

## MEMORIA FINAL DE PROYECTO

HAPPYPET



**CICLO FORMATIVO DE GRADO SUPERIOR**

**DESARROLLO DE APLICACIONES WEB**

### AUTORES

MARCOS SANTOS GARCÍA-MIGUEL  
ROBERTO DANIEL FLORÍN MIRZA  
RAÚL GALÁN ORELLANA

### TUTOR

ROBERTO GONZÁLEZ

### COORDINADOR

ALBERTO GARAY



## **Licencia**

**Esta obra está bajo una licencia Reconocimiento-Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.**

## **DEDICATORIAS/AGRADECIMIENTOS**

Javier Matute, que se tuvo que ir del grupo, y aun así, nos ha ayudado a resolver alguna incidencia en el proyecto.

## ÍNDICE

INTRODUCCIÓN	5
ALCANCE DEL PROYECTO Y ANÁLISIS PREVIO	6
ESTUDIO DE VIABILIDAD	8
ANÁLISIS	11
DISEÑO	16
IMPLEMENTACIÓN	20
PRUEBAS	25
EXPLOTACIÓN	27
DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN	32
CONCLUSIONES	33
FUENTES	34

# INTRODUCCIÓN

Este documento recoge el trabajo realizado para el **módulo de Proyecto** del CFGS en Desarrollo de Aplicaciones Web.

Este módulo profesional complementa la formación establecida para el resto de los módulos profesionales que integran el título en las funciones de **análisis** del contexto, **diseño** o **desarrollo** del proyecto y organización de la **ejecución**.

El proyecto consiste en el desarrollo de una tienda online de mascotas, diseñada para ser utilizada por usuarios de todas las edades. La tienda ofrece una amplia variedad de productos, desde alimentos hasta complementos, y permite a los usuarios registrarse, consultar, comprar y comentar los distintos productos disponibles en el catálogo. Los administradores tienen privilegios adicionales y son responsables de agregar, actualizar y eliminar productos y usuarios. El objetivo principal del proyecto es proporcionar una experiencia amigable e intuitiva para los usuarios.

# ALCANCE DEL PROYECTO Y ANÁLISIS PREVIO

El propósito de este proyecto es desarrollar una aplicación llamada HappyPet que permita a los usuarios comprar productos para animales de compañía de manera conveniente y satisfactoria. El objetivo general del proyecto es proporcionar una plataforma en línea que ofrezca una amplia gama de productos para mascotas y facilite el proceso de compra a los usuarios.

El alcance del proyecto incluye las siguientes características y funcionalidades:

Catálogo de productos: HappyPet mostrará un catálogo completo de productos para mascotas, incluyendo alimentos, juguetes, accesorios, productos de cuidado y más. Los usuarios podrán navegar por las categorías y explorar los productos disponibles.

Búsqueda y filtros: Los usuarios podrán aplicar filtros para refinar los resultados según la categoría. Esto facilitará la localización y selección de productos deseados.

Detalles del producto: Para cada producto, se mostrará información detallada que incluirá descripción, nombre, precio, categoría y calificaciones/reseñas de otros usuarios.

Carrito de compras: Los usuarios podrán agregar productos al carrito de compras mientras exploran el catálogo. Podrán ajustar las cantidades, eliminar productos.

Gestión de pedidos: Los usuarios podrán realizar un seguimiento de sus pedidos, recibir actualizaciones sobre el estado de envío y acceder al historial de compras. También podrán realizar devoluciones y solicitar asistencia al cliente si es necesario.

El desarrollo de este proyecto se ha llevado a cabo en varias fases: estudio de viabilidad, análisis, diseño, implementación y pruebas, y explotación o ejecución. A continuación se detallan las actividades/tareas/procedimientos de cada una de estas fases.

# ESTUDIO DE VIABILIDAD

En esta fase se considera si el proyecto se puede realizar teniendo en cuenta las circunstancias internas y externas de la empresa, las diferentes soluciones posibles y los recursos de los cuales se dispone.

Para ello se hace una valoración del estado actual del sistema y de los requisitos del cliente, se presentará un estudio de soluciones alternativas y la solución elegida por el cliente.

## 1.1 ***Estado actual del sistema***

En el contexto de HappyPet, no se proporciona información sobre un sistema existente previo. Por lo tanto, no hay un estado actual del sistema para describir. HappyPet es un proyecto nuevo y se está desarrollando desde cero para ofrecer una aplicación de compra de productos para mascotas.

## 1.1 ***Resumen de requisitos del cliente***

Registro de usuarios: Los clientes deben poder registrarse en la aplicación proporcionando información básica como nombre, dirección de correo electrónico y contraseña.

Búsqueda y navegación de productos: Los usuarios deben poder buscar productos para mascotas utilizando filtros como categoría. Además, se debe proporcionar una interfaz intuitiva de navegación para explorar los productos disponibles.

Detalles del producto: Los usuarios deben poder ver información detallada sobre cada producto, incluyendo descripción, precio, imagen, opiniones de otros usuarios, etc.

Carrito de compras: Los usuarios deben poder agregar productos al carrito de compras y realizar modificaciones en términos de cantidad, eliminar productos, etc. Además, se debe calcular el precio total de los productos en el carrito.



Valoraciones y opiniones de productos: Los usuarios deben poder dejar valoraciones y opiniones sobre los productos que han adquirido. Esto permitirá a otros usuarios tomar decisiones informadas al realizar compras.

Interfaz de administración: El administrador del sistema debe tener acceso a una interfaz de administración para gestionar productos, categorías, usuarios, pedidos, etc.

### 1.1 ***Posibles soluciones***

Otras aplicaciones de comercio electrónico: Hay varias plataformas de comercio electrónico populares, como Amazon, eBay, Alibaba, entre otras, que ofrecen una amplia gama de productos para mascotas.

### 1.1 ***Solución elegida***

Aplicación de comercio electrónico.

La aplicación HappyPet se centra exclusivamente en productos para animales, lo que la hace más adecuada para atender las necesidades específicas de los clientes que buscan artículos relacionados con sus mascotas. Al enfocarse en un nicho de mercado, la aplicación puede ofrecer una experiencia más personalizada y adaptada a las necesidades de los amantes de las mascotas.

### 1.1 ***Planificación temporal de las tareas del proyecto HappyPet***

Análisis de requisitos y diseño: En esta etapa se recopilarán y analizarán los requisitos del cliente, se definirá la arquitectura de la aplicación y se diseñará la interfaz de usuario. Se estima que esta etapa llevará aproximadamente 2 semanas.

Desarrollo del back-end: En esta etapa se implementará la lógica de negocio y la gestión de datos de la aplicación. Se estiman 4 semanas para esta etapa.

Desarrollo del front-end: En esta etapa se construirá la interfaz de usuario de la aplicación utilizando tecnologías web como HTML, CSS y JavaScript. Se estima que esta etapa tomará alrededor de 4 semanas.

Integración y pruebas: En esta etapa se realizarán pruebas de integración para asegurar el correcto funcionamiento de todos los componentes de la aplicación. Se estima que esta etapa llevará aproximadamente 2 semanas.

### 1.1 ***Planificación de los recursos a utilizar***

Para solventar los problemas que plantea el proyecto HappyPet es necesario que todo el equipo nos formemos en las nuevas herramientas que vamos a utilizar, entre las nuevas herramientas se encuentra el framework angular, ya que ninguno de nosotros sabemos utilizarlo de forma correcta.

# ANÁLISIS

En esta fase se establecerán los requisitos del sistema:

Registro de usuarios: Los usuarios deben poder registrarse en el sistema proporcionando información básica como nombre, dirección de correo electrónico y contraseña.

Inicio de sesión: Los usuarios registrados deben poder iniciar sesión en el sistema utilizando su dirección de correo electrónico y contraseña.

Catálogo de productos: El sistema debe mostrar un catálogo de productos disponibles para la compra, que incluya información detallada de cada producto, como nombre, descripción, precio.

Búsqueda y filtrado de productos: Los usuarios deben poder buscar productos por categoría.

Carrito de compras: Los usuarios deben poder agregar productos al carrito de compras y gestionar su contenido, incluyendo la cantidad de cada producto seleccionado.

Gestión de pedidos: Los usuarios deben poder ver el historial de sus pedidos realizados, incluyendo detalles como fecha, productos comprados y estado de entrega.

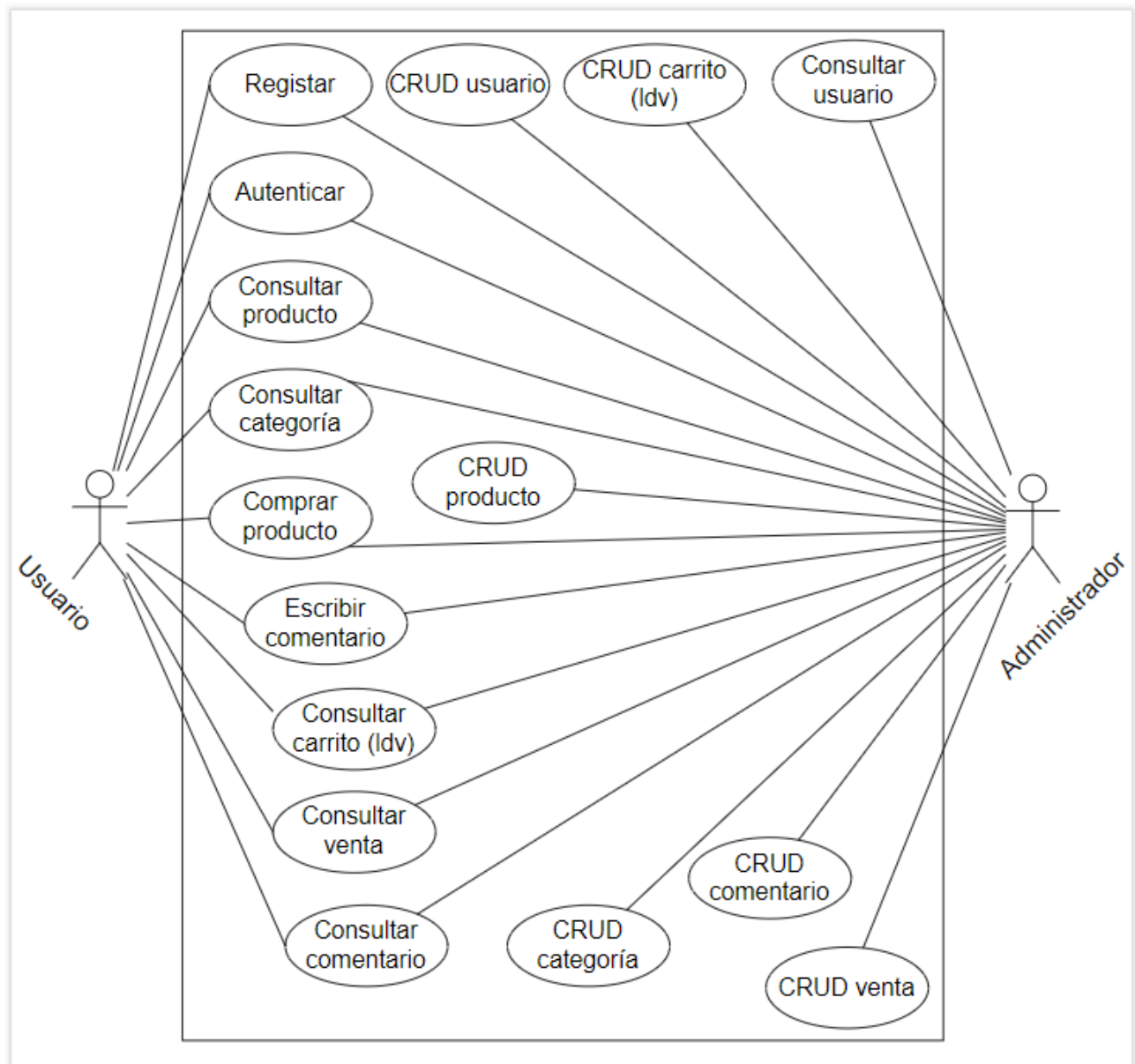
Gestión de cuentas de usuario: Los usuarios deben tener la capacidad de gestionar su información personal, como dirección de envío, datos de pago y preferencias de notificaciones.

Sistema de valoraciones y reseñas: Los usuarios deben poder dejar valoraciones y reseñas de los productos que han comprado, para que otros usuarios puedan tomar decisiones informadas.

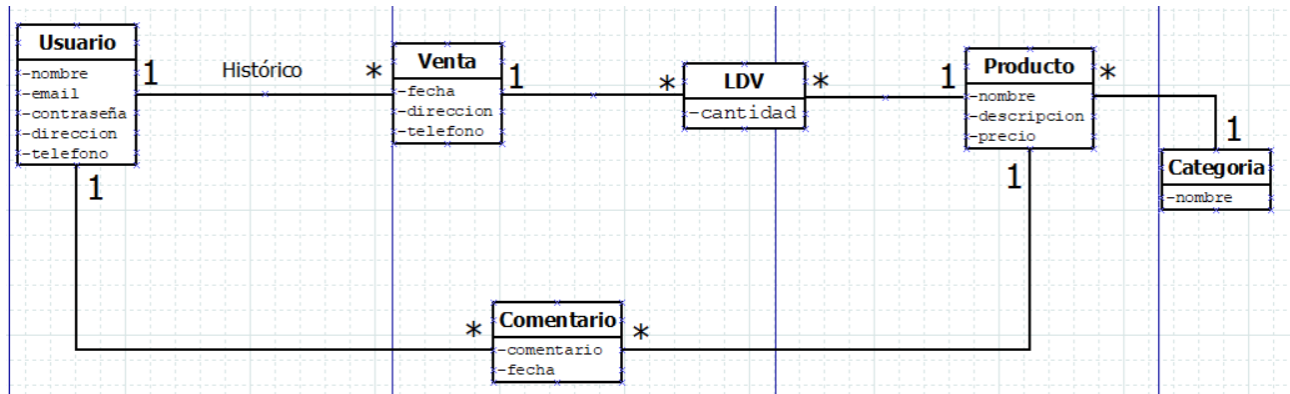
Administración de productos: Los administradores del sistema deben tener la capacidad de agregar, editar y eliminar productos del catálogo, así como actualizar su información y gestionar la disponibilidad.

Administración de categorías: Los administradores deben poder crear, editar y eliminar categorías de productos, para organizar y clasificar los productos en el catálogo.

### 1.1 Diagrama de casos de uso



## 1.2 Modelo de datos



### 1.1 Requisitos funcionales

Son aquellos que determinan qué tareas tiene que hacer el sistema.

Registro de usuarios: Los usuarios deben poder registrarse en el sistema proporcionando información básica.

Inicio de sesión: Los usuarios registrados deben poder iniciar sesión en el sistema.

Búsqueda y filtrado de productos: Los usuarios deben poder buscar productos por categoría.

Visualización de detalles de productos: Los usuarios deben poder ver información detallada de cada producto, como nombre, descripción, precio .

Agregar productos al carrito: Los usuarios deben poder agregar productos al carrito de compras.

Gestión del carrito de compras: Los usuarios deben poder ver el contenido del carrito de compras, editar la cantidad de productos y eliminar productos.

Gestión de pedidos: Los usuarios deben poder ver el historial de sus pedidos realizados, incluyendo detalles como fecha, productos comprados y estado de entrega.

Valoración y reseñas de productos: Los usuarios deben poder dejar valoraciones y reseñas de los productos que han comprado.

Administración de productos: Los administradores deben poder agregar, editar y eliminar productos del catálogo.

Administración de categorías: Los administradores deben poder crear, editar y eliminar categorías de productos.

## 1.2 ***Requisitos no funcionales***

Usabilidad: El sistema debe ser intuitivo y fácil de usar para los usuarios.

Rendimiento: El sistema debe ser rápido y eficiente en la búsqueda de productos, gestión del carrito y proceso de compra.

Adaptabilidad: El sistema debe ser compatible con diferentes dispositivos y navegadores web.

# DISEÑO

En esta fase se realiza una aproximación al diseño tecnológico de la solución.

Registro de usuarios: Se diseñará una página de registro donde los usuarios podrán proporcionar su información básica, como nombre, correo electrónico y contraseña.

Se implementará la validación de los datos ingresados por el usuario para asegurar la integridad de los datos registrados.

Inicio de sesión: Se desarrollará una página de inicio de sesión donde los usuarios podrán ingresar sus credenciales para acceder al sistema.

Se utilizará un mecanismo de autenticación, como tokens o sesiones, para verificar la identidad de los usuarios y permitirles el acceso.

Búsqueda y filtrado de productos: Se implementará una funcionalidad de búsqueda que permita a los usuarios buscar productos por nombre o categoría.

Se diseñará una interfaz de usuario que muestre los resultados de búsqueda y permita a los usuarios aplicar filtros para refinar los resultados.

Visualización de detalles de productos: Se creará una página de detalles de productos donde los usuarios podrán ver información detallada de cada producto, como nombre, descripción, precio y disponibilidad.

Se incluirán imágenes de los productos para una mejor visualización.

Agregar productos al carrito: Se diseñará una funcionalidad que permita a los usuarios agregar productos al carrito de compras desde la página de detalles de productos.

Se implementará una gestión del carrito en el lado del cliente para almacenar los productos seleccionados.



Gestión del carrito de compras: Se desarrollará una página de carrito de compras que mostrará los productos seleccionados por el usuario.

Se permitirá al usuario editar la cantidad de productos y eliminar productos del carrito.

Gestión de pedidos: Se creará una funcionalidad que permita a los usuarios ver el historial de sus pedidos realizados.

Se mostrarán detalles como la fecha del pedido, los productos comprados y el estado de entrega.

Valoración y reseñas de productos:

Se implementará una funcionalidad que permita a los usuarios dejar valoraciones y reseñas de los productos que han comprado.

Se mostrarán las valoraciones y reseñas en la página de detalles de productos para ayudar a otros usuarios en su decisión de compra.

Administración de productos: Se desarrollará una interfaz de administración para que los administradores puedan agregar, editar y eliminar productos del catálogo.

Se implementarán formularios y validaciones para garantizar la integridad de los datos ingresados.

Administración de categorías: Se diseñará una funcionalidad que permita a los administradores crear, editar y eliminar categorías de productos.

Se proporcionarán opciones para asignar productos a categorías.

### 1.1 ***Estructura de la aplicación***

La estructura de la aplicación HappyPet se basa en una aplicación web de comercio electrónico. La aplicación constará de los siguientes componentes principales:

Front-end (Cliente): Angular

Back-end (Servidor): SpringBoot

### 1.1 ***Componentes del sistema / arquitectura de red***

#### Servidor web:

El servidor web es el componente central de la arquitectura, encargado de recibir las solicitudes de los clientes y procesarlas para generar las respuestas correspondientes.

Estará implementado utilizando un framework de back-end como Spring Boot.

Será responsable de la lógica de negocio, la comunicación con la base de datos y la integración con otros servicios externos.

#### Base de datos:

La base de datos es el componente encargado de almacenar y gestionar los datos de la aplicación.

Se utilizará una base de datos relacional como MySQL

#### Conexión de red:

La arquitectura de red implica la conexión de los diferentes componentes del sistema para permitir la comunicación entre ellos.

El servidor web estará conectado a la base de datos a través de una conexión de red para enviar y recibir datos.

Los clientes web se conectarán al servidor web a través de Internet utilizando protocolos estándar como HTTP o HTTPS.

### 1.1 ***Herramientas y tecnologías utilizadas.***

XAMPP se ha utilizado para proporcionar el entorno de desarrollo local, Angular ha sido utilizado para el desarrollo del front-end de la aplicación, Spring Boot ha sido utilizado para el desarrollo del back-end, y MySQL ha sido utilizado como el sistema de gestión de bases de datos. Estas herramientas y tecnologías combinadas han permitido la creación de una aplicación web completa y funcional.

# IMPLEMENTACIÓN

Desarrollo de la página web: Se implementó la interfaz de usuario de la aplicación utilizando el framework Angular. Se crearon los componentes, servicios y enrutamientos necesarios para permitir la navegación y la interacción con los usuarios. Se utilizó HTML, CSS y TypeScript para definir la estructura, el diseño y la lógica de la página web.

Creación de la base de datos: Se diseñó el esquema de la base de datos utilizando MySQL Workbench. Se crearon las tablas necesarias para almacenar la información de los productos, categorías y usuarios. Se definieron las relaciones entre las tablas y se establecieron las restricciones de integridad necesarias.

Carga de datos: Se realizó la carga inicial de datos en la base de datos. Se insertaron registros de ejemplo para los productos, categorías y usuarios, con el fin de tener datos para probar el funcionamiento de la aplicación.

Configuración del entorno: Se configuraron los archivos de configuración necesarios para que la aplicación funcione correctamente. Se establecieron las conexiones con la base de datos, se configuraron las rutas y permisos de acceso, y se establecieron las variables de entorno necesarias para la ejecución del servidor.

Implementación de funcionalidades: Se implementaron las funcionalidades específicas de la aplicación, como la visualización y búsqueda de productos, la gestión de categorías, el registro y autenticación de usuarios, y la realización de compras. Se utilizaron los servicios y métodos proporcionados por Angular y Spring Boot para llevar a cabo estas funcionalidades.

Pruebas y depuración: Se realizaron pruebas exhaustivas de todas las funcionalidades de la aplicación para garantizar su correcto funcionamiento. Se corrigieron los errores y se depuraron las partes problemáticas del código.

### 1.1 ***Implementación del modelo de datos***

#### Tabla "producto":

Columnas: idProducto (clave primaria), nombre, precio, descripción, idCategoría (clave foránea).

Esta tabla almacena la información de los productos disponibles en la aplicación, como su nombre, precio, descripción y la categoría a la que pertenecen.

#### Tabla "categoría":

Columnas: idCategoría (clave primaria), nombre.

Esta tabla contiene las categorías de los productos, como "Comida para perros", "Juguetes para gatos", etc. Cada producto está asociado a una categoría a través de su campo "idCategoría".

#### Tabla "usuario":

Columnas: idUsuario (clave primaria), nombre, correo, contraseña, dirección, teléfono.

Esta tabla almacena la información de los usuarios de la aplicación, como su nombre, correo electrónico, contraseña, dirección y número de teléfono.

#### Tabla "comentario":

Columnas: idComentario (clave primaria), texto, producto\_id (clave foránea hacia la tabla "producto"), usuario\_id (clave foránea hacia la tabla "usuario").

Esta tabla permite almacenar los comentarios realizados por los usuarios sobre los productos. Cada comentario está asociado a un producto y a un usuario a través de los campos "producto\_id" y "usuario\_id".

#### Tabla "ldv" (línea de venta):

Columnas: idLDV (clave primaria), cantidad, product\_id (clave foránea hacia la tabla "producto"), venta\_id (clave foránea hacia la tabla "venta").

Esta tabla representa las líneas de venta, que contienen la información detallada de los productos vendidos en cada transacción. Cada línea de venta está relacionada con un producto y una venta a través de los campos "product\_id" y "venta\_id".

#### Tabla "venta":

Columnas: idVenta (clave primaria), dirección, fecha, teléfono, historico\_id (clave foránea hacia la tabla "histórico").

Esta tabla registra la información de las ventas realizadas en la aplicación, incluyendo la dirección de entrega, la fecha de la venta, el número de teléfono del cliente y el historial asociado a la venta.

### 1.1 ***Carga de datos***

Hemos añadido los datos a mano para probar los servicios con la aplicación postman y más adelante lo hemos añadido desde la parte cliente con el angular

## 1.1 Configuraciones realizadas en el sistema

```
happypet/pom.xml X
19 <groupId>org.galan.spring.pap2023</groupId>
20 <artifactId>pap</artifactId>
21 <version>0.0.1-SNAPSHOT</version>
22 <name>PAP_2023</name>
23 <description>PAP del año 2023</description>
24 <properties>
25 <java.version>17</java.version>
26 </properties>
27 <dependencies>
28 <dependency>
29 <groupId>org.springframework.boot</groupId>
30 <artifactId>spring-boot-starter-data-jpa</artifactId>
31 </dependency>
32
33 <dependency>
34 <groupId>org.springframework.boot</groupId>
35 <artifactId>spring-boot-starter-thymeleaf</artifactId>
36 </dependency>
37
38 <dependency>
39 <groupId>org.springframework.boot</groupId>
40 <artifactId>spring-boot-starter-web</artifactId>
41 </dependency>
42
43 <dependency>
44 <groupId>com.mysql</groupId>
45 <artifactId>mysql-connector-j</artifactId>
46 <scope>runtime</scope>
47 </dependency>
48
49 <dependency>
50 <groupId>org.springframework.boot</groupId>
51 <artifactId>spring-boot-starter-test</artifactId>
52 <scope>test</scope>
53 </dependency>
54
55 <dependency>
56 <groupId>org.springframework.boot</groupId>
57 <artifactId>spring-boot-starter-security</artifactId>
58 </dependency>
59
60 <dependency>
61 <groupId>commons-io</groupId>
62 <artifactId>commons-io</artifactId>
63 <version>2.11.0</version>
64 </dependency>
65
66 <dependency>
67 <groupId>org.projectlombok</groupId>
68 <artifactId>lombok</artifactId>
69 <scope>provided</scope>
70 </dependency>
71 <dependency>
72 <groupId>io.jsonwebtoken</groupId>
73 <artifactId>jjwt-api</artifactId>
74 <version>0.11.2</version>
75 </dependency>
76 </dependencies>
```

## 1.1 ***Implementaciones de código realizadas***

### Front-end (Angular):

En el directorio src/app, se encuentran los diferentes componentes, servicios y directivas desarrollados para la interfaz de usuario.

El archivo app-routing.module.ts contiene la configuración de las rutas de navegación en la aplicación.

En los componentes, se ha utilizado la sintaxis de enlace de datos (`{{ }}`) para mostrar dinámicamente los datos en las vistas.

Se han aplicado estilos utilizando CSS y se han utilizado clases de Bootstrap para la apariencia y el diseño de la aplicación.

### Back-end (Spring Boot):

En el directorio src/main/java, se encuentra la estructura del proyecto Spring Boot con los controladores, servicios, modelos y repositorios.

Los controladores (`@RestController`) manejan las solicitudes HTTP y definen los puntos finales de la API.

Los servicios implementan la lógica de negocio y se comunican con los repositorios para realizar operaciones en la base de datos.

Los modelos representan las entidades de la base de datos y se utilizan para la serialización y deserialización de datos.

Los repositorios proporcionan métodos predefinidos para realizar operaciones de consulta y manipulación de datos en la base de datos.

### Base de datos (MySQL):

En las tablas relacionales, se han establecido las claves primarias y las claves externas para mantener la integridad referencial.



# PRUEBAS

Son muchas las pruebas que pueden realizarse en un proyecto para eliminar los posibles errores y garantizar su correcto funcionamiento. Los casos de prueba establecen las condiciones/variables que permitirán determinar si los requisitos establecidos se cumplen o no.

A continuación se detallan algunos de los casos de prueba que se ejecutarán para comprobar la correcta construcción de este proyecto.

## 1.1 *Casos de pruebas*

12/06/23:

12/06/23 Marcos Santos Pruebas:

Caso de prueba:

Prueba CRUD usuarios y administrados

Identificador del caso de prueba (nombre único):

prueba#1

Descripción:

Se probará el CRUD del usuario administrador y de los demás usuarios

Breve explicación sobre el objetivo del caso de prueba:

El objetivo es comprobar que funcionan todos los requisitos establecidos

Condiciones de ejecución:

Que el servidor este en ejecución.

Entrada:

I.E.S. Rey Fernando VI

Curso: 2022/2023

Datos del carrito y venta, y datos de todos los edits.

Resultado esperado:

- Valor esperado para el correcto funcionamiento del proyecto: Esperamos que todo funcione correctamente.
- Resultado obtenido: Todo ha salido bien menos al realizar un get de los comentarios de un producto.
- Valor de salida al ejecutar el caso de prueba.: Error
- Evaluación: En general buena.
- Comparación del valor esperado y obtenido: En general el resultado esperado es positivo ya que el error tiene fácil solución.
- 
- 12/06/23 Marcos Santos Pruebas:
- Caso de prueba.
- Evaluación: Negativa.
- Comparación del valor esperado y obtenido: Esperamos un valor positivo y hemos obtenido uno negativo.
- Posible causa de error: En el servicio de comentarios al filtrar comentarios por productos o en el angular.
- Posible corrección: Arreglar el servicio y revisar el angular, además revisar el restController con postman.
- Áreas afectadas: El cliente al filtrar los comentarios por producto.
- Detallar qué áreas se verán afectados al implementar la corrección: masProductoComponent.html y productoService.

# EXPLOTACIÓN

La implantación es la fase más crítica del proyecto ya que el sistema entra en producción, es decir opera en un entorno real, con usuarios reales.

## 1.1 ***Planificación***

### Preparación del entorno de desarrollo:

- Configurar el servidor de aplicaciones.
- Instalar y configurar la base de datos.
- Instalar las herramientas necesarias, como XAMPP, Angular y Spring Boot.

### Desarrollo e implementación de la aplicación:

- Desarrollar los módulos y funcionalidades de la aplicación, siguiendo las especificaciones y requisitos establecidos.
- Realizar pruebas unitarias y de integración para garantizar el correcto funcionamiento de la aplicación.
- Realizar ajustes y correcciones necesarias según los resultados de las pruebas.

### Carga de datos:

- Crear una base de datos de prueba con datos de ejemplo.

Cargar los datos de productos, categorías y usuarios en la base de datos.

Capacitación y formación del personal:

### Implementación en producción:

- Configurar el entorno de producción, incluyendo el despliegue del servidor de aplicaciones y la configuración de la base de datos.
- Realizar pruebas finales en el entorno de producción para garantizar su estabilidad y rendimiento.

## 1.1 ***Preparación para el cambio***

### Permisos y autorizaciones:

- Identificar los roles y permisos necesarios para cada usuario dentro de la aplicación.
- Establecer políticas de acceso y seguridad para proteger los datos y funcionalidades sensibles.
- Configurar los niveles de autorización y asignar los permisos correspondientes a cada usuario o grupo de usuarios.

### Procedimientos de actuación:

- Documentar y establecer procedimientos claros para la ejecución de las actividades relacionadas con la aplicación.
- Definir los flujos de trabajo y los pasos necesarios para llevar a cabo tareas comunes, como la gestión de pedidos o el registro de usuarios.

## 1.1 ***Manual de usuario***

### 1. Introducción

HappyPet es una aplicación web diseñada para facilitar la compra de productos para mascotas. Este manual del usuario proporciona una guía detallada sobre cómo utilizar la aplicación y aprovechar al máximo sus funcionalidades.

### 2. Registro e inicio de sesión

Para acceder a HappyPet, visita nuestro sitio web y haz clic en "Registrarse" para crear una cuenta.

Proporciona la información requerida en el formulario de registro y haz clic en "Registrarse". Después de registrarte, inicia sesión con tu correo electrónico y contraseña en la página de inicio de sesión.

### 3. Explorar categorías y productos

En la página de inicio, encontrarás una lista de categorías de productos. Haz clic en una categoría para ver los productos relacionados.

Dentro de una categoría, podrás ver una lista de productos con su nombre, imagen y precio. Haz clic en un producto para obtener más detalles, como descripción, características y reseñas de otros usuarios.

### 4. Agregar productos al carrito

Cuando encuentres un producto que deseas comprar, haz clic en el botón "Agregar al carrito".

Selecciona la cantidad deseada y el producto se añadirá al carrito de compras.

Puedes repetir este proceso para agregar más productos al carrito.

### 5. Realizar el pedido

Para realizar el pedido, haz clic en el botón del carrito en la parte superior de la página para acceder al resumen del carrito de compras.

Revisa los productos agregados al carrito, las cantidades y los precios totales.

Si es necesario, puedes eliminar productos o ajustar las cantidades en esta página.

Cuando estés listo, haz clic en el botón "Realizar pedido" y sigue las instrucciones para completar.

#### 1.1 ***Implantación propiamente dicha***

Accedemos a la aplicación del proyecto.

-Proporcionamos la URL o el enlace a la página web.

-Verificamos que la aplicación esté en funcionamiento y accesible.

Realizamos una prueba de funcionalidad.

- Comprobamos que todas las funcionalidades principales del sistema están disponibles y funcionando correctamente.
- Realizamos acciones típicas del usuario y verificamos que los resultados sean los esperados.
- Verificamos la conexión con la base de datos.
- Comprobamos que el sistema puede acceder y utilizar la base de datos correctamente.
- Realizamos una acción que involucre la lectura o escritura de datos en la base de datos y verificamos que se realice correctamente.

### 1.1 ***Pruebas de implantación***

#### Pruebas de accesibilidad:

Se verificó que los usuarios pudieran acceder al sistema utilizando los dispositivos y navegadores recomendados.

Se comprobó que no existieran restricciones de acceso y que los usuarios tuvieran los permisos adecuados.

#### Pruebas de funcionalidad:

Se verificó que todas las funcionalidades del sistema estuvieran disponibles y se comportaran según lo esperado.

Se realizaron pruebas exhaustivas en cada módulo o componente del sistema para identificar posibles errores o comportamientos inesperados.

#### Pruebas de rendimiento:

Se evaluó el rendimiento del sistema bajo condiciones de carga y tráfico simulado para verificar su capacidad de respuesta y tiempos de respuesta.

Se monitoreó el consumo de recursos (CPU, memoria, ancho de banda) para identificar posibles cuellos de botella o problemas de rendimiento.

Pruebas de seguridad:

Se evaluó la seguridad del sistema, incluyendo el manejo de sesiones, autenticación y autorización de usuarios.

Se verificó que no existieran vulnerabilidades conocidas y se implementaron medidas de seguridad apropiadas.

Pruebas de integración:

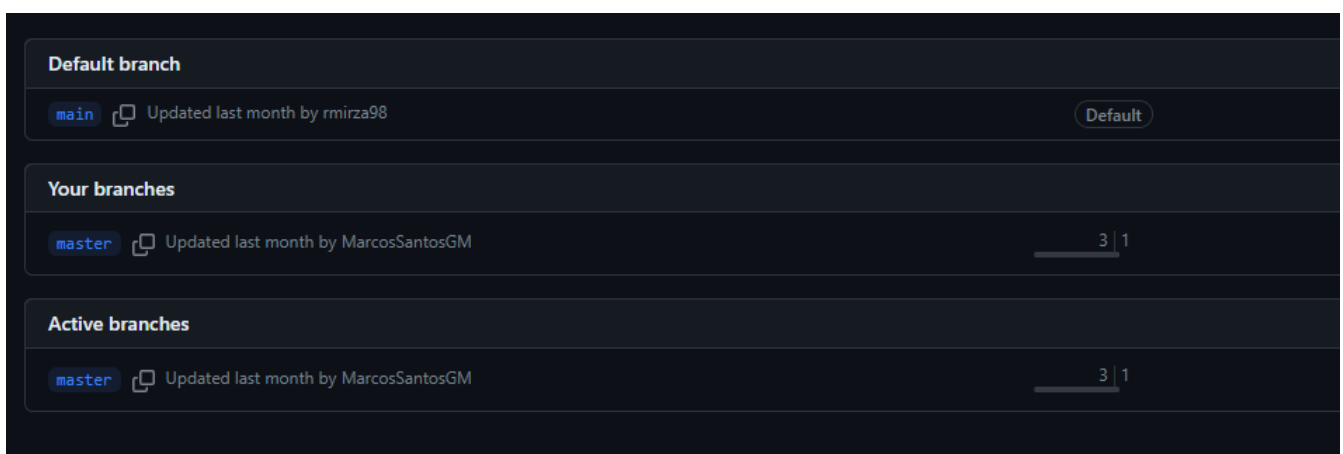
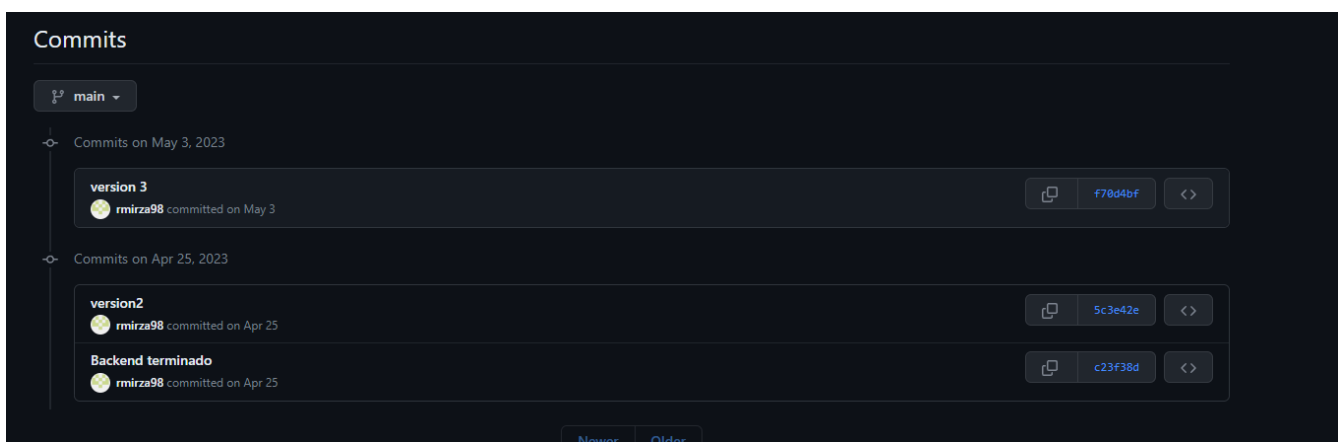
Se comprobó la correcta integración del sistema con otros sistemas o servicios utilizados por el cliente.

Se verificó que los datos se intercambiaran correctamente y que no se produjeran conflictos o errores en la integración.

# DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN

A lo largo del ciclo de vida del proyecto se producirán cambios e incidencias que deberán controlarse y registrarse.

Hemos utilizado GitHub para el control de versiones con varias ramas.





# CONCLUSIONES

Durante el desarrollo del proyecto "HappyPet", se ha logrado implementar una aplicación que permite comprar productos para animales. Se han cumplido los objetivos establecidos, como la creación de las funcionalidades de registro de usuarios, gestión de categorías y productos, así como la visualización y compra de productos.

Personalmente, consideramos que el proyecto ha sido exitoso. Se ha logrado desarrollar una aplicación funcional que cumple con los requisitos establecidos por el cliente. La interfaz de usuario es intuitiva y permite a los usuarios navegar fácilmente por la tienda, agregar productos al carrito y realizar compras.

En cuanto a las posibles ampliaciones, una de ellas sería mejorar el sistema de inicio de sesión implementando JWT (JSON Web Tokens) para proporcionar una autenticación más segura y permitir la gestión de sesiones de usuario. Esto mejoraría la seguridad del sistema y brindaría una experiencia de usuario más fluida. Además, crear una gestión de errores.

Además, se podrían agregar funcionalidades adicionales, como la implementación de un panel de administración para gestionar métodos de pago y envío. Estas ampliaciones podrían enriquecer la aplicación y brindar más opciones y comodidades a los usuarios.

# FUENTES

## 1.1 **Legislación**

DAW

Enseñanzas mínimas: Real Decreto 686/2010, de 20 de mayo (BOE 12/06/2010)

[http://pdf/IFCS03/titulo/RD20100686\\_TS\\_Desarrollo\\_Aplicaciones\\_Web.pdf](http://pdf/IFCS03/titulo/RD20100686_TS_Desarrollo_Aplicaciones_Web.pdf)

Currículo: [Decreto 1/2011, de 13 de enero \(BOCM 31/01/2011\)](#)

[http://pdf/IFCS03/curriculo/D20110001\\_TS\\_Desarrollo\\_Aplicaciones\\_Web.pdf](http://pdf/IFCS03/curriculo/D20110001_TS_Desarrollo_Aplicaciones_Web.pdf)

Definición de procedimientos de control y evaluación:

- <http://www.xperta.es/es/descripcion.asp>
- <http://www.xperta.es/es/aquienvadirigido.asp>
- <http://churriwifi.wordpress.com/2010/04/10/gestion-de-incidencias/>
- [http://es.wikipedia.org/wiki/Control\\_de\\_versiones](http://es.wikipedia.org/wiki/Control_de_versiones)
- 

## 1.1 **Bibliografía**

- 1 <https://angular.io/docs>
- 2 <https://spring.io/guides/gs/rest-service/>
- 3 <https://getbootstrap.com/docs/5.3/getting-started/introduction/>