

Memoria de Proyecto Fin de CFGS

Desarrollo de Aplicaciones WEB

23/01/2024

IES Bernaldo de Quirós
Ernesto Estrada Álvarez

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Índice

Planteamiento e idea del proyecto	4
Justificación	4
Desarrollo de la idea y propuesta de valor	4
A quién va dirigido y objetivos.....	5
Objetivos generales	5
Objetivos específicos	6
Análisis de costes	6
Plan de financiación	7
Plan de recursos humanos	8
Plan de prevención de riesgos	9
Requisitos	11
Introducción	11
Descripción general	11
Requisitos específicos	11
Requerimientos funcionales	11
Interfaces de Usuario	12
Interfaces Hardware	12
Requerimientos de Rendimiento	13
Obligaciones de diseño.....	14
Estándares cumplidos.....	14
Limitaciones Hardware.....	14
Seguridad	14
Facilidades de mantenimiento.....	14
Portabilidad	14
Otros Requerimientos	15
API de Restaurantes - Estructura y Funcionamiento	15
Capas Principales de la API.....	15
Funcionalidades Extendidas de la API	17
Gestión de Sidrerías	17
Endpoints y Funcionalidades.....	17
Gestión de Usuarios.....	18
Endpoints y Funcionalidades.....	18
Gestión de Reseñas.....	19

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Endpoints y Funcionalidades.....	19
Gestión de Imágenes	20
Endpoints y Funcionalidades.....	20
Roles y Autorización	20
Configuración y Seguridad mediante token JWT	21
Autenticación con JWT.....	21
Roles y Permisos	21
Protección de Rutas	21
Manejo de Seguridad en el Código	22
Gestión de Contraseñas.....	22
Configuración de CORS.....	22
Flujos de Autenticación	22
Buenas Prácticas	23
Frontend y diseño de la interfaz	24
Funcionamiento del frontend.....	24
Ficheros JS y para qué sirven	24
Bootstrap y hoja de estilos	33
Interfaz responsive para dispositivos	33
Conclusión final.....	40

Planteamiento e idea del proyecto

Se trata de una aplicación web que te permita buscar sidrerías asturianas según unos filtros, por ejemplo la ubicación, nombre, clase de sidra, escanciado...

Tendrás la opción de registrarse que será opcional a la hora de utilizar la aplicación, podrás registrarte como cliente o como hostelero de una sidrería

También te permite guardar tus sidrerías favoritas, y la sidra que más te guste, puedes elaborar tus propias reseñas y valoraciones a parte podrás consultar las reseñas de los demás.

De cada sidrería tendrás información detallada e imágenes del local y de su oferta gastronómica.

El gerente podrá modificar las imágenes, menú y carta de su sidrería

En la página principal tendrás un mapa de Google Maps marcando tu ubicación y las sidrerías cercanas.

Justificación

Esta idea surgió debido a que el turismo en Asturias ha crecido este año en un 13% cada vez más personas vienen a Asturias a disfrutar de su cultura y gastronomía, la sidra es la bebida tradicional asturiana como experiencia y sería un buen gancho para atraer a nuevos clientes a las sidrerías. Muchos hosteleros estarían interesados en promocionar sus negocios y muchos clientes en buscar sitios nuevos.

Desarrollo de la idea y propuesta de valor

La idea principal para el desarrollo de la aplicación es crear una plataforma online gastronómica que permita a los usuarios consultar los restaurantes más populares de la gastronomía asturiana, podrá filtrar por ubicación nombre, precio medio por comensal. Se podrá acceder a la aplicación a través de un login y registrarse en la aplicación si no lo estas. También podrás utilizar la aplicación sin darte de alta, aunque no podrás sacarle partido a toda la funcionalidad. Todos los usuarios podrán ver todos los restaurantes alojados en la plataforma y ver su oferta gastronómica y reviews publicadas. Solo los usuarios registrados podrán además de la funcionalidad anterior crear sus propias reviews y dar su valoración del sitio, guardar sus sidrerías favoritas además de editar su perfil.

Por último, existirá un usuario gerente que tendrá toda la funcionalidad anterior además de registrar su restaurante podrá editarlo, descripción fotos, precio por comensal... Y también podrá realizar una contestación a las reviews de su restaurante.

A quién va dirigido y objetivos

Esta aplicación va dirigida a personas mayores de edad de Asturias o turistas que pasen un período de tiempo en la zona. El objetivo de desarrollar esta aplicación es que se descubra nuevos lugares de gastronomía asturiana ayudando a promocionar estos negocios.

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Objetivos generales

1. **Lanzar Exitosamente la Aplicación:**
 - Lograr un lanzamiento efectivo de la aplicación web "Echa un culín" en el mercado asturiano.
2. **Posicionamiento en el Mercado:**
 - Establecer la aplicación como la principal plataforma de referencia para la promoción de restaurantes asturianos, ganando reconocimiento y confianza entre los usuarios.
3. **Generación de Ingresos:**
 - Obtener ingresos sostenibles a través de la publicidad dentro de la aplicación y la participación activa de los restaurantes asociados.
4. **Crecimiento de Usuarios:**
 - Aumentar de manera constante la base de usuarios, fomentando la descarga y el uso frecuente de la aplicación.
5. **Fomentar la Participación del Usuario:**
 - Incentivar la participación activa de los usuarios mediante la creación de reseñas, compartiendo imágenes y promoviendo la interacción social en la plataforma.

Objetivos específicos

1. **Desarrollo Tecnológico:**
 - Garantizar el desarrollo eficiente y efectivo de la aplicación, asegurando la compatibilidad con múltiples dispositivos y una experiencia de usuario óptima.
2. **Registro de Establecimientos:**
 - Lograr la participación activa de restaurantes asturianos en la aplicación, facilitando el registro de sus perfiles y promociones.
3. **Campañas de Marketing:**

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

- Implementar estrategias de marketing digital efectivas para aumentar la visibilidad de la aplicación y atraer nuevos usuarios.
- 4. **Interfaz de Usuario Atractiva:**
 - Diseñar una interfaz de usuario atractiva e intuitiva que fomente la exploración y participación constante.
- 5. **Monetización Efectiva:**
 - Establecer y optimizar modelos de monetización a través de la publicidad y servicios Premium para los restaurantes.
- 6. **Análisis de Datos y Retroalimentación:**
 - Implementar un sistema robusto de análisis de datos para comprender el comportamiento del usuario y obtener comentarios valiosos para la mejora continua.
- 7. **Alianzas Estratégicas:**
 - Establecer alianzas estratégicas con asociaciones gastronómicas, organismos turísticos y otros actores clave para fortalecer la posición de la aplicación.
- 8. **Atención al Cliente:**
 - Garantizar un servicio de atención al cliente eficiente para resolver consultas, problemas técnicos y proporcionar asistencia general a los usuarios y establecimientos registrados.
- 9. **Adaptación Continua:**
 - Mantener la aplicación actualizada con las últimas tendencias tecnológicas y necesidades del mercado, asegurando su relevancia a lo largo del tiempo.
- 10. **Fomentar la Lealtad del Usuario:**
 - Implementar programas de fidelización y recompensas para fomentar la lealtad de los usuarios y promover el uso continuo de la aplicación.

Análisis de costes

Los costes fijos mensuales para "Echa un culín" son de 3200 euros siendo los 500 euros para el hosting, 200 euros de mantenimiento y actualización, 500 euros de gastos generales y 2000 euros para salario del desarrollador. Estos denotan los costos básicos para el funcionamiento de proyecto y el mantenimiento constante.

Actualmente, los ingresos provienen de dos fuentes principales: la que son 800 euros los meses de publicidad integrada; y las suscripciones premium que suponen un coste de 5 euros por usuario al mes, que recaudan 500 euros al mes, gracias a 100 usuarios. Lo que suma un total de 1300 € al mes, lo que deja un déficit de 1900 € para afrontar los costes fijos.

Para lograr obtener beneficios únicamente de las suscripciones de pago, se necesitarían 380 usuarios más al coste actual de 5 euros por suscripción, llegando a una cifra total de 480 suscriptores. A este objetivo podrían añadirse otras que permitan incrementar los ingresos publicitarios, realzar la visibilidad de la aplicación o incluir nuevas características que justifiquen un aumento del precio de la suscripción.

Por lo tanto, el análisis pone énfasis en la mejora de fuentes de ingresos. Pasando al crecimiento de la base de usuarios y, la búsqueda de efectivas prácticas de monetización, el proyecto puede llegar a ser rentable y alcanzar su sostenibilidad financiera a más largo plazo.

Plan de financiación

Por la posibilidad de financiar mi empresa "Echa un culín, una aplicación web sobre gastronomía asturiana, opto por no utilizar financiación propia como la principal fuente.

Mi capital inicial es un premio de lotería y es de 30.000 €. Esta cantidad se considera una suma razonable para poder atender los gastos de inicio y las aportaciones que requiere la creación de la empresa, así como su funcionamiento en los primeros seis meses importantes. De este total, he optado por inversión 15.450 € para el negocio de flujos operativos y de inversión durante tres primeros meses, como se reflejó en cálculo anterior.

Como cualquier emprendedor, mi inversión inicial no se reduce a la aportación económica. Y enseguida también incluyo mi tiempo para la experiencia y destrezas en el área de desarrollo de software, lo que son aspectos importantes en el desarrollo y en el mantenimiento de la aplicación. Así mismo, decido reinvertir el salario que presumo que tengo durante los primeros meses en la empresa. Esto significa que al menos una parte del dinero del capital inicial se gasta en mi sueldo como un desarrollador lo que da al proyecto el financiamiento sustento inicial y asegura las condiciones para que el trabajo avance de manera constante.

En esta fase inicial no he buscado financiamiento externo ni he incluido socios inversores. Esta decisión me proporciona la máxima autoridad en las decisiones empresariales y planificación del proyecto. No obstante, lo cual, estoy dispuesto a replantearme esta visión en el futuro al enfocar la empresa más fuerte y exitosa, y así llegar a resultar como atractiva para el inversor. Esta sería especialmente útil si se requiere de más recursos para crecer con la compañía o ampliar su portafolio.

Mi enfoque de financiamiento se basa en lograr la máxima financiación propia posible manteniendo la mayor independencia posible de cualquier financiamiento externo. Para mí, el

enfoque que estoy siguiendo me permite comenzar el proyecto con el dinero en el banco, y, al mismo tiempo, lo suficientemente flexibles en la implementación de la estrategia de la empresa para ser capaz de cambiar en función de las condiciones del mercado y las reacciones de los usuarios a la aplicación.

Plan de recursos humanos

Varios aspectos relacionados con la planificación del personal en "Echa un Culín" hace referencia a la estrategia, en términos de mantener al proyecto adecuadamente preparado desde sus inicios y durante su crecimiento. Inicialmente, el equipo estará compuesto por un único miembro clave: el desarrollador web, junto con los gerentes de los restaurantes asociados y la comunidad de usuarios.

El desarrollador web que ocuparé yo en persona se encargará tanto de diseñar como de desarrollar la estructura y las características de la plataforma. Esta función implica la integración de características técnicas, la solución de cuestiones y el apoyo constante para asegurar que el usuario experimente una interacción fluida y efectiva con la aplicación. Asimismo, esta posición tendrá una labor crucial en el proceso de incorporación de nuevas tecnologías de manera que la plataforma se mantenga sostenible en términos de tecnológica.

No será un experto en gastronomía el que se encargue de lidiar con las posibles variantes sobre la oferta de los restaurantes, sino que los gerentes de estos últimos se encargaran de proponer lo que se puede cocinar directamente en la aplicación. Lo que refleja a su vez en lo que se refiere a la presencia en redes sociales es la publicación de sus menús, fotografías de los platos ofertados y demás datos pertinentes que destaquen su posición dentro del sector gastronómico. Además, los usuarios serán cruciales en la promoción de restaurantes y reseñas y puntuaciones que ayudarán a otros usuarios a visualizar los restaurantes más destacados.

A medida que la empresa siga creciendo y el número de usuarios aumente, se buscará contratar nuevo personal en marketing, atención al cliente y administración. Esto asegurará el buen

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

desempeño tanto para los restaurantes que forman parte de la plataforma como los clientes que acceden a ella para disfrutar de los restaurantes y el crecimiento estable y sano del negocio.

Este plan de expansión será enfocado en el desempeño financiero y en la demanda del mercado, de tal manera que "Echa un Culín" pueda cambiar y crecer según a lo que requiera el negocio e a lo que los usuarios esperen.

Plan de prevención de riesgos

La presente estrategia de Riesgo para "Echa un Culín" se hace imprescindible para conocer, analizar y controlar los riesgos potenciales que puedan repercutir en el aterrizaje y puesta en marcha de la aplicación. Este plan aborda diferentes tipos de riesgos: técnicos, de producción, financiero, legal y de mercado.

Riesgos Técnicos

Para disminuir posibles efectos negativos causados por desempeño técnico deteriorado se establecerá un plan detallado de prueba y mantención. Esta se engloba en realizar constantes comprobaciones de la accesibilidad y operatividad de la aplicación, así como en la implementación de garantías de seguridad y en la toma de copias de seguridad de los datos con cierta frecuencia. Así mismo se definirá un plan de Respuesta Rápida en caso de presentar alguna avería técnica que pueda dificultar o generar inconvenientes a los mismos usuarios, lo que se buscará evitar en la medida de lo posible.

Riesgos Operativos

Ya que al principio el equipo es pequeño y es en gran medida comprometido por un programador único, esta condición es una oportunidad operativa potencial. Para minimizarlo, se realizará la documentación de todos los flujos críticos que tengan relación con el desarrollo, mantenimiento y actualización de la plataforma. Esto garantizará que en caso de problema el conocimiento necesario esté al alcance para continuar las operaciones sin interrupción.

Riesgos Financieros

Las posibles pérdidas financieras incluyen escasez de cash-flow con el cual la empresa no puede satisfacer los costes operativos. Para prevenirla, se llevará a cabo un estudio financiero a profundidad, también se establecerá una garantía de seguro. También se aplicarán fuentes

de financiamiento, tales como, anuncios, suscripciones y posibles inversores externos y garantizar la viabilidad financiera en las primeras fases del proceso.

Riesgos Legales

Para no caer en peligros legales, "Echa un Culín" seguirá muy de cerca las regulaciones que tienen que ver con la privacidad de datos, el comercio electrónico y publicidad. Un abogado será contratado para revisar los términos y condiciones de servicio, así como las políticas de privacidad para asegurar que la empresa ofrezca sus servicios bajo la ley que sea correcta.

Riesgos de Mercado

La oferta de servicios del sector digital gastronómico es muy competitiva. Para el control de los riesgos de mercado se llevará un estudio permanente de las tendencias del sector y la competencia. Además, también se adoptará un marketing adaptativo que tendrá en cuenta las necesidades de los usuarios y las condiciones del mercado para mantener la relevancia de la aplicación.

Con este plan, "Echa un Culín" estará capacitado de encontrar y manejar riesgos de manera anticipada lo que garantizará su estabilidad y solidez a futuro.

Requisitos

Introducción

La especificación de requisitos es un paso crucial en el desarrollo de una aplicación, ya que define las necesidades y expectativas de los usuarios. En esta sección detallaré los requisitos necesarios para el correcto funcionamiento de la aplicación, asegurando que se cumplan los objetivos del proyecto y se brinde una experiencia de usuario satisfactoria.

Descripción general

El proyecto en cuestión está orientado a la creación de una aplicación web específica con énfasis en la presentación de la gastronomía asturiana con posibilidad de seleccionar restaurantes y sidrerías locales, así como obtener información sobre sus platos y ofrecer recomendaciones en función de sus experiencias personales. La aplicación presentará varios tipos de usuarios: comensales, gerentes de restaurantes y administradores con funcionalidades y prioridades propias. El propósito principal es ofrecer una herramienta fácil de usar y a la vez efectiva que permita coordinar en forma satisfactoria a los usuarios y los lugares de comida en Asturias para hacer el proceso de búsqueda interesante y encantador de la cultura culinaria.

Requisitos específicos

Los requisitos específicos del sistema se dividen en diferentes categorías para comprender mejor las necesidades y funcionalidades del proyecto. A continuación, se detallan los requisitos funcionales, que describen las acciones que el sistema debe ser capaz de realizar para satisfacer los objetivos del usuario.

Requerimientos funcionales

- Gestión de Usuarios:
 - Registro de nuevos usuarios como clientes, gerentes o administradores.
 - Inicio de sesión para acceder a las funcionalidades del sistema controlando el rol del usuario
- Gestión de Sidrerías:
 - Registro y edición de sidrerías por parte del gerente
 - Visualizar las sidrerías disponibles en la aplicación
- Gestión de Reviews:
 - Añadir reviews por parte de los usuarios registrados y valorar su visita al establecimiento.
 - Contestar las reviews por parte del gerente.
- Gestión perfil Usuario:
 - Edición del perfil por parte del gerente y del cliente.
 - Lista de sidrerías favoritas por parte de los usuarios registrados.

Interfaces de Usuario

Las interfaces de los usuarios constituyen la parte visible y accesible del sistema. Deben ser intuitivas y amigables para facilitar su uso por parte de los clientes y gerentes. Los requisitos específicos son:

- **Registro de Usuarios:** Debe haber un formulario de registro que permita a los usuarios crear una cuenta proporcionando la información necesaria, como nombre, correo electrónico y contraseña. Te puedes registrar como cliente o gerente de un restaurante
- **Inicio de Sesión:** Se debe proporcionar una interfaz de inicio de sesión que permita a los usuarios acceder al sistema introduciendo su correo electrónico y contraseña previamente registrados.
- **Perfil de usuario:** Los usuarios registrados en el sistema podrán personalizar su perfil, los clientes tendrán una lista de sidrerías favoritas y los gerentes un listado de las contestaciones a las reviews que han hecho.

Interfaces Hardware

Aunque el proyecto se centra en una aplicación web, es importante considerar los dispositivos hardware en los que los usuarios pueden acceder a la aplicación. Lo más habitual es que estos sean dispositivos móviles, por lo que la aplicación debe ser compatible y responsive con una variedad de dispositivos móviles, como teléfonos inteligentes y tabletas, para garantizar una experiencia de usuario consistente y óptima en diferentes plataformas. No se requiere hardware específico más allá de un dispositivo con capacidad de acceder a internet.

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Requerimientos de Rendimiento

- **Tiempo de Respuesta:** La aplicación debe cargar la página principal en menos de 3 segundos.
- **Escalabilidad:** El sistema debe poder llegar a manejar hasta 500 usuarios activos simultáneamente sin degradación del rendimiento. Este número de usuarios es importante para que la aplicación llegue a ser totalmente rentable.
- **Disponibilidad:** La aplicación debe estar operativa y accesible para los usuarios el 99.9% del tiempo durante un año.

Obligaciones de diseño

Estándares cumplidos

La aplicación debe cumplir con las normas de accesibilidad web ([WCAG 2.1](#)). Por las que debe ser perceptible, operable, comprensible y robusta

Uso de estándares de seguridad web para proteger datos sensibles ([OWASP](#)), que ayudan a evitar vulnerabilidades y a realizar pruebas de seguridad.

Limitaciones Hardware

No existen limitaciones significativas en cuanto a hardware, siempre que los dispositivos sean capaces de ejecutar un navegador web moderno. En cuanto a memoria, ningún dispositivo tendrá problemas ya que la carpeta de despliegue de la aplicación sólo ocupa 100Mb.

Seguridad

El registro de la aplicación consta de protección de datos personales, incluidas contraseñas, mediante cifrado. Esta acción se realiza mediante la librería de Spring; Spring security, método que nos facilita tener control y seguridad al procesar datos. Se llaman así porque nos permite filtrar los parámetros que recibimos, de modo que solamente los parámetros que indicamos son tomados en cuenta por nuestro controlador. Además de validar que endpoints son accesibles según el rol de cada usuario.

Facilidades de mantenimiento

Las facilidades de mantenimiento de la aplicación se basan en partir de un código limpio y ordenado, además de estar programado en un lenguaje de bajo nivel y por tanto fácilmente comprensible. Además, cuenta con un control de versiones en Git para gestionar el desarrollo del proyecto.

Portabilidad

La aplicación deberá ser portable entre distintos servidores y ser fácilmente desplegable en distintas plataformas de hosting.

Otros Requerimientos

Usabilidad: La interfaz debe ser intuitiva y fácil de usar, con una curva de aprendizaje mínima.

Fiabilidad: El sistema debe ser robusto y capaz de recuperarse rápidamente de fallos.

API de Restaurantes - Estructura y Funcionamiento

Este documento describe la estructura y funcionamiento de la API de restaurantes desarrollada en Spring Boot. La API permite gestionar restaurantes, menús, imágenes, reseñas y usuarios con funcionalidades completas, incluyendo autenticación y autorización.

Capas Principales de la API

La arquitectura de la API se organiza en varias capas que permiten una separación clara de responsabilidades, haciendo el sistema más mantenible, escalable y fácil de entender. A continuación, se describen las funciones y responsabilidades de cada capa.

1. Controller

La capa de **Controller** es el punto de entrada para las solicitudes HTTP. Su responsabilidad principal es procesar las peticiones entrantes, realizar validaciones iniciales y delegar la lógica de negocio a la capa de servicio. También se encarga de estructurar las respuestas adecuadas que se enviarán al cliente.

- **Responsabilidad:** Coordinar la interacción entre el cliente y la lógica de negocio.
- **Ejemplo:**
 - Procesar solicitudes de creación, actualización o eliminación de entidades.
 - Validar datos de entrada y enviar respuestas HTTP con códigos de estado adecuados.

2. Service

La capa de Service contiene la lógica de negocio y las reglas específicas del dominio de la aplicación. Es la encargada de gestionar las operaciones complejas que pueden involucrar varios repositorios o procesos de validación. En esta capa, se procesan los datos antes de interactuar con la base de datos o preparar la respuesta para el cliente.

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

- **Responsabilidad:** Implementar la lógica de negocio, asegurar que las reglas del dominio se cumplan, y servir como intermediario entre los controladores y los repositorios.
- **Ejemplo:**
 - Calcular promedios de valoración de restaurantes.
 - Gestionar relaciones entre entidades, como asignar reseñas a un restaurante.

3. Repository

La capa de **Repository** es responsable de interactuar directamente con la base de datos. Utiliza Spring Data JPA para simplificar el acceso y manejo de los datos, como guardar, actualizar, buscar o eliminar registros. Esta capa abstrae las operaciones CRUD básicas y permite crear consultas personalizadas mediante nombres de métodos o consultas JPQL.

- **Responsabilidad:** Acceso directo y eficiente a la base de datos.
- **Ejemplo:**
 - Obtener restaurantes por nombre o ubicación.
 - Buscar reseñas asociadas a un usuario o restaurante.

4. Model

La capa de **Model** define las entidades que representan las tablas de la base de datos. Estas clases son fundamentales para la interacción con JPA, ya que especifican las relaciones entre tablas y los mapeos necesarios. Las entidades suelen contener propiedades que se mapean directamente a columnas en la base de datos.

- **Responsabilidad:** Representar las estructuras de datos persistentes y las relaciones entre ellas.
- **Ejemplo:**
 - Definir entidades como **Restaurante**, **Usuario** o **Review**.
 - Especificar relaciones como uno a muchos o muchos a muchos

Funcionalidades Extendidas de la API

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

La API está diseñada para gestionar sidrerías, usuarios, reseñas y favoritos con diferentes niveles de acceso según el rol del usuario: **CLIENTE** o **GERENTE**. A continuación, se describen las funcionalidades con mayor detalle.

Gestión de Sidrerías

Endpoints y Funcionalidades

1. **Crear Sidrería (Rol: GERENTE):**
 - **Método:** POST
 - **URL:** /api/sidrerias/gerente/{id}
 - **Descripción:** Permite a un gerente crear una sidrería asociada a su cuenta. Se envían datos de texto (JSON) e imágenes como parte del formulario.
 - **Requisitos:**
 - Debes estar autenticado como GERENTE.
 - Los datos enviados incluyen el nombre, ubicación, precio por comensal, y otras configuraciones específicas de la sidrería.
2. **Actualizar Sidrería (Rol: GERENTE):**
 - **Método:** PUT
 - **URL:** /api/sidrerias/{id}
 - **Descripción:** Modifica los detalles de una sidrería existente, incluyendo datos y la gestión de imágenes.
 - **Requisitos:**
 - Solo el gerente propietario puede modificar su sidrería.
3. **Obtener Lista de Sidrerías (Público):**
 - **Método:** GET
 - **URL:** /api/sidrerias
 - **Descripción:** Devuelve un listado de sidrerías con posibilidad de aplicar filtros opcionales:
 - **nombre:** Buscar sidrerías por nombre.
 - **ubicacion:** Buscar por ubicación.
 - **minValoracion:** Filtrar por valoración mínima.
 - **maxPrecioComensal:** Filtrar por precio máximo por comensal.
 - **Respuesta:**
 - Detalles básicos como nombre, ubicación, valoración, imágenes.
4. **Eliminar Sidrería (Rol: GERENTE):**
 - **Método:** DELETE
 - **URL:** /api/sidrerias/{id}
 - **Descripción:** Permite al gerente eliminar una sidrería asociada a su cuenta. Esto también elimina las imágenes y reseñas relacionadas.

Gestión de Usuarios

Endpoints y Funcionalidades

1. **Registro de Usuarios (Público):**
 - **Método:** POST
 - **URL:** /api/auth/register
 - **Descripción:** Permite a nuevos usuarios registrarse como CLIENTE o GERENTE.
 - **Cuerpo:** JSON con nombre, email, contraseña, y rol deseado.
2. **Autenticación de Usuarios (Público):**
 - **Método:** POST
 - **URL:** /api/auth/login
 - **Descripción:** Autentica a un usuario y devuelve un token JWT para usar en las solicitudes protegidas.
 - **Cuerpo:** JSON con email y contraseña.
3. **Actualizar Perfil de Usuario (Autenticado):**
 - **Método:** PUT
 - **URL:** /api/usuarios/{id}
 - **Descripción:** Permite actualizar datos del perfil, como nombre, avatar, y contraseña.
 - **Requisitos:**
 - Solo el propio usuario puede modificar su perfil.
4. **Consultar Favoritos (Rol: CLIENTE):**
 - **Método:** GET
 - **URL:** /api/usuarios/{id}/sidrerias/favoritos
 - **Descripción:** Devuelve una lista de sidrerías marcadas como favoritas por el usuario.
5. **Añadir Sidrería a Favoritos (Rol: CLIENTE):**
 - **Método:** POST
 - **URL:** /api/sidrerias/favorito/{sidreriaId}
 - **Descripción:** Permite a un usuario marcar una sidrería como favorita.
6. **Eliminar Sidrería de Favoritos (Rol: CLIENTE):**
 - **Método:** DELETE
 - **URL:** /api/sidrerias/favorito/{sidreriaId}
 - **Descripción:** Quita una sidrería de los favoritos del usuario.

Gestión de Reseñas

Endpoints y Funcionalidades

1. **Crear Reseña para una Sidrería (Rol: CLIENTE):**
 - **Método:** POST
 - **URL:** /api/sidrerias/{id}/reviews
 - **Descripción:** Los clientes pueden dejar una reseña con título, contenido, valoración, y fecha.
2. **Actualizar Reseña (Rol: CLIENTE):**
 - **Método:** PUT
 - **URL:** /api/reviews/{id}
 - **Descripción:** Modifica una reseña creada por el cliente. Solo el autor puede modificarla.
3. **Eliminar Reseña (Rol: CLIENTE):**
 - **Método:** DELETE
 - **URL:** /api/reviews/{id}
 - **Descripción:** Permite eliminar una reseña creada por el cliente.
4. **Responder a una Reseña (Rol: GERENTE):**
 - **Método:** PUT
 - **URL:** /api/reviews/{id}/respuesta
 - **Descripción:** Los gerentes pueden responder a las reseñas que se hayan dejado en sus sidrerías.
5. **Listar Reseñas de un Usuario (Rol: CLIENTE):**
 - **Método:** GET
 - **URL:** /api/reviews/usuario/{usuarioId}
 - **Descripción:** Devuelve todas las reseñas creadas por un usuario.

Gestión de Imágenes

Endpoints y Funcionalidades

1. **Subir Imágenes para una Sidrería (Rol: GERENTE):**
 - **Método:** POST
 - **URL:** /api/sidrerias/{id}/imagenes
 - **Descripción:** Permite a un gerente subir imágenes de su sidrería. Las imágenes se guardan en el servidor y se enlazan con la entidad correspondiente.
2. **Eliminar Imágen (Rol: GERENTE):**
 - **Método:** DELETE
 - **URL:** /api/imagenes/{id}
 - **Descripción:** Permite eliminar una imagen específica asociada a una sidrería.
3. **Ver Imágenes (Público):**
 - **Método:** GET
 - **URL:** /imagenes/{ruta}
 - **Descripción:** Devuelve una imagen almacenada en el servidor.

Roles y Autorización

1. **CLIENTE:**
 - Puede:
 - Crear y modificar sus propias reseñas.
 - Gestionar sus sidrerías favoritas.
 - Consultar información pública de las sidrerías.
2. **GERENTE:**
 - Puede:
 - Crear y gestionar sus propias sidrerías.
 - Responder a reseñas.
 - Subir y gestionar imágenes de las sidrerías.
3. **Público (No Autenticado):**
 - Puede:
 - Consultar la lista de sidrerías.
 - Ver información básica de las sidrerías (ubicación, valoración, precio, etc.).
 - Ver imágenes.

Configuración y Seguridad mediante token JWT

La configuración de seguridad para la API de gestión de sidrerías utiliza **Spring Security** y **JWT Authentication** para proporcionar un entorno seguro y flexible, donde las solicitudes se manejan según los roles y permisos definidos. Este enfoque permite controlar el acceso a diferentes recursos de manera eficiente, sin depender de sesiones en el servidor, lo que es ideal para aplicaciones modernas y escalables.

Autenticación con JWT

El sistema de autenticación está basado en **JSON Web Tokens (JWT)**, una tecnología que permite autenticar a los usuarios mediante tokens compactos y seguros. Estos tokens se generan al registrarse o iniciar sesión, y se incluyen en cada solicitud autenticada mediante el encabezado **Authorization**. Un token JWT contiene información codificada como el ID del usuario, su rol y la fecha de expiración, lo que elimina la necesidad de mantener sesiones activas en el servidor. Cuando una solicitud llega al servidor, el filtro de autenticación **JwtAuthFilter** valida el token, asegurándose de que sea genuino y no haya expirado.

Roles y Permisos

La API define roles claros para los usuarios: **ADMIN**, **GERENTE** y **CLIENTE**, cada uno con diferentes niveles de acceso.

- El rol de **ADMIN** tiene control completo sobre todos los recursos, permitiéndole gestionar usuarios, reseñas y sidrerías sin restricciones.
- Por su parte, el **GERENTE** puede administrar exclusivamente las sidrerías que posee, incluyendo la creación, edición y eliminación de estas, así como responder a las reseñas de los clientes.
- Finalmente, el **CLIENTE** tiene un rol más limitado, enfocado en la interacción con los recursos públicos, la gestión de sus favoritos, y la creación y actualización de sus propias reseñas.

Esta división asegura que cada usuario solo pueda interactuar con los recursos que le correspondan, manteniendo un control estricto sobre los accesos.

Protección de Rutas

La seguridad de la API se implementa dividiendo las rutas en públicas y protegidas. Las rutas públicas incluyen aquellas necesarias para el registro y la autenticación de usuarios, así como la consulta de información general sobre las sidrerías. Por ejemplo, los endpoints para listar todas las sidrerías o acceder a imágenes estáticas están disponibles para cualquier usuario. En

contraste, las rutas protegidas requieren autenticación mediante un token válido y están restringidas según el rol del usuario. Así, un gerente podrá acceder a los endpoints relacionados con la administración de sus sidrerías, mientras que un cliente solo podrá gestionar sus favoritos y reseñas.

Manejo de Seguridad en el Código

El método de configuración de la seguridad utiliza el componente **SecurityFilterChain**. Este método establece varias capas de protección, como la desactivación de CSRF (que no es necesaria debido al uso de JWT), la habilitación de CORS para permitir solicitudes desde el frontend y la autorización de las rutas según los permisos definidos. Adicionalmente, la política de sesiones es completamente stateless, lo que significa que el servidor no guarda información de las sesiones activas; cada solicitud debe incluir un token válido para ser procesada.

Un aspecto clave es el filtro **JwtAuthFilter**, que intercepta todas las solicitudes protegidas. Este filtro extrae el token JWT del encabezado de la solicitud, lo valida y, si es correcto, configura el contexto de seguridad con la autenticación del usuario. Esto permite que las operaciones posteriores se realicen con el nivel de acceso correspondiente al rol del usuario.

Gestión de Contraseñas

Para garantizar la seguridad de las credenciales de los usuarios, las contraseñas se encriptan utilizando **BCryptPasswordEncoder** antes de almacenarlas en la base de datos. Al iniciar sesión, la contraseña proporcionada por el usuario se compara con la versión encriptada almacenada, asegurando que solo aquellos con las credenciales correctas puedan autenticarse.

Configuración de CORS

El sistema también incluye una configuración de **Cross-Origin Resource Sharing (CORS)** que permite solicitudes desde el dominio del frontend especificado. Esto es especialmente importante en aplicaciones distribuidas donde el frontend y el backend están en dominios diferentes. La configuración de CORS permite todos los métodos HTTP (**GET**, **POST**, **PUT**, etc.) y cabeceras necesarias, asegurando una comunicación fluida entre el cliente y el servidor.

Flujos de Autenticación

El flujo de autenticación comienza con el registro, donde un usuario envía su información, como nombre, email y contraseña. La contraseña se encripta antes de almacenarse, y al finalizar el registro exitoso, se genera un token JWT que se devuelve al usuario para futuras

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

solicitudes. En el inicio de sesión, el usuario proporciona sus credenciales, que se validan contra la base de datos. Si las credenciales son correctas, se genera un nuevo token JWT.

En las solicitudes protegidas, el cliente debe incluir este token JWT en el encabezado **Authorization**. El filtro **JwtAuthFilter** valida el token antes de procesar la solicitud. Si el token es válido y pertenece al usuario correcto, el sistema permite el acceso; de lo contrario, devuelve un error 401.

Buenas Prácticas

El diseño de seguridad de esta API sigue varias buenas prácticas:

- **Definición clara de roles y permisos**, asegurando que cada usuario solo pueda acceder a los recursos correspondientes.
- **Encriptación robusta de contraseñas** con **BCrypt**, lo que dificulta ataques incluso si la base de datos se ve comprometida.
- **División de rutas en públicas y protegidas**, simplificando la gestión de accesos.
- **Política de sesiones stateless**, lo que reduce la carga en el servidor y mejora la escalabilidad.
- **Expiración de tokens JWT**, lo que aumenta la seguridad al limitar la duración de los tokens válidos.

En conjunto, esta configuración proporciona una solución segura, escalable y bien organizada para gestionar la autenticación y autorización en la API.

Frontend y diseño de la interfaz

Funcionamiento del frontend

Para la creación del frontend he decidido limitarme a utilizar las herramientas con las que ya me desenvuelvo sin dificultad, a pesar de que en los inicios me plantee usar un framework como Angular, ya que mi objetivo era centrarme más en la optimización del backend. Las herramientas que he utilizado son HTML5, CSS3 Bootstrap, y JQuery.

El funcionamiento general de todos los ficheros .js de la aplicación consiste en realizar una petición ajax a la API que devuelva o envíe los datos que se quieran mostrar o procesar en la interfaz. Mientras tanto los ficheros html solo se encargan de mostrar la cabecera con el menú de navegación y el pie de página, el resto del código html se genera con JQuery.

Ficheros JS y para qué sirven

login.js y signup.js

Los scripts de **login** y **signup** gestionan el inicio de sesión y el registro de usuarios. Ambos recogen datos del formulario, validan el correo y la contraseña, y envían una solicitud AJAX al servidor (a `/api/auth/login` o `/api/auth/register`). Si la respuesta es exitosa, almacenan la información del usuario y su token en `sessionStorage` y redirigen a la página correspondiente. En caso de error, muestran alertas con mensajes adecuados.

index.js

Este script gestiona la carga y visualización dinámica de una lista de sidrerías, ajustando el contenido según si el usuario está autenticado.

Resumen del funcionamiento:

1. Carga inicial:

- Verifica si hay un usuario autenticado obteniendo sus datos del `sessionStorage`.
- Configura la URL de la API para obtener sidrerías, personalizándola si el usuario está autenticado.

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

2. Solicitud AJAX:

- Realiza una solicitud **GET** para obtener la lista de sidrerías.
- Si el usuario está autenticado, añade un token en los encabezados para acceso autorizado.

3. Visualización de sidrerías:

- Genera dinámicamente tarjetas que muestran información de cada sidrería (nombre, ubicación, valoración, imagen, etc.).
- Incluye un botón para marcar o desmarcar la sidrería como favorita.

4. Interacción con el usuario:

- **Click en una sidrería:** Redirige a su página específica.
- **Click en el botón de favorito:** Envía una solicitud **POST** para cambiar el estado de favorito de la sidrería, actualizando la interfaz en tiempo real.

5. Gestión de favoritos:

- Si el usuario no está autenticado, se le redirige a la página de inicio de sesión al intentar interactuar con favoritos.

mi-cuenta.js

Inicio y validación:

- Obtiene al usuario de **sessionStorage**.
- Si no está autenticado, redirige al inicio.

Secciones principales:

- **Detalles del usuario:** Muestra avatar, nombre completo y botón para editar.
- **Favoritos:** Carga y muestra las sidrerías favoritas del usuario en tarjetas interactivas.
- **Reseñas:** Lista reseñas con valoración, texto, fecha, opciones para editar/eliminar o muestra respuestas de las sidrerías.

Funciones clave:

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

- Generación de estrellas de valoración.
- Navegación a detalles de sidrerías y edición/eliminación de datos o reseñas.

mi-sidreria.js

Inicio y validación:

- Se verifica al usuario en `sessionStorage`.
- Si el usuario es válido, se obtiene la sidrería asociada al gerente.
- Si no, redirige al inicio de sesión.

Secciones principales:

- **Detalles de la sidrería:**
 - Muestra el nombre, ubicación, valoración, precio por comensal, servicio de escanciado, imágenes y enlace a la carta.
 - Incluye un botón para editar los datos de la sidrería.
- **Comentarios y reseñas:**
 - Lista las reseñas realizadas a la sidrería, mostrando la valoración, título, contenido y nombre del usuario que realizó la reseña.
 - Ofrece opciones para responder, editar o eliminar respuestas a las reseñas directamente desde la interfaz.

Funciones clave:

- **Gestión de respuestas:**
 - Responder a comentarios mediante un formulario.
 - Editar o cancelar respuestas previamente realizadas.
 - Eliminar respuestas enviadas.
- **Edición de datos:**
 - Navegación para modificar los datos generales de la sidrería.

pagina-sidreria.js

La página muestra información detallada sobre una sidrería específica y permite a los usuarios consultar sus reseñas, ver el menú y agregar comentarios, si están autenticados.

1. Carga inicial:

- Al cargar la página, se obtiene el parámetro **id** desde la URL.
- Si el **id** es válido:
 - Se realiza una solicitud AJAX a la API para obtener los datos de la sidrería y sus reseñas.
 - Si ocurre un error, muestra un mensaje indicando el fallo.

2. Función principal: **mostrarDetallesSidreria**

Crea y organiza los elementos de la página con los datos obtenidos.

Componentes:

1. **Detalles de la sidrería:**
 - Muestra el nombre, ubicación, valoración (con estrellas generadas dinámicamente), precio por comensal y si ofrece servicio de escanciado.
 - Incluye una galería de imágenes y un enlace al menú de la sidrería.
2. **Reseñas de usuarios:**
 - Presenta las reseñas en un diseño organizado:
 - Avatar, nombre y apellidos del usuario.
 - Título, contenido y valoración (representada con estrellas).
 - Respuesta del gerente, si existe.
 - Si no hay reseñas, indica que no hay comentarios disponibles.
3. **Botón "Añadir reseña":**
 - Permite al usuario agregar una nueva reseña.

- Redirige a la página de creación de reseñas si el usuario está autenticado; si no, redirige al inicio de sesión.

3. Funciones auxiliares:

1. **generarEstrellas(valoracion):**

- Crea un elemento HTML con estrellas llenas, medias o vacías según la valoración.

2. **clickOnCrearReview(idSidreria):**

- Verifica si hay un usuario autenticado:
 - Si lo hay, redirige a la página para crear una reseña.
 - Si no, lleva al inicio de sesión.

review-crear.js y review-editar.js

Verificación del Usuario: Al cargar la página, ambos procesos comienzan verificando si el usuario está autenticado, leyendo los datos desde **sessionStorage**. Si no se encuentra un usuario autenticado, el sistema redirige automáticamente al inicio de sesión.

Crear Reseña:

- **Formulario de Creación:** Si el usuario está autenticado y se proporciona un **idSidreria** en la URL, el formulario de creación de reseñas se activa. El usuario puede ingresar un título, una valoración (en forma de número), y el contenido de su comentario.
- **Envío del Formulario:** Al enviar el formulario, los datos de la reseña se envían mediante una solicitud **POST** a la API del servidor para crear una nueva reseña para la sidrería correspondiente.
- **Redirección:** Si la creación es exitosa, el sistema redirige al usuario a la página de detalles de la sidrería para visualizar la nueva reseña.

Editar Reseña:

- **Carga de la Reseña:** Si se proporciona un `idReview` válido en la URL, el sistema verifica si el usuario autenticado tiene esa reseña. Realiza una solicitud `GET` a la API para cargar las reseñas del usuario y filtra por el `idReview` específico. Si se encuentra la reseña, los campos del formulario se rellenan automáticamente con los datos existentes (título, valoración y contenido).
- **Envío del Formulario:** Al enviar el formulario de edición, los datos actualizados se envían mediante una solicitud `PUT` a la API para actualizar la reseña en el servidor.
- **Redirección:** Si la actualización es exitosa, el sistema redirige al usuario a la página anterior.

sidreria-crear.js y sidreria-editar.js

Verificación del Usuario: Ambas acciones comienzan con la verificación del usuario autenticado, que se obtiene de `sessionStorage`. Si no hay un usuario autenticado, se redirige al inicio de sesión o a la página principal.

Crear Sidrería:

- **Formulario de Creación:** El formulario permite al usuario introducir el nombre, ubicación, precio por comensal y si ofrece servicio de escanciado. También se puede agregar imágenes de la sidrería a través de un campo de carga de archivos.
- **Envío del Formulario:** Al enviar el formulario, los datos del formulario (incluyendo imágenes) se recopilan en un objeto `FormData` y se envían mediante una solicitud `POST` a la API del servidor para crear una nueva sidrería.
- **Redirección:** Si la creación es exitosa, el usuario es redirigido a la página de inicio con un mensaje de éxito.

Editar Sidrería:

- **Cargar Datos Existentes:** Al cargar la página de edición, se realiza una solicitud **GET** para obtener los detalles de la sidrería actual del usuario. Estos datos se muestran en el formulario de edición, permitiendo que el usuario los modifique. Se cargan las imágenes actuales y la carta (si existen).
- **Envío del Formulario:** Al enviar el formulario de edición, los datos modificados (incluyendo nuevas imágenes) se recopilan en un objeto **FormData** y se envían mediante una solicitud **PUT** a la API para actualizar los detalles de la sidrería.
- **Redirección:** Si la actualización es exitosa, el usuario es redirigido a la página anterior con un mensaje de éxito.

usuario-editar.js

Verificación del Usuario: El código comienza con la comprobación de si el usuario está autenticado, obteniendo los detalles del usuario desde **sessionStorage**. Si no hay un usuario autenticado, se redirige al usuario a la página de inicio de sesión.

Cargar Datos del Usuario:

- Se realiza una solicitud **GET** a la API del servidor para obtener los datos actuales del usuario autenticado, utilizando su **id** y el **token** de autenticación.
- Los datos obtenidos (nombre, apellidos, correo electrónico) se cargan automáticamente en el formulario, permitiendo que el usuario vea y edite su información. Además, si el usuario tiene un avatar (imagen de perfil), este se muestra junto a un botón para eliminarlo.

Envío del Formulario de Edición:

- Cuando el usuario envía el formulario, los datos se recopilan en un objeto **FormData**, que incluye los campos básicos (nombre, apellidos, email) y el archivo del avatar (si se ha seleccionado uno nuevo).
- La solicitud **PUT** se envía a la API para actualizar la información del perfil del usuario en el servidor.

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

- Si la actualización es exitosa, el usuario recibe un mensaje de confirmación y es redirigido a la página anterior.
- Si ocurre un error, se muestra un mensaje de error con los detalles del problema.

Manejo del Avatar:

- Si el usuario ya tiene un avatar, se muestra en el formulario junto con la opción de eliminarlo. Si se selecciona un nuevo archivo de avatar, este se incluirá en la solicitud de actualización.
- El avatar se muestra como una imagen pequeña, con un botón para eliminarla si el usuario lo desea.

sesion.js

Verificación de Usuario en `sessionStorage`:

- Al cargar la página, se verifica si hay un usuario autenticado en el `sessionStorage` mediante `JSON.parse(sessionStorage.getItem('user'))`.
- Si existe un objeto de usuario, se asume que el usuario está autenticado. Si no, se considera que no está autenticado.

Condiciones para Mostrar u Ocultar Botones:

- Si el usuario está autenticado:
 - Se **ocultan** los botones de **Login** y **Registro** (`btnLogin` y `btnSignUp`) usando `style.display = 'none'`.
 - Se **muestran** los botones de **Mi Cuenta** (`btnMiCuenta`) y **Cerrar Sesión** (`btnCerrarSesion`) con `hidden = false`.
 - Si el usuario tiene el rol de "GERENTE", se **muestra también** el botón de **Mi Sidrería** (`btnMiSidreria`).
- Si el usuario no está autenticado:
 - Se **muestran** los botones de **Login** y **Registro** (`btnLogin` y `btnSignUp`) utilizando `style.display = 'inline-block'`.

- Se **ocultan** los botones de **Mi Cuenta**, **Mi Sidrería** y **Cerrar Sesión** (`btnMiCuenta`, `btnMiSidreria`, `btnCerrarSesion`) utilizando `hidden = true`.

Evento de Cierre de Sesión:

- Se asigna un evento de **click** al botón **Cerrar Sesión** (`btnCerrarSesion`), que al hacer clic ejecuta lo siguiente:
 - Se eliminan los elementos `user` y `miSidreria` del `sessionStorage` usando `sessionStorage.removeItem('user')` y `sessionStorage.removeItem('miSidreria')`, lo que efectivamente cierra la sesión del usuario.
 - Luego, se redirige al usuario a la página principal (o cualquier otra página deseada) usando `window.location.href = '../html/index.html'`.

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

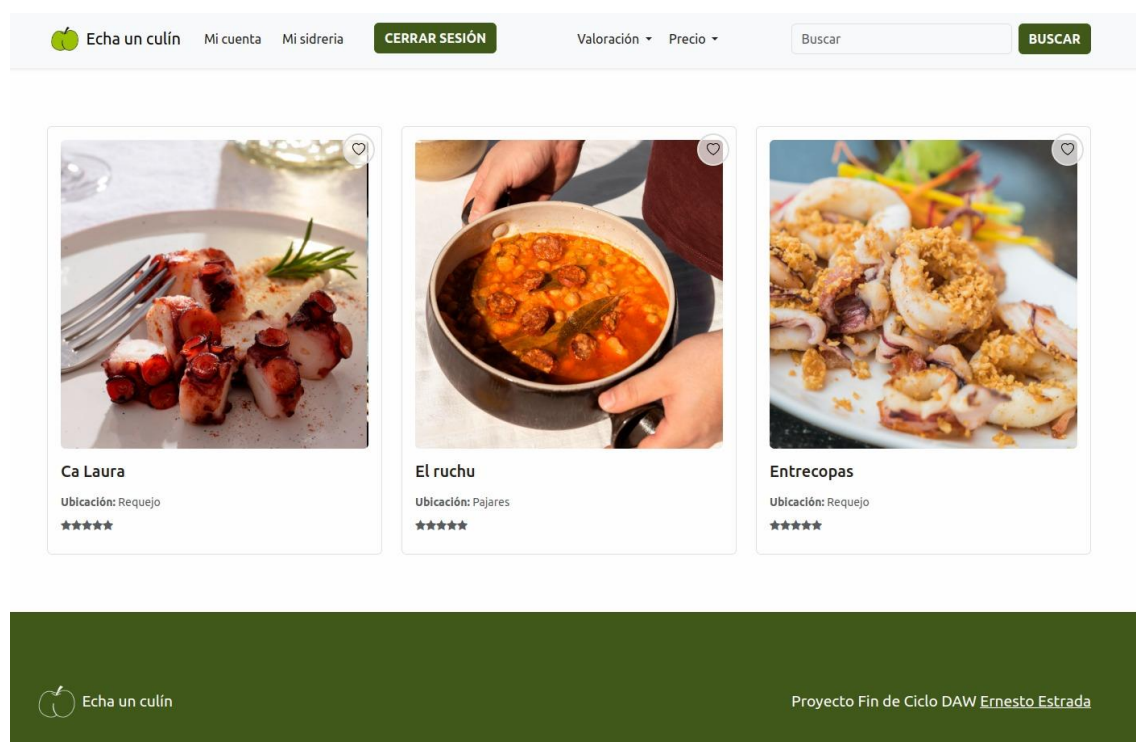
Bootstrap y hoja de estilos

El grueso de los estilos de mi aplicación se aplican con Bootstrap ya que facilita mucho que la interfaz sea responsive pero añadí también una hoja de estilos para personalizar y añadir algunos estilos de bootstrap. Personalizo las variables de color y detalles como las estrellas de las valoraciones.

Interfaz responsive para dispositivos

index

escritorio

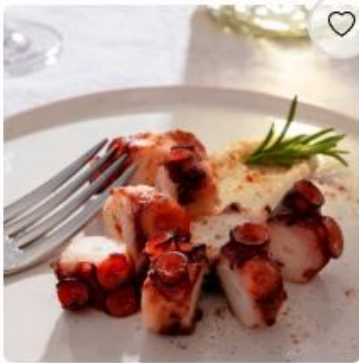


ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

tablet



Echa un culín



Ca Laura

Ubicación: Requejo

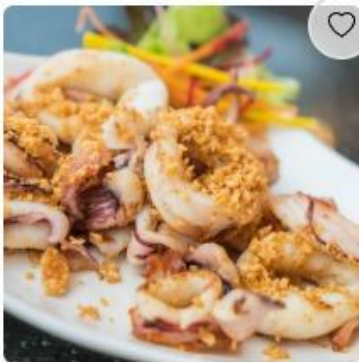
★★★★★



El ruchu

Ubicación: Pajares

★★★★★



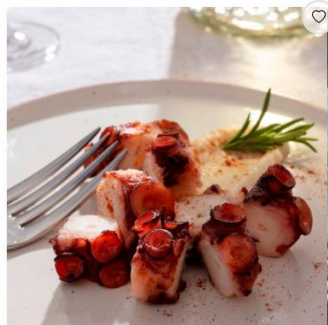
Entrecopas

Ubicación: Requejo

★★★★★

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

movil



Ca Laura


Ubicación: Requejo

★★★★★



ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Pagina sidreria

 Echa un culín

Mi cuenta

Mi sidreria

CERRAR SESIÓN

Valoración ▾

Precio ▾

Buscar

BUSCAR

El ruchu

Ubicación: Pajares

Valoración: ★★★★★

Precio por comensal: €12

Servicio de escanciado



Reseñas:

AÑADIR RESEÑA



Reseñas:

AÑADIR RESEÑA

 Roberto Fernandez

Muy buena sidreria

★★★★★

Me encanta este sitio

 Maria Gutierrez

Ya no es lo que era


★★★★★

Antes me encantaba este sitio, ahora ya no escancian tan bien

 Echa un culín

Proyecto Fin de Ciclo DAW Ernesto Estrada

Crear Review

 Echa un culín Mi cuenta Mi sidrería CERRAR SESIÓN Valoración ▾ Precio ▾ Buscar BUSCAR

Crear una Reseña


Título

Valoración

Completa este campo


Comentario

PUBLICAR RESEÑA

 Echa un culín Proyecto Fin de Ciclo DAW Ernesto Estrada

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Mi cuenta

 Echa un culín

Mi cuenta

Mi sidrería

CERRAR SESIÓN

Valoración ▾ Precio ▾

Buscar

BUSCAR

 Roberto Fernandez

EDITAR

Mis favoritos:



El ruchu

Ubicación: Pajares

Valoración: ★★★★★



Entrecopas

Ubicación: Requejo

Valoración: ★★★★★

Mis reseñas:

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

Mi sidreria



VER CARTA

Reseñas



Roberto Fernandez

La mejor sidreria

Valoración: 4 ★

Siempre voy cuando quiero tomar unes

Escribe tu respuesta...

RESPONDER



Echa un culín

Proyecto Fin de Ciclo DAW Ernesto Estrada



VER CARTA

Reseñas



Roberto Fernandez

La mejor sidreria

Valoración: 4 ★

Siempre voy cuando quiero tomar unes

Respuesta: muchas gracias por venir Roberto

EDITAR RESPUESTA

ELIMINAR RESPUESTA



Echa un culín

Proyecto Fin de Ciclo DAW Ernesto Estrada

Conclusión final

La realización del proyecto de gestión de sidrerías representa un hito significativo en el desarrollo de una plataforma moderna y funcional, diseñada para conectar a gerentes de sidrerías con sus clientes en un entorno digital seguro, eficiente y dinámico. Este proyecto no solo abarca las necesidades básicas de administración y operación de sidrerías, sino que también prioriza la experiencia del usuario y la escalabilidad del sistema, integrando prácticas modernas de desarrollo de software y tecnologías innovadoras.

Uno de los aspectos más destacados del proyecto es su arquitectura bien definida, que separa claramente las responsabilidades en capas como controladores, servicios, repositorios y modelos. Esto asegura un código limpio, modular y fácil de mantener, lo que es crucial para futuros desarrollos y actualizaciones. La implementación de DTOs y validaciones garantiza que los datos fluyan de manera consistente y segura entre las capas, minimizando errores y mejorando la experiencia de usuario.

Además, el uso de **Spring Security** con autenticación basada en **JWT** asegura que el sistema sea robusto frente a amenazas de seguridad, como accesos no autorizados. La división clara de roles entre administradores, gerentes y clientes permite un control granular sobre los permisos, asegurando que cada usuario interactúe únicamente con los recursos que le corresponden. Esta funcionalidad resulta particularmente valiosa en un sistema donde los datos personales y las opiniones de los usuarios deben protegerse cuidadosamente.

El diseño del sistema también tiene en cuenta la escalabilidad y la interoperabilidad. La implementación de rutas públicas y protegidas, junto con una configuración de **CORS** adecuada, facilita la integración con diferentes plataformas frontend, permitiendo que el sistema crezca y se adapte a las necesidades futuras. Además, la política stateless para las sesiones reduce la carga del servidor, haciéndolo más eficiente y preparado para manejar un gran volumen de solicitudes.

En términos de funcionalidad, el sistema no solo se limita a la gestión administrativa, sino que también fomenta la interacción y participación de los usuarios a través de la creación de reseñas, la gestión de favoritos y la posibilidad de responder comentarios. Este enfoque

ALUMNO/A: Ernesto Estrada Álvarez
CURSO: Desarrollo de Aplicaciones WEB
PROYECTO

fortalece la relación entre las sidrerías y sus clientes, promoviendo una mayor lealtad y satisfacción.

Finalmente, este proyecto demuestra cómo una planificación cuidadosa, el uso de tecnologías modernas y un enfoque centrado en el usuario pueden combinarse para crear una solución práctica y escalable. No solo satisface las necesidades actuales, sino que también se posiciona como una plataforma adaptable para evolucionar junto con el mercado y las demandas de los usuarios. La gestión de sidrerías ha sido modernizada con éxito, brindando un entorno digital que conecta personas, tradiciones y negocios de manera eficiente y segura.