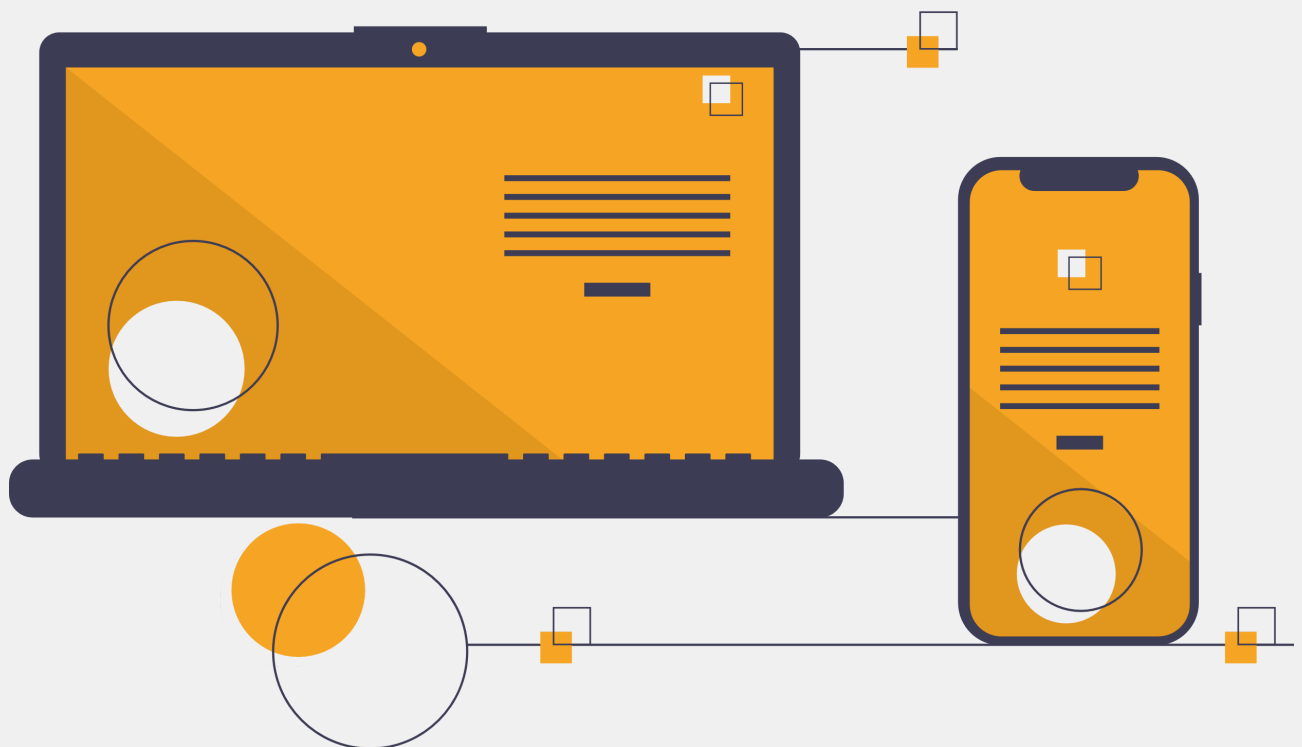


Cafeteria Mallorca

Documentacion

Trabajo Realizado por: Javier Calderón Garrote
Tutor Asignado : Sergio Elvira Perez
Curso: 22/24



Abstract

El objetivo principal de este proyecto es desarrollar una tienda online para ampliar las ventas de la tienda física de embutidos y jamones que posee mi familia en su restaurante. En un contexto de auge del comercio electrónico, esta iniciativa busca no solo aumentar las ventas, sino también alcanzar nuevos mercados y mejorar la accesibilidad de los productos.

Para ello, he utilizado tecnologías modernas y robustas. Angular ha sido elegido para el desarrollo del frontend por su capacidad para crear interfaces de usuario dinámicas y responsivas, asegurando una experiencia atractiva y fácil de usar. En el backend, he empleado Spring Boot, que facilita el desarrollo de aplicaciones Java, creando una API REST eficiente para la interacción entre frontend y backend. PostgreSQL ha sido la base de datos seleccionada por su fiabilidad y rendimiento.

Herramientas adicionales como JPA para la gestión de la persistencia de datos, Lombok para reducir el código repetitivo, y DevTools para el desarrollo ágil, han sido integradas para optimizar el proceso de desarrollo. Además, se ha implementado un sistema de pagos seguro utilizando la API de PayPal, conocida y confiable para los usuarios.

En resumen, este proyecto utiliza tecnologías avanzadas para ofrecer una solución completa y eficiente que digitaliza el negocio familiar y lo posiciona para un crecimiento sostenible en el comercio electrónico.

Abstract

The primary goal of this project is to develop an online store to expand sales of my family's physical store, which sells cured meats and hams within their restaurant. In the context of booming e-commerce, this initiative aims not only to increase sales but also to reach new markets and enhance product accessibility.

To achieve this, I have utilized modern and robust technologies. Angular was chosen for frontend development due to its ability to create dynamic and responsive user interfaces, ensuring an attractive and user-friendly experience. For the backend, I employed Spring Boot, which simplifies the development of Java applications, creating an efficient REST API for interaction between frontend and backend. PostgreSQL was selected as the database for its reliability and performance.

Additional tools such as JPA for data persistence management, Lombok to reduce boilerplate code, and DevTools for agile development have been integrated to optimize the development process. Moreover, a secure payment system using the PayPal API, known and trusted by users, has been implemented.

In summary, this project leverages advanced technologies to provide a comprehensive and efficient solution that digitizes the family business and positions it for sustainable growth in e-commerce.

ÍNDICE

Abstract	2
1. Justificación del Proyecto	6
1. Expansión del Mercado	6
2. Adaptación a Nuevas Tendencias de Consumo	6
3. Diversificación de Canales de Venta	6
4. Mejora de la Competitividad	7
5. Facilitación de la Gestión de Pedidos y Pagos	7
7. Flexibilidad y Escalabilidad	7
8. Experiencia Personal en Tecnologías Relevantes	7
2. Introducción	8
Selección de Tecnologías	8
3. Objetivos del Proyecto de Tienda Online	11
Objetivo General	11
Objetivos Específicos	11
4.Desarrollo	14
Organización y Control de Versiones en el Proyecto de Tienda Online	14
Diseño y Desarrollo de la Interfaz de Usuario	16
Documentación del Sistema de Autenticación y Seguridad con JWT	18
Tipos de Usuario en la Aplicación	22
Pantalla Principal de Cafetería Mallorca: Un Vistazo Detallado	25
Página de Carrito de Compra	28
Página del CRUD de Productos	30
Página del CRUD de Categorías:	33
5. Conclusion	36
6. Bibliografía	37
7. Mejoras Futuras	38
8. Problemas Encontrados	39

1. Justificación del Proyecto

La realización de este proyecto de desarrollo de una tienda online para la venta de embutidos y jamones del restaurante familiar se justifica por varias razones clave que subrayan su importancia y necesidad en el contexto actual:

A. Expansión del Mercado

El principal motivo para desarrollar esta tienda online es la oportunidad de expandir el alcance del negocio familiar más allá de su ubicación física. El restaurante cuenta con una clientela local, pero la venta online permite acceder a un mercado mucho más amplio, incluyendo clientes de otras ciudades e incluso de otros países. Esta expansión es esencial para aumentar las ventas y asegurar un crecimiento sostenido del negocio.

B. Adaptación a Nuevas Tendencias de Consumo

En los últimos años, ha habido un notable incremento en las compras online, impulsado en parte por la conveniencia y seguridad que ofrece este canal, especialmente en tiempos de restricciones sanitarias. Adaptarse a estas nuevas tendencias de consumo es crucial para cualquier negocio que desee mantenerse competitivo y relevante en el mercado actual.

C. Diversificación de Canales de Venta

Contar con una tienda online no solo diversifica los canales de venta, sino que también mitiga el riesgo asociado a depender exclusivamente del comercio físico. Este enfoque multicanal permite que el negocio siga generando ingresos incluso en situaciones donde la tienda física podría estar limitada, como en casos de pandemias o desastres naturales.

D. Mejora de la Competitividad

El mercado de embutidos y jamones es altamente competitivo. Tener una presencia online robusta no solo mejora la visibilidad del negocio, sino que también lo posiciona mejor frente a la competencia. Una tienda online profesional y bien diseñada puede atraer a nuevos clientes y fidelizar a los existentes, aumentando así la competitividad del negocio en el mercado.

E. Facilitación de la Gestión de Pedidos y Pagos

Implementar un sistema online de gestión de pedidos y pagos simplifica significativamente la operativa del negocio. La automatización de estos procesos reduce el riesgo de errores, mejora la eficiencia y ofrece una experiencia más cómoda tanto para los clientes como para el personal del restaurante. Integrar métodos de pago como PayPal también añade una capa de seguridad y confianza en las transacciones.

F. Flexibilidad y Escalabilidad

El desarrollo de una plataforma online ofrece una flexibilidad que no se puede lograr con una tienda física. Las actualizaciones y mejoras pueden implementarse rápidamente sin interrumpir las operaciones. Además, la tienda online es escalable, lo que significa que puede crecer junto con el negocio, adaptándose a nuevas necesidades y demandas sin requerir una inversión significativa en infraestructura física.

G. Experiencia Personal en Tecnologías Relevantes

Este proyecto también se justifica desde una perspectiva personal y profesional. Durante mis prácticas, adquirí conocimientos y habilidades en tecnologías como Angular y Spring Boot, que deseo aplicar en un proyecto real. Esta experiencia no solo me permite consolidar mis habilidades, sino también contribuir significativamente al éxito del negocio familiar.

2. Introducción

En la era digital actual, la transformación hacia el comercio electrónico es crucial para los negocios que buscan expandir su alcance y mejorar su accesibilidad. Este Trabajo de Fin de Grado (TFG) surge de la necesidad de digitalizar un negocio familiar que integra un restaurante con una pequeña tienda de productos. La meta es expandir significativamente el mercado a través de internet, utilizando tecnologías avanzadas que no solo permitan mejorar la interacción con los clientes, sino también optimizar la gestión interna del negocio. Este proyecto se enfoca en la creación de una plataforma online que integre de manera eficaz el restaurante y la tienda, facilitando un acceso más amplio a nuestros productos y servicios.

Selección de Tecnologías

La elección de las tecnologías para este proyecto fue un proceso deliberado y estratégico, orientado a maximizar la eficiencia y la experiencia del usuario. Angular, Spring Boot y PostgreSQL fueron seleccionados como los pilares tecnológicos de este proyecto debido a sus robustas capacidades y su compatibilidad comprobada en el desarrollo de aplicaciones web complejas.

Angular

fue elegido para el desarrollo del front-end por su potente entorno basado en componentes y su habilidad para construir interfaces ricas e interactivas utilizando HTML, CSS y TypeScript. Angular es reconocido por facilitar la construcción de aplicaciones de página única (SPA), que ofrecen experiencias fluidas y dinámicas a los usuarios.



Spring Boot,

utilizado para el back-end, destaca por su configuración automática y su amplia gama de funcionalidades, que permiten un desarrollo rápido y sin complicaciones. Integrado con herramientas como Spring Data JPA para la persistencia de datos y Spring Web para la creación de servicios web, Spring Boot facilita un manejo eficiente de las operaciones del servidor y la integración con otros sistemas y tecnologías.

Un avance significativo en este proyecto fue la creación de una **API REST** en el back-end utilizando Spring Boot. Esta API es crucial en la arquitectura de la aplicación al permitir una comunicación efectiva y estructurada entre el front-end y el back-end. Las operaciones críticas como la gestión de pedidos, usuarios y transacciones son manejadas a través de esta API, asegurando interacciones fluidas y seguras. La implementación de una API REST ha sido decisiva para mejorar la escalabilidad y mantenibilidad de la plataforma, facilitando futuras actualizaciones y adaptaciones sin interrumpir el funcionamiento existente.

La utilización de **Lombok** en Spring Boot ha optimizado significativamente el proceso de desarrollo. Lombok permite reducir el código boilerplate necesario para definir modelos y métodos en Java, como getters, setters y constructores. Esta herramienta ha mejorado la legibilidad del código y ha permitido a los desarrolladores concentrarse en implementar la lógica de negocio compleja en lugar de preocuparse por la verbosidad del código.



PostgreSQL

fue seleccionado como el sistema de gestión de bases de datos relacional debido a su robustez, escalabilidad y licencia abierta. Su uso en conjunto con **pgAdmin 4**, una herramienta de gestión de bases de datos, simplifica las tareas de administración, configuración y mantenimiento, asegurando que los datos del



negocio se manejen de manera segura y eficiente.

Desafíos y Superación

El camino hacia la finalización de este proyecto presentó varios desafíos. Inicialmente, el vasto alcance y la complejidad técnica del proyecto parecían desalentadores, lo que llevó a una pausa en su desarrollo. Este período de incertidumbre fue superado durante una fase intensiva de formación en el centro de trabajo, donde el uso diario y constante de estas tecnologías demostró su eficacia. La experiencia adquirida no solo facilitó el manejo de las herramientas, sino que también reforzó la confianza en su capacidad para llevar a cabo el proyecto con éxito.

3. Objetivos del Proyecto de Tienda Online

El proyecto tiene como finalidad fundamental la creación de una plataforma de comercio electrónico robusta y funcional para un negocio familiar que integra un restaurante y una tienda de productos. Esta tienda online está diseñada para expandir el mercado del negocio, mejorar la accesibilidad de los clientes a sus productos y optimizar los procesos internos de gestión. A continuación se detallan los objetivos generales y específicos que guiarán el desarrollo y la implementación del proyecto.

Objetivo General

- **Desarrollar una solución integral de comercio electrónico** que integre de manera efectiva las operaciones de un restaurante y una tienda física en una única plataforma online, proporcionando una interfaz de usuario amigable y eficiente para facilitar la interacción del cliente y la gestión administrativa.

Objetivos Específicos

1. **Crear una interfaz de usuario atractiva y funcional:**
 - Desarrollar una interfaz de usuario utilizando el framework Angular, que ofrezca una experiencia de usuario coherente y fluida a través de una aplicación de página única (SPA).
 - Implementar un diseño responsive que se adapte a diferentes dispositivos y tamaños de pantalla, asegurando que todos los usuarios puedan acceder a la tienda con facilidad desde cualquier dispositivo móvil o de escritorio.
 - Utilizar Tailwind CSS para estilizar la aplicación, aprovechando su sistema de diseño utilitario para una personalización rápida y mantenimiento eficiente del código CSS

2. Establecer una comunicación robusta entre el cliente y el servidor:

- Construir una API REST con Spring Boot que maneje todas las interacciones entre el front-end y el back-end, incluyendo la gestión de productos, pedidos y usuarios.
- Asegurar que la API sea segura, escalable y mantenible, utilizando las mejores prácticas de codificación y patrones de diseño apropiados.

3. Gestionar los datos de manera eficiente:

- Integrar PostgreSQL como sistema de gestión de base de datos relacional para almacenar y gestionar de manera eficiente todos los datos necesarios, incluyendo detalles de productos, usuarios y transacciones.
- Implementar Spring Data JPA para simplificar la integración de la base de datos con el back-end, facilitando operaciones CRUD (crear, leer, actualizar, eliminar) complejas y consultas a la base de datos.

4. Proporcionar capacidades de administración en la plataforma:

- Desarrollar módulos de administración dentro de la plataforma que permitan a los usuarios con privilegios de administrador gestionar el contenido de la tienda, incluyendo la adición, modificación y eliminación de productos y categorías.
- Crear dashboards administrativos que ofrezcan análisis e informes en tiempo real sobre las ventas, tendencias de pedidos y comportamientos de los clientes.

5. Implementar una solución de pago segura:

- Integrar la API de PayPal para procesar los pagos, ofreciendo a los usuarios una manera segura y confiable de realizar transacciones online.
- Asegurar que el proceso de checkout sea sencillo y directo, minimizando los pasos necesarios para completar una compra y reduciendo la tasa de abandono de carritos.

6. Asegurar la aplicación y los datos de los usuarios:

- Utilizar JSON Web Tokens (JWT) para la autenticación y autorización en la plataforma, asegurando que solo los usuarios autenticados puedan acceder a ciertas funcionalidades.
- Implementar medidas de seguridad adicionales como HTTPS, sanitización de entradas y validación de formularios para proteger contra vulnerabilidades comunes de seguridad web.

7. Documentar exhaustivamente el sistema y proporcionar capacitación:

- Elaborar documentación detallada del sistema que incluya guías de usuario, especificaciones técnicas y manuales de operación para los administradores del sistema.
- Organizar sesiones de capacitación para los usuarios finales y administradores del sistema, asegurando que puedan utilizar la plataforma de manera efectiva y eficiente.

Estos objetivos están diseñados para construir una plataforma de comercio electrónico que no solo resuelva las necesidades actuales del negocio, sino que también sea adaptable a futuras expansiones o cambios en el modelo de negocio. El cumplimiento de estos objetivos específicos contribuirá significativamente al éxito del proyecto y al crecimiento continuo del negocio en el entorno digital.

4.Desarrollo

Organización y Control de Versiones en el Proyecto de Tienda Online

El control de versiones es un componente crítico del desarrollo de nuestro proyecto, gestionado a través de GitHub, una plataforma que permite un control detallado y colaborativo del código fuente. A continuación se detalla la estrategia adoptada para el control de versiones y la gestión de cambios en nuestro repositorio:

1. Estructura de Ramas:

- **Rama Main:** Configurada como protegida para evitar modificaciones directas, es utilizada exclusivamente para el lanzamiento de versiones estables del proyecto. La integración a esta rama se realiza a través de pull requests, asegurando que todo el código haya pasado por una revisión exhaustiva y pruebas antes de su despliegue final.
- **Rama Develop:** Actúa como la principal rama de desarrollo, donde todas las nuevas características, correcciones y mejoras son integradas inicialmente. Esta rama sirve como un entorno de prueba para validar la interoperabilidad y estabilidad de nuevas funcionalidades antes de su consolidación en la rama main.

2. Protección de Ramas y Políticas de Merge:

- La rama main está configurada con políticas de protección que incluyen revisiones obligatorias de código por parte de otros desarrolladores y pruebas automatizadas que deben pasar exitosamente antes de permitir la fusión de código.
- Se implementaron hooks de pre-commit y pre-push para asegurar que las pruebas y linters se ejecuten antes de cada commit y push, ayudando a mantener la calidad del código en niveles óptimos.

3. Uso de Pull Requests y Code Reviews:

- Todos los cambios significativos en el código se realizan a través de pull requests, lo que facilita la revisión de código por parte de otros desarrolladores y contribuye a mejorar la calidad y seguridad del software.
- Las discusiones sobre mejoras o cambios propuestos se gestionan dentro de la interfaz de GitHub, permitiendo un seguimiento claro y estructurado de las decisiones y cambios.

4. Gestión del Proyecto con GitHub Projects:

- Para mejorar la organización y el seguimiento del progreso, se creó un proyecto tipo Board dentro de GitHub. Este Board se utiliza para visualizar y gestionar el flujo de trabajo de las tareas y características en diferentes etapas de desarrollo, desde la planificación hasta la finalización.
- El uso de un Board permite una asignación clara de responsabilidades y plazos, facilitando la distribución eficiente del tiempo y recursos. Cada tarea o issue se asocia con un card que puede moverse entre las columnas de "To Do", "In Progress", y "Done", asegurando un seguimiento visual del avance de cada elemento del proyecto.

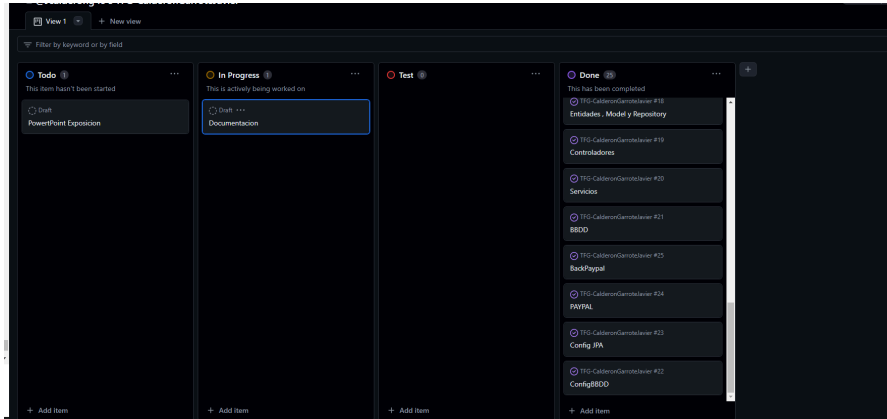
Enlace al repositorio de GitHub para un acceso directo y revisión del código:
[Repositorio del TFG - CalderonGarroteJavier](#)

Ejemplo de Gestión de Proyecto Utilizando GitHub Projects

En el Board de GitHub Projects para este proyecto, podemos ver el estado actual de varias tareas:

- **En la columna "To Do"** se encuentra el ítem "PowerPoint Exposición", indicando que esta tarea aún no ha sido iniciada.
- **En la columna "In Progress"** figura "Documentación", mostrando que este componente está actualmente en desarrollo y no se ha completado aún.

- La columna **"Done"** contiene múltiples tareas como "Entidades", "Model y Repository", "Controladores", "Servicios", "BBDD", entre otros, indicando que estas partes del proyecto ya están completas.



Este método de organización ayuda no solo a mantener un control claro sobre el progreso del trabajo sino también a facilitar la comunicación y coordinación entre los miembros del equipo, proporcionando una visualización en tiempo real del estado del desarrollo.

Esta estructura y metodología aseguran un desarrollo eficiente y ordenado, proporcionando un marco de trabajo claro para todos los involucrados en el proyecto y mejorando significativamente la gestión del ciclo de vida del software.

Diseño y Desarrollo de la Interfaz de Usuario

Para la creación de la tienda online de este proyecto, se puso especial énfasis en el diseño de la interfaz de usuario (UI), con el objetivo de ofrecer una experiencia de usuario (UX) clara, intuitiva y eficiente. El diseño sencillo y la estructura responsive de la aplicación se han implementado utilizando el framework Tailwind CSS, una herramienta altamente versátil y personalizable que facilita la adaptabilidad a distintos tamaños de pantalla y dispositivos.

Uso de Tailwind CSS:

Tailwind CSS se ha seleccionado como la solución principal para el diseño frontend debido a sus características únicas que permiten un control granular del estilo de la aplicación sin la necesidad de salir del contexto del HTML:

- **Escala Responsiva:** Tailwind facilita la implementación de un diseño responsive mediante clases utilitarias predefinidas que ajustan el estilo de los componentes según el tamaño de la pantalla, asegurando que la aplicación sea accesible y estéticamente agradable en dispositivos móviles, tabletas y ordenadores de escritorio.
- **Personalización y Extensibilidad:** A través de su configuración, Tailwind permite extender sus funcionalidades de manera que se pueden añadir nuevos componentes y utilidades específicas del proyecto, manteniendo un equilibrio entre el diseño personalizado y el rendimiento.

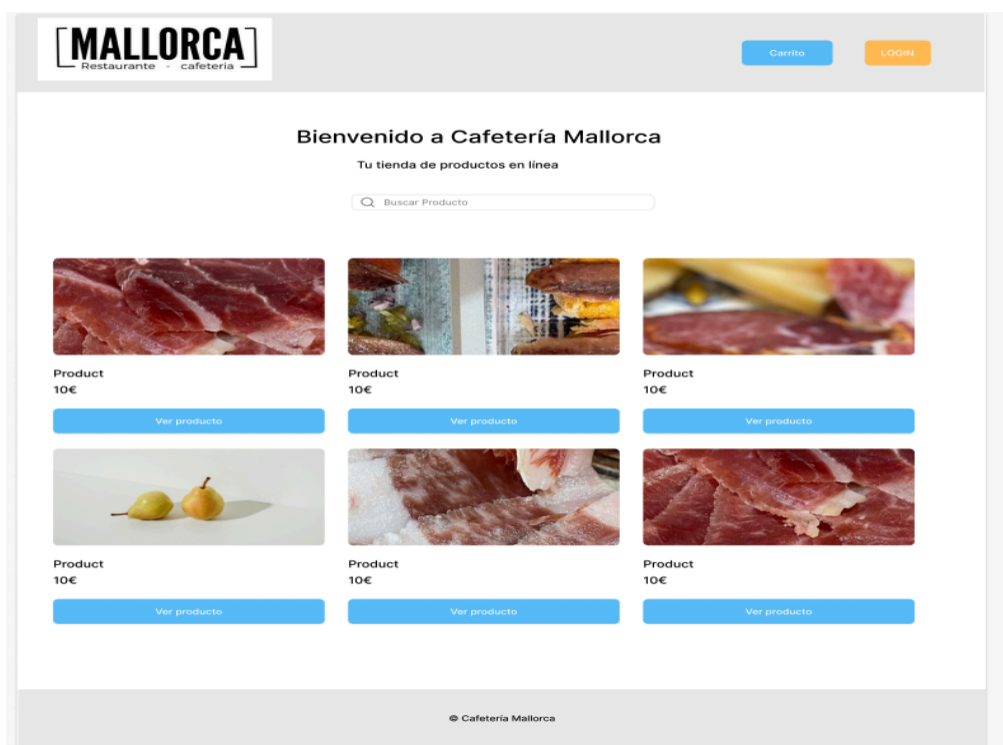
Uso de Figma para el Diseño Preliminar:

Para el diseño preliminar de la interfaz, se utilizó **Figma**, una herramienta de diseño vectorial y prototipado que permite a los diseñadores construir interfaces de usuario de manera colaborativa y en tiempo real. Figma se destacó por las siguientes razones:

- **Prototipado Interactivo:** Permite crear maquetas interactivas que simulan la navegación y las interacciones del usuario, lo cual es esencial para validar ideas de diseño y mejorar la usabilidad antes de la implementación técnica.
- **Colaboración en Tiempo Real:** Figma facilita la colaboración entre los miembros del equipo de diseño y desarrollo, permitiendo compartir avances, recibir feedback y realizar ajustes en tiempo real, lo cual acelera el proceso de diseño y asegura una mayor coherencia en la visión del proyecto.
- **Integración con Herramientas de Desarrollo:** Figma ofrece la posibilidad de exportar el diseño en formatos que son directamente utilizables por los desarrolladores (como código CSS), simplificando el proceso de transición del diseño al desarrollo.

Conclusión sobre el Diseño de la Interfaz

El enfoque utilizado para el diseño y desarrollo de la interfaz de la tienda online garantiza no solo una apariencia moderna y profesional, sino también una experiencia de usuario coherente y accesible, clave para el éxito de cualquier plataforma de comercio electrónico. Este enfoque estratégico en el diseño, apoyado por herramientas de vanguardia como Tailwind CSS y Figma, pone de manifiesto el compromiso del proyecto con la calidad y la innovación tecnológica.



Documentación del Sistema de Autenticación y Seguridad con JWT

Introducción

El sistema de autenticación y seguridad implementado en este proyecto utiliza JSON Web Tokens (JWT) para manejar la autenticación y autorización de los usuarios. Este método asegura que solo los usuarios autorizados puedan acceder a ciertos recursos del sistema. A continuación, se detalla el funcionamiento de los componentes principales involucrados en el proceso de login, registro, autenticación y seguridad.

Configuración de Seguridad

SecurityConfig

La clase SecurityConfig configura la seguridad de la aplicación utilizando Spring Security. Aquí se definen las políticas de seguridad, los filtros y la configuración de CORS y CSRF.

Componentes Clave:

1. Autorización de Peticiones HTTP:

- Se configuran las rutas y roles de acceso para determinar quién puede acceder a qué recursos en la aplicación.
- Por ejemplo, las rutas administrativas requieren el rol ADMIN, mientras que las rutas de pedidos y pagos requieren el rol USER.
- Las políticas de autorización se definen específicamente, estableciendo reglas de acceso para cada ruta.

2. Configuración de CORS:

- Se permite el acceso desde cualquier origen, métodos y encabezados.
- Esto es crucial para aplicaciones que interactúan con APIs desde diferentes dominios, evitando problemas de seguridad relacionados con CORS.
- La configuración de CORS se maneja mediante la definición de una política CORS global para la aplicación.

3. Desactivación de CSRF:

- CSRF (Cross-Site Request Forgery) se desactiva para aplicaciones REST.
- Las aplicaciones REST suelen manejar la seguridad de forma diferente a las aplicaciones web tradicionales, por lo que desactivar CSRF es una práctica común.
- Esto permite que las aplicaciones REST, que utilizan tokens para la autenticación, no requieran la protección CSRF que las aplicaciones web tradicionales necesitan.

4. Filtros de Seguridad:

- Se añade un filtro de autorización JWT para procesar los tokens y validar la autenticidad de las solicitudes.
- Este filtro se inserta en la cadena de filtros de Spring Security, lo que permite interceptar las solicitudes antes de que lleguen a los controladores para realizar la autenticación y autorización.

5. Codificación de Contraseñas:

- Se utiliza BCrypt para la codificación de contraseñas, proporcionando un alto nivel de seguridad al almacenar las contraseñas de los usuarios.
- BCrypt es un algoritmo de hashing que hace que las contraseñas sean más seguras al convertirlas en una cadena irreconocible de caracteres.

JWTAuthorizationFilter

El JWTAuthorizationFilter intercepta solicitudes HTTP para validar tokens JWT presentes en las cabeceras de autorización.

Funcionamiento:

1. Verificación de la Existencia del Token:

- El filtro comprueba si la solicitud contiene un token JWT en la cabecera de autorización.
- Si no se encuentra un token, la solicitud continúa sin autenticación adicional.

2. Validación del Token:

- El token se valida utilizando la clave secreta definida en la aplicación.
- Si el token es válido, se establece la autenticación en el contexto de seguridad de Spring.
- Esto incluye extraer el nombre de usuario y las autoridades (roles) del token.

3. Manejo de Excepciones:

- Se gestionan las excepciones relacionadas con JWT, como la expiración del token o su malformación.
- En caso de error, se devuelve un estado HTTP 403 (Forbidden) al cliente.

Controladores de Login y Registro

LoginController

El LoginController maneja solicitudes de inicio de sesión, autentica al usuario y genera un token JWT si las credenciales son válidas.

Proceso de Login:

1. Autenticación del Usuario:

- Las credenciales del usuario se autentican utilizando el gestor de autenticación de Spring Security.
- Esto implica verificar si el nombre de usuario y la contraseña proporcionados son correctos.

2. Generación del Token JWT:

- Si las credenciales son válidas, se genera un token JWT para el usuario autenticado.
- Este token contiene información como el ID del usuario, sus roles y una fecha de expiración.

3. Respuesta al Cliente:

- Se devuelve al cliente un objeto JSON que contiene el ID del usuario, el token JWT y posiblemente otra información relevante.
- El cliente puede almacenar este token y utilizarlo en futuras solicitudes para acceder a recursos protegidos por el sistema.

RegistrationController

El RegistrationController maneja solicitudes de registro de nuevos usuarios.

Proceso de Registro:

1. Encriptación de la Contraseña:

- Antes de almacenar la contraseña en la base de datos, se encripta utilizando un algoritmo seguro como BCrypt.
- Esto garantiza que las contraseñas de los usuarios estén protegidas incluso si la base de datos es comprometida.

2. Registro del Usuario:

- Se crea una nueva entrada en la base de datos para el usuario registrado.
- Esto incluye información como el nombre de usuario, la dirección de correo electrónico y otros detalles proporcionados durante el registro.

3. Respuesta al Cliente:

- Se devuelve al cliente una respuesta que confirma que el registro fue exitoso, junto con cualquier información adicional relevante.
- Por ejemplo, se puede devolver el usuario recién creado con un estado HTTP 201 (CREATED).

Servicios y Utilidades JWT

JWTGenerator

El JWTGenerator crea tokens JWT para usuarios autenticados.

Funcionamiento:

1. Generación del Token:

- Cuando un usuario se autentica correctamente, se genera un token JWT para ese usuario.
- Este token contiene información como el ID del usuario, sus roles y una fecha de expiración.
- La información se codifica en el token utilizando un algoritmo seguro y se firma con una clave secreta para garantizar su integridad.

JWTValidate

El JWTValidate proporciona métodos estáticos para validar y procesar tokens JWT.

Funcionamiento:

1. Existencia del Token:

- Antes de procesar una solicitud, se verifica si existe un token JWT en la cabecera de autorización.
- Si no se encuentra un token, se deniega el acceso y se devuelve un estado HTTP 401 (Unauthorized).

2. Validación del Token:

- Una vez que se confirma la existencia del token, se valida su integridad y se decodifica para extraer la información relevante

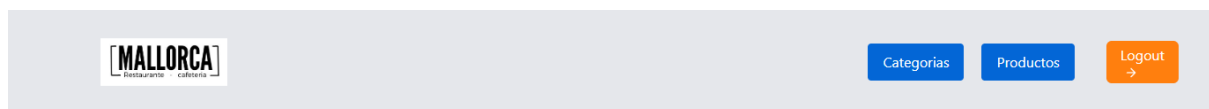
Tipos de Usuario en la Aplicación

La aplicación diseñada utiliza Tailwind CSS para un diseño responsivo que se adapta a diferentes dispositivos, facilitando la interacción del usuario con la plataforma. Se emplea Angular para la construcción de la interfaz, donde se diferencian dos tipos principales de usuarios: **Administrador** y **Usuario Normal**. Este diseño ha sido previamente esquematizado y probado usando Figma, lo que permitió previsualizar el diseño y comportamiento de la interfaz antes de su implementación.

Administrador

El usuario **Administrador** tiene capacidades extendidas dentro de la aplicación. En la barra de navegación, este tipo de usuario tiene acceso a las opciones de "Categorías" y "Productos", que le permiten administrar estos elementos. Estas opciones le permiten crear, editar, y eliminar productos o categorías según sea necesario.

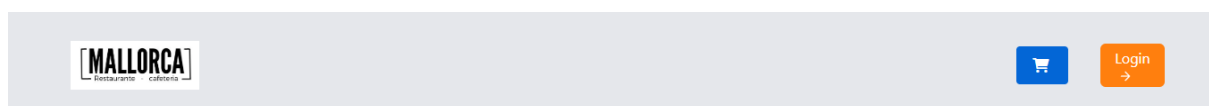
Ejemplo de la Barra de Navegación para el Administrador:



Usuario Normal

El **Usuario Normal** tiene una vista simplificada. En su barra de navegación aparece el botón de "Carrito" que le permite ver y gestionar sus productos seleccionados antes de finalizar la compra, y el botón de "Logout" para cerrar sesión.

Ejemplo de la Barra de Navegación para el Usuario Normal:



Detección de Tipo de Usuario

La aplicación determina si el usuario es un **Administrador** o un **Usuario Normal** basándose en la información contenida en el token JWT (JSON Web Token) que se almacena en el `sessionStorage` tras el inicio de sesión. Este token incluye los permisos y tipo de usuario, permitiendo al frontend adaptar la interfaz de usuario de acuerdo al nivel de acceso que posee el usuario.

Código TypeScript para el Manejo de Sesión:

```
ngOnInit(): void { no usages
  const token = this.sessionStorage.getItem(key: 'token');
  if (token && token.type === UserType.ADMIN) {
    this.isAdmin = true;
    this.isAuthenticated = true; // El usuario está autenticado
  }
}

logout() { Show usages
  // Elimina el token de la sesión y establece el estado de autenticación en falso
  this.sessionStorage.removeItem(key: 'token');
  this.sessionStorage.removeItem(key: 'user');
  this.isAuthenticated = false;
}
```

Este código ayuda a controlar la visualización de componentes en la interfaz dependiendo del tipo de usuario autenticado, asegurando que cada usuario acceda solo a las funciones que le son permitidas.

Documentación del Componente Footer

El componente Footer es una parte esencial de la interfaz de usuario en una aplicación Angular, diseñado para proporcionar información de pie de página de manera coherente en todas las páginas del sitio web. Su función principal es ofrecer una ubicación fija para mostrar detalles como derechos de autor, enlaces importantes o cualquier otra información relevante que se desee destacar en la parte inferior de la interfaz.

Pantalla Principal de Cafetería Mallorca:

La página principal de "Cafetería Mallorca", diseñada para proporcionar una experiencia de usuario óptima y eficiente, es la entrada al mundo de productos que ofrece el establecimiento. Esta pantalla ha sido meticulosamente desarrollada utilizando el framework Tailwind CSS, asegurando así un diseño responsivo que se adapta perfectamente a cualquier dispositivo, desde móviles hasta ordenadores de escritorio.

Diseño y Estructura de la Interfaz

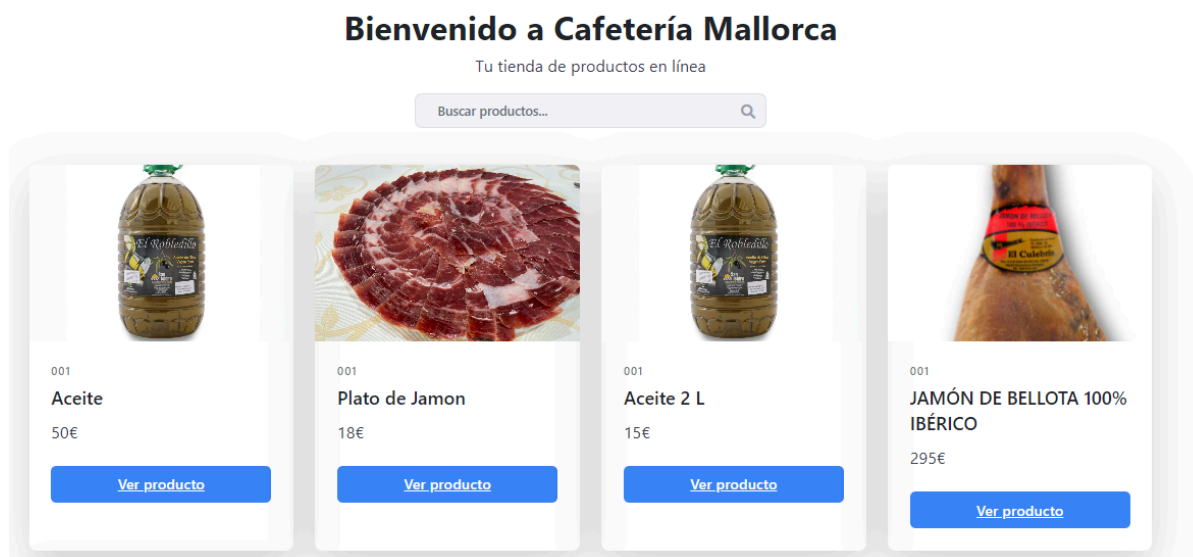
El diseño de la página principal está centrado en la simplicidad y la accesibilidad, permitiendo a los usuarios de todas las edades y habilidades navegar por la tienda con facilidad:

- **Barra de Búsqueda:** Estratégicamente colocada en la parte superior de la página, permite a los usuarios buscar rápidamente productos específicos, reduciendo el tiempo necesario para encontrar lo que buscan. Esta barra acepta entradas de texto y responde dinámicamente a las búsquedas, filtrando los productos en tiempo real.
- **Listado de Productos:** Cada producto se muestra dentro de una tarjeta que incluye una imagen clara y atractiva del producto, el nombre del mismo, el precio, y un botón de acción que dice "Ver producto". Estas tarjetas están diseñadas para ser lo suficientemente grandes para ser fáciles de leer, pero lo suficientemente compactas para no abrumar visualmente al usuario.
- **Diseño Responsivo:** Utilizando Tailwind CSS, el diseño responsivo asegura que la página se vea bien en una variedad de dispositivos y resoluciones. Las tarjetas de productos se reorganizan según el tamaño de la pantalla, asegurando una presentación óptima tanto en vertical como en horizontal.

Funcionalidad Detrás de la Interfaz

La funcionalidad de la página principal no sólo se limita a mostrar productos, sino que también facilita una interacción intuitiva:

- **Carga Dinámica de Productos:** Al visitar la página principal, una solicitud HTTP GET se envía al backend específicamente al endpoint `/api/products`. Esta solicitud es manejada por Spring Boot en el backend, que consulta la base de datos PostgreSQL para recuperar los productos y enviarlos de vuelta al frontend en formato JSON.
- **Filtrado y Búsqueda:** La barra de búsqueda interactúa con el modelo de datos en Angular, filtrando los productos en tiempo real según la entrada del usuario. Esto se maneja a través de enlaces de datos bidireccionales proporcionados por Angular, lo que permite una experiencia de usuario fluida y receptiva.
- **Acciones del Usuario:** Cada tarjeta de producto tiene un botón "Ver producto" que redirige al usuario a una página de detalles del producto. Esta acción proporciona una ruta fácil para que los usuarios obtengan más información sobre un producto específico, incluyendo descripciones detalladas, opciones de configuración y precios.



Diseño y Funcionalidad de la Página de Detalle del Producto

En la página de detalle de producto de "Cafetería Mallorca", la interfaz está diseñada para proporcionar una vista clara y detallada del producto seleccionado, mejorando la experiencia del usuario al facilitar la decisión de compra. Aquí te explico cómo se presenta cada elemento en esta página:

Estructura y Funcionalidad

1. Imagen del Producto:

- **Ubicación:** Lado izquierdo de la página.
- **Descripción:** Se muestra una imagen en tamaño completo del producto para que los usuarios puedan ver claramente lo que están considerando comprar. La alta resolución asegura que todos los detalles sean visibles.

2. Detalles del Producto:

- **Ubicación:** Lado derecho de la página.
- **Contenido:** Incluye el nombre del producto, una descripción detallada, y el precio. Estos detalles están dispuestos claramente para informar eficazmente al cliente sobre las características y el valor del producto.
- **Interacción de cantidad:** Los usuarios pueden seleccionar cuántas unidades del producto desean añadir a su carrito, proporcionando una pequeña interfaz con opciones para aumentar o disminuir la cantidad.

3. Añadir al Carrito:

- **Botón:** Ubicado debajo de las opciones de cantidad, este botón permite a los usuarios agregar fácilmente el producto seleccionado al carrito de compras. Al hacer clic en este, el producto se añade al carrito con la cantidad especificada, facilitando continuar con el proceso de compra o volver para seguir explorando otros productos.

Cafeteria Mallorca Producto



Plato de Jamon

Plato de Jamón Ibérico de Guijuelo. Cebo de campo, criado en libertad. Peso 250gr.
+ bolsita de picos andaluces. Los jamones de Carrasco Guijuelo están en nuestra carta desde hace un par de décadas. Un jamón excelente a un precio sorprendente.

18€

Cantidad:

1

Añadir al carrito

Página de Carrito de Compra

El carrito de compra es una parte fundamental de las tiendas en línea donde los usuarios pueden ver, ajustar y revisar los productos que desean comprar antes de realizar la compra final. Permite a los usuarios recopilar artículos, revisar precios y cantidades, y proceder con el pago de manera conveniente.

DATOS ORDEN
Nombre: Javier Calderon Garrote
Correo: jcalderong49@gmail.com
Dirección: Calle Cabarco 7D

Resumen Orden
Total: 18 €
[Pagar con PayPal](#)

PRODUCTOS

Producto	Precio	Cantidad	Total	Eliminar
Plato de Jamon	18	1	18	Quitar

Datos de la Orden:

Esta sección recopila información vital sobre el usuario y la orden en curso. Además de los datos básicos como el nombre y la dirección de correo electrónico, también se pueden incluir detalles de envío, como la dirección de entrega y cualquier instrucción especial. Esta información es crucial para garantizar que el pedido se entregue de manera precisa y oportuna. Además, la capacidad de editar estos datos directamente en el carrito proporciona una experiencia más fluida al usuario, ya que les permite corregir cualquier error o actualizar la información según sea necesario

DATOS ORDEN

Nombre: Javier Calderon Garrote

Correo: jcalderong49@gmail.com

Dirección: Calle Cabarco 7D

sin salir de la página.

Productos:

Aquí es donde los usuarios pueden ver en detalle todos los productos que han agregado al carrito. Cada artículo se muestra con su nombre, descripción, imagen, precio unitario y cantidad seleccionada. Esto permite a los usuarios revisar cuidadosamente los productos antes de proceder al pago. Además, se pueden incluir opciones para personalizar ciertos productos, como seleccionar diferentes tamaños o colores. Esto proporciona una experiencia de compra más personalizada y flexible.

La capacidad de eliminar productos individualmente del carrito es una característica importante. A menudo, los usuarios pueden cambiar de opinión sobre ciertos productos o decidir que ya no desean comprarlos. Permitirles eliminar elementos fácilmente sin tener que eliminar todo el pedido y comenzar de nuevo mejora la experiencia del usuario y fomenta la finalización de la compra.

PRODUCTOS

Producto	Precio	Cantidad	Total	Eliminar
Plato de Jamon	18	1	18	<button>Quitar</button>


Resumen de la Orden:

En esta sección se muestra un resumen claro y conciso de la orden, incluido el subtotal de los productos seleccionados, los impuestos aplicables y cualquier cargo adicional, como los gastos de envío. Proporcionar esta información de manera transparente y comprensible es fundamental para mantener la confianza del usuario y evitar sorpresas desagradables al finalizar la compra.

La opción de pago con PayPal ofrece a los usuarios una forma conveniente y segura de completar la transacción. Al integrar la API de PayPal directamente en la página del carrito, se simplifica el proceso de pago y se reduce la fricción del usuario. Esto es especialmente beneficioso para los usuarios que ya tienen cuentas de PayPal, ya que les permite realizar pagos de manera rápida y segura sin tener que ingresar información de pago adicional.

Resumen Orden

Total: 18 €

 Pagar con PayPal

Calderón Garrote Javier

Página del CRUD de Productos

La página del CRUD (Crear, Leer, Actualizar, Eliminar) de productos está diseñada para permitir a los administradores gestionar eficazmente los productos de una tienda en línea. A continuación, se ofrece una explicación detallada del diseño y la funcionalidad de esta página.

Contenedor Principal

El contenedor principal de la página incluye el título y las rutas de navegación. El título, ubicado en el centro, indica claramente que se trata de la sección de gestión de productos. Las rutas de navegación, o breadcrumbs, permiten a los usuarios rastrear su ubicación dentro del sitio y regresar fácilmente a secciones anteriores, como la página principal de administración.

Barra de Búsqueda y Botón de Crear Producto

Debajo del encabezado, se encuentra una barra de búsqueda y un botón para crear nuevos productos. La barra de búsqueda es una herramienta fundamental que permite a los administradores filtrar productos rápidamente según términos específicos, como el nombre o la descripción del producto. El botón "Crear Producto" redirige a un formulario de creación de productos, facilitando la adición de nuevos ítems al inventario.

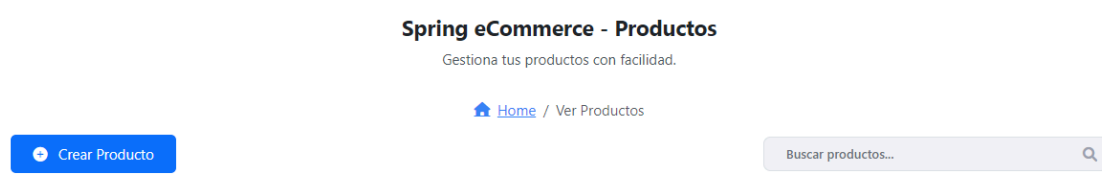

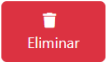


Tabla de Productos

La sección principal de la página muestra una tabla que enumera todos los productos disponibles. Esta tabla incluye varias columnas que muestran información básica sobre cada producto, como el código, nombre, descripción y precio. Además, cada fila de la tabla contiene botones para editar y eliminar productos:

- **Código, Nombre, Descripción, Precio:** Estas columnas proporcionan una visión general de la información esencial de cada producto.
- **Botón de Editar:** Al hacer clic en este botón, el administrador es redirigido a un formulario de edición con los datos del producto pre-rellenados, lo que facilita la actualización de la información.
- **Botón de Eliminar:** Este botón permite al administrador eliminar el producto de la base de datos. Antes de proceder con la eliminación, generalmente se solicita una confirmación para evitar borrados accidentales.

Código	Nombre	Descripción	Precio	Editar	Eliminar
001	Aceite 2 L	Aceite de Oliva Virgen Extra, Selección Cosecha Temprana, variedad Picual. Calidad superior, extraído en frío a menos de 27°C mediante centrifugación de la pasta de aceitunas y sólo mediante procedimientos mecánicos	15		

Paginación

Para mejorar la navegación a través de una gran cantidad de productos, la tabla incluye una funcionalidad de paginación. Esto significa que solo se muestran unos pocos productos por página, y los administradores pueden navegar entre las páginas usando botones de "Anterior" y "Siguiete", así como saltar a páginas específicas. La paginación es crucial para mantener la interfaz de usuario limpia y manejable, especialmente cuando se trabaja con grandes bases de datos.

Crear Producto

La funcionalidad para crear un nuevo producto se inicia desde el botón "Crear Producto". Al presionar este botón, el administrador es llevado a una nueva página con un formulario detallado. Este formulario incluye varios campos obligatorios, como la categoría del producto, código, nombre, descripción, precio e imagen. Cada campo tiene validaciones para asegurarse de que no se pueda enviar el formulario sin completar la información necesaria:

- **Categoría:** Un menú desplegable permite seleccionar la categoría del producto. Si el campo no se completa, se muestra un mensaje de error indicando que es obligatorio.
- **Código:** Un campo numérico que requiere la entrada de un código único para el producto.
- **Nombre:** Un campo de texto para ingresar el nombre del producto.
- **Descripción:** Un área de texto donde se debe ingresar una descripción detallada del producto.
- **Precio:** Un campo numérico para el precio del producto.
- **Imagen:** Un campo de carga de archivos para subir una imagen del producto.

Cada campo está acompañado de mensajes de error que se muestran si el campo no cumple con las validaciones requeridas (por ejemplo, si el campo está vacío o si el formato no es correcto).

Crear Producto

ELIJA UNA CATEGORÍA:

CÓDIGO:

Ingrese el Código del producto

NOMBRE:

Ingrese el nombre del producto

DESCRIPCIÓN:

Ingrese la descripción del producto

PRECIO:

0

IMAGEN:

Seleccionar archivo Ningún archivo seleccionado

Editar Producto

Al hacer clic en el botón de edición en la tabla de productos, el administrador es redirigido al mismo formulario utilizado para crear productos, pero con los campos pre-rellenados con la información del producto seleccionado. Esto facilita la modificación de los detalles del producto sin necesidad de reingresar toda la información desde cero. El administrador puede actualizar cualquier campo y guardar los cambios, que se reflejan inmediatamente en la base de datos.

Eliminar Producto

La eliminación de un producto se realiza mediante el botón de eliminar en cada fila de la tabla. Al hacer clic en este botón, se suele mostrar un cuadro de confirmación para asegurarse de que el administrador realmente desea eliminar el producto. Esta confirmación es una medida preventiva para evitar la eliminación accidental de productos. Una vez confirmado, el producto se elimina de la base de datos y se actualiza la lista de productos para reflejar esta eliminación.

Página del CRUD de Categorías:


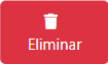
La página del CRUD de Categorías está diseñada para facilitar a los administradores la gestión eficiente de las categorías de productos en una tienda en línea. A continuación, se ofrece una explicación detallada del diseño y la funcionalidad de esta página.

Contenedor Principal: El contenedor principal de la página incluye un título central que indica claramente la sección de gestión de categorías.

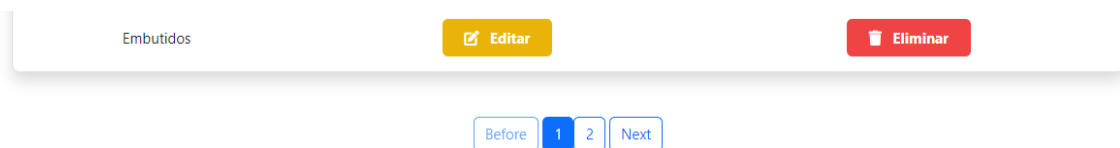
Barra de Búsqueda y Botón de Crear Categoría: Debajo del encabezado, se encuentra una barra de búsqueda y un botón para crear nuevas categorías. La barra de búsqueda es una herramienta fundamental que permite a los administradores filtrar categorías rápidamente según términos específicos, como el nombre de la categoría. El botón "Crear Categoría" redirige a un formulario de creación de categorías, facilitando la adición de nuevas categorías al sistema.

Tabla de Categorías: La sección principal de la página muestra una tabla que enumera todas las categorías disponibles. Esta tabla incluye varias columnas que muestran información básica sobre cada categoría, como el nombre y una descripción breve. Además, cada fila de la tabla contiene botones para editar y eliminar categorías:

- **Nombre:** Esta columna proporciona una visión general de los nombres de las categorías disponibles.
- **Botón de Editar:** Al hacer clic en este botón, el administrador es redirigido a un formulario de edición con los datos de la categoría pre-rellenados, lo que facilita la actualización de la información.
- **Botón de Eliminar:** Permite al administrador eliminar la categoría de la base de datos. Antes de proceder con la eliminación, generalmente se solicita una confirmación para evitar borrados accidentales.

Código	Nombre	Descripción	Precio	Editar	Eliminar
001	Aceite 2 L	Aceite de Oliva Virgen Extra, Selección Cosecha Temprana, variedad Picual. Calidad superior, extraído en frío a menos de 27°C mediante centrifugación de la pasta de aceitunas y sólo mediante procedimientos mecánicos	15		

Paginación: Para mejorar la navegación a través de una gran cantidad de categorías, la tabla incluye una funcionalidad de paginación. Esto significa que solo se muestran unos pocos elementos por página, y los administradores pueden navegar entre las páginas usando botones de "Anterior" y "Siguiente", así como saltar a páginas específicas. La paginación es crucial para mantener la interfaz de usuario limpia y manejable, especialmente cuando se trabaja con grandes conjuntos de datos.



Crear Categoría: La funcionalidad para crear una nueva categoría se inicia desde el botón "Crear Categoría". Al presionar este botón, el administrador es llevado a una nueva página con un formulario detallado. Este formulario incluye un campo obligatorio. Cada campo tiene validaciones para asegurarse de que no se pueda enviar el formulario sin completar la información necesaria.

Cafeteria Mallorca Categorías

[Categorías](#) / [Crear Categoría](#)

Crear Categoría

Nombre:

Ingrese el nombre de la categoría

Guardar

Editar Categoría: Al hacer clic en el botón de edición en la tabla de categorías, el administrador es redirigido al mismo formulario utilizado para crear categorías, pero con los campos pre-rellenados con la información de la categoría seleccionada. Esto facilita la modificación de los detalles de la categoría sin necesidad de reingresar toda la información desde cero. El administrador puede actualizar cualquier campo y guardar los cambios, que se reflejan inmediatamente en la base de datos.

Eliminar Categoría: La eliminación de una categoría se realiza mediante el botón de eliminar en cada fila de la tabla. Al hacer clic en este botón, se muestra un cuadro de confirmación para asegurarse de que el administrador realmente desea eliminar la categoría. Esta confirmación es una medida preventiva para evitar la eliminación accidental de categorías. Una vez confirmado, la categoría se elimina de la base de datos y se actualiza la lista de categorías para reflejar esta eliminación.

5. Conclusion

Mi proyecto de desarrollo ha sido un viaje increíble de aprendizaje y superación. Al principio, decidí utilizar Angular y Spring, pero me enfrenté a desafíos abrumadores. La complejidad del proyecto me hizo considerar alternativas como Drupal para simplificar el proceso. Sin embargo, decidí persistir con Angular y Spring gracias a la formación que recibí en el Centro de Trabajo, donde adquirí valiosos conocimientos.

A lo largo del proyecto, me encontré con obstáculos técnicos y conceptuales que pusieron a prueba mis habilidades y conocimientos. Cada desafío superado representó una valiosa oportunidad de crecimiento. Aprendí a resolver problemas complejos, a optimizar el rendimiento del código y a trabajar de manera eficiente con las herramientas y frameworks seleccionados.

Con el tiempo, desarrollé una comprensión más profunda de los principios de desarrollo de software y cómo aplicarlos en un entorno práctico. Aprendí a planificar y estructurar proyectos a gran escala, a colaborar con otros desarrolladores y a gestionar eficazmente el tiempo y los recursos disponibles.

Aunque el camino fue difícil y el proyecto requirió mucho tiempo y esfuerzo, el resultado final fue gratificante. Completar con éxito este proyecto es un logro del que me siento muy orgulloso. No solo demuestra mi capacidad para enfrentar desafíos técnicos, sino también mi dedicación y determinación para alcanzar mis metas.

En resumen, este proyecto ha sido una experiencia transformadora que me ha permitido crecer tanto personal como profesionalmente. Me ha enseñado la importancia de la perseverancia, la resiliencia y el trabajo duro, y estoy emocionado de aplicar lo que he aprendido en futuros proyectos y desafíos.

6. Bibliografía

- Academia de Desarrollo Web. (2023). "Construyendo una Tienda Online con Angular y Spring Boot" https://www.youtube.com/watch?v=_wSmeve6Sz8.
- Documentación oficial de Angular. Disponible en: <https://docs.angular.lat/docs>.
- Documentación oficial de Spring Boot. Disponible en: <https://docs.spring.io/spring-boot/index.html>.
- Documentación oficial de Tailwind CSS v2. Disponible en: <https://v2.tailwindcss.com/docs>.
- "¿Qué es JSON Web Token y cómo funciona?" OpenWebinars. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>.

7. Mejoras Futuras

Página Todo en Uno: Estoy trabajando en un concepto para centralizar toda la información relevante sobre mi restaurante, tienda y servicios en un único espacio digital. Creo que esta integración facilitará a mis clientes tener una visión global de lo que ofrezco y tomar decisiones más informadas sobre cómo interactuar con mi negocio en línea.

Reservas Simplificadas: Estoy desarrollando un sistema de reserva de mesas que sea intuitivo y fácil de usar para mis clientes. La idea es proporcionarles una experiencia sin complicaciones donde puedan reservar una mesa con unos pocos clics o toques en la pantalla. Esto no solo mejorará la comodidad para mis clientes, sino que también optimizará mi proceso interno de gestión de reservas.

Diversificación de Métodos de Pago: Estoy ampliando las opciones de pago disponibles en mi plataforma en línea para adaptarme a las preferencias de una variedad más amplia de clientes. Además de las opciones tradicionales como tarjetas bancarias, estoy integrando métodos de pago alternativos como Bizum y transferencia bancaria. Esta diversificación debería hacer que el proceso de compra sea más accesible y conveniente para una audiencia más amplia.

Mejora de la Estética del Sitio Web: Estoy enfocando mis esfuerzos en mejorar el aspecto visual y la estética general de mi sitio web. Esto implica revisar el diseño, la disposición de los elementos y la selección de imágenes para crear una experiencia más atractiva y agradable para mis visitantes. Mi objetivo es transmitir profesionalismo y confianza a través de una presencia en línea más pulida y atractiva.

8.Problemas Encontrados

Problema 1: Falta de Familiaridad con las Tecnologías Utilizadas

Descripción: Al inicio del proyecto, me encontré con dificultades debido a mi escasa experiencia en las tecnologías seleccionadas. Esta carencia generó incertidumbre y me hizo considerar abandonar el proyecto debido a la complejidad percibida.

Solución: Para superar este obstáculo, me sumergí en recursos de formación en línea, como vídeos en YouTube y documentación proporcionada por la comunidad de desarrolladores. Con determinación y dedicación, adquirí el conocimiento necesario para abordar los desafíos del proyecto.

Problema 2: Dificultades con la Integración de PayPal

Descripción: Durante la integración de la API de PayPal, me enfrenté a varios desafíos, especialmente en la configuración y la implementación de la autenticación y la seguridad necesarias para procesar pagos de manera segura.

Solución: Profundicé en la documentación disponible y busqué soluciones en diversas fuentes en línea. A través de la perseverancia y la experimentación, logré resolver los problemas y completar la integración de PayPal con éxito.

Problema 3: JWT (JSON Web Token)

Descripción: La implementación y gestión de JSON Web Token (JWT) para la autenticación y seguridad en el proyecto presentó desafíos significativos.

Solución: Dedicando tiempo al estudio detallado de la especificación de JWT y consultando recursos educativos en línea, fortalecí mi comprensión y resolví los problemas específicos que surgieron. La colaboración con expertos en el tema fue fundamental para superar estos obstáculos y completar la implementación de JWT con éxito.

Problema 4: GitHub: Problemas de Ramas y Versiones

Descripción: En el desarrollo del proyecto, enfrenté dificultades en la gestión de ramas y versiones en GitHub, lo que afectó la colaboración efectiva entre los miembros del equipo y la organización del código.

Solución: Estudiando detenidamente la documentación oficial de GitHub y aprovechando recursos educativos en línea, implementamos un flujo de trabajo sólido. Con la colaboración activa, mejoramos significativamente la colaboración y la organización del código en el proyecto.