias de este.

Este trabajo se centra en la creación de una aplicación que permite realizar un seguimiento y llevar un control sobre la cantidad de inventario enfocando al ecosistema empresarial en el sector hostelero.

Esta aplicación se ha desarrollado para que las empresas puedan llevar un correcto inventario y facilitarles la realización de la búsqueda de recursos disponibles, así como ver cuál ha sido la evolución de estos a lo largo del tiempo.

El sistema dispone de una página web a la que los trabajadores de la empresa pueden acceder para ver, almacenar o eliminar los recursos de la empresa, además de consultar el estado e histórico de estos.

La implementación se ha realizado utilizando un entorno de ejecución orientado a eventos asíncronos Node.js, que permite controlar las solicitudes realizadas por los usuarios. Este servidor está alojado en una instancia en la nube de Amazon Web Services, a la que se redirige todo el tráfico entrante gracias a un sistema de nombre de dominios (DNS), donde está registrada la IP de la máquina donde está desplegado el sistema.

Summary

In this Final Degree work is the development of an application that allows inventory control of a warehouse in business environments.

Given the great importance of the hospitality sector in the global industry and particularly in Spain, a proper management of available resources will mean a positive business activity that will prevent losses and boost profits.

This work will focus on creating an application that allows to track and keep track of the amount of inventory focusing on the business ecosystem in the hospitality sector.

This application has been developed with the objective that companies can keep a correct inventory and make it easier for them to search for available resources, as well as to see how they have evolved over time.

The system has a web page that company employees can access to view, store or delete the company's resources, as well as view their status and history.

The implementation has been done using an asynchronous event-driven execution environment Node.js, which allows to control the requests made by the users. This server is hosted on a cloud instance of Amazon Web Services, to which all incoming traffic is redirected thanks to a domain name system (DNS), where the IP of the machine where the system is deployed is registered.

Índice de contenido

1.	Introducción	1
1.1.	Estructura de la memoria	2
1.2.	Descripción de la documentación técnica	3
2.	Objetivos	5
2.1.	Objetivos del sistema	5
2.2.	Objetivos personales.....	6
3.	Conceptos teóricos.....	7
3.1.	Frontend.....	7
3.2.	Framework	7
3.3.	Backend	7
3.4.	Amazon web Services (AWS).....	8
3.5.	Servidor DNS	9
3.6.	Servidor web	10
3.7.	Certificado SSL.....	11
4.	Herramientas y técnicas utilizadas.....	13
4.1.	Cloud Computing.....	13
4.1.1.	Amazon Web Services	13
4.2.	Herramientas de control de versiones.....	13
4.2.1.	Git y GitHub.....	13
4.3.	Herramientas CASE	13
4.3.1.	REM	13
4.3.2.	Visual Paradigm.....	14
4.3.3.	Microsoft Projects	14
4.3.4.	EZEstimate.....	14
4.4.	Entornos de desarrollo	15
4.4.1.	Visual Studio Code.....	15
4.5.	Herramientas utilizadas en el servidor.....	15
4.5.1.	Node.js.....	15
4.5.2.	NPM.....	15
4.5.3.	Express.....	15
4.5.4.	Mongoose.....	15
4.5.5.	Mongodb	16
4.5.6.	Override.....	16
4.5.7.	CSS	16

4.5.8.	Nodemailer	16
4.5.9.	Generate password	16
4.5.10.	Forever	16
4.6.	Lenguajes de programación	17
4.6.1.	HTML	17
4.6.2.	JSON	17
4.6.3.	UML	17
5.	Aspectos relevantes del desarrollo	19
5.1.	Marco de trabajo.....	19
5.2.	Estimación de coste y esfuerzo	19
5.2.1.	UUCW (Unadjusted Use Case Weight)	20
5.2.2.	UAW (Unadjusted Actor Weight)	21
5.2.3.	TCF (Technical Complexity Factor)	21
5.2.4.	ECF (Environment Complexity Factor)	22
5.3.	Planificación temporal.....	24
5.3.1.	Fase Inicio	25
5.3.2.	Fase Elaboración.....	26
5.3.3.	Fase Construcción	27
5.3.4.	Fase Transición	28
5.4.	Especificación de requisitos	29
5.4.1.	Actores	29
5.4.2.	Objetivos del sistema	30
5.4.3.	Requisitos de información.....	31
5.4.4.	Requisitos no funcionales	32
5.4.5.	Requisitos funcionales.....	33
5.5.	Análisis del sistema	36
5.5.1.	Modelo de dominio	36
5.5.2.	Paquete de análisis.....	37
5.5.3.	Realización de casos de Uso	38
5.6.	Diseño del sistema	39
5.6.1.	Modelo de diseño.....	39
5.6.2.	Modelo de despliegue.....	43
5.7.	Implementación	44
5.8.	Pruebas.....	45
5.9.	Descripción de la aplicación	46
5.9.1.	Gestión de usuarios.....	46

5.9.2.	Gestión de inventario.....	51
5.9.3.	Gestión de productos.....	54
5.9.4.	Gestión de alertas	58
6.	Conclusiones.....	61
7.	Líneas de trabajo futuras.....	63
8.	Referencias.....	65

Índice de Tablas

Tabla 1. Factores de complejidad técnica.....	21
Tabla 2. Factores de complejidad del entorno.....	22
Tabla 3. Actores.....	29
Tabla 4. Objetivos del sistema.	30
Tabla 5. Requisitos de información.....	31
Tabla 6. Requisitos no funcionales.....	32
Tabla 7: Solicitar registro.....	34
Tabla 8: Casos de uso.	35

Índice de figuras

Figura 1: Instancia EC2.	8
Figura 2: Servidor DNS.....	9
Figura 3: sites-available.....	10
Figura 4: configuración nginx.	10
Figura 5: Certificado SSL.....	11
Figura 6: Letsencrypt directory.	11
Figura 7: Configuración nginx.....	12
Figura 8: Visual paradigm.....	14
Figura 9: Fase Inicio.....	25
Figura 10: Fase Elaboración.....	26
Figura 11: Fase Construcción.	27
Figura 12: Fase Transición.....	28
Figura 13: Diagrama de paquetes.	33
Figura 14: Diagrama de casos de uso.....	33
Figura 15: Modelo de dominio.....	36
Figura 16: Paquete de análisis.....	37
Figura 17: Solicitar registro.	38
Figura 18: MVC.....	40
Figura 19: Diagrama de paquetes.	41
Figura 20: Diagrama de secuencia.	42
Figura 21: Modelo de despliegue.....	43
Figura 22: Solicitar registro.	46
Figura 23: Olvidar contraseña.	47
Figura 24: Iniciar sesión.....	47
Figura 25: Inicio.....	47
Figura 26: Personal.....	48
Figura 27: Peticiones.....	48
Figura 28: Perfil.....	49
Figura 29: Cambiar contraseña.	49
Figura 30: Eliminar cuenta.	50
Figura 31: Inventario.....	51
Figura 32: Añadir inventario.....	51
Figura 33: Modificar inventario.....	52
Figura 34: Eliminar inventario.....	52
Figura 35: Filtrar.....	53
Figura 36: Código QR.....	53
Figura 37: Inventario filtrado.	53
Figura 38: Productos.....	54
Figura 39: Añadir productos.....	54
Figura 40: Modificar producto.....	55
Figura 41: Eliminar producto.....	55
Figura 42: Alerta eliminar producto.....	55
Figura 43: Gráficas producto.....	56
Figura 44: Evolución temporal.....	56
Figura 45: Estadísticas.....	57

Figura 46: Alertas. 58

Figura 47: Activar alertas automáticas..... 58

Figura 48: Desactivar alertas automáticas. 59

1. Introducción

El sector hostelero representa uno de los sectores más importantes de la industria mundial. En particular, en España, un país donde dicho sector representa a nivel estatal más de 315000 empresas y representa un 6.4% de del PIB del país, con un volumen de ventas de 130841 millones de euros, es necesario realizar esfuerzos y aplicaciones que faciliten el trabajo de las empresas de dicho sector en minimizar costes y tiempo e impulsar las ganancias de estas [1].

Un correcto inventariado es importante para mantener una buena administración de existencias dentro de los almacenes de una empresa. Un inventario insuficiente puede suponer una pérdida de ventas por la imposibilidad de cubrir la demanda, lo que puede ser una consecuencia de desconocer en qué productos se hacen mayores o menores movimientos. Por otro lado, un exceso de inventario puede desencadenar también pérdidas, debido a los costes de almacenamiento o incluso a la saturación de la capacidad de almacenaje. La clave para un control de inventario adecuado es obtener una comprensión sobre la demanda de los productos.

Existen ventajas de usar un sistema de gestión del inventariado de los productos de una empresa:

- Conocimiento claro de qué se puede ofrecer y qué no se puede ofrecer a los clientes. Esto ofrecerá una mayor credibilidad y será parte fundamental para ofrecer un buen servicio.
- Facilidad para cumplir los plazos de entrega, ya que se reduce la probabilidad de aparición de sorpresas e imprevistos.
- Conoceremos en todo momento la cantidad exacta del inventario del que se dispone, minimizando el riesgo de extravío de productos.
- En caso de robo, un inventario controlado puede facilitar el proceso de denuncia.
- Permite descubrir mermas en algún producto en el establecimiento.

Es de vital importancia para la supervivencia de una empresa utilizar herramientas que faciliten realizar un correcto inventariado y que automaticen funciones que permitan minimizar errores y generar informaciones con valor.

Por ello, este Trabajo de Fin de Grado está orientado a realizar una aplicación que nos permita realizar un correcto inventario dentro de este entorno empresarial, la hostelería. Como caso de uso para el diseño *software* se ha considerado un establecimiento donde tendremos una Sala Principal y un Almacén, donde almacenaremos inventario y querremos realizar el inventariado y la gestión de personal de este; si bien se ha procurado realizar un diseño generalizable que permite extender el sistema para otros escenarios.

1.1. Estructura de la memoria

A continuación, se detalla la estructura que se sigue a lo largo del documento con las diferentes fases de desarrollo de este.

- **Introducción:** Se realiza una breve presentación del proyecto y se hace enfoque en la importancia de realizar un buen inventario de los productos que generan producto en un ecosistema empresarial.
- **Objetivos:** En este apartado se reflejan los objetivos buscados a la hora de realizar este Trabajo de Fin de Grado.
- **Conceptos teóricos:** Se presenta una fundamentación básica para la comprensión del trabajo a lo largo del resto de la memoria y de los anexos.
- **Técnicas y herramientas:** Se realiza una breve descripción de las técnicas con las metodologías y herramientas de trabajo que se han seguido y utilizado a lo largo del proyecto para su creación y desarrollo.
- **Aspectos relevantes del desarrollo:** En este apartado se tratan los aspectos más destacables del desarrollo del proyecto a lo largo de las diferentes fases de desarrollo.
- **Conclusiones:** En este apartado se exponen unas conclusiones sobre cómo ha sido el desarrollo del proyecto, así como conclusiones generales personales de cómo se ha llevado la creación de este.
- **Referencias bibliográficas:** Se presentan las fuentes utilizadas para la creación de la presente memoria.

1.2. Descripción de la documentación técnica

Para la ampliación de la información y facilitar la profundización del desarrollo de este proyecto, se detallan todas las actividades realizadas a lo largo de las fases de creación en diferentes anexos, que se presentan de la siguiente manera:

- **Anexo I - Planificación temporal:** Se realiza una planificación temporal del desarrollo del proyecto siguiendo el marco de trabajo de Proceso Unificado.
- **Anexo II - Especificación de Requisitos:** Se realiza la identificación de objetivos y requisitos del sistema.
- **Anexo III - Análisis del sistema:** Se realiza el desarrollo de los casos de uso, diagramas de secuencia, modelo de dominio y paquetes de análisis del sistema a partir de los requisitos obtenidos en el anexo previo.
- **Anexo IV - Diseño del sistema:** Se realiza un refinado del dominio del problema, llegando al dominio de solución, realizando un diseño de arquitectura del sistema como el diseño de interfaz y pruebas.
- **Anexo V - Documentación técnica:** En este anexo se documenta la implementación del código con el objetivo de explicar y facilitar las funcionalidades del código para otros programadores.
- **Anexo VI - Manual de usuario:** Se realiza un documento que permita al usuario comprender las funcionalidades de la aplicación de una manera clara.

2. Objetivos

En este apartado se detallan los objetivos que se deben de cumplir y desarrollar en el sistema para su validación. Además, serán expuestos los objetivos personales que han sido definidos para este proyecto.

2.1. Objetivos del sistema

Los objetivos principales planteados como obligatorios para el desarrollo de este proyecto de Fin de Grado son:

- **Gestión de usuarios:** El sistema será capaz de almacenar y gestionar datos de los usuarios que se registren dentro de la aplicación. Los usuarios tendrán la capacidad de registrarse y de iniciar sesión dentro del sistema. Se debe controlar el acceso de estos usuarios a las distintas funcionalidades y datos de la aplicación en función de los permisos que se le hayan asociado a sus respectivos roles dentro de la misma.
- **Gestión de inventario:** El sistema será capaz de almacenar y gestionar los inventarios introducidos por los usuarios poder mostrarles el inventario disponible de cada recurso que haya sido almacenado. Se deberá de darla posibilidad a los usuarios de realizar filtrado del inventario en caso de el usuario quiera enfocarse en buscar alguna característica del este en específico.
- **Gestión de productos:** El sistema será capaz de almacenar y gestionar los productos creados por los usuarios. Estos, deben tener asociado un tipo que habrá sido definido por los creadores del sistema y que permita realizar un ordenado de los productos añadidos en el sistema. Para cada producto se generan estadísticas y gráficas que dan valor a la hora de saber cómo ha sido la evolución del producto a lo largo del tiempo.
- **Gestión de alertas:** El sistema será capaz de enviar avisos a los usuarios cuando ocurran situaciones que les afecten. El sistema debe enviar una notificación una vez se haya aceptado o se haya rechazado su solicitud de entrada a una empresa. Se deben enviar notificaciones cuando el administrador haya decidido eliminar el usuario de la información de la empresa. Además, se generarán alertas cada vez que el inventario del producto esté en un valor superior o inferior a unos límites que serán establecidos por los usuarios del sistema. Estas situaciones de alerta pueden generar avisos a los administradores del sistema, según se haya especificado en las propiedades de los productos.
- **Usabilidad:** Se prima que el uso de la aplicación sea cómoda, fácil e intuitiva, para que los usuarios no tengan problemas a la hora de navegar por ella.

2.2. Objetivos personales

En cuanto a lo personal, en este proyecto espero adquirir gran cantidad de conocimientos relacionados con el desarrollo web que pueden ser de gran utilidad en la continuación del mundo laboral. Durante estos años de grado los estudiantes de ingeniería informática hemos aprendido sobre la creación de aplicaciones web, manejo de bases de datos, comunicaciones de aplicaciones, pero en ningún momento hemos llegado a combinar todos estos conocimientos. Realizar una práctica general en la que utilicemos todos estos conocimientos adquiridos a lo largo de los años más, seguro gran cantidad que iremos sumando a lo largo del desarrollo de todo el proyecto, supone una gran motivación de hacer un buen trabajo en el que demostrar todo lo aprendido.

Además, como se ha comentado en apartados anteriores, tener un control de inventario es de gran utilidad para las empresas ya que, cuanto más vayan creciendo estas, más cantidad de recursos tendrán que gestionar y más importante se volverá esta tarea de inventariado. Poder observar las evoluciones de manera gráfica, tener todos los recursos centralizados y poder tener un buen control de todos estos puede quitar grandes problemas de cabeza y generar la utilidad que buscamos crear en este proyecto. Estos problemas se han generado en entornos empresariales en los que personalmente he trabajado por lo que sirve de motivación para realizar un buen trabajo y que en caso de volver a aparecer, pudieran eliminarse de manera sencilla.

Por último, este trabajo supone terminar el grado en Ingeniería Informática y finalizar uno de los objetivos que lleva conmigo unos cuantos años. Permite seguir adquiriendo experiencias y conocimientos que puedan ponerse en práctica en la etapa laboral en la que estoy actualmente.

3. Conceptos teóricos

3.1. Frontend

Parte de la aplicación web que interactúa con los usuarios. Es conocida como el lado del cliente. Será lo que se vea en la pantalla cuando se acceda al sitio web en el que esté la aplicación. Se encarga de la experiencia del usuario. Las interfaces deben ser fáciles de usar, atractivas y funcionales.

3.2. Framework

Estructura previa que puede ser aprovechada para el desarrollo de un proyecto. En este proyecto es utilizado el *framework* para aplicaciones web **bootstrap**. Este *framework* combina CSS y Javascript y permite estilizar elementos de una página web. Esta herramienta proporciona interactividad en la página ya que ofrece una serie de componentes que facilitan la comunicación con el usuario. Estos elementos serán entre varios menús de navegación, controles de página o barras de progreso. Además, este framework asegura que el diseño funciona para distintos navegadores y que el código cumple ciertas normas estándar.

3.3. Backend

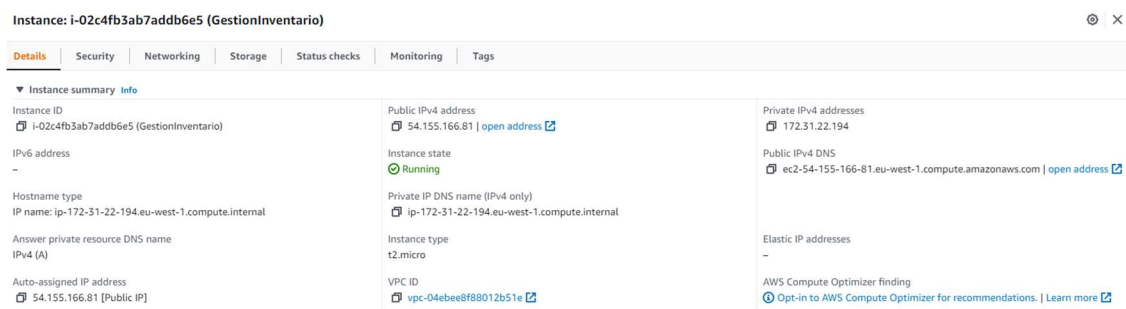
Parte de la aplicación en la que se toman datos, se procesan y se envía la información a los usuarios. Es conocida como el lado del servidor. Esta parte es la encargada de toda la lógica para que el sistema funciona. Consiste en un conjunto de acciones que pasan dentro de una web pero que no podremos ver.

3.4. Amazon web Services (AWS)

AWS [2] es una plataforma en la nube ofrecida por la empresa Amazon la cual ofrece servicios integrales para la computación y almacenamiento en la nube entre otros servicios de *cloud computing* que puedan ser utilizado por los usuarios para el desarrollo de servicios o aplicaciones. Los servicios en la nube permiten alojar el servidor en una página que pueda ser accesible para todo el mundo desde cualquier parte del mundo.

Para el desarrollo del trabajo se utiliza el servicio llamado *Elastic Compute Cloud (EC2)*, servicio que permite alquilar ordenadores virtuales donde aplicaciones. Estos ordenadores son llamados Instancias y cada una dispone de una dirección privada y pública a las que se puede acceder desde cualquier navegador.

Para la instancia EC2 de este proyecto han asignado lo siguientes valores representados en la Figura 1.



Instance: i-02c4fb3ab7addb6e5 (GestionInventario)		
Details	Security	Networking
▼ Instance summary info		
Instance ID i-02c4fb3ab7addb6e5 (GestionInventario)	Public IPv4 address 54.155.166.81 open address	Private IPv4 addresses 172.31.22.194
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-155-166-81.eu-west-1.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-22-194.eu-west-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-22-194.eu-west-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 54.155.166.81 [Public IP]	VPC ID vpc-04ebee8f88012b51e	

Figura 1: Instancia EC2.

Dirección pública: 54.155.166.81

Dirección privada: 172.31.22.194

3.5. Servidor DNS

Software que se encarga de traducir direcciones IP a nombre de domino. En este proyecto, se utiliza un servidor DNS gratuito llamado NO-IP [3] en el cual se ha registrado un nombre del *host* con el que se quiere que los usuarios de la aplicación accedan. A este nombre de *host* se ha asignado una dirección IP Objetivo a la cual son redirigidos a los usuarios de la aplicación del proyecto.

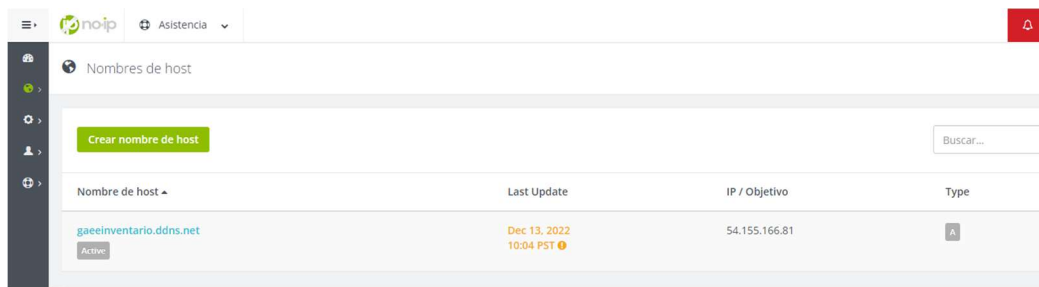


Figura 2: Servidor DNS.

Con esto, se garantiza que todos los usuarios que busquen en el navegador gaeeinventario.ddns.net [4] sean redirigidos a la máquina **54.155.166.81** donde se está ejecutándose la aplicación.

3.6. Servidor web

Para la aplicación se utiliza un servidor web llamado Nginx [5] que actúa como proxy y que permite redirigir tráfico entrante hacia un proceso o puerto que donde se está ejecutando la aplicación dentro del servidor. Los servidores web son los encargados de la entrega de las aplicaciones web. Responden a las peticiones HTTPS realizadas por los usuarios desde un navegador web.



```
ubuntu@ip-172-31-22-194: /etc/nginx/sites-available
GNU nano 6.2

server_name gaeinventario.ddns.net;
location / {
    proxy_pass http://localhost:3000;
}
```

Figura 3: sites-available.

En la práctica, todo el tráfico entrante para la dirección gaeinventario.ddns.net es redirigido y procesado por la aplicación que esté corriendo en el puerto 3000. En el puerto 3000 está ejecutándose el servidor principal de la aplicación el cual da respuesta a todas las peticiones entrantes por el nginx.

Dentro del fichero de configuración de nginx se configura que todo el tráfico entrante para la dirección gaeinventario.ddns.net es redirigido a su https. El código 301 antes de la dirección https es un código de estado que indica que el host ha sido capaz de comunicarse con el servidor, pero el recurso solicitado ha sido movido a otra dirección permanente.



```
server {
    if ($host = gaeinventario.ddns.net) {
        return 301 https://$host$request_uri;
    }

    listen 80 default_server;
    listen [::]:80 default_server;

    server_name gaeinventario.ddns.net;
    return 404;
}
```

Figura 4: configuración nginx.

3.7. Certificado SSL

Un certificado SSL es un protocolo de encriptación. Este certificado asegura que la forma de conexión entre cliente y servidor se realiza de manera segura y con un cifrado de datos. Todos los datos compartidos entre cliente y servidor son codificados. Para garantizar que un sitio web está asegurado con un certificado SSL es necesario comprobar que las conexiones realizadas se mantienen mediante el navegador web a la URL empezando con HTTPS.



Figura 5: Certificado SSL.

Para la creación del certificado SSL que es utilizado por el servidor, se hace uso de Let's Encrypt [6], una autoridad de certificación gratuita que proporciona certificados X.509 para el cifrado de Seguridad a Nivel de Transporte. Esta organización ha creado los certificados que son almacenados en el sistema y son expuestos por el servidor nginx al usuario cuando intente acceder al sistema.

Los certificados son almacenados de la manera mostrada en la Figura 6.

```
root@ip-172-31-22-194:/etc/letsencrypt/live/gaeinventario.ddns.net# cat README
This directory contains your keys and certificates.

`privkey.pem` : the private key for your certificate.
`fullchain.pem` : the certificate file used in most server software.
`chain.pem` : used for OCSP stapling in Nginx >=1.3.7.
`cert.pem` : will break many server configurations, and should not be used
              without reading further documentation (see link below).

WARNING: DO NOT MOVE OR RENAME THESE FILES!
Certbot expects these files to remain in this location in order
to function properly!

We recommend not moving these files. For more information, see the Certbot
User Guide at https://certbot.eff.org/docs/using.html#where-are-my-certificates.
```

Figura 6: Letsencrypt directory.

El nginx está configurado como en la Figura 7 para manera para poder presentar el certificado a los usuarios que accedan al dominio.

```
listen [::]:443 ssl ipv6only=on; # managed by Certbot
listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/gaeinventario.ddns.net/fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/gaeinventario.ddns.net/privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
```

Figura 7: Configuración nginx.

4. Herramientas y técnicas utilizadas

4.1. Cloud Computing

4.1.1. Amazon Web Services

Se utiliza esta plataforma en la nube para publicar el servidor donde da servicio la aplicación. Este servicio permite alquilar ordenadores virtuales a los que llamaremos Instancias. En estas instancias se sube el código de la aplicación y se ejecuta para que esta máquina del servicio requerido. Alquilar los servicios de Amazon genera costes.

4.2. Herramientas de control de versiones

4.2.1. Git y GitHub

Git [7] es un sistema de control de versiones que permite el clonado de proyectos en un repositorio de control de versiones en un repositorio local. Este repositorio local plenamente funcional permite trabajar sin conexión o de forma remota con facilidad.

GitHub [8] es una plataforma que permite alojar el proyecto y utilizar un sistema de control de versiones. Con esta herramienta se trabaja para subir y bajar el código desde la máquina local donde se escribe el código hasta la Instancia en Amazon que será la responsable de tener el servicio abierto.

4.3. Herramientas CASE

Para la utilización de esta herramienta se ha utilizado la licencia que brinda la Universidad de Salamanca en el grado de Ingeniería Informática para sus estudiantes.

4.3.1. REM

REM [9] es una herramienta utilizada para la gestión de requisitos durante el desarrollo del proyecto. Esta, proporciona plantillas para la documentación de objetivos, actores dentro del sistema o casos de uso entre varios elementos que componen la elicitación de requisitos.

4.3.2. Visual Paradigm

Visual Paradigm [10] es una herramienta capaz de dar soporte al ciclo de vida completo del desarrollo de *software* ya que permite crear diagramas de modelado UML (Pressman, 2002). Esta herramienta se utiliza durante la fase de especificación de requisitos, análisis de requisitos y diseño del sistema. Estos diagramas aparecerán a lo largo de las memorias.

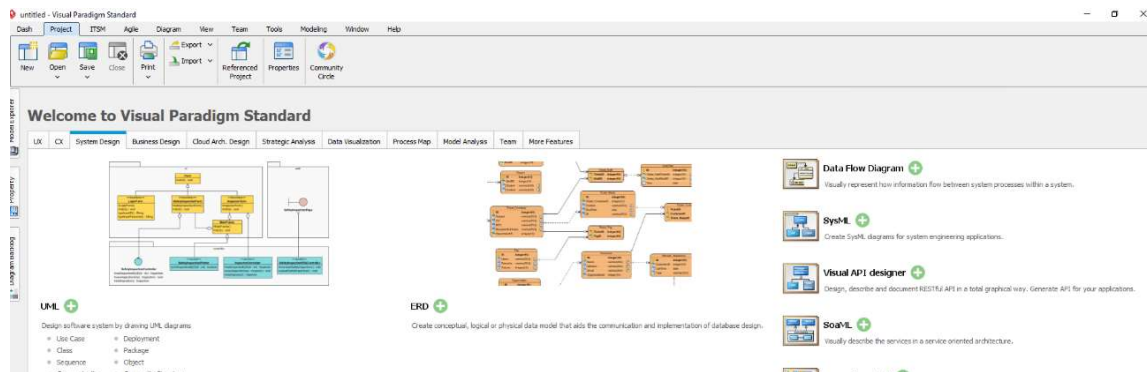


Figura 8: Visual paradigm.

4.3.3. Microsoft Projects

Microsoft Projects [11] es una herramienta que permite realizar una gestión de proyecto mediante planificaciones temporales, asignación de tareas, duraciones de estas, disponibilidad de los recursos y elaboración automática de un diagrama de Gant cuyo objetivo será ver de manera gráfica el tiempo de dedicación previsto para tareas o actividades durante un total determinado. Esta herramienta se utiliza para planificar proyectos con un marco de trabajo de Proceso Unificado como el que se ha desarrollado durante este proyecto.

4.3.4. EZEstimate

EZEstimate es una herramienta que permite realizar una estimación de la duración de proyectos en función de la complejidad de casos de uso creados en la fase de especificación de recursos. Esta herramienta se utiliza en la fase de planificación temporal del proyecto.

4.4. Entornos de desarrollo

4.4.1. Visual Studio Code

Visual Studio Code [12] es un editor de código fuente que permite editar el código fuente, depurarlo y testearlo, así como un control de versiones integrado de GIT. Aporta soporte para gran cantidad de lenguajes de programación.

4.5. Herramientas utilizadas en el servidor

4.5.1. Node.js

Node.js [13] es un entorno de tiempo de ejecución para la capa de servidor basado en el lenguaje de programación JavaScript asíncrono. Utilizada entrada y salida de datos en una arquitectura orientada a eventos.

4.5.2. NPM

NPM [14] es un sistema de gestión de paquetes por defecto para Node.js. Este permite instalar módulos y automatizar la gestión de dependencias

4.5.3. Express

Express [15] es una infraestructura de aplicaciones que permite configurar el *framework* para que la aplicación pueda trabajar con diferentes motores de plantillas. Se utilizan plantillas Handlebars en Javascript que permiten obtener un código más organizado y reutilizar secciones de código así ayudando a mantener el proyecto de manera óptima.

4.5.4. Mongoose

Mongoose [16] es una biblioteca de programación orientada a objetos que permite realizar conexiones entre mongodb y el marco de aplicación web.

4.5.5. Mongoddb

Sistema de bases de datos orientado a documentos. MongoDB [17] almacena estructura de datos BSON (especificación similar a JSON) con un esquema dinámico haciendo que la integración de los datos sea fácil y rápida.

4.5.6. Override

Override es un módulo de Node.js que permite extender la funcionalidad de los formularios. Esta extensión permite enviar métodos como post y delete en dichos formularios. Sin esta funcionalidad solamente se pueden utilizar los métodos post y get.

4.5.7. CSS

Lenguaje utilizado para determinar el aspecto del contenido de nuestra página web. Esto pasa por elementos de color, estilos, y su código se complementa con HTML para crear páginas web completas.

4.5.8. Nodemailer

NodeMailer [18] es un módulo de Node.js que permite añadir automatización el envío de correos en el servidor web. Este, es utilizado para enviar mensajes a la hora de registrar usuarios, la hora de recordar la contraseña a usuarios que la hayan olvidado o alertas sobre los productos que hayan generado esta situación.

4.5.9. Generate password

Generate password es un módulo de Node.js que permite generar contraseñas de manera aleatoria. Este módulo tiene uso en la aplicación a la hora de generar una nueva contraseña que es enviada a los usuarios en caso de que estos la soliciten debido a la pérdida de la contraseña antigua.

4.5.10. Forever

Forever [19] es un módulo de Node.js que permite ejecutar scripts de manera continua. Este módulo es utilizado para ejecutar el script que realiza el despliegue de la aplicación y que hace que esta esté disponible durante todas las horas del día.

4.6. Lenguajes de programación

4.6.1. HTML

Lenguaje utilizado para la elaboración de páginas web. Es un lenguaje de marcado de hipertexto utilizado para estructurar y que describe cómo debe de mostrarse el contenido de una web asignando etiquetas con el contenido. HTML es uno de los lenguajes más comunes a la hora de crear sitios y aplicaciones web.

4.6.2. JSON

Formato de texto sencillo para el intercambio de datos. La información almacenada en el sistema de base de datos utiliza este formato.

En la Figura 9. Formato JSON. se muestra el formato con el que son almacenados los datos.



```
[{"_id": {"$oid": ""}, "nombre": "", "actual": "", "fecha": "", "maximo": "", "minimo": ""}]
```

Figura 9. Formato JSON.

4.6.3. UML

Es un lenguaje gráfico que permite visualizar, especificar, construir y documentar un sistema. Este es utilizado a lo largo de la vida del proyecto *software*.

5. Aspectos relevantes del desarrollo

En este apartado se detallan los aspectos más relevantes de las fases que se dividen en el desarrollo de un sistema.

5.1. Marco de trabajo

El marco de trabajo utilizado durante el desarrollo del *software* es el Proceso Unificado. Este, permite un desarrollo iterativo e incremental. En el proceso unificado se repiten una serie de ciclos de desarrollo que constituyen la vida de un sistema. Cada ciclo de desarrollo concluye con una versión entregable del producto.

Cada ciclo se divide en cuatro fases:

- **Fase de inicio:** Se define el alcance del proyecto y se desarrollan los casos de negocio.
- **Fase de elaboración:** Se planifica el proyecto, se especifican en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema.
- **Fase de construcción:** Se realiza la construcción del proyecto y se define la funcionalidad inicial.
- **Fase de transición:** Se obtiene una versión beta del *software* y se incorporan las mejoras sugeridas en la versión

El seguimiento del Proceso Unificado permite realizar un desarrollo controlado y ordenado debido al seguimiento a la planificación realizada en la hora de la fase de Inicio. Una correcta planificación y un desarrollo iterativo permite la realización de cambios a lo largo de todo el proceso de vida del proyecto que garantizan una integración continua de mejoras. El proceso unificado permite crear *software* de calidad.

5.2. Estimación de coste y esfuerzo

La estimación del coste y esfuerzo es una de las primeras tareas que han de realizarse a la hora del desarrollo de un proyecto. El objetivo es obtener una estimación que puede ser inferior o superior del tiempo de desarrollo de un sistema. Con esta estimación se pueden distribuir los tiempos que son necesarios emplear para las tareas que se tengan definidas en la planificación temporal de una manera más correcta.

Se utiliza el método de Puntos de Caso de Uso (UCP) que permite medir la funcionalidad de un sistema en base a los casos de uso y obtener una estimación en horas de la cantidad de esfuerzo necesario. Se consideran actores, escenarios y factores técnicos y de entorno.

Para calcular esta estimación se utiliza el programa EZEstimate, utilizado en la práctica de estimación de esfuerzo de la asignatura Gestión de Proyectos.

Se puede encontrar más información acerca de la realización de este proceso de estimación en el *Anexo 1 – Plan de proyecto* que podremos encontrar en esta misma memoria.

Para calcular el esfuerzo se utiliza como métrica los puntos de casos de uso. Se evalúan la funcionalidad del sistema, se estima el esfuerzo requerido a la hora de realizar el desarrollo y se tienen en cuenta los actores que participan en el sistema y factores técnicos y de entorno.

Esta métrica requiere el cálculo de tres variables:

- UUCP (Unadjusted Use Case Points)
 - UUCW (Unadjusted Use Case Weight)
Considera el número y complejidad de los casos de uso.
 - UAW (Unadjusted Actor Weight)
Considera el número y complejidad de los actores.
- TCF: Factores de complejidad técnica

5.2.1. UUCW (Unadjusted Use Case Weight)

Se considera el número y complejidad de los casos de uso.

- Número de pasos de los escenarios.
- Complejidad de la interfaz de usuario.
- Número de identidades de la base de datos a la que se accede.
- Número de clases implicadas.

Los distintos tipos de complejidad se definen con:

Simple: Se realizan 3 transacciones o menos.

Medio: Se realizan de 4 a 7 transacciones.

Complejo: Se realizan más de 7 transacciones.

5.2.2. UAW (Unadjusted Actor Weight)

Se considera el número y complejidad de los actores:

Simple: El actor es un sistema que se comunica con la aplicación mediante una API.

Medio: El actor es un sistema y la aplicación se comunica con él mediante un protocolo.

Complejo: Persona con una interfaz gráfica.

5.2.3. TCF (Technical Complexity Factor)

Se definen trece factores de complejidad técnica. A cada factor se le asigna un peso (W) acorde a su impacto y a su complejidad percibida (F) correspondiente a la percepción de complejidad que tiene el equipo de desarrollo.

Tabla 1. Factores de complejidad técnica.

Factores de complejidad técnica		
T1. Sistemas distribuidos	T6. Facilidad de instalación	T11. Características especiales de seguridad
T2. Rendimiento	T7. Facilidad de uso	
T3. Eficiencia del usuario final	T8. Portabilidad	T12. Acceso directo a terceras partes.
T4. Procesamiento interno complejo	T9. Facilidad de cambio	T13. Se requiere entrenamiento especial del usuario
T5. Reusabilidad	T10. Concurrencia	

5.2.3.1. Justificación de los factores de complejidad técnica

- Sistemas distribuidos: 0.
- Eficiencia del usuario final: 1.
- Procesamiento interno complejo: 0.
- Reusabilidad: 1.
- Facilidad de instalación: 0.
- Facilidad de uso: 1.
- Portabilidad: 0.
- Facilidad de cambio: 3.
- Concurrencia: 3.
- Características especiales de seguridad: 2.
- Acceso directo a terceras partes: 0.
- Se requiere entrenamiento especial del usuario: 0.

Cálculo de TFC.

$$TCF = C_1 + C_2 \sum_{i=1}^{13} W_i F_i$$

Ecuación 1. TCF.

5.2.4. ECF (Environment Complexity Factor)

Se definen ocho factores de complejidad del entorno. A cada factor se le asigna un peso (W) acorde a su impacto y a su complejidad percibida (F) correspondiente a la percepción de complejidad que tiene el equipo de desarrollo.

Tabla 2. Factores de complejidad del entorno.

Factores de complejidad del entorno	
E1. Familiaridad con UML	E5. Experiencia en orientación a objetos
E2. Trabajadores a tiempo parcial	E5. Motivación
E3. Capacidad de los analistas	E7. Dificultad del lenguaje de programación
E4. Experiencia en la aplicación	E8. Estabilidad de los requisitos

- Familiaridad con UML: 4.
- Trabajadores a tiempo parcial: 5.
- Capacidad de los analistas: 1.
- Experiencia en la aplicación: 2.
- Experiencia en orientación a objetos: 2.
- Motivación: 4.
- Dificultad del lenguaje de programación: 2.
- Estabilidad de los requisitos: 0.

Cálculo de ECF.

$$ECF = C_1 + C_2 \sum_{i=1}^8 W_i F_i$$

Ecuación 2. ECF.

Los UCP se calculan mediante la utilización de la Ecuación 3.

$$UCP = UUCP * TFC * ECF$$

Ecuación 3. UCP.

Para la estimación del esfuerzo a partir de UCP se utiliza la Ecuación 4.

$$Esfuerzo = UCP * F$$

Ecuación 4. Ecuación.

Siendo F: Factor de conversión (número de horas de persona por UCP).

Esto, realiza el cálculo de una estimación del esfuerzo en número de horas de persona que son necesarias emplear para la realización del proyecto.

The screenshot displays the EZEstimate software interface, which is used for estimating project effort. It is divided into several sections:

- Module:** A dropdown menu currently shows 'Actores'. Below it are 'Add Module' and 'Delete' buttons.
- Summary:** Contains input fields for 'Total Modules' (4), 'Use cases' (Simple: 26, Average: 6, Complex: 0), and 'Actors' (Simple: 0, Average: 1, Complex: 2). It also has an 'Excel Report' button and a 'Generate Report' button.
- Add Actor / Use case:** A section for adding new items, with fields for 'Actor / Use case Name', 'Select Type', and 'Complexity', followed by an 'Add' button.
- Tech / Env Factors:** Includes 'Set Tech Factor' and 'Set Env Factors' buttons.
- Estimation Summary:** A list of calculated values:
 - UAW: 8
 - UUCW: 190
 - UUPC = UAW + UUCW: 198
 - TFactor: 12
 - EFactor: 8
 - TCF = 0.6 + (.01*TFactor): 0.72
 - EF = 1.4 + (-0.03*EFactor): 1.16
 - UCP = UUPC*TCF*EF: 165,3696
 - Total Effort@** 6 Hrs/UCP: 992,2176
- Use case / Actor List:** A table listing all identified use cases and actors.

Id	Module	Type	Name	complexity
1	Actores	Actor	Usuario sin regi...	Average
10	Gestion de usu...	Usecase	Eliminar usuario	Simple
11	Gestion de usu...	Usecase	Peticiones	Simple
12	Gestion de usu...	Usecase	Aceptar	Simple
13	Gestion de usu...	Usecase	Rechazar	Simple
14	Gestion de usu...	Usecase	Perfil	Simple
15	Gestion de usu...	Usecase	Actualizar camb...	Simple
16	Gestion de usu...	Usecase	Cambiar contra...	Simple
17	Gestion de usu...	Usecase	Eliminar cuenta	Simple
18	Gestion de usu...	Usecase	Salir	Simple
19	Gestion de inve...	Usecase	Inventario	Simple
2	Actores	Actor	Usuario registra...	Complex
20	Gestion de inve...	Usecase	Añadir inventario	Average
21	Gestion de inve...	Usecase	Modificar invent...	Average
22	Gestion de inve...	Usecase	Eliminar inventa...	Simple
23	Gestion de inve...	Usecase	Filtrar	Simple
24	Gestion de prod...	Usecase	Productos	Simple
25	Gestion de prod...	Usecase	Añadir producto	Average

Figura 10. Resultados EZEstimate.

El *software* de estimación de costes calcula con los datos que las horas necesarias para la realización de este proyecto son **992**. Estas horas se aproximan a las horas empleadas por parte del desarrollador durante todo el proceso de creación del trabajo.

5.3. Planificación temporal

Este apartado tiene como objetivo dejar documentada la especificación de requisitos del sistema que vamos a desarrollar. Para ello, se utiliza la metodología de Durán Y Bernárdez con las herramientas REM para las tablas que se realizan para los requisitos y la herramienta Visual Paradigm para la realización de diagramas de casos de uso.

Se ha detallado toda la información sobre el desarrollo de la planificación temporal en el *Anexo I - Plan del Proyecto Software*.

Para la realización de esta planificación se ha seguido el marco de trabajo de Proceso Unificado estudiado durante las asignaturas de la rama de gestión de proyectos. Este modelo se basa en el proceso iterativo e incremental en el que se divide el proceso de creación del *software* en cuatro fases: Inicio, Elaboración, Construcción y Transición.

En los siguientes apartados se muestran las tareas que se planifican tener en cada fase del proyecto.

5.3.1. Fase Inicio

Fase en la que se realiza la definición del proyecto. Se realizan las tareas principales como plantear un flujo de trabajo correcto, realizar las especificaciones de requisitos y realizar una investigación de las tecnologías que serán utilizadas a lo largo de la elaboración del proyecto.

Modo de tarea ▼	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin ▼
📌	➤ Inicio	28,8 días?	jue 01/09/22	lun 26/09/22
📌	➤ Iteración 1	28,8 días?	jue 01/09/22	lun 26/09/22
📌	➤ Modelado de Negocio	3,2 días	jue 01/09/22	vie 02/09/22
📌	Reunión con tutores	1,6 días	jue 01/09/22	jue 01/09/22
📌	Planear flujo de trabajo	1,6 días	vie 02/09/22	vie 02/09/22
📌	➤ Requisitos	19,8 días	lun 05/09/22	mié 21/09/22
📌	Establecimiento de los objetivos del sistema	3,2 días	lun 05/09/22	mié 07/09/22
📌	Definición de actores	4,8 días	mié 07/09/22	lun 12/09/22
📌	Establecimiento de requisitos de información	4,8 días	lun 12/09/22	mié 14/09/22
📌	Establecimiento de requisitos funcionales	4,8 días	jue 15/09/22	mar 20/09/22
📌	Establecimiento de requisitos no funcionales	3,2 días	lun 19/09/22	mié 21/09/22
📌	➤ Análisis	5 días	mar 20/09/22	vie 23/09/22
📌	Investigación de tecnologías	3,2 días	mar 20/09/22	jue 22/09/22
📌	Investigación del hardware	3,2 días	jue 22/09/22	vie 23/09/22
📌	➤ Diseño	3,2 días	vie 23/09/22	lun 26/09/22
📌	Diseño del inventario	3,2 días	vie 23/09/22	lun 26/09/22
📌	Fin iteración 1	0,6 días?	jue 01/09/22	jue 01/09/22
📌	Fin de Inicio	0,6 días?	jue 01/09/22	jue 01/09/22

Figura 11: Fase Inicio.

5.3.2. Fase Elaboración

Durante esta fase se realiza la planificación del proyecto. Se realiza una estimación del esfuerzo que supone la realización del desarrollo *software*. Se realiza el establecimiento de requisitos y documentación de los casos de uso ajustándose a las estimaciones.

	▸ Elaboración	30,4 días?	mar 27/09/22	vie 21/10/22
	▸ Iteración 2	30,4 días	mar 27/09/22	vie 21/10/22
	▸ Modelado de Negocio	4,6 días	mar 27/09/22	jue 29/09/22
	Reunión con tutores	1,6 días	mar 27/09/22	mar 27/09/22
	Estimación de esfuerzo	1,6 días	mar 27/09/22	mar 27/09/22
	Planificación Temporal	1,6 días	mié 28/09/22	jue 29/09/22
	▸ Requisitos	14,6 días	jue 29/09/22	mié 12/10/22
	Establecimiento de requisitos restantes	4,8 días	jue 29/09/22	mar 04/10/22
	Diagramas de casos de uso	4,8 días	mié 05/10/22	vie 07/10/22
	Documentación de los casos de uso	4,8 días	lun 10/10/22	mié 12/10/22
	▸ Análisis	9,6 días	mié 12/10/22	mié 19/10/22
	Paquete de análisis	3,2 días	mié 12/10/22	jue 13/10/22
	Realización de casos de uso: Gestión de Usuarios	4,8 días	jue 13/10/22	lun 17/10/22
	Realización de casos de uso: Gestión de Productos	3,2 días	vie 14/10/22	lun 17/10/22
	Realización de casos de uso: Gestión de Inventario	4,8 días	lun 17/10/22	mié 19/10/22
	▸ Diseño	3,2 días	mié 19/10/22	jue 20/10/22
	Estudio de patrones de diseño	3,2 días	mié 19/10/22	jue 20/10/22
	▸ Implementación	3,2 días	jue 20/10/22	vie 21/10/22
	Construcción del inventario	3,2 días	jue 20/10/22	vie 21/10/22
	▸ Pruebas	1,6 días	vie 21/10/22	vie 21/10/22
	Pruebas del inventario	1,6 días	vie 21/10/22	vie 21/10/22
	Fin iteración 2	1,6 días	vie 21/10/22	vie 21/10/22
	Fin Elaboración	0,6 días?	vie 21/10/22	vie 21/10/22

Figura 12: Fase Elaboración.

5.3.3. Fase Construcción

Durante la fase de construcción se realiza la evolución de los requisitos y casos de uso hasta un producto entregable. Se realiza una refinación de los detalles menores de los casos de uso y se procede a la construcción del producto. Una vez construido, se procede a la realización de pruebas este para comprobar que todo funciona y se cumplan los requisitos propuestos al inicio del proyecto.

→	▸ Construcción	54,4 días?	vie 21/10/22	mié 07/12/22
→	▸ Iteración 3	54,4 días?	vie 21/10/22	mié 07/12/22
→	▸ Modelado de Negocio	3,2 días	vie 21/10/22	lun 24/10/22
→	Reunión con Tutores	3,2 días	vie 21/10/22	lun 24/10/22
→	▸ Diseño	4,8 días	lun 24/10/22	mié 26/10/22
→	Realización de casos de uso: Gestión de Usuarios	1,6 días	lun 24/10/22	lun 24/10/22
→	Realización de casos de uso: Gestión de Productos	1,6 días	mar 25/10/22	mar 25/10/22
→	Realización de casos de uso: Gestión de Inventario	1,6 días	mié 26/10/22	mié 26/10/22
→	▸ Construcción	46,4 días	mié 26/10/22	lun 05/12/22
→	Construcción Gestión de Usuarios	19,2 días	mié 26/10/22	jue 10/11/22
→	Construcción Gestión de Productos	19,2 días	jue 10/11/22	vie 25/11/22
→	Construcción Gestión de Inventario	11,2 días	vie 25/11/22	lun 05/12/22
→	▸ Pruebas	4,8 días	lun 05/12/22	mié 07/12/22
→	Pruebas unitaria de los componentes	4,8 días	lun 05/12/22	mié 07/12/22
→	Fin iteración 3	0,6 días?	lun 05/12/22	lun 05/12/22
→	Fin Construcción	0,6 días?	lun 05/12/22	lun 05/12/22

Figura 13: Fase Construcción.

5.3.4. Fase Transición

Por último, en la fase de transición se realiza la finalización de la documentación necesaria de entregar para el proyecto, el despliegue de la aplicación del sistema y las últimas pruebas donde se comprueba que no existen fallos esperados en las ejecuciones del sistema.

→	▸ Transición	27,2 días?	lun 12/12/22	mar 03/01/23
→	▸ Iteración 4	27,2 días	lun 12/12/22	mar 03/01/23
→	▸ Modelo de Negocio	6,4 días	lun 12/12/22	jue 15/12/22
→	Finalizar documentación	4,8 días	lun 12/12/22	mié 14/12/22
→	Reunión tutores	1,6 días	jue 15/12/22	jue 15/12/22
→	▸ Implementación	12,8 días	vie 16/12/22	mar 27/12/22
→	Despliegue del sistema	3,2 días	vie 16/12/22	lun 19/12/22
→	Corrección de fallos	11,2 días	lun 19/12/22	mar 27/12/22
→	▸ Pruebas	9,6 días	mar 27/12/22	mar 03/01/23
→	Pruebas del sistema completo	4,8 días	mar 27/12/22	jue 29/12/22
→	Pruebas con los fallos correctiso	6,4 días	jue 29/12/22	mar 03/01/23
→	Fin iteración 4	0,6 días?	mar 03/01/23	mar 03/01/23
→	Fin de Transición	0,6 días?	mar 03/01/23	mar 03/01/23

Figura 14: Fase Transición.

5.4. Especificación de requisitos

En la especificación de requisitos se recogen los objetivos y requisitos funcionales que tiene el sistema. Todos estos requisitos están expuestos mediante tablas basadas en la metodología de Durán y Bernárdez. Se exponen los requisitos de los casos de uso, los requisitos no funcionales, los requisitos de información entre otros requisitos del sistema.

Para cada apartado de especificaciones es representada una tabla que sirve de ejemplo del resto de tablas que se podrán encontrar en cada uno de estos.

En el *Anexo II - Especificación de Requisitos del Software* se recoge la información correspondiente a esta fase.

5.4.1. Actores

Actores que intervienen dentro del sistema:

- Usuario sin registrar.
- Usuario registrado.
- Usuario administrador.

La Tabla 3 representa una tabla de actores del sistema.

Tabla 3. Actores.

ACT-0001	Usuario sin registrar
Versión	1.0
Autores	• Daniel González Martín
Fuentes	-
Descripción	Este actor representa a un usuario genérico que accede al sistema. Deberá realizar una petición que deberá ser aceptada por los administradores para que pueda acceder al sistema.
Comentarios	Ninguno

5.4.2. Objetivos del sistema

En este apartado se recogen las tablas donde se serán descritos los objetivos principales del sistema que deben ser alcanzados para satisfacer los requisitos planteados.

Los principales objetivos del sistema serán:

- Gestión de Usuarios.
- Gestión de Inventario.
- Gestión de Productos.
- Gestión de Alertas.

La Tabla 4 representa una tabla de objetivos del sistema.

Tabla 4. Objetivos del sistema.

OBJ-0001	Gestión de Usuarios
Versión	1.0
Autores	• Daniel González Martín
Fuentes	-
Descripción	
Subobjetivos	El Sistema deberá ser capaz de funcionar simultáneamente en varios dispositivos.
Importancia	Alta
Urgencia	Alta
Estado	Alta
Estabilidad	Alta
Comentarios	Ninguno

5.4.3. Requisitos de información

En este apartado se recogen los datos mínimos que se deberán almacenar en el sistema.

- Usuario registrado.
- Inventario.
- Configuración.
- Producto.
- Producto gráfica.
- Producto temporal.
- Alerta.

En estas tablas se detallan los datos específicos de los que se deben disponer en la aplicación.

La

Tabla 5 representa una tabla de requisitos de información del sistema.

IRQ-0006	Alerta	
Versión	1.0	
Autores	• Daniel González Martín	
Fuentes	-	
Dependencias	Ninguna	
Descripción	El sistema deberá ser capaz de almacenar las alertas que se generen dentro del sistema cuando el valor de los productos creados en el sistema no esté entre los límites establecidos por los usuarios.	
Datos específicos	• Nombre (String) • Tipo (String) • Actual (Number) • Alerta (String)	
Tiempo de vida	Medio	Máximo
	Indefinido	
Ocurrencias simultáneas	Medio	Máximo
	Intermedio	
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 5. Requisitos de información.

5.4.4. Requisitos no funcionales

En este apartado se especifican los requisitos no funcionales que se requieren en el sistema que se ha sido desarrollado. Los requisitos no funcionales definen aquellas restricciones que se deben cumplir en el proyecto y con los que se garantiza un diseño seguro con el que se cumplen ciertos estándares de calidad.

- Simultaneidad.
- Disponibilidad.
- Interfaz clara.
- Tiempo de respuesta.
- Usabilidad.
- Copia de seguridad.
- Base de datos.

La Tabla 6 representa una tabla de requisito no funcional del sistema.

Tabla 6. Requisitos no funcionales.

NFR-0001	Protección de datos
Versión	1.0
Autores	• Daniel González Martín
Fuentes	-
Dependencias	
Descripción	El Sistema deberá ser capaz de funcionar simultáneamente en varios dispositivos.
Importancia	Alta
Urgencia	Alta
Estado	Alta

5.4.5. Requisitos funcionales

Los requisitos funcionales son recogidos en forma de casos de uso. Estos representan las diferentes funcionalidades y servicios del sistema.

Estos casos de uso son agrupados en diferentes paquetes que se definen en un diagrama de paquetes que permite obtener una visión inicial del sistema.

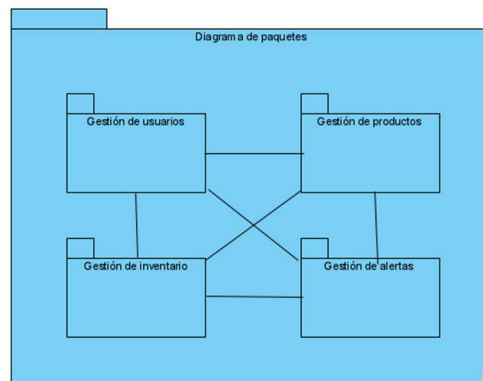


Figura 15: Diagrama de paquetes.

Se define un diagrama de casos de uso para cada paquete donde se especifican los casos de uso que pertenecen a cada paquete. En la **¡Error! No se encuentra el origen de la referencia.** se podrán ver el Diagrama de Casos de uso del paquete de Gestión de Usuarios.

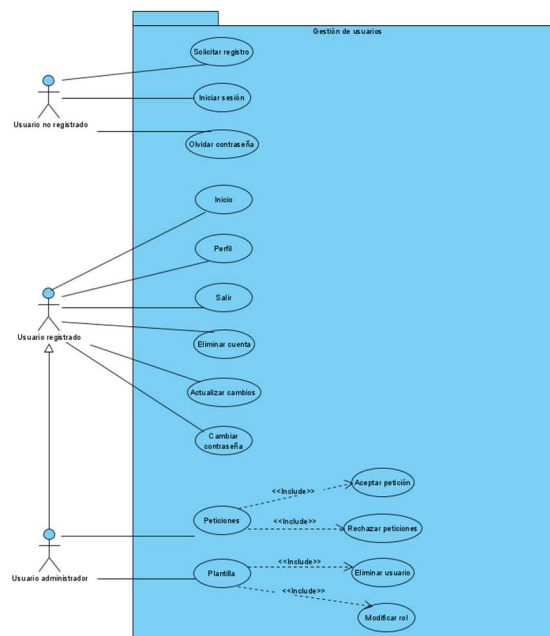


Figura 16: Diagrama de casos de uso.

Estos casos de uso y la información de cada uno se recogen en el *Anexo II: Especificación de Requisitos del Software*.

La Tabla 7 representa un caso de uso del sistema.

Tabla 7: Solicitar registro.

UC-0001	Solicitar registro	
Versión	1.0 (10/09/2022)	
Autores	·Daniel González Martín	
Fuentes	Enunciado	
Dependencias	<u>[OBJ-0001] Gestión de usuarios</u>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>usuario</i> haya hecho clic en 'Solicitar registro'.	
Precondición	El usuario ha enviado su formulario de registro, rellenado correctamente, al sistema cuando haya terminado de registrarse.	
Secuencia normal	Paso	Acción
	1	El sistema mostrará un mensaje en la pantalla en el que confirmará al usuario que se ha enviado una petición para entrar al sistema a los administradores.
	2	El sistema enviará al usuario un correo electrónico para confirmar que se ha realizado el envío a los administradores para que acepten su entrada al sistema.
	3	El sistema almacenará en la base de datos los datos del usuario almacenando un "No" en aceptado. El usuario no podrá acceder al sistema hasta que un administrador acepte la petición.
Postcondición	El usuario se añadirá al sistema.	
Excepciones	Paso	Acción
	1	Si el correo introducido por el usuario está repetido en el sistema, no se realizará ningún proceso de aviso al correo ni se introducirá un nuevo dato en la base de datos.
Rendimiento	Paso	Tiempo máximo
	-	-
Importancia	Importante	
Urgencia	Vital	
Estado	Inmediatamente	
Estabilidad	Alta	
Comentarios	Ninguno	

En la Tabla 8 se representa la agrupación de los casos de uso en los diferentes paquetes definidos en el sistema.

Tabla 8: Casos de uso.

Gestión de usuarios	Gestión de inventario	Gestión de productos	Gestión de Alertas
[UC-0001] Solicitar registro	[UC-0016] Inventario	[UC-0021] Productos	[UC-0029] Alertas
[UC-0002] Olvidar contraseña	[UC-0017] Añadir inventario	[UC-0022] Añadir producto	[UC-0030] Activar alertas automáticas
[UC-0003] Iniciar sesión	[UC-0018] Modificar inventario	[UC-0023] Modificar producto	[UC-0031] Desactivar alertas automáticas
[UC-0004] Inicio	[UC-0019] Eliminar inventario	[UC-0024] Eliminar producto	[UC-0032] Avisar administradores
[UC-0005] Personal	[UC-0020] Filtrar	[UC-0025] Gráficas	
[UC-0006] Modificar rol		[UC-0026] Cambiar gráfica	
[UC-0007] Eliminar usuario		[UC-0027] Estadísticas	
[UC-0008] Peticiones		[UC-0028] Cambiar estadísticas	
[UC-0009] Aceptar			
[0010] Rechazar			
[UC-0011] Perfil			
[UC-0012] Actualizar cambios			
[UC-0013] Cambiar contraseña			
[UC-0014] Eliminar cuenta			
[UC-0015] Salir			

5.5. Análisis del sistema

Una vez recogidos los requisitos en la fase de especificación de requisitos, se procede a realizar una fase de análisis de estos en la que se refina la información de estos y en la que se obtienen los casos de uso y una vista arquitectónica de una manera más específica.

En el *Anexo III – Análisis del sistema* se especifica el primer modelo de dominio, la transformación de los casos de uso en diagramas de secuencia, las clases de análisis y una vista arquitectónica del modelo.

5.5.1. Modelo de dominio

En el modelo de dominio se recogen las necesidades definidas en los requisitos de información y las representa de una manera más conceptual. Se determinan cuales son las entidades que lo componen, con sus atributos y relaciones entre ellas.

El diagrama del modelo de dominio se muestra en la Figura 17.

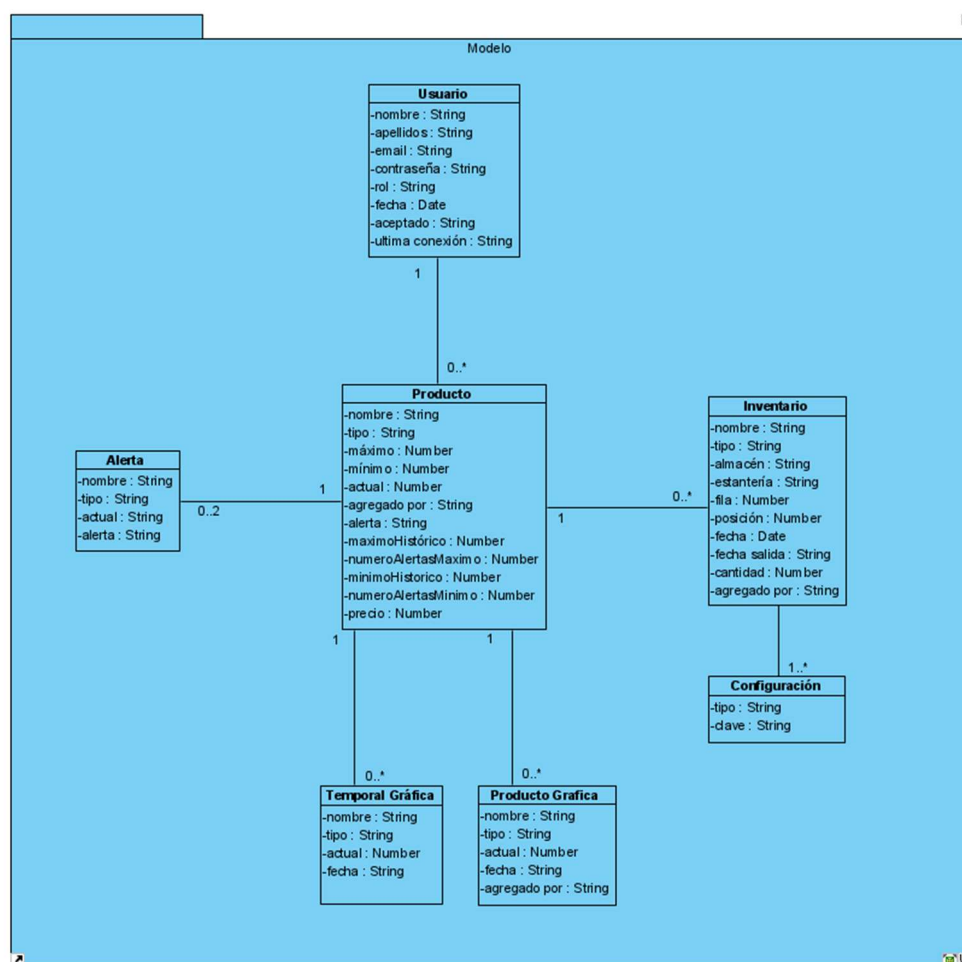


Figura 17: Modelo de dominio.

En el que se encuentran las siguientes clases:

- Usuario: Usuarios del sistema.
- Inventario: Inventario almacenado en el sistema.
- Producto: Productos creados en el sistema.
- Configuración: Configuraciones del establecimiento.
- Alerta: Alertas activas generadas en situaciones donde la cantidad de producto esté por encima o por debajo de los valores óptimos establecidos.
- TemporalGráfica: Entrada generada diariamente sobre la cantidad de presente de los productos.
- ProductoGráfica: Entrada generada cada vez que se realiza algún movimiento sobre el inventario de algún producto.

5.5.2. Paquete de análisis

Los paquetes de análisis escenifican la descomposición del sistema *software*. Permite realizar un análisis de cómo están distribuidas las clases de análisis.

Se utilizan estas clases para la realización de los casos de uso en los diagramas de secuencia.

Durante el desarrollo del proyecto se utiliza un patrón Modelo Vista Controlador utilizado en gran cantidad de asignaturas a lo largo de la carrera.

El diagrama de paquetes se muestra en la Figura 18.

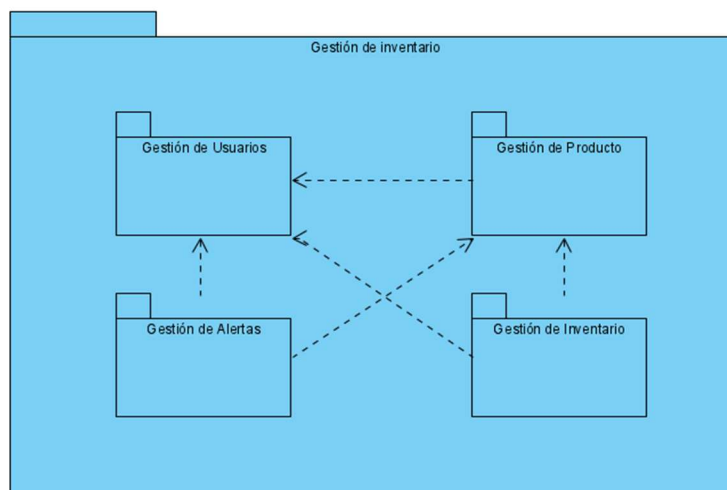


Figura 18: Paquete de análisis.

Se encuentran los siguientes paquetes:

- Gestión de Usuarios: Clases de análisis relacionadas con la lógica de Usuarios.
- Gestión de Inventario: Clases de análisis relacionadas con el Inventario del sistema.
- Gestión de Productos: Clases de análisis relacionadas con los Productos del sistema.
- Gestión de Alertas: Clases de análisis relacionadas con las Alertas del sistema.

5.5.3. Realización de casos de Uso

Los diagramas de secuencia muestran la realización de caso de uso a nivel de análisis. Este diagrama permite observar cómo los actores del sistema interactúan con las vistas de este y cómo esta realiza la comunicación con el controlador.

Se muestra un diagrama de secuencia en la Figura 19 donde un Usuario no registrado realiza la solicitud para acceder al sistema.

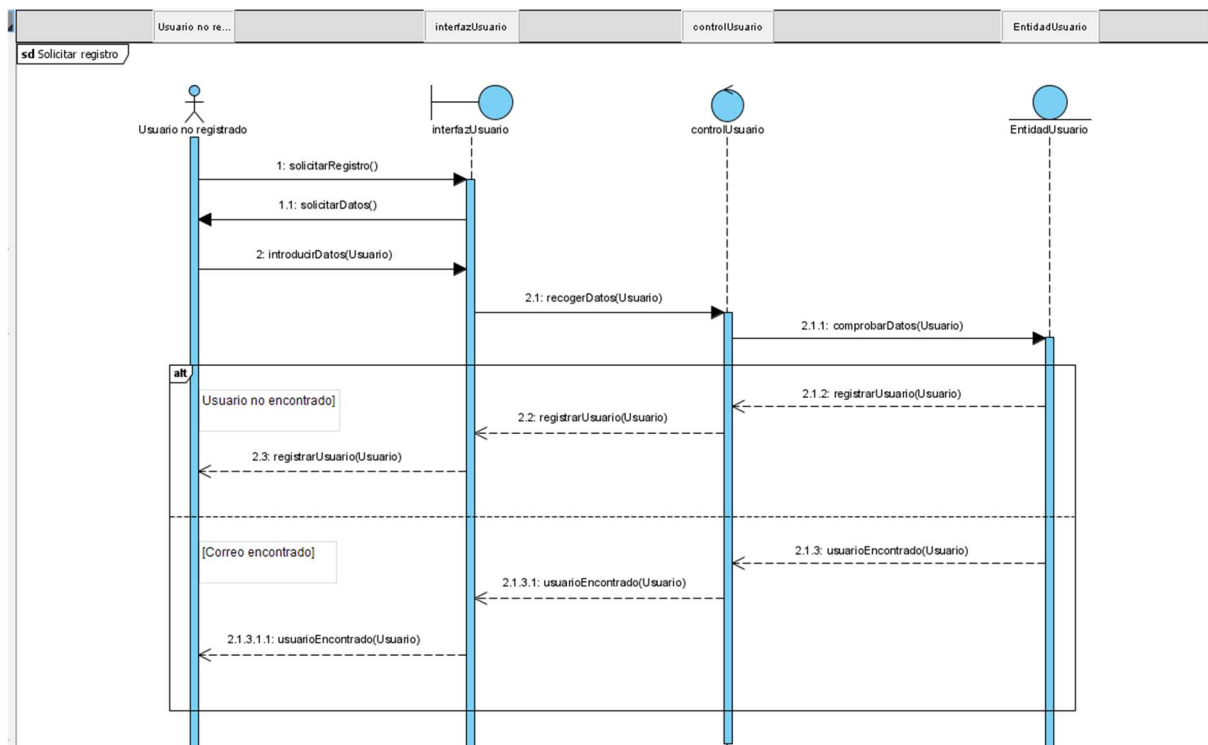


Figura 19: Solicitar registro.

En este ejemplo, el Usuario realiza la Solicitud de Registro. El usuario debe introducir los datos que se le especifiquen en el formulario de Solicitud. Serán Nombre, Apellidos, Correo y Contraseña. Una vez introducidos, estos datos son enviados y recogidos por el Controlador. El controlador solicita datos al modelo de Usuario y en caso de que el correo insertado por el usuario en el formulario coincida con algún correo almacenado en el sistema. Esto, genera un fallo que produce que no se registre la solicitud.

En caso de que no exista coincidencia con ningún correo almacenado en el sistema, se registra la solicitud del usuario y se almacenan los datos de este en la base de datos del sistema.

La solicitud debe ser aprobada o denegada por un usuario con rol Administrador para que el usuario que ha realizado la solicitud pueda acceder o no en el sistema.

5.6. Diseño del sistema

Supone la fase de mayor fase definición de requerimientos del sistema ya que representa la última fase de comenzar con la construcción del sistema.

5.6.1. Modelo de diseño

El modelo de diseño recoge la arquitectura elegida para el sistema y la especificación de los componentes con sus relaciones. Permite describir los modelos y las clases de análisis con un mayor detalle. En este apartado se tiene como objetivo detallar de una manera más concreta los datos expuestos en el modelo de análisis y conseguir una menor abstracción a la hora de realizar la implementación de los elementos del sistema.

5.6.1.1. Patrones arquitectónicos

El patrón de arquitectura elegido para la realización del sistema ha sido el patrón Modelo Vista Controlador (MCV) el cual nos permite separar los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes.

- El modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. Es el responsable acceder a la capa de almacenamiento, definir las reglas de negocio y llevar un registro de las vistas y controladores del sistema.
- La vista compone la información que es enviada al cliente y los mecanismos de interacción con este. Es el responsable de recibir datos del modelo y mostrarlos al usuario.
- El controlador actúa como intermedio entre Modelo y Vista. Gestiona el flujo de información entre ellos. Es el responsable de recibir los elementos de entrada y contendrá reglas de gestión de eventos.

Este patrón facilita el buen desarrollo de la aplicación debido al funcionamiento del servidor. La utilización de un servidor ejecutado orientado a eventos asíncronos permite que sea el controlador la parte que reciba las peticiones de los usuarios y quien maneje la información pedida al modelo. El controlador realiza la pedida de datos al modelo y es el encargado de enviarlos a la vista para que esta pueda mostrar la información pedida por los usuarios.

En la Figura 20 se muestra un esquema del patrón de arquitectura MVC:

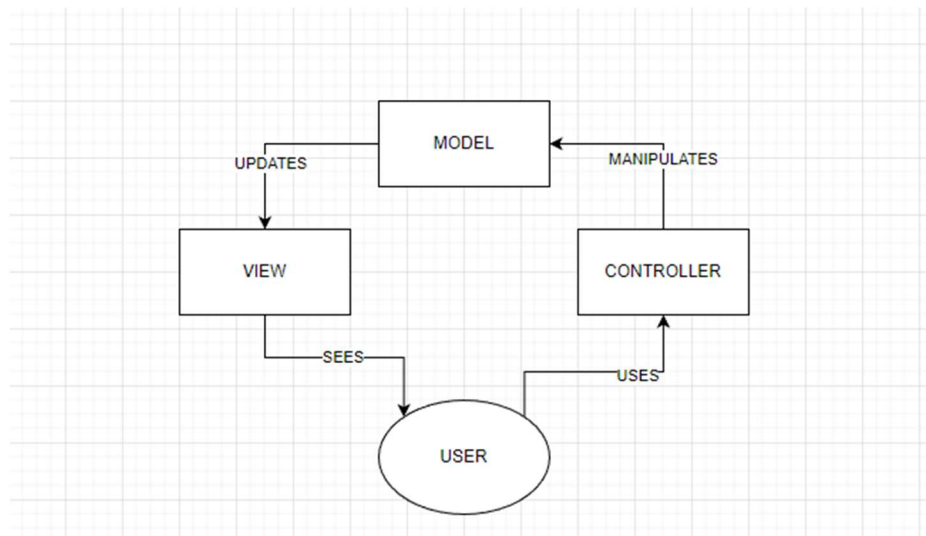


Figura 20: MVC.

5.6.1.2. Subsistema de diseño

En este apartado se representa la descomposición del sistema en módulo y las relaciones que existen entre ellos. Los módulos de la descomposición seguirán los siguientes pasos:

1. El usuario realiza una operación.
2. Routes recibe la petición y será quien haga el acceso al controlador.
3. El controlador realiza las operaciones necesarias sobre el modelo.
4. Se actualiza la vista con los cambios del modelo.
5. Se muestra al usuario la vista con los cambios realizados.

En la Figura 21 se puede observar el diagrama de paquetes que representa estos subsistemas.

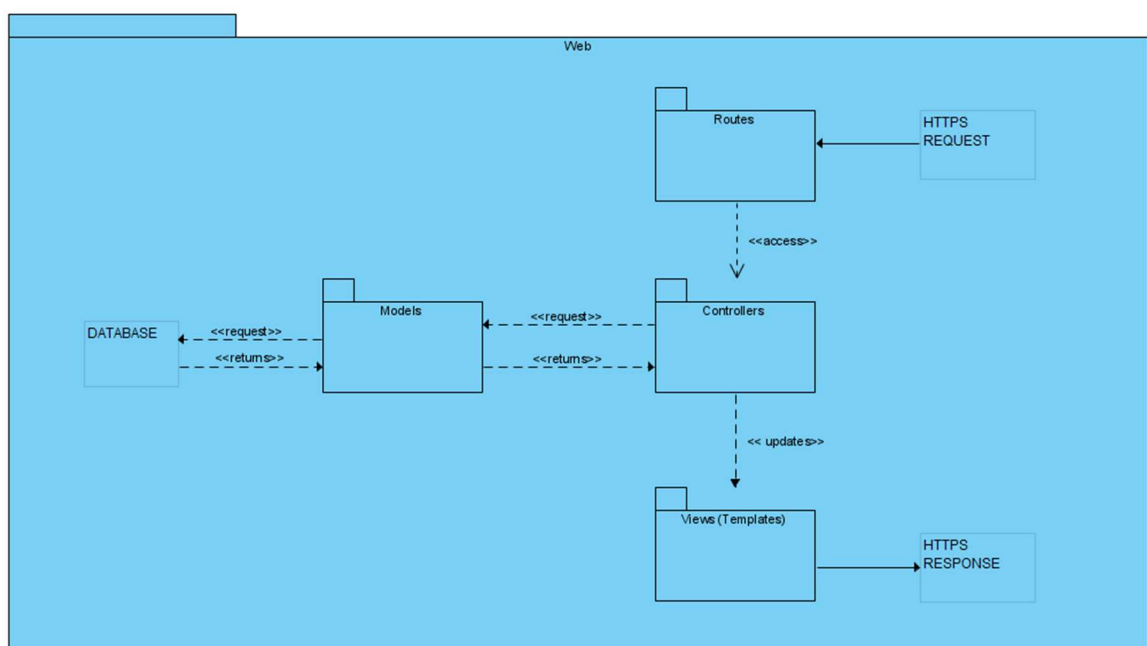


Figura 21: Diagrama de paquetes.

5.6.1.3. Realización de casos de uso

En la fase de diseño se refinan los casos de uso realizados en la fase de análisis permitiendo realizar el enfoque en el dominio de la solución y realizando una disminución del grado de abstracción.

En la Figura 22 se muestra un diagrama de secuencia realizado en el que se puede observar el paso de mensajes entre los actores y los diferentes objetos del sistema.

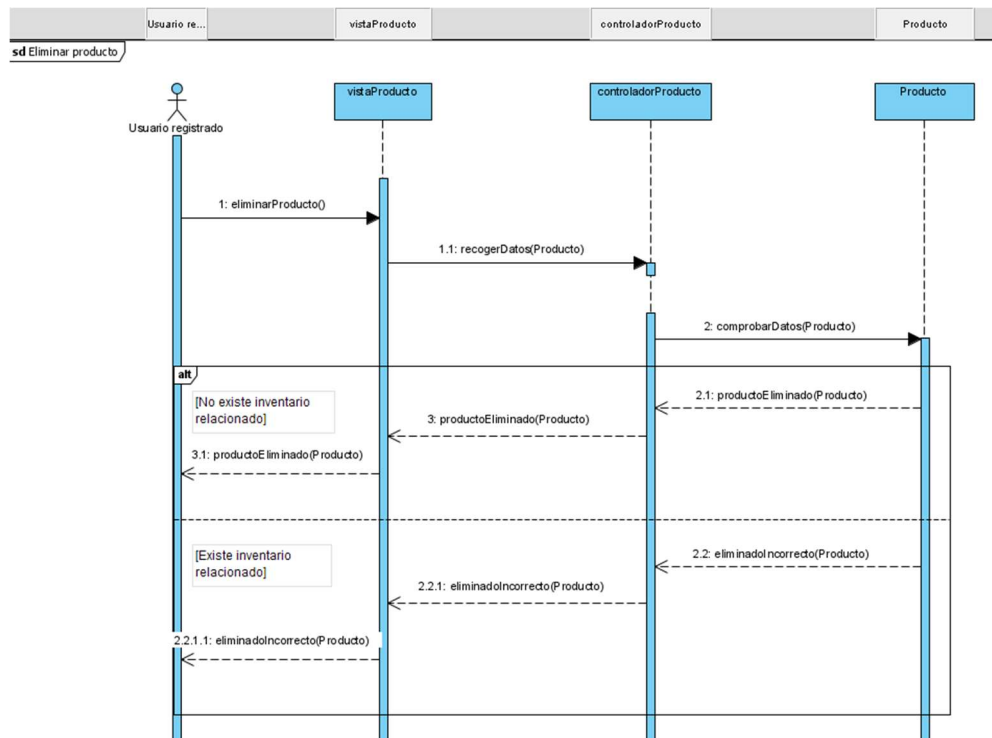


Figura 22: Diagrama de secuencia.

En la Figura 22 se puede ver el Diagrama de Secuencia del caso de uso Eliminar Producto donde el Usuario Registrado realiza una eliminación del producto. Una vez realizada la petición por la vista, el controlador es quién pida información al Modelo Producto para comprobar que no existe inventario creado con dicho producto.

En caso de que exista inventario creado, el sistema no permite al usuario realizar el borrado y se muestra una alerta de que no es posible realizar la eliminación del producto del sistema.

En caso de que no exista inventario creado para dicho producto, el sistema procede con el borrado del producto y muestra una alerta de que el producto ha sido eliminado del sistema de manera correcta.

5.6.2. Modelo de despliegue

El modelo de despliegue permite representar la distribución en despliegue de los elementos del sistema. Cada nodo representa un elemento *software* que forma parte del sistema final de producción.

En el *Anexo IV – Diseño del sistema software* se especifican los nodos presentes en el despliegue de la aplicación.

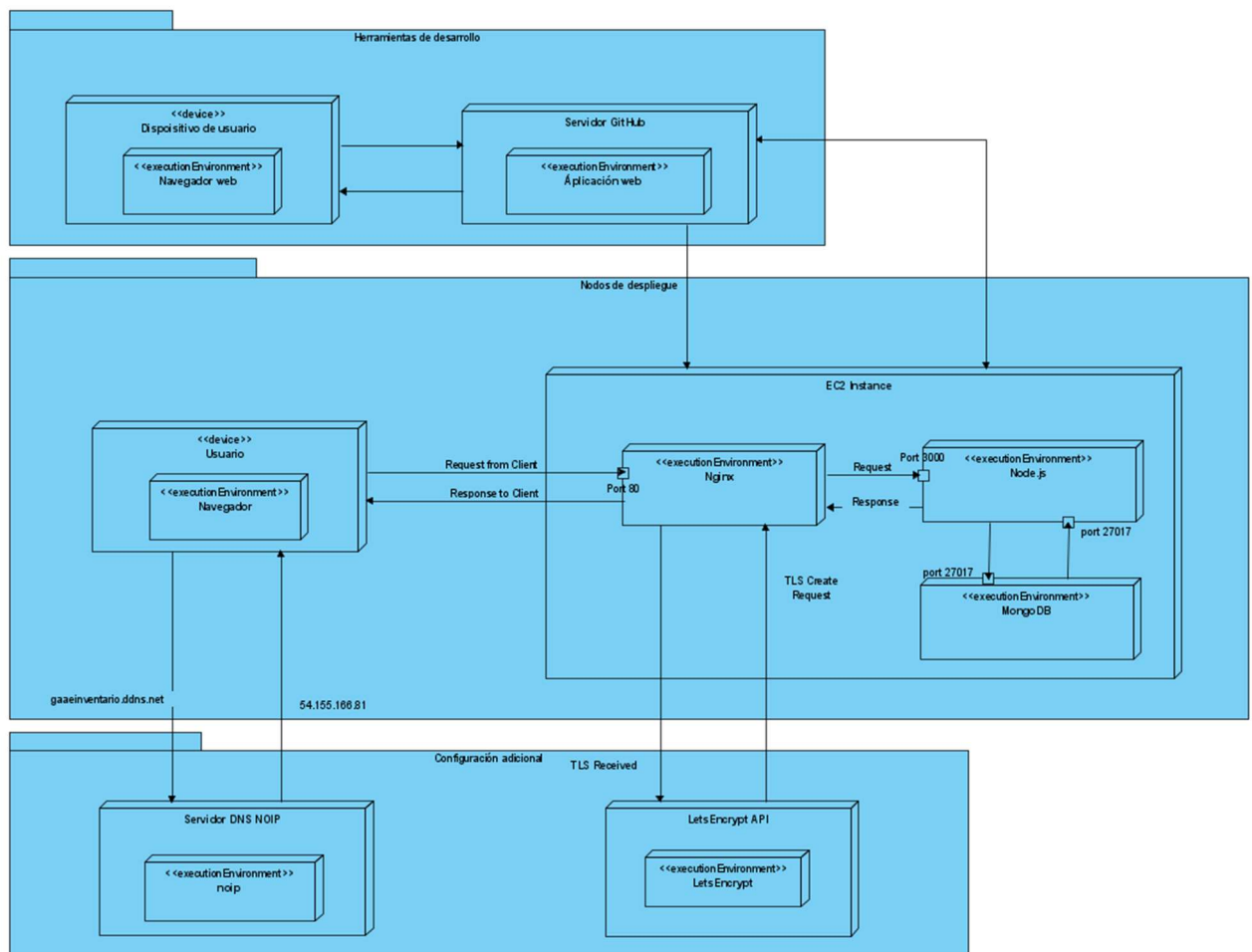


Figura 23: Modelo de despliegue.

5.7. Implementación

En la fase de implementación se realiza la codificación del sistema tomando de referencia para el resultado final todos los resultados obtenidos en la fase de diseño. Se trata de una fase donde se utilizan y se ponen en prácticas las técnicas, herramientas especificadas a lo largo de todos los anexos del proyecto.

La implementación se puede dividir en las siguientes partes:

- **Aplicación:** Se lleva a cabo la implementación de todas las funcionalidades descritas a lo largo de la fase de diseño y desarrollado todas las tablas e ideas en una aplicación que pueda utilizarse. A la hora de realizar el diseño de la aplicación se buscaba realizar una aplicación que permitiera el registro de usuarios y realizar un correcto seguimiento de inventario en un entorno empresarial. Se puso como caso de uso el inventario de un bar. Para ello, se realizaron casos de uso donde se especificaron funcionalidades relacionadas con Usuarios, Inventario, Productos y Alertas. En esta parte se realiza toda la codificación y construcción de la aplicación para que las funcionalidades buscadas puedan realizarse y por lo tanto la aplicación tenga el sentido y uso que se busca de ella.
- **Servidor:** En esta parte se realiza todo el proceso de creación de Servidor donde la aplicación realiza sus funciones. En esta parte se realiza la puesta en punto del servidor en la plataforma de computación bajo la nube AWS donde realiza las funciones la aplicación. Para ello es necesario utilizar el servicio EC2 que permite el alquiler de máquinas virtuales. Una vez alquilado el servidor, es necesario utilizar un *software* de control de versiones que permita realizar intercambio de código entre las máquinas donde se realizan las pruebas de manera local y la máquina donde se realiza el despliegue de la aplicación. Una vez configurado esto, es necesario realizar el registro pertinente de la máquina en algún sistema de nombre de dominios que permite que los usuarios puedan acceder de manera correcta al sistema. Una vez registrado el nombre de la aplicación y comprobado que el tráfico llega a la máquina, es necesario redirigirlo al puerto donde la aplicación esté escuchando y garantice que las acciones pedidas por los usuarios sean realizadas. Por último, es necesario instalar certificados SSL para las comunicaciones entre clientes y servidores para garantizar la seguridad de los datos que serán intercambiados entre los usuarios y la aplicación.

Una vez hecho el despliegue de la aplicación en el servidor, se puede decir que el sistema se ha desplegado de manera correcta y está operativo para los usuarios finales.

5.8. Pruebas

Las pruebas representan una parte fundamental del desarrollo de proyectos. Unas pruebas correctas permiten realizar una comprobación de que todas las funcionalidades del sistema se están realizando de manera correcta y, por lo tanto, se están cumpliendo los objetivos marcados para la aplicación. El marco de trabajo de Proceso Unificado ha permitido realizar pruebas sobre la aplicación en cada una de las fases de las que este se compone debido a las reuniones periódicas que se han desarrollado entre tutores y alumnos. Estas reuniones han permitido encontrar problemas en las funcionalidades o cambios en estas que han ido surgiendo a lo largo del proceso de creación. Este proceso ha contribuido a realizar una mejora continua e iterativa.

Para cada una de las funcionalidades del sistema se han realizado pruebas de validación durante y a la finalización del proyecto. Esto ha permitido comprobar que las funcionalidades realizan las acciones esperadas de manera correcta.

Para finalizar, se han realizado pruebas globales con la finalidad de conocer si el sistema general se comporta de forma correcta una vez se hayan integrado todos los componentes del sistema.

Se han realizado pruebas para todas las acciones descritas y buscadas a la hora de realizar el diseño del sistema para comprobar que el funcionamiento de la aplicación es el correcto.

5.9. Descripción de la aplicación

En este apartado se muestran las principales funcionalidades del sistema desarrollado. Se detallan todas las funcionalidades en el *Anexo VI - Manual del usuario*.

Las principales funcionalidades del sistema son divididas en cuatro:

- Gestión de Usuarios.
- Gestión de Inventario.
- Gestión de Productos.
- Gestión de Alertas.

5.9.1. Gestión de usuarios

En esta parte del sistema se realizan las funcionalidades que tengan que ver con el almacenamiento y gestión de los usuarios.

Por una parte, se realizan las funcionalidades que tengan que Inicio de sesión de Usuarios en la aplicación con la Solicitud de Registro, Olvido de contraseña, Inicio de sesión.

Por una segunda parte, se realizan las funciones de gestión de Usuarios dentro de la aplicación donde los Administradores podrán realizar las funcionalidades de aceptar y rechazar las Solicitudes del Sistema, y movimiento en los roles de los Usuarios dentro del sistema.

5.9.1.1. Solicitar registro

En esta funcionalidad, el Usuario no registrado en el sistema debe rellenar dicho formulario para generar una solicitud a los Administradores que deberán aceptar o rechazar. Una vez aceptada, el Usuario que ha realizado la petición obtiene el acceso a la aplicación. En caso de ser rechazada se elimina la entrada generada por el Usuario. Una vez resuelta la solicitud se envía un correo al correo electrónico especificado por el usuario para con el que se informa de la resolución de la solicitud.

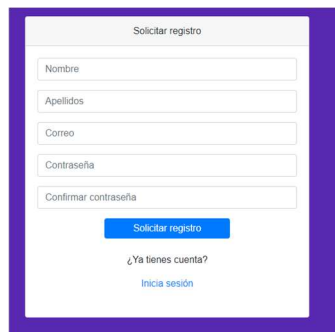
El formulario, titulado "Solicitar registro", está contenido en un recuadro con un borde púrpura. Incluye campos de entrada para "Nombre", "Apellidos", "Correo", "Contraseña" y "Confirmar contraseña". Debajo de estos campos hay un botón azul que dice "Solicitar registro". En la parte inferior del formulario, se encuentra el texto "¿Ya tienes cuenta?" con un enlace azul "Inicia sesión" debajo de él.

Figura 24: Solicitar registro.

5.9.1.2. Olvidar contraseña

En esta funcionalidad, el usuario en el sistema debe rellenar un formulario donde se le pide el correo electrónico para generar una nueva contraseña que será enviada a este correo. En este correo se recomendará al usuario realizar un cambio de la contraseña a una propia.

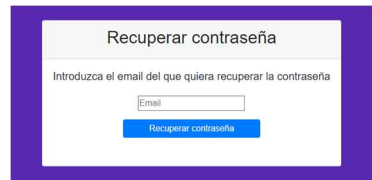
El formulario tiene un título "Recuperar contraseña" en un recuadro gris. Debajo, un texto indica: "Introduzca el email del que quiera recuperar la contraseña". Hay un campo de entrada "Email" y un botón azul "Recuperar contraseña".

Figura 25: Olvidar contraseña.

5.9.1.3. Iniciar sesión

En esta funcionalidad se realiza la comprobación si el correo introducido y la contraseña coinciden con algún usuario registrado en el Sistema.

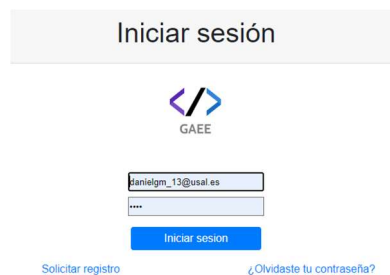
El formulario tiene un título "Iniciar sesión" en un recuadro gris. Debajo está el logo de GAEE. Hay dos campos de entrada: uno para el correo electrónico (contiene "danielgm_13@usal.es") y otro para la contraseña (contiene "****"). Un botón azul "Iniciar sesión" está debajo. En la parte inferior, hay dos enlaces: "Solicitar registro" a la izquierda y "¿Olvidaste tu contraseña?" a la derecha.

Figura 26: Iniciar sesión.

Una vez comprobado que el usuario y la contraseña se han introducido de manera correcta, se redirige al Usuario al Inicio de la aplicación.

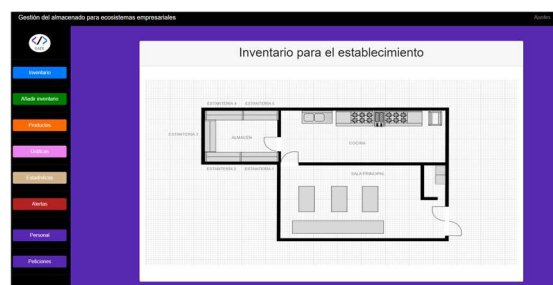


Figura 27: Inicio.

5.9.1.4. Personal

Funcionalidad solamente disponible para Administradores. En esta funcionalidad los usuarios administradores pueden realizar cambio en el rol de otros usuarios del sistema o realizar la eliminación de alguno de estos.

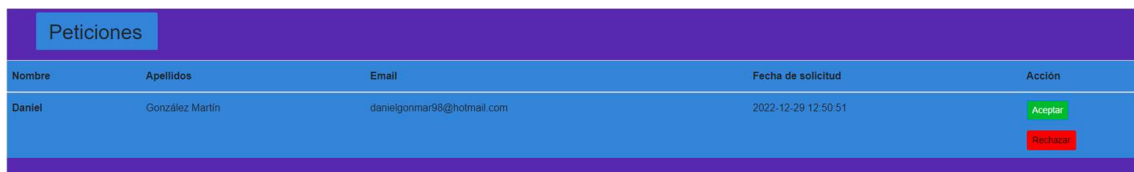


Gestión del almacenado para ecosistemas empresariales						Ajustes
Personal						
Nombre	Apellidos	Email	Última conexión	Rol	Acción	
Daniel	González	danielgonmar98@hotmail.com	null	Usuario	Modificar rol	Eliminar usuario

Figura 28: Personal.

5.9.1.5. Peticiones

Funcionalidad solamente disponible para administradores. En esta funcionalidad los usuarios administradores pueden aceptar o rechazar peticiones pendientes de usuarios que solicitan entrar en el sistema.



Peticiones				
Nombre	Apellidos	Email	Fecha de solicitud	Acción
Daniel	González Martín	danielgonmar98@hotmail.com	2022-12-29 12:50:51	Aceptar Rechazar

Figura 29: Peticiones.

5.9.1.6. Perfil

En esta funcionalidad se muestran a los usuarios la información almacenada en el sistema respecto a su perfil. Se da la opción de realizar cambios en el Nombre y Apellidos y realizar un cambio de contraseña y, un eliminado de la cuenta.



Figura 30: Perfil.

5.9.1.7. Cambiar contraseña

Funcionalidad que permite realizar un cambio de contraseña a los usuarios. Estos deben introducir la antigua contraseña para confirmar que se desea realizar el cambio de contraseña y dos veces la nueva contraseña para asegurar la nueva contraseña que este quiere.

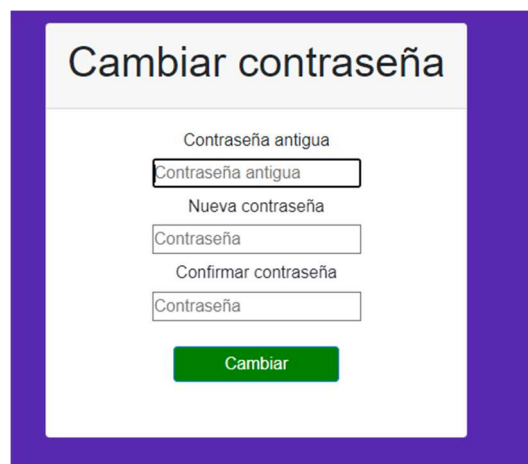
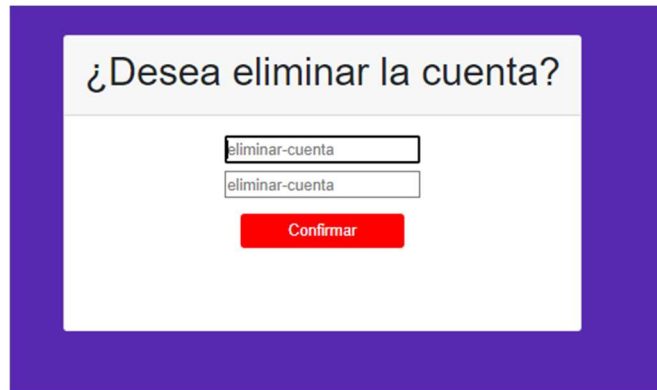


Figura 31: Cambiar contraseña.

5.9.1.8. Eliminar cuenta

Funcionalidad que permite a los usuarios realizar el eliminado de su cuenta del sistema. Se realiza la comprobación de que el usuario desea realizar el eliminado mediante un formulario en el que el usuario debe introducir dos veces eliminar-cuenta para realizar la eliminación.

El formulario está centrado en una pantalla con un fondo morado. El título "¿Desea eliminar la cuenta?" está en un recuadro gris. Debajo, hay dos campos de texto blancos con el texto "eliminar-cuenta" en gris. El segundo campo está deshabilitado. Abajo de los campos hay un botón rojo con el texto "Confirmar" en blanco.

¿Desea eliminar la cuenta?

eliminar-cuenta

eliminar-cuenta

Confirmar

Figura 32: Eliminar cuenta.

5.9.2. Gestión de inventario

En esta parte del sistema se realiza las funciones que tengan que ver con el almacenamiento y búsqueda de inventario almacenado en el sistema.

Se realizan las acciones para realizar el añadido, modificaciones, eliminación y filtrado del inventario almacenado por los Usuarios dentro de la aplicación.

5.9.2.1. Inventario

Se muestra el inventario almacenado en el sistema.

Inventario para el establecimiento								
		Desde:		Hasta:				
		dd/mm/aaaa		dd/mm/aaaa				
<Tipo>	<Nombre>	<Salas>	<Estanteria>	<Fila>	<Posición>	<Última modificación>	<Cantidad>	Filtrar
Bebida	Fanta de Naranja	Almacén	Estanteria 1	Fila 1	Posición 1	2023-01-07 13:37:44	5	Modificar Eliminar
Bebida	Fanta de Naranja	Almacén	Estanteria 2	Fila 1	Posición 1	2023-01-07 13:37:52	5	Modificar Eliminar
Bebida	Fanta de Naranja	Almacén	Estanteria 3	Fila 1	Posición 1	2023-01-07 13:37:59	5	Modificar Eliminar
Bebida	Fanta de Naranja	Almacén	Estanteria 4	Fila 1	Posición 1	2023-01-07 13:38:25	5	Modificar Eliminar

Figura 33: Inventario.

5.9.2.2. Añadir inventario

Funcionalidad que permite realizar el añadido de Inventario en el sistema. Se muestra un formulario donde los usuarios deben introducir la información requerida para realizar el añadido del inventario.

Añadir inventario

Almacén

Bebida

Fanta de Naranja

Estanteria 1

Fila 1

Posición 1

Cantidad:

5

Añadir

+ Añadir producto

Figura 34: Añadir inventario

5.9.2.3. Modificar inventario

Funcionalidad que permite realizar cambios en inventario ya añadido en el sistema. Se ofrece la posibilidad de realizar movimientos en la localización y en la cantidad de este.



Modificar inventario

Sala principal

Bebida

Fanta de Naranja

Cantidad:

5

Modificar

Figura 35: Modificar inventario.

5.9.2.4. Eliminar inventario

Funcionalidad que permite realizar el eliminado de inventario en el sistema.

Inventario eliminado correctamente .

Figura 36: Eliminar inventario.

5.9.2.5. Filtrar

El sistema permite hacer un filtrado en las columnas que los usuarios deseen. Se puede realizar un filtrado de una o varias columnas a la vez y se muestra el inventario que cumpla con las condiciones insertadas.

En el ejemplo de la Figura 37 se realiza un filtrado del inventario que esté registrado en la Estantería 1.



The screenshot shows a web application titled 'Inventario para el establecimiento'. It features a search bar with 'Desde:' and 'Hasta:' date pickers. Below is a row of dropdown filters: '<Tipo>', '<Nombre>', '<Salas>', '<Estantería>', '<Fila>', '<Posición>', '<Última modificación>', and '<Cantidad>'. A 'Filtrar' button is on the right. The table below displays two items, both filtered by 'Estantería 1':

<Tipo>	<Nombre>	<Salas>	<Estantería>	<Fila>	<Posición>	<Última modificación>	<Cantidad>	
Bebida	Fanta de Naranja	Almacén	Estantería 1	Fila 1	Posición 1	2023-01-07 13:37:44	5	Modificar Eliminar
Bebida	Coca cola	Almacén	Estantería 1	Fila 1	Posición 1	2023-01-11 10:58:07	8	Modificar Eliminar

Figura 37: Filtrar.

5.9.2.6. Código QR

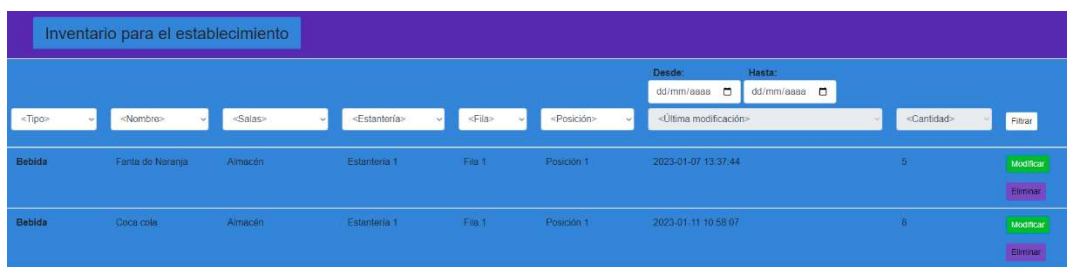
Se aprovecha la funcionalidad del filtrado y se genera códigos QR que son impresos en cada una de las estanterías. Estos códigos realizan el filtrado por Estantería directamente. Cada código QR tiene asignado una estantería a la que filtrar.

En Figura 38 se muestra el código QR generado [20] para el filtrado de la primera Estantería.



Figura 38. Ejemplo de código QR para la Estantería 1.

Una vez accedido a la página de este código, se redirigirá al inventario con el filtrado realizado.



This screenshot is identical to the one in Figure 37, showing the same filtered inventory table with two items under 'Estantería 1'.

<Tipo>	<Nombre>	<Salas>	<Estantería>	<Fila>	<Posición>	<Última modificación>	<Cantidad>	
Bebida	Fanta de Naranja	Almacén	Estantería 1	Fila 1	Posición 1	2023-01-07 13:37:44	5	Modificar Eliminar
Bebida	Coca cola	Almacén	Estantería 1	Fila 1	Posición 1	2023-01-11 10:58:07	8	Modificar Eliminar

Figura 39. Inventario filtrado.

5.9.3. Gestión de productos

En esta parte del sistema se realizan las funciones que tienen que ver con el almacenamiento de los productos almacenado en el sistema.

Se realizan las acciones para hacer el añadido, modificaciones, eliminación del producto almacenado por el Sistema. Para realizar el eliminado de un producto es necesario que no exista inventario almacenado.

5.9.3.1. Productos

Se muestran los productos almacenados en el sistema. Se tienen las funcionalidades de Añadir Producto, Modificar Producto y Eliminar.

Productos disponibles								
Nombre	Tipo	Mínimo	Máximo	Actual	Precio	Alertas automáticas	Estado	Acción
Fanta de Naranja	Bebida	5	20	30	3 €	Si		Modificar producto Eliminar
Coca cola	Bebida	5	20	0	1 €	No		Modificar producto Eliminar
Huevos	Comida	10	20	0	2 €	No		Modificar producto Eliminar
Sillas	Mobiliario	5	20	5	1 €	No		Modificar producto Eliminar

Figura 40: Productos.

5.9.3.2. Añadir producto

Funcionalidad que permite realizar el añadido de producto en el sistema. Se muestra un formulario donde los usuarios deben introducir la información requerida para realizar el añadido del producto.

Añadir producto

Precio en €

Mínimo deseado

Máximo deseado

☐ Activar alertas automáticas

Añadir

Figura 41: Añadir productos.

5.9.3.3. Modificar producto

Funcionalidad que permite realizar cambios en un producto ya añadido en el sistema. Se ofrece la posibilidad de realizar cambios en el precio, límites superiores e inferiores que se recomienda tener del producto y la posibilidad de activar las alertas automáticas.

Formulario de modificación de producto. El título es 'Modificar producto'. Hay un campo de texto con 'Fanta de Naranja'. Debajo es un menú desplegable con 'Bebida'. Luego, 'Precio en €' con un campo de texto que contiene '3,2'. A continuación, 'Mínimo deseado' con un campo de texto que contiene '5'. Luego, 'Máximo deseado' con un campo de texto que contiene '20'. Hay una casilla de verificación marcada con 'Activar alertas automáticas'. En la parte inferior hay un botón azul que dice 'Modificar'.

Figura 42: Modificar producto.

5.9.3.4. Eliminar producto

Funcionalidad que permite realizar el eliminado del producto del sistema.

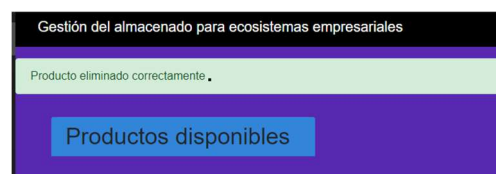


Figura 43: Eliminar producto.

En caso de que exista inventario registrado asociado al producto que se desea eliminar, no se podrá realizar el borrado del producto.



Figura 44: Alerta eliminar producto.

5.9.3.5. Gráficas

Funcionalidad donde se muestran tres gráficas sobre los productos a los usuarios.

Se muestra una gráfica de evolución temporal del producto donde existe una entrada por día en la gráfica para poder ver la evolución de inventario de manera diaria respecto al máximo y al mínimo diario.

Se muestra una gráfica individual del producto donde existe entradas de todos los movimientos que se han hecho en inventario sobre el producto. Se genera una entrada por cada creación, modificación y eliminado de inventario. Se muestra la fecha en la que se ha realizado y quién ha sido la persona que ha realizado dicho movimiento.

Por último, se muestra una gráfica donde se refleja la evolución del precio del producto. Se genera una entrada para la gráfica diariamente para el precio de ese día del producto. Esta gráfica nos permite realizar un seguimiento acerca del precio.



Figura 45: Gráficas producto.

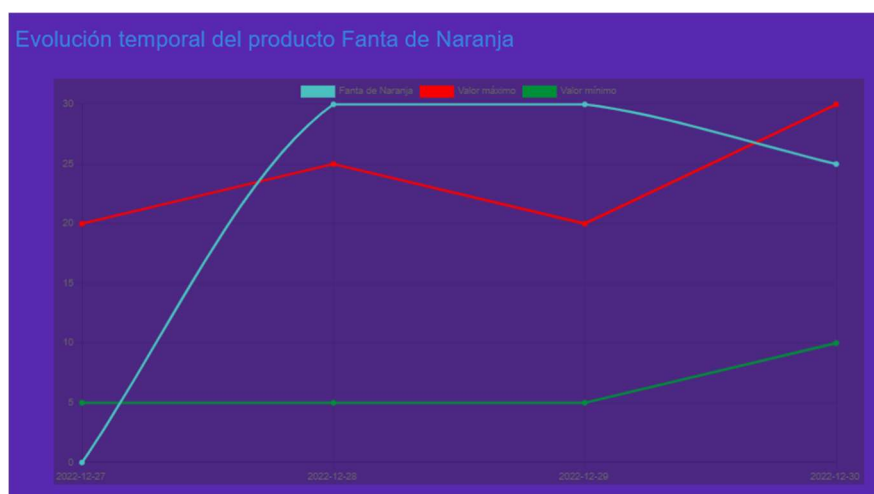


Figura 46: Evolución temporal del stock frente al mínimo y máximo recomendados.

5.9.3.6. Estadísticas

En esta funcionalidad se muestra al usuario estadísticas sobre el producto seleccionado. Se pueden ver ciertas estadísticas acerca de los productos de manera individual como cuando ha sido añadido, tipo de Producto, etc.

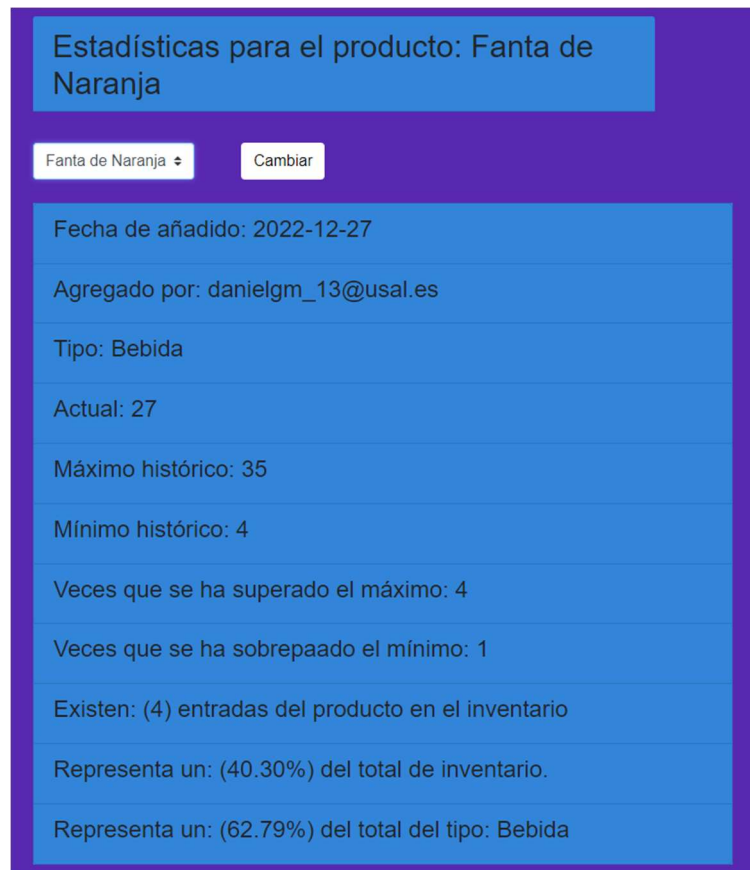


Figura 47: Estadísticas.

5.9.4. Gestión de alertas

En esta funcionalidad se muestran las alertas activas en el sistema. Estas son generadas cuando el total de inventario de cualquier producto esté por encima o por debajo de los límites máximo y mínimos establecidos.

5.9.4.1. Alertas

Tabla con las alertas activas.

Existen las funcionalidades de activar y desactivar las alertas automáticas de todos los productos del sistema y de realizar un aviso mediante correo electrónico por alguna de las alertas activas para los administradores.



Producto	Alerta	Actual	Acción
Prueba	El producto Prueba está por debajo del límite de inventario	0	Avisar administradores
Huevos	El producto Huevos está por debajo del límite de inventario	0	Avisar administradores

Figura 48: Alertas.

5.9.4.2. Activar alertas automáticas

Se activan las alertas automáticas para todos los productos.



Figura 49: Activar alertas automáticas.

5.9.4.3. Desactivar alertas automáticas

Se desactivan las alertas automáticas para todos los productos.



Figura 50: Desactivar alertas automáticas.

6. Conclusiones

Una vez realizadas todas las fases de desarrollo y finalizado el proyecto “Gestión del almacenado para ecosistemas empresariales” se recogen unas conclusiones a cerca del trabajo realizado durante todos estos meses.

En primer lugar, podemos afirmar que se han logrado y cumplido todos los objetivos funcionales y no funcionales definidos al principio del proyecto. La gestión de usuarios representaba un objetivo principal en el sistema y se han implementado las funcionalidades suficientes para satisfacer las necesidades que se presentaban al principio del proyecto. Se ha conseguido desarrollar un sistema completo que nos permite realizar una gestión de los productos almacenados en el ecosistema empresarial propuesto. Se ha realizado un sistema que nos permite realizar un seguimiento de la evolución temporal del producto y ver cómo su inventario va variando a lo largo del tiempo. Otro de los puntos importantes propuesto como parte del diseño del sistema, y que también se ha cumplido, era que este fuera capaz de realizar alertas que nos permitieran conocer y avisar al usuario en circunstancias donde la cantidad del producto generara situaciones de conflicto, que pudieran generar pérdidas o problemas en un futuro.

Durante el desarrollo del proyecto se han mantenido seguimientos frecuentes con los tutores, por lo que se ha realizado una aproximación a lo que en un futuro puede ser un cliente al que se le deberá de realizar un *software* en específico y cumplir los requisitos que este nos especifique.

En cuanto a conclusiones referentes sobre el desarrollo del *software*, se han utilizado gran cantidad de técnicas y herramientas actuales utilizadas en una gran cantidad de empresas. En consecuencia, todo el desarrollo de este proyecto facilita y favorece una entrada al mundo laboral en un futuro próximo.

A lo largo de la creación del proyecto se ha utilizado la experiencia obtenida a lo largo de todos los años utilizando conocimientos adquiridos a lo largo de las asignaturas estudiadas.

En un punto no tan técnico, este trabajo ha sido una gran experiencia que me ha permitido realizar una evolución tanto en lo personal como en lo profesional debido a la gestión realizada de conocimientos y emociones a lo largo de la creación del proyecto.

En definitiva, este trabajo de Fin de Grado ha permitido agrupar todos los conocimientos adquiridos durante todos estos años y sumarle gran cantidad de información nueva para crear un proyecto como este. Todo este aprendizaje será de gran valor para etapas futuras.

Este proyecto ayuda y motiva a seguir aprendiendo de este mundo de la Ingeniería Informática y de en un futuro poder aportar a la sociedad toda la experiencia adquirida con los años.

7. Líneas de trabajo futuras

Adicionalmente, se plantean unas futuras líneas de trabajo donde se podría seguir realizando desarrollo.

Actualmente para realizar algunos cambios de configuración en el sistema es necesario acceder a la base de datos y realizar los cambios en ella, por lo que se propone:

- Realizar la posibilidad que algún Usuario Administrador pueda realizar cambios en la imagen del establecimiento. Esto permitirá que en caso de que existan cambios en el Almacén o Sala principal algún usuario administrador del sistema pueda realizar el cambio.
- Igual ocurre con las configuraciones como el número de estanterías, filas y posiciones. Actualmente no existe manera de realizar cambios en estas configuraciones sin que el Administrador de la máquina en la que está ejecutándose la aplicación realice los cambios. Sería recomendable dar una opción para los Usuarios Administradores para que puedan realizar estos cambios y configurar esto a sus preferencias.

Los tipos de Producto actualmente están definidos como Bebida, Comida y Mobiliario a pesar de que el modelo está preparado para añadir nuevos tipos de Productos se propone:

- Crear una interfaz que proporcione la posibilidad para que algún Usuario Administrador pueda realizar cambios sobre los tipos de Productos del sistema a su gusto.

En cuanto a la gestión de Inventario, el sistema está definido para que solamente se almacene un inventario para un establecimiento por lo que se propondría para una línea de trabajo futura:

- Que cada Usuario, Inventario y Producto tengan asignado un número de Inventario y que cada Usuario solo pueda crear, modificar y realizar movimientos de estas entradas para el Inventario al que pertenece. Esto permitiría que un sistema almacenara varios inventarios en caso de que fuera necesario.

Esto permitiría que grandes cadenas con varios establecimientos pudieran realizar un correcto inventariado en varios de sus establecimientos y no solamente en uno.

8. Referencias

- [1] H. d. España, «Hostelería de España,» [En línea]. Available: <https://www.cehe.es/#:~:text=La%20hostelería%20está%20formada%20por,PIB%20de%20la%20economía%20española..>
- [2] A. W. Services, «Amazon Web Services,» [En línea]. Available: <https://aws.amazon.com/es/>.
- [3] No-IP, «No-IP,» [En línea]. Available: <https://www.noip.com/es-MX>.
- [4] gaeinventario, «gaeinventario,» [En línea]. Available: <https://gaeinventario.ddns.net>.
- [5] Nginx, «Nginx,» [En línea]. Available: <https://www.nginx.com>.
- [6] L. Encrypt, «Let's Encrypt,» [En línea]. Available: <https://letsencrypt.org/es/>.
- [7] Git, «Git,» [En línea]. Available: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>.
- [8] GitHub, «GitHub,» [En línea]. Available: <https://github.com>.
- [9] H. REM, «Herramienta REM,» [En línea]. Available: http://www.lsi.us.es/descargas/descarga_programas.php?id=3.
- [10] V. Paradigm, «Visual Paradigm,» [En línea]. Available: <https://www.visual-paradigm.com>.
- [11] M. Projects, «Microsoft Projects,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>.
- [12] V. S. Code, «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com>.
- [13] NodeJS, «NodeJS,» [En línea]. Available: <https://nodejs.org/en/>.
- [14] NPM, «NPM,» [En línea]. Available: <https://www.npmjs.com>.
- [15] Express, «Express,» [En línea]. Available: <https://expressjs.com/es/>.
- [16] Mongoose, «Moongose,» [En línea].
- [17] MondoDB, «MondoDB,» [En línea]. Available: <https://www.mongodb.com>.
- [18] Nodemailer, «Nodemailer,» [En línea]. Available: <https://nodemailer.com/about/>.
- [19] Forever, «Forever,» [En línea]. Available: <https://www.npmjs.com/package/forever>.
- [20] G. d. QR, «Generador de RQ,» [En línea]. Available: <https://www.codigos-qr.com/generador-de-codigos-qr/>.
- [21] M. Moreno García, «Apuntes de la asignatura "Gestión de proyectos",» (2022).

[22 r. Pressman, «Ingeniería del Software: Un enfoque práctico.,» 2021.

[23 G. Ivan Jacobson, «El proceso unificado de desarrollo de software.,» 2000.