

proyecto final del ciclo

INDICE

I. Introducción

- a. Presentación del proyecto
- b. Objetivos del proyecto
- c. Justificación del proyecto

II. Análisis de requerimientos

- a. Identificación de necesidades y requerimientos
- b. Identificación de público
- c. Estudio de mercado y competencia

III. Diseño y planificación

- a. Definición de la arquitectura del proyecto
- b. Diseño de la interfaz de usuario
- c. Planificación de las tareas y los recursos necesarios

IV. Implementación y pruebas

- a. Desarrollo de las funcionalidades del proyecto
- b. Pruebas unitarias y de integración
- c. Corrección de errores y optimización del rendimiento

V. Documentación

- a. Documentación técnica
- b. Documentación de usuario
- c. Manual de instalación y configuración

VI. Mantenimiento y evolución

- a. Plan de mantenimiento y soporte
- b. Identificación de posibles mejoras y evolución del proyecto
- c. Actualizaciones y mejoras futuras

VII. Conclusiones

- a. Evaluación del proyecto
- b. Cumplimiento de objetivos y requisitos
- c. Lecciones aprendidas y recomendaciones para futuros proyectos

VIII. Bibliografía y referencias

- a. Fuentes utilizadas en el proyecto
- b. Referencias y enlaces de interés

Introducción:

1. Presentación del proyecto:

- La página de anuncios que estamos desarrollando tiene como objetivo principal proporcionar un espacio en línea donde los usuarios puedan interactuar de manera eficiente para publicar y buscar anuncios de una amplia gama de categorías. Desde la venta de productos, nuestra plataforma se concibe como un centro virtual donde la comunidad pueda conectar, explorar y participar en diversas oportunidades.
- En este sentido, la página estará diseñada para alojar anuncios variados que abarquen desde artículos de segunda mano hasta propiedades inmobiliarias,. Esto garantiza que la plataforma sea útil y relevante para una amplia gama de usuarios, independientemente de sus necesidades o intereses específicos.
- La audiencia objetivo de nuestra página de anuncios es diversa y abarca a personas de todas las edades y antecedentes, que tienen diferentes motivaciones para usar la plataforma. aquellos que buscan oportunidades de compra y venta, nuestra página está diseñada para satisfacer las necesidades de una amplia audiencia.

2. Objetivos del proyecto:

El proyecto tiene como objetivo principal desarrollar una plataforma de anuncios en línea que satisfaga las necesidades de los usuarios y proporcione una experiencia integral. Para lograr este objetivo principal, se han establecido los siguientes objetivos específicos:

1. Proporcionar una plataforma fácil de usar para publicar y buscar anuncios: El primer objetivo es garantizar que la plataforma sea accesible y amigable para los usuarios, permitiéndoles navegar fácilmente por la página, publicar sus propios anuncios y encontrar información relevante de manera rápida y eficiente.
2. Fomentar la interacción entre compradores y vendedores: Se busca promover un entorno colaborativo donde los usuarios puedan comunicarse entre sí, realizar consultas sobre productos y establecer relaciones comerciales sólidas. Esto incluye facilitar la comunicación directa entre compradores y vendedores a través de herramientas de mensajería integradas en la plataforma.
3. Generar ingresos a través de opciones publicitarias: Otro objetivo clave del proyecto es monetizar la plataforma mediante la implementación de opciones publicitarias. Esto incluye la inclusión de espacios publicitarios dentro de la página de anuncios para que los anunciantes puedan promocionar sus productos, lo que a su vez genera ingresos para el mantenimiento y la mejora continua de la plataforma.

Además de estos objetivos principales, se han identificado algunas funcionalidades clave que serán fundamentales para el éxito de la plataforma. Estas incluyen:

- **Sistema de registro y inicio de sesión para usuarios:** Se implementará un sistema de registro seguro y un proceso de inicio de sesión para que los usuarios puedan crear y gestionar sus cuentas personalizadas en la plataforma.
- **Capacidad de publicar y gestionar anuncios:** Los usuarios tendrán la capacidad de crear, editar y eliminar sus propios anuncios, así como gestionar la información relacionada con ellos, como imágenes, descripciones y detalles de contacto.

- **Funciones de búsqueda avanzada:** Se desarrollará un sistema de búsqueda avanzada que permita a los usuarios filtrar y encontrar anuncios específicos según sus criterios de búsqueda, como categoría, ubicación, precio, etc.
- **Sistema de mensajería para facilitar la comunicación entre usuarios:** Se integrará un sistema de mensajería dentro de la plataforma para que los usuarios puedan comunicarse de manera directa y segura entre sí, facilitando la negociación y la coordinación de transacciones.

Estos objetivos y funcionalidades clave están diseñados para garantizar que la plataforma de anuncios cumpla con las expectativas de los usuarios, fomente la interacción y la participación activa, y genere ingresos de manera sostenible a largo plazo.

3. Justificación del proyecto:

La necesidad de una plataforma centralizada donde los usuarios puedan encontrar y publicar anuncios de manera conveniente es evidente en el panorama actual. El crecimiento exponencial del comercio electrónico y la digitalización de las transacciones comerciales han generado una demanda cada vez mayor de espacios virtuales que faciliten la interacción entre compradores y vendedores.

En este contexto, el desarrollo de nuestra página de anuncios se justifica por varias razones fundamentales:

1. **Centralización y conveniencia:** En un mercado saturado de información dispersa, es crucial contar con una plataforma centralizada que reúna una amplia variedad de anuncios en un solo lugar. Esto simplifica el proceso de búsqueda para los usuarios, ya que no necesitan navegar por múltiples sitios web o aplicaciones para encontrar lo que están buscando.
2. **Facilitación del comercio electrónico:** Con el auge del comercio electrónico, los consumidores y los vendedores buscan plataformas que les permitan realizar transacciones de manera rápida, segura y conveniente. Nuestra página de anuncios proporcionará un entorno seguro y confiable donde los usuarios puedan comprar, vender e intercambiar una amplia gama de productos y servicios con facilidad.
3. **Atención a una demanda creciente:** El mercado en línea está experimentando un crecimiento constante, impulsado por la comodidad y la accesibilidad que ofrece el comercio electrónico. Nuestra página de anuncios se posiciona para satisfacer esta demanda creciente, brindando a los usuarios una plataforma robusta y eficiente para sus necesidades de compra, venta y búsqueda de productos.

Análisis de requerimientos:

1. Identificación de necesidades y requerimientos:

1. Necesidades de los usuarios:

- **Publicadores de anuncios:**
 - Necesitan una plataforma fácil de usar para publicar sus anuncios de manera efectiva.

- Desean un proceso de publicación rápido y sencillo, con opciones de personalización para sus anuncios.
- Buscan herramientas para administrar y editar sus anuncios de manera conveniente.
- **Usuarios que buscan anuncios:**
 - Esperan una experiencia de búsqueda intuitiva y eficiente para encontrar anuncios relevantes.
 - Quieren filtros avanzados y opciones de búsqueda para refinar sus resultados de búsqueda.
 - Valoran la autenticidad y la veracidad de los anuncios, así como la capacidad de contactar fácilmente con los publicadores.

2. Requerimientos funcionales:

- **Publicación de anuncios:**
 - Los usuarios deben poder crear cuentas y acceder a un panel de control para gestionar sus anuncios.
 - Se requiere un formulario de publicación de anuncios con campos para título, descripción, categoría, precio, etc.
 - Debe haber opciones para cargar imágenes o archivos relacionados con el anuncio.
- **Búsqueda y filtrado de anuncios:**
 - Los usuarios deben poder buscar anuncios utilizando palabras clave, categorías o filtros específicos.
 - Se necesita un sistema de filtrado avanzado que permita a los usuarios refinar los resultados de búsqueda según sus preferencias.
- **Interacción con anuncios:**
 - Los usuarios deben poder ver los detalles completos de un anuncio individual.
 - Se requiere un sistema de comentarios o preguntas para que los usuarios puedan interactuar con los publicadores.
 - Debe haber opciones para que los usuarios guarden anuncios favoritos o comparten en redes sociales.

3. Requerimientos no funcionales:

- **Seguridad:**
 - Se debe implementar un sistema de autenticación y autorización seguro para proteger los datos de los usuarios y la integridad del sistema.
 - Todas las transacciones y comunicaciones deben estar encriptadas para garantizar la privacidad y seguridad de la información.
- **Rendimiento:**
 - La plataforma debe ser capaz de manejar un alto volumen de tráfico y consultas sin experimentar tiempos de carga prolongados.
 - Se deben implementar técnicas de optimización para garantizar un rendimiento óptimo del sitio web.
- **Escalabilidad:**
 - La arquitectura de la aplicación debe ser escalable, lo que permitirá agregar nuevas funcionalidades y manejar un aumento en la cantidad de usuarios sin problemas.
 - Se debe considerar el uso de tecnologías y herramientas que faciliten la escalabilidad, como el uso de servicios en la nube y la implementación de patrones de diseño adecuados.

4. Integración con tecnologías específicas de Spring Boot:

- Utilización de Spring Security para la gestión de la seguridad y la autenticación de usuarios.
- Implementación de Spring Data JPA para el acceso a la base de datos y la gestión de entidades.
- Uso de Thymeleaf como motor de plantillas para la generación dinámica de páginas HTML.
- Implementación de Spring MVC para el desarrollo del controlador y la lógica de negocio de la aplicación.

○

2. Identificación de público:

1. Usuarios potenciales de la plataforma:

- **Publicadores de anuncios:**
 - Empresas y comercios locales que desean promocionar sus productos Personas individuales que buscan vender artículos de segunda mano.
- **Usuarios que buscan anuncios:**
 - Personas que están interesadas en comprar productos en línea.
 - Consumidores que buscan ofertas o descuentos en una variedad de categorías, desde bienes raíces hasta vehículos .
 - Usuarios que buscan información sobre productos.

2. Características demográficas y comportamientos:

- **Publicadores de anuncios:**
 - Edades variadas, desde jóvenes emprendedores hasta empresarios establecidos y jubilados.
 - Diversos niveles de educación y experiencia en tecnología, desde usuarios principiantes hasta expertos en internet.
 - Intereses y necesidades específicas según la naturaleza de sus productos o servicios, como la ubicación geográfica, el presupuesto disponible y el público objetivo.
- **Usuarios que buscan anuncios:**
 - Amplio rango de edades y perfiles demográficos, desde jóvenes estudiantes hasta personas mayores en busca de servicios especializados.
 - Diferentes niveles de experiencia en compras en línea, desde consumidores habituales hasta aquellos menos familiarizados con la tecnología.
 - Comportamientos de búsqueda diversos, desde usuarios que buscan información detallada sobre productos específicos hasta aquellos que exploran diferentes categorías en busca de inspiración.

3. Adaptación de la plataforma para satisfacer las necesidades del público:

- **Personalización de la experiencia de usuario:**
 - Implementación de perfiles de usuario que permitan a los publicadores gestionar sus anuncios y a los usuarios guardar sus búsquedas favoritas.
 - Ofrecimiento de recomendaciones personalizadas basadas en el historial de búsqueda y los intereses de cada usuario.
- **Diseño intuitivo y accesible:**
 - Desarrollo de una interfaz de usuario intuitiva y fácil de usar, con navegación clara y funciones de búsqueda avanzadas.
 - Adopción de un diseño "responsive" que garantice una experiencia consistente en todos los dispositivos, incluyendo computadoras de escritorio, tabletas y dispositivos móviles.
- **Comunicación efectiva:**
 - Incorporación de funciones de mensajería o comentarios que faciliten la comunicación entre los publicadores y los usuarios interesados en sus anuncios.
 - Integración de sistemas de notificación para alertar a los usuarios sobre nuevos anuncios relevantes o mensajes importantes.
 -

3. Estudio de mercado y competencia:

- Se ha realizado un análisis exhaustivo del mercado de páginas de anuncios similares para comprender las tendencias, preferencias del usuario y áreas de oportunidad.
- La competencia en el mercado de páginas de anuncios varía en términos de diseño, funcionalidades y alcance geográfico. Se han identificado oportunidades para diferenciar nuestra página mediante características únicas y una experiencia de usuario mejorada.

Diseño y planificación:

1. Definición de la arquitectura del proyecto:

1. Arquitectura general:

- **Arquitectura de tres capas:**

- Se implementará una arquitectura de tres capas que consta de la capa de presentación (Frontend), la capa de lógica de negocio (Backend) y la capa de almacenamiento de datos (Base de datos).
- Esta arquitectura modular facilita la escalabilidad, el mantenimiento y la reutilización de componentes.

2. Capa de presentación (Frontend):

- **Tecnologías utilizadas:**

- Se desarrollará el Frontend utilizando tecnologías web estándar como HTML5, CSS3 y JavaScript.
- Se utilizará Thymeleaf como motor de plantillas para generar vistas dinámicas y renderizar datos desde el Backend.

- **Diseño responsive:**

- Se implementará un diseño responsive que garantice una experiencia de usuario consistente en una variedad de dispositivos, incluyendo computadoras de escritorio, tabletas y dispositivos móviles. tailwind es un framework como (bootstrap)

- **Interacción asíncrona:**

- Se utilizarán técnicas asíncronas como peticiones typescript para mejorar la velocidad de carga y la fluidez de la interfaz de usuario.

3. Capa de lógica de negocio (Backend):

- **Framework Spring Boot:**

- Se utilizará Spring Boot para el desarrollo del Backend debido a su facilidad de configuración, su amplio ecosistema de bibliotecas y su soporte para la construcción de aplicaciones web robustas.

- **Seguridad:**

- Se implementará Spring Security para gestionar la autenticación de usuarios, la autorización de accesos y la protección contra vulnerabilidades de seguridad.

- **API RESTful:**

- Se expondrá una API RESTful para permitir la comunicación entre el Frontend y el Backend, utilizando principios REST para la creación, lectura, actualización y eliminación (CRUD) de recursos.

4. Capa de almacenamiento de datos (Base de datos):

- **Base de datos relacional:**

- Se utilizará una base de datos relacional para almacenar la información de los usuarios, los anuncios y otros datos relacionados con la plataforma.

- Se emplea MySQL o Oracle o Postgres como sistemas de gestión de bases de datos, debido a su amplia adopción y compatibilidad con Spring Boot.
- **ORM con Spring Data JPA:**
 - Se utilizará Spring Data JPA para facilitar el acceso a la base de datos y la manipulación de objetos Java como entidades persistentes.
 - Se mapean las relaciones entre entidades mediante anotaciones JPA para asegurar la integridad de los datos y mantener la consistencia de la base de datos.

2. Diseño de la interfaz de usuario:

1. Diseño visual:

- **Aspecto y sensación modernos:**
 - Se diseñará una interfaz de usuario moderna y atractiva que refleje profesionalismo y facilite la navegación intuitiva para los usuarios.
- **Paleta de colores y tipografía:**
 - Se seleccionará una paleta de colores atractiva y coherente que transmita la marca y mejore la legibilidad de la interfaz.
 - Se utilizarán tipos de letra legibles y accesibles para mejorar la experiencia de lectura del contenido.
- **Elementos visuales coherentes:**
 - Se mantendrá la coherencia en el diseño de todos los elementos visuales, incluyendo botones, formularios, íconos y barras de navegación, para una experiencia de usuario uniforme.

2. Navegación y estructura:

- **Menús de navegación claros:**
 - Se implementarán menús de navegación claros y fáciles de entender para ayudar a los usuarios a encontrar rápidamente la información que están buscando.
- **Estructura jerárquica:**
 - Se establecerá una estructura de navegación jerárquica que organice el contenido de manera lógica y facilite la exploración de diferentes secciones de la plataforma.
- **Breadcrumbs y links relacionados:**
 - Se incluirán breadcrumbs y enlaces relacionados para orientar a los usuarios dentro del sitio y facilitar la navegación hacia atrás o hacia adelante en la jerarquía de páginas.

3. Funcionalidades principales:

- **Búsqueda y filtrado avanzados:**
 - Se proporcionarán opciones de búsqueda avanzada y filtros para permitir a los usuarios refinar los resultados de búsqueda según sus necesidades específicas.
- **Visualización de anuncios:**
 - Se diseñarán páginas de detalle de anuncios que muestren información detallada sobre cada anuncio de manera clara y atractiva, incluyendo imágenes, descripciones y detalles relevantes.
- **Formularios de publicación de anuncios:**
 - Se desarrollarán formularios de publicación de anuncios intuitivos y fáciles de usar que guíen a los usuarios a través del proceso de creación de anuncios de manera paso a paso.

4. Interacción y retroalimentación:

- **Feedback visual:**
 - Se proporcionarán mensajes de retroalimentación visual para confirmar acciones exitosas, alertar sobre errores o proporcionar instrucciones claras sobre cómo proceder en determinadas situaciones.
- **Interacción intuitiva:**
 - Se implementarán elementos interactivos como botones, deslizadores y menús desplegables de manera intuitiva y consistente para mejorar la experiencia de usuario y fomentar la participación.
- **Comunicación efectiva:**
 - Se integrarán funciones de mensajería o comentarios que permitan a los usuarios comunicarse entre sí y proporcionar retroalimentación sobre los anuncios.

5. Diseño responsive:

- **Compatibilidad multiplataforma:**
 - Se garantizará que la interfaz de usuario sea compatible con una variedad de dispositivos y tamaños de pantalla, incluyendo computadoras de escritorio, tabletas y dispositivos móviles.
- **Adaptación dinámica:**
 - Se utilizarán técnicas de diseño responsive para adaptar dinámicamente la disposición y el contenido de la interfaz según el tamaño y la resolución de la pantalla del dispositivo del usuario.
 -

3. Planificación de las tareas y los recursos necesarios:

1. Identificación de tareas:

- **Análisis y diseño:**
 - Definir los requisitos del proyecto.
 - Realizar el diseño de la arquitectura y la interfaz de usuario.
 - Identificar las tecnologías y herramientas necesarias.
- **Desarrollo:**
 - Configurar el entorno de desarrollo.
 - Implementar la lógica de negocio del Backend.
 - Desarrollar el Frontend y la interfaz de usuario.
 - Integrar la base de datos y realizar pruebas de integración.
- **Pruebas:**
 - Realizar pruebas unitarias y de integración.
 - Realizar pruebas de usabilidad y rendimiento.
 - Corregir errores y realizar ajustes según sea necesario.

2. Estimación de tiempos:

- **Análisis y diseño:**
 - Duración estimada: 1 semana..
- **Desarrollo:**
 - Configuración del entorno de desarrollo: 1 día.
 - Implementación del Backend: 1 semanas.
 - Desarrollo del Frontend: 1 semanas.
 - Integración y pruebas: 1 semanas.
- **Pruebas:**
 - Pruebas unitarias e integración: 2 días.

- Pruebas de usabilidad y rendimiento: 2 días.
-

3. Asignación de recursos:

• Recursos técnicos:

- Hardware: Equipos de desarrollo con capacidades adecuadas para ejecutar el entorno de desarrollo y las pruebas.
- Software: Herramientas de desarrollo como IDEs, control de versiones, git herramientas de diseño, draw.io , bases de datos , etc.

Implementación y pruebas

a. Desarrollo de las funcionalidades del proyecto:

- Durante esta etapa, se implementarán todas las funcionalidades identificadas en la fase de diseño. Esto incluirá el desarrollo del frontend y backend de la plataforma, así como la integración con la base de datos.
- Para el frontend, se construirán las páginas y componentes necesarios utilizando las tecnologías web mencionadas anteriormente, como HTML, CSS , tailwind, JavaScript , con framework angular. Se asegurará de que la interfaz de usuario sea intuitiva, receptiva y cumpla con los requisitos de diseño establecidos.
- En el backend, se configurarán los controladores y servicios utilizando Spring Boot. Se implementarán las operaciones CRUD (Crear, Leer, Actualizar, Eliminar, buscar, paginar) para los anuncios y usuarios, así como la lógica de negocio necesaria para la funcionalidad de la plataforma.
- Se integrarán todas las partes del sistema, asegurando que la comunicación entre el frontend y el backend sea fluida y eficiente.

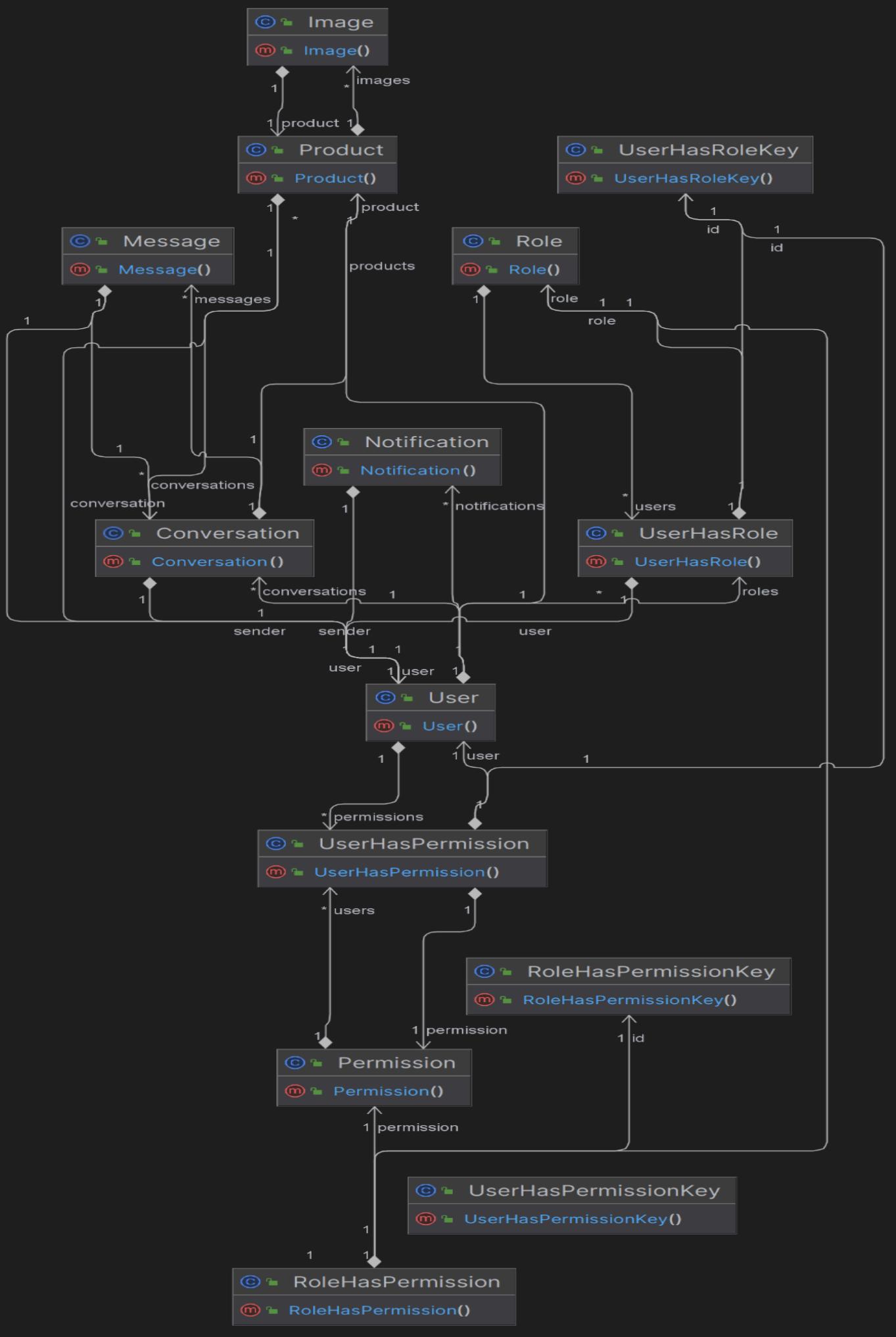
b. Pruebas unitarias y de integración:

- Se llevarán a cabo pruebas unitarias para cada componente del sistema, tanto en el frontend como en el backend. Las pruebas unitarias garantizarán que cada función individual opere correctamente según lo previsto.
- Además, se realizan pruebas de integración para verificar que todos los componentes del sistema funcionen correctamente juntos como un todo. Esto incluirá pruebas de extremo a extremo para simular el flujo de usuario completo, desde la creación de una cuenta hasta la publicación y búsqueda de anuncios.
- Durante estas pruebas, se identificarán y corregirán cualquier error o comportamiento inesperado que se encuentre en el sistema.

c. Corrección de errores y optimización del rendimiento:

- Despues de completar las pruebas, se abordarán todos los errores y problemas identificados. Esto puede implicar cambios en el código, ajustes en la configuración del servidor o actualizaciones en la base de datos.
- Además de corregir errores, se buscarán oportunidades para optimizar el rendimiento del sistema. Esto puede incluir la optimización del tiempo de carga de la página, la reducción del uso de recursos del servidor y la mejora de la velocidad de respuesta de la aplicación.
- Una vez que se hayan realizado todas las correcciones y optimizaciones necesarias, se realizarán pruebas adicionales para verificar que el sistema funcione según lo esperado y que se hayan logrado mejoras significativas en el rendimiento.

Relaciones De los Entities en spring boot



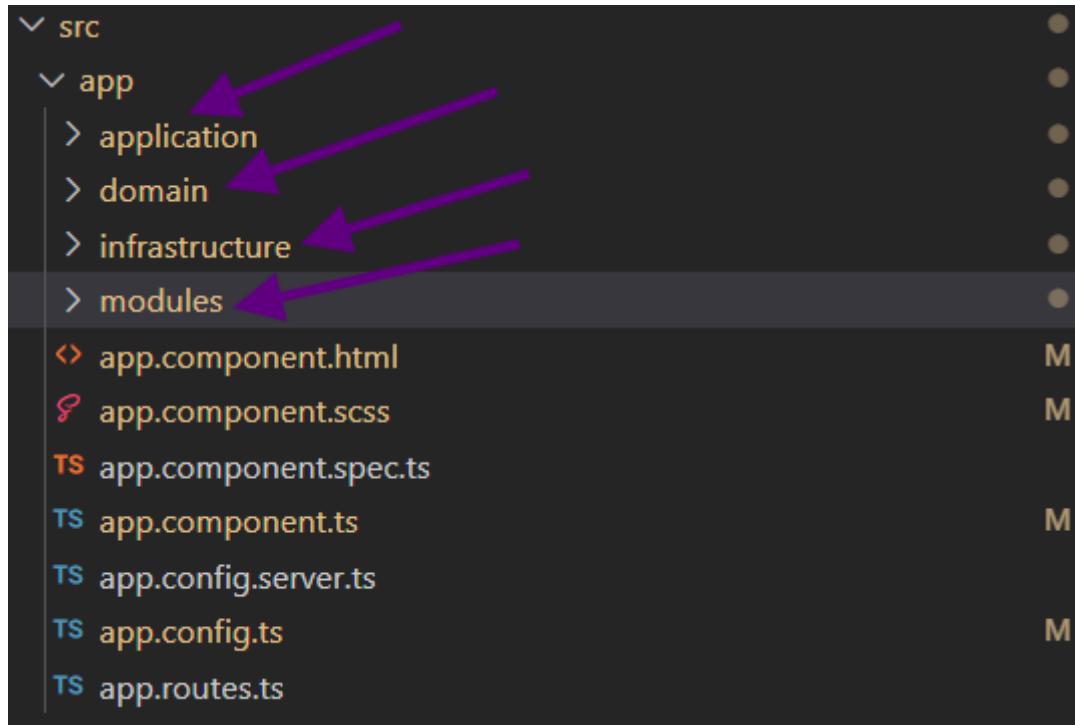
V. Documentación

a. Documentación técnica

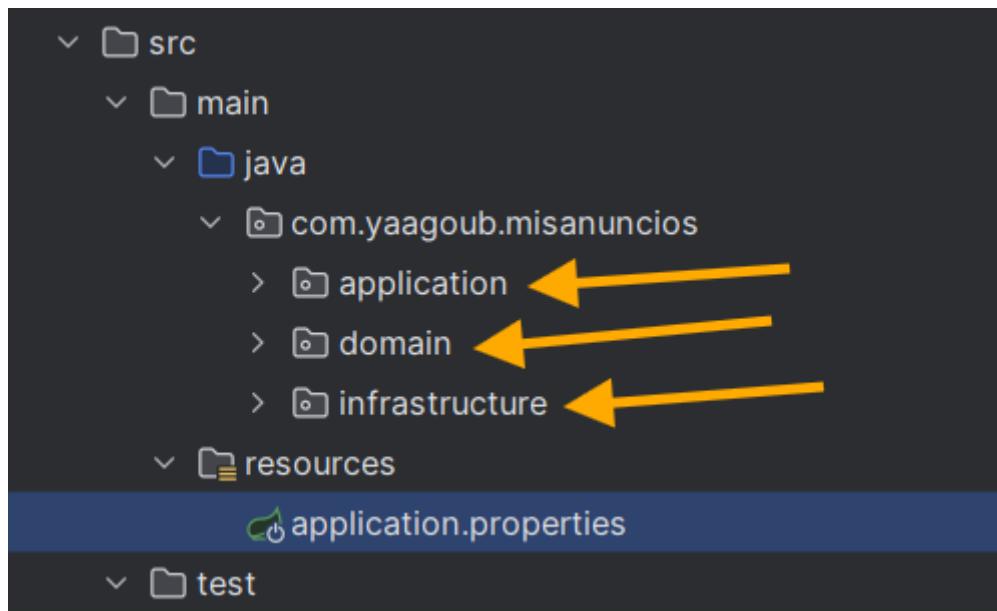
Angular y spring boot (Arquitectura Hexagonal)

proyecto hecho con arquitectura Hexagonal separado en tres capas application , domain , infrastructure modules

1) angular



2) spring boot

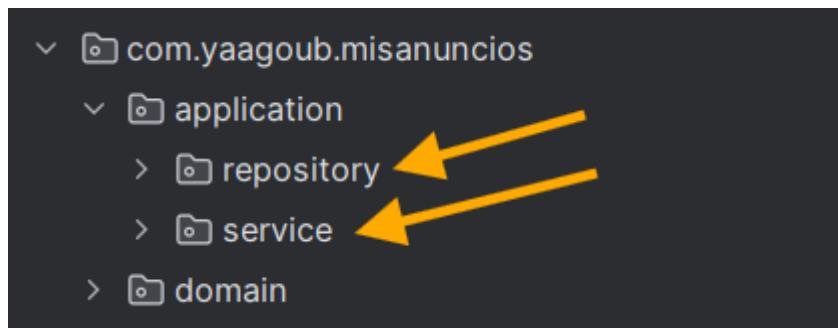


capa application : La capa de aplicación es el responsable de implementar los casos de uso y lógica de negocio específicos de nuestra aplicación

1) angular



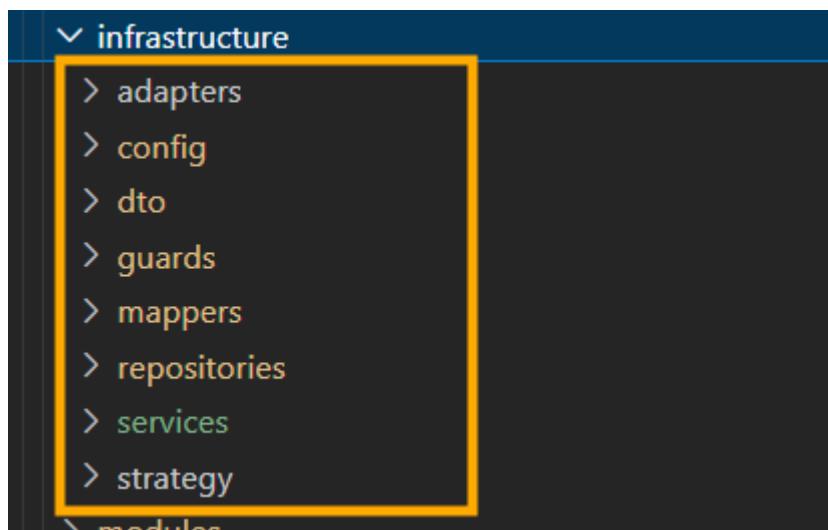
2) spring boot



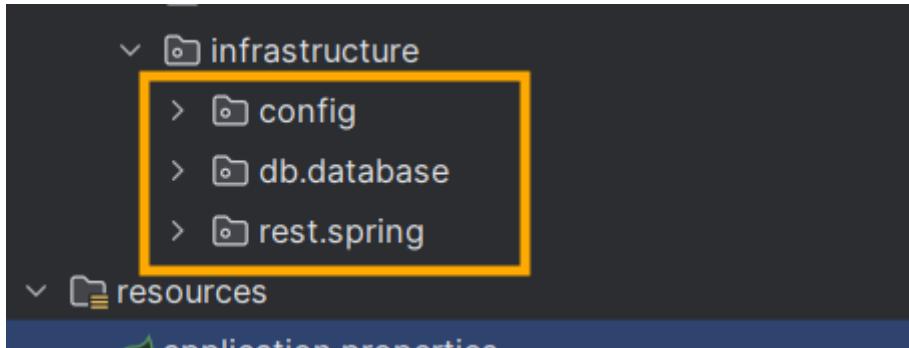
capa de infrastructure

La capa de infraestructura contiene todos los detalles técnicos y aspectos de bajo nivel, como el acceso a bases de datos, interacción con servicios externos, interfaces de usuario

1) angular

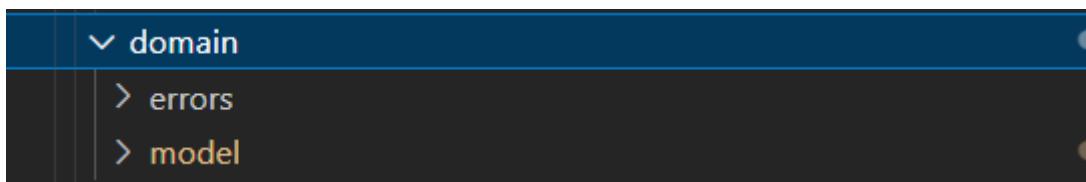


2) spring boot

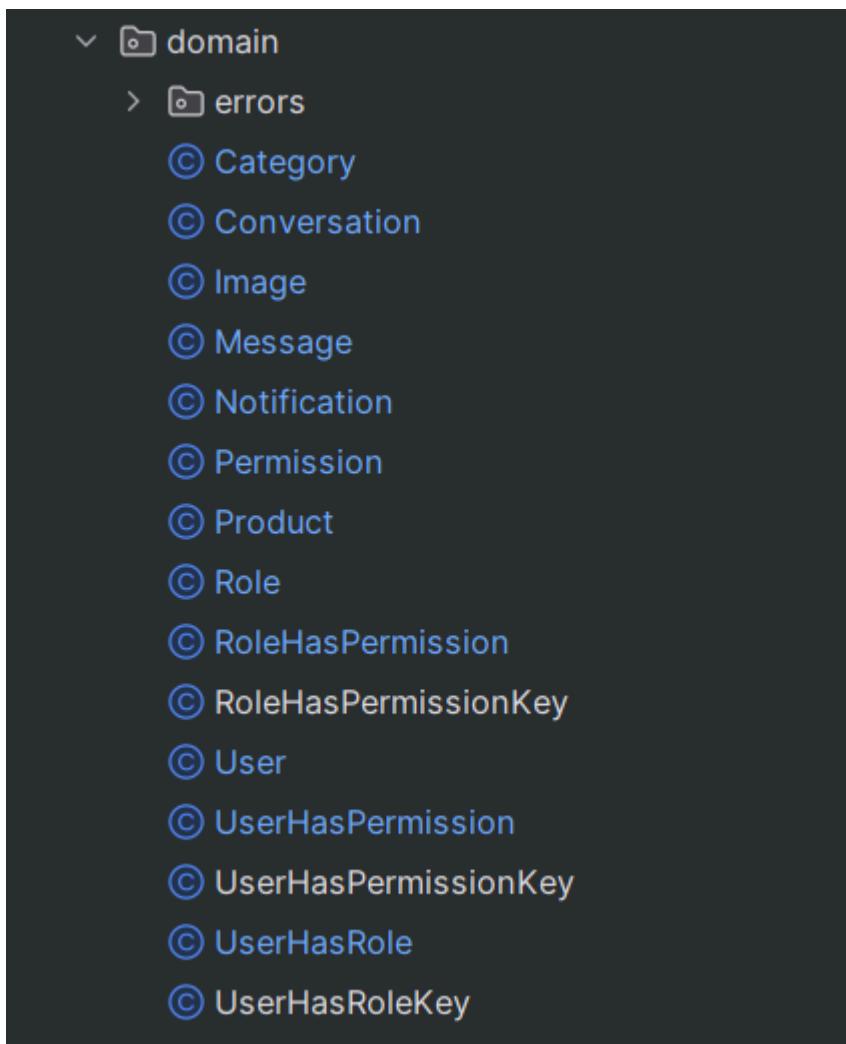


Capa de Dominio : La capa de dominio representa el corazón de la aplicación y su función principal es representar y encapsular las entidades centrales y los conceptos de dominio

1) angular



2) spring boot



- Tecnologías utilizadas: Spring Boot, Spring Data JPA, Spring Security, MySQL., angular , websocket
- Especificaciones técnicas de los servicios y APIs RESTful.

```
▲ OYaagoub
@GetMapping(@RequestMapping("notifications"))
public ResponseEntity<Object> getNotifications() throws JsonProcessingException {
    Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
    Object principal = authentication.getPrincipal();
    if(principal instanceof User){
        User user = (User) principal;
        var response = notificationService.getNotificationsByUser(user.getId()).stream().map(notification
            return notificationDtoMapper.toDto(notification,context);
        })
        .collect(Collectors.toList());
        return ResponseEntity.ok().body(response);
    }else{
        return ResponseEntity.ok().body("Authentication request");
    }
}
```

- Detalles de configuración del entorno de desarrollo y producción.

c. Manual de instalación y configuración

- Instalación del servidor de aplicaciones: Configuración de Spring Boot.

- clonar el proyecto desde el repositorio

<https://github.com/OYaagoub/repositorioYaagoub>

- installer dependencias del angular y spring boot

para spring boot (dentro la carpeta del proyecto)

```
mvn install
```

para angular (dentro la carpeta del proyecto)

```
npm install
```

- crear la base de datos con nombre “misanuncios” en Mysql, instalar MYSQL (XAMPP)

cambiar los variables del entorno

```
spring.datasource.url=${SPRING_DATASOURCE_URL}&createDatabaseIfNotExist=true
spring.datasource.username=${SPRING_DATASOURCE_USERNAME}
spring.datasource.password=${SPRING_DATASOURCE_PASSWORD}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

con estos datos

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/misanuncios
2 spring.datasource.username=root
3 spring.datasource.password=
```

cambiar o añadir la dominio del front-end para websocket en el proyecto de spring Boot (atencion al puerto)

```
J WebSocketConfig.java ×
src > main > java > com > yaagoub > misanuncios > infrastructure > config > authentication > J WebSocketConfig.java
1 package com.yaagoub.misanuncios.infrastructure.config.authentication;
2 import org.springframework.context.annotation.Configuration;
3 import org.springframework.messaging.simp.config.ChannelRegistration;
4 import org.springframework.messaging.simp.config.MessageBrokerRegistry;
5 import org.springframework.security.config.annotation.web.messaging.MessageSecurityMetadataSourceRegistry;
6 import org.springframework.security.config.annotation.web.socket.AbstractSecurityWebSocketMessageBrokerConfigurer;
7 import org.springframework.web.socket.config.annotation.EnableWebSocketMessageBroker;
8 import org.springframework.web.socket.config.annotation.StompEndpointRegistry;
9 import org.springframework.web.socket.config.annotation.WebSocketMessageBrokerConfigurer;
10
11 @Configuration
12 @EnableWebSocketMessageBroker
13 public class WebSocketConfig implements WebSocketMessageBrokerConfigurer {
14
15     @Override
16     public void configureMessageBroker(MessageBrokerRegistry config) {
17         config.enableSimpleBroker("/topic");
18         config.setApplicationDestinationPrefixes("/app");
19     }
20
21     @Override
22     public void registerStompEndpoints(StompEndpointRegistry registry) {
23         registry.addEndpoint("/chat").setAllowedOrigins("http://localhost:4201","http://localhost:4200").withSockJS();
24     }
25
26
27 }
```

cambiar al dominio del microservicio en proyecto angular

en mi caso hay que cambiar **192.168.1.133:8080** al **localhost:8080**

```

    > application
    > domain
    < infrastructure
      > adapters
      < config
        TS config.ts
          > dto
          > guards
          > mappers
          > repositories
          > services

```

```

21   socketUrl: "//192.168.1.133:8080/chat",
22 };
23 */
24 export const config: Config = {
25   apiUrl: "http://192.168.1.133:8080/api",
26   contentUrl: "http://192.168.1.133:8080/api/v3/content",
27   authUrl: "http://192.168.1.133:8080/api/v3/auth",
28   auth: "token",
29   socketUrl: "//192.168.1.133:8080/chat",
30 };
31

```

para ejecutarlo

para angular

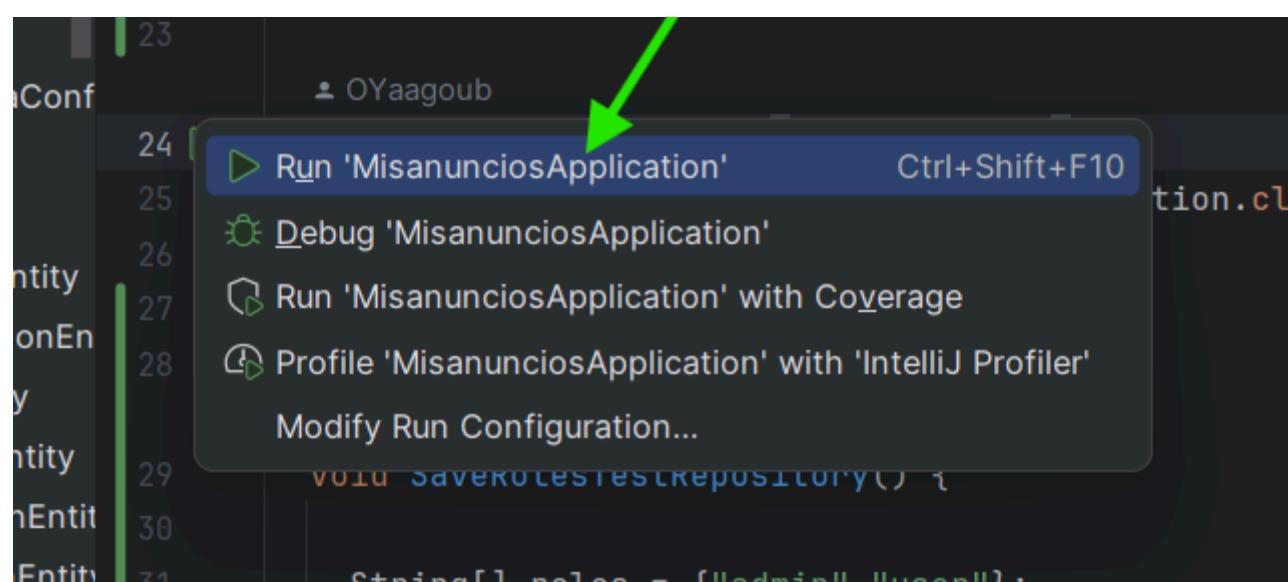
```
ng serve
```

para spring boot

```

3   > import ...
15
16 @SpringBootApplication(scanBasePackages = "com.yaagoub.misanuncios.infrastructure")
17 @EntityScan(basePackages = "com.yaagoub.misanuncios.infrastructure.db.database.model")
18 public class MisanunciosApplication {
19   @Autowired
20   private SpringDataRoleRepository springDataRoleRepository;
21   @Autowired
22   private SpringDataCategoryRepository springDataCategoryRepository;
23
24   public static void main(String[] args) {
25     SpringApplication.run(MisanunciosApplication.class);
26   }
27
28
1 usage now*

```



Para poder desplegarlo en kubernetes

- clonar el proyecto desde el repositorio

<https://github.com/OYaagoub/repositorioYaagoub>

El Deployment de un Proyecto de angular Docker hub + Kubernetes 😊

1) crear un fichero Dockerfile

```
FROM node:alpine

WORKDIR /usr/src/app

COPY . /usr/src/app

RUN npm install -g @angular/cli

RUN npm install

CMD ["ng", "serve", "--host", "0.0.0.0"]
```

2 . construir la imagen docker hub

```
docker build -t yaagoub/misanuncios-ui:latest .
```

3) crear un namespace para aplicaciones de front-end en kubernetes

```
apiVersion: v1
kind: Namespace
metadata:
  name: angular
```

4) crear un deployment para a aplicación , atención con los puertos

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: misanuncios-ui
  namespace: angular
  labels:
    app: misanuncios-ui
spec:
  replicas: 1
  selector:
    matchLabels:
      app: misanuncios-ui
```

```
template:
  metadata:
    labels:
      app: misanuncios-ui
  spec:
    containers:
      - name: misanuncios-ui
        image: yaagoub/misanuncios-ui:latest
        ports:
          - containerPort: 4200
```

5) crear un service de tipo Load Balancer

```
apiVersion: v1
kind: Service
metadata:
  namespace: angular
  name: misanuncios-ui
spec:
  type: LoadBalancer
  ports:
    - port: 4201
      targetPort: 4200
  selector:
    app: misanuncios-ui
```

6) y ahora tenemos todo esta preparado arrancamos el fichero del deployment y service

con comando :

```
kubectl apply -f file.yaml
```

7)

intentaremos meter <http://localhost:4201/>

Coches

Furgonetas,Diesel,Coches eléctricos,Audi A5,BMW X5,Chevrolet,Citroen Xsara,Fiat Coupe, Ford,Ford Fiesta,Hyundai Getz,Jeep,Lada Niva,Mazda,Mercedes Vito, Mitsubishi Montero,Nissan,Nissan Qashqai,Opel Corsa,Peugeot 3008,Porsche,Renault Clio,Renault Scenic,Renault Trafic,SEAT,SEAT Cordoba,Smart,Suzuki Samurai,Suzuki Vitara,Toyota,Volkswagen,Volkswagen Passat,Volvo,Camión

El Deployment de un Proyecto de Spring Boot Docker hub + Kubernetes 😍😎😊😴

1) configurar variables de entorno en application.yaml/properties

```
1  spring.datasource.url=${SPRING_DATASOURCE_URL}
2  spring.datasource.username=${SPRING_DATASOURCE_USERNAME}
3  spring.datasource.password=${SPRING_DATASOURCE_PASSWORD}
```

2) crear un fichero para Dockerfile

```
FROM openjdk:17-jdk-slim
VOLUME /tmp
COPY misanuncios/target/misanuncios-0.0.1-SNAPSHOT.jar app.jar
EXPOSE 8080
CMD ["java", "-jar", "app.jar"]
```

3) generar paquete .jar del proyecto

```
./mvnw clean package -DskipTests
```

4) construir la imagen del docker

```
docker build -t yaagoub/misanuncios:latest .
```

5) crear namespace en kubernetes

```
apiVersion: v1
kind: Namespace
metadata:
  name: springboot
```

6) crear un deployment con variables de entorno si no puedes usar volt engineer para gestionarlo

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: misanuncios
  namespace: springboot
  labels:
    app: misanuncios
spec:
  replicas: 1
```

```
  selector:
    matchLabels:
      app: misanuncios
  template:
    metadata:
      labels:
        app: misanuncios
  spec:
    containers:
      - name: misanuncios
        image: yaagoub/misanuncios:latest
        ports:
          - containerPort: 8080
        env:
          - name: SPRING_DATASOURCE_URL
            value: "jdbc:mysql://10.97.11.232:3306/misanuncios"
          - name: SPRING_DATASOURCE_USERNAME
            value: "root"
          - name: SPRING_DATASOURCE_PASSWORD
            value: "1234"
```

7) crearemos el service de tipo LoadBalancer

```
apiVersion: v1
kind: Service
metadata:
  namespace: springboot
  name: misanuncios
spec:
```

```

type: LoadBalancer
ports:
  - port: 8080
    targetPort: 8080
selector:
  app: misanuncios

```

8) comprobaremos intentando meter al swagger

<http://localhost:8080/swagger-ui/index.html>

9) el siguiente es hacer un deploy al front-end angular

Crear database usando Kubernetes

1) iniciar cluster

```
C:\Users\oyaag>minikube start --cpus 4 --memory 5192
W0607 17:52:13.389554 18664 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\oyaag\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.
* minikube v1.33.1 on Microsoft Windows 11 Home 10.0.22631.3593 Build 22631.3593
* Automatically selected the docker driver. Other choices: virtualbox, ssh
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.44 ...
* Creating docker container (CPUs=4, Memory=5192MB) ...
* Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
- Generating certificates and keys ...
- Booting up control plane ...
- Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

2) habilitar el controlador ingress y abrir el túnel para loadbalancer

```
C:\Users\oyaag>minikube addons enable ingress
W0607 17:53:47.520819 29436 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\oyaag\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.
* ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
* After the addon is enabled, please run "minikube tunnel" and your ingress resources would be available at "127.0.0.1"
- Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
- Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
- Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
* Verifying ingress addon...
* The 'ingress' addon is enabled

C:\Users\oyaag>minikube tunnel
W0607 18:01:12.510422 12664 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\oyaag\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.
* Tunnel successfully started

* NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

* Starting tunnel for service mysql.
```

3) crear namespace

```
apiVersion: v1
kind: Namespace
metadata:
  name: database
```

4) crear PersistentVolume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mysql-pv
  namespace: database
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
```

```
hostPath:  
  path: "/mnt/data"
```

5) crear PersistentVolumeClaim

```
apiVersion: v1  
kind: PersistentVolumeClaim  
metadata:  
  name: mysql-pvc  
  namespace: database  
spec:  
  accessModes:  
    - ReadWriteOnce  
  resources:  
    requests:  
      storage: 1Gi
```

6) crear Deployment

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: mysql  
  labels:  
    app: mysql  
  namespace: database  
spec:  
  selector:  
    matchLabels:  
      app: mysql  
  strategy:  
    type: Recreate  
  template:  
    metadata:  
      labels:  
        app: mysql  
    spec:  
      containers:  
        - image: mysql:5.7  
          name: mysql  
          env:  
            - name: MYSQL_ROOT_PASSWORD  
              value: "1234"  
            - name: MYSQL_DATABASE  
              value: "misanuncios" # Ensure this environment variable is set  
      ports:  
        - containerPort: 3306  
          name: mysql
```

```

volumeMounts:
- name: mysql-persistent-storage
  mountPath: /var/lib/mysql
volumes:
- name: mysql-persistent-storage
  persistentVolumeClaim:
    claimName: mysql-pvc

```

7) crear Load Balancer Service

```

apiVersion: v1
kind: Service
metadata:
  namespace: database
  name: mysql
spec:
  type: LoadBalancer
  ports:
    - port: 3306
      targetPort: 3306
  selector:
    app: mysql

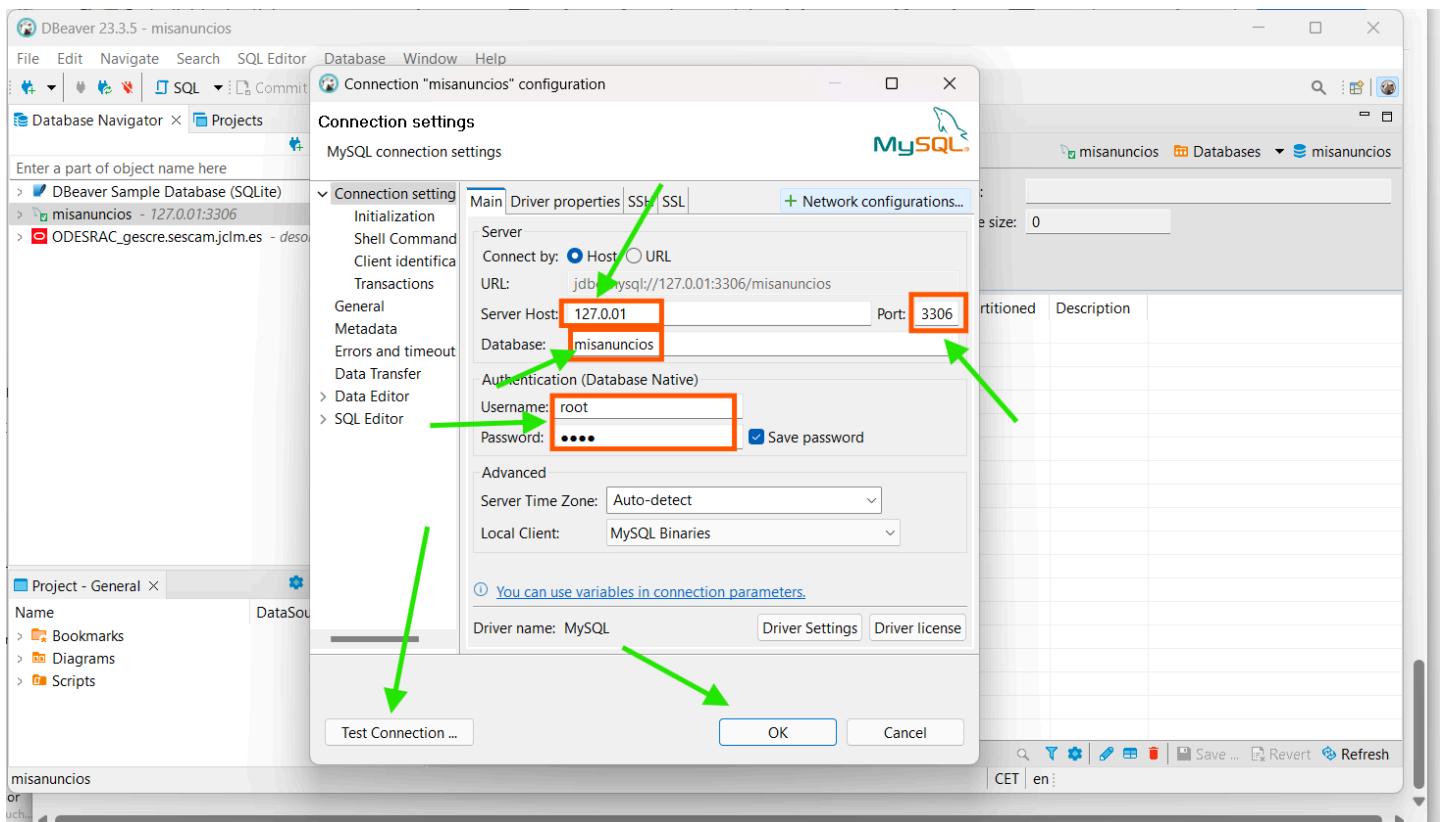
```

8) instalar el lens , si no quieres usarlo mete al terminal y ejecuta el comando : minikube dashboard

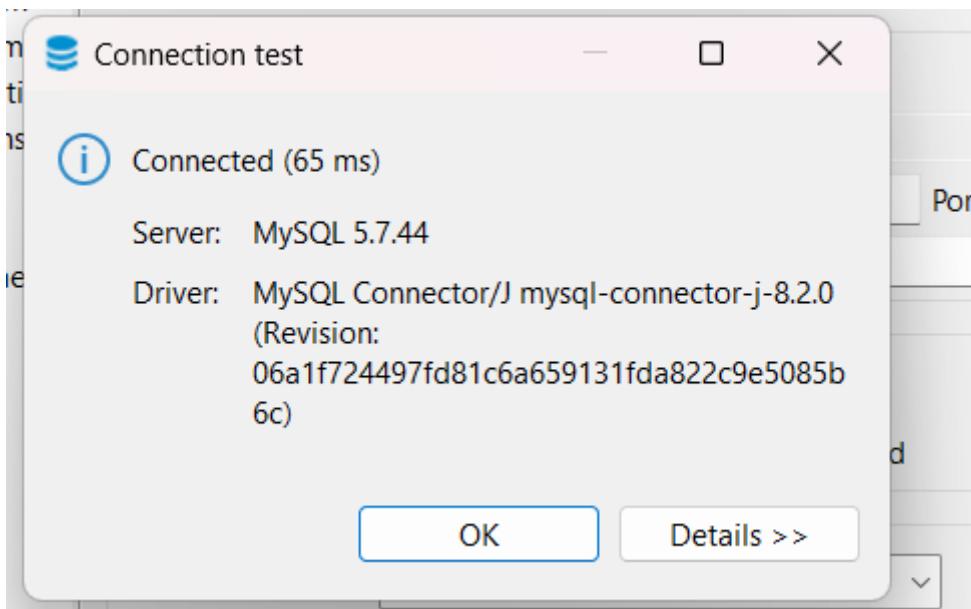
Name	Namespace	Type	Cluster IP	Ports	External IP	Selector	Age	Status
mysql	database	LoadBal	10.97.11.232	3306:30628/TCP	127.0.0.1	app=mysql	8m27s	Active

9) el host es external IP el puerto en nuestro caso es 3306 y el usuario es root viene por defecto y la contraseña y el nombre de la base de datos es

```
name: mysql
env:
- name: MYSQL_ROOT_PASSWORD
  value: "1234"
- name: MYSQL_DATABASE
  value: "misanuncios" # En
```



10) resultados final



b. Documentación de usuario

- Guía de inicio: Cómo registrarse y crear una cuenta.

1) Iniciar sesión

The diagram illustrates the user interaction flow for logging in:

1. Click on the 'Log in' button.
2. Enter the email address into the 'Correo' input field.
3. Enter the password into the 'Contraseña' input field.
4. Click on the 'Iniciar sesión' button.

Labels: Misanuncios, Inicio, Productos, Sobre me, Log in, Sign up, Iniciar sesión, Correo, Contraseña, Olvidar Contraseña?, Iniciar sesión.

Misanuncios

Proporcionamos una plataforma para que pueda anunciar libremente sus productos como desee respetando a los usuarios de nuestra plataforma.

Inicio Products Sobre Me

© 2021-2025 Misanuncios™ All Rights Reserved

2) registrarse

Misanuncios

Inicio Productos Sobre me

Log in Sign up

The diagram shows a registration form titled "Crear Una Cuenta". The fields are: email (a@a.com), Contraseña (****), Confirmar la Contraseña (12), and a checkbox labeled "I accept the Terms and Conditions". A large blue button at the bottom right says "Registrarse". Five numbered circles (1-5) with arrows point to specific elements: 1 points to the "Sign up" link; 2 points to the "email" field; 3 points to the "Contraseña" field; 4 points to the "Confirmar la Contraseña" field; and 5 points to the "Registrarse" button.

Crear Una Cuenta

email
a@a.com

Contraseña

Confirmar la Contraseña
12

I accept the [Terms and Conditions](#)

Registrarse

- Publicación de anuncios:

- Pasos para crear,

Misanuncios

Inicio Productos Marketplace Sobre me

Logout

Nuevo Producto

Mis Productos Chats notificaciones

Neuvo Producto

1

2

Titulo

Precio

\$2999

Cargar Imagen

La Imagen seleccionada debe ser de estos tipos de formatos: **JPG, PNG, or GIF**.

3

4

5

6

Product Description

Write product description here

+ Añadir Producto

Rechazar

nota: para borrar un imagen solo tienes que pulsar sobre la imagen que quieres borrar

editar o borrar producto





Furgoneta Renault Trafic 2021

★★★★★ 5.0

\$23.650

Update

Ver

Delete

- Búsqueda y filtrado: Cómo buscar y filtrar anuncios.

Resulado de búsqueda

1

2

3

4

5

Category

- Coches
- Motos
- Motor y Accesorios
- Moda y Accesorios
- Inmobiliaria
- Tecnología y Electrónica
- Deporte y Ocio
- Hogar y Jardín
- Otros

- Gestión de la cuenta: Configuración de perfil, recuperación de contraseñas.
 - Uso del sistema de mensajería: Cómo comunicarse con otros usuarios.
- 1) Elegir producto y pula “ver producto”

Furgoneta Renault Trafic 2021

5.0

\$23.650

Ver

- 2) pulsar el chat, y empieza chatear con vendedor



Furgoneta Renault Trafic 2021

Introducción a. Presentación del proyecto b. Objetivos del proyecto c. Justificación del proyecto II. Análisis de requerimientos a. Identificación de necesidades y requerimientos b. Identificación de público c. Estudio de mercado y competencia III. Diseño y planificación a. Definición de la arquitectura del proyecto b. Diseño de la interfaz de usuario c. Planificación de las tareas y los recursos necesarios IV. Implementación y pruebas a. Desarrollo de las funcionalidades del proyecto b. Pruebas unitarias y de integración c. Corrección de errores y optimización del rendimiento V. Documentación a. Documentación técnica b. Documentación de usuario c. Manual de instalación y configuración VI. Mantenimiento y evolución a. Plan de mantenimiento y soporte b. Identificación de posibles mejoras y evolución del proyecto c. Actualizaciones y mejoras futuras VII. Conclusiones a. Evaluación del proyecto b. Cumplimiento de objetivos y requisitos c. Lecciones aprendidas y recomendaciones para futuros proyectos VIII. Bibliografía y referencias a. Fuentes utilizadas en el proyecto b.

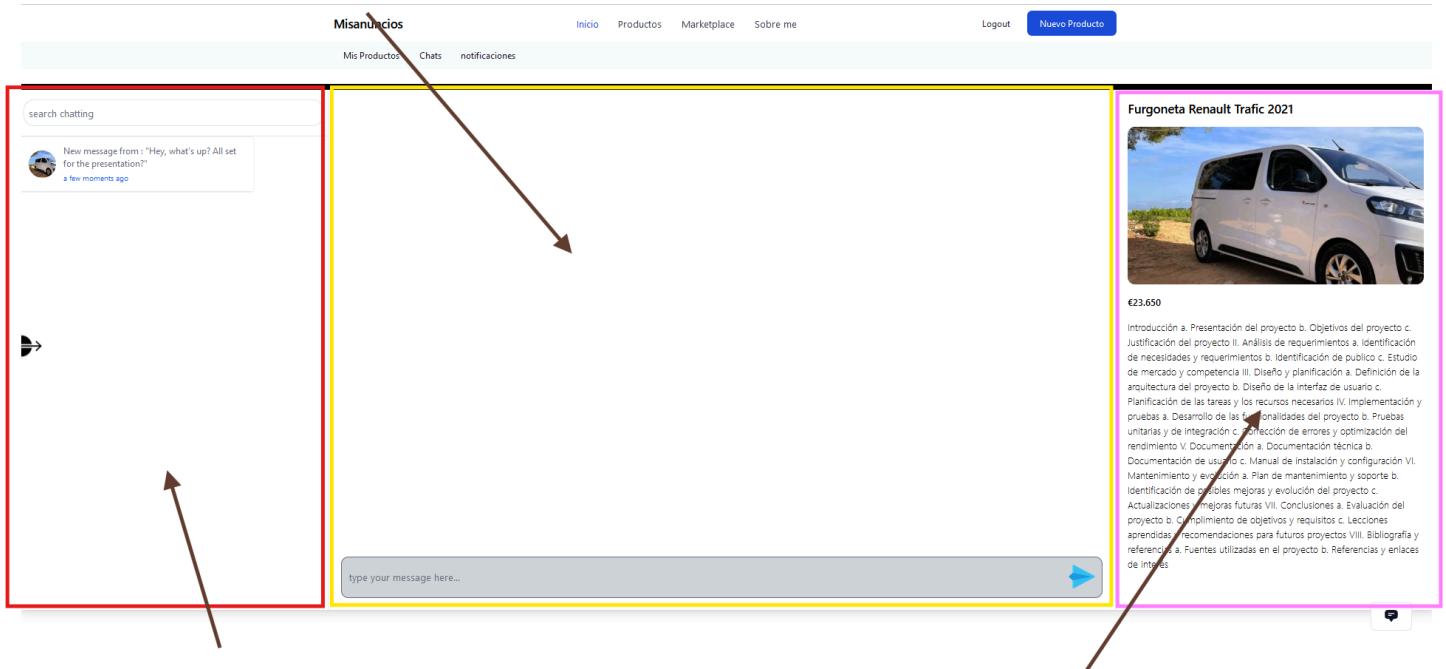
\$23.650

★★★★★ 117 reviews

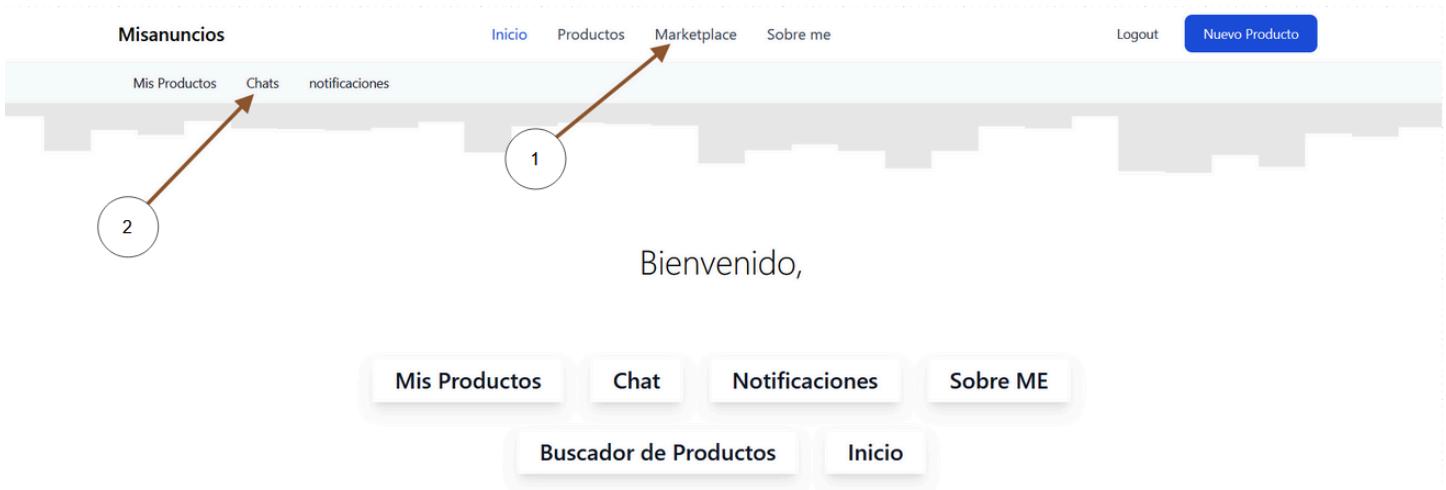
Chat

en el chat sale tres parte la primera :

- parte de conversaciones
- parte :del producto
- parte de acciones con vendedor



para ir al chat :



para el ver si te ha llegado un mensaje te va a llegar en la parte del notification

[Mis Productos](#)[Chats](#)[notificaciones](#)

Mensaje desde Chat

Tienes nuevo Mensaje!: me interesa este pr0ducto 🔥

NUEVO

2024-06-01 11:59:22



Mensaje desde Chat

Tienes nuevo Mensaje!: que pasa 🔥

NUEVO

2024-06-01 11:56:12



Mensaje desde Chat

Tienes nuevo Mensaje!: hfdfbrftfy ftdfu tvfrt dtyjctrf cut 🔥

NUEVO

2024-06-01 11:55:41



Mensaje desde Chat

Tienes nuevo Mensaje!: hola 🔥

NUEVO

2024-06-01 11:54:51



Mensaje desde Chat

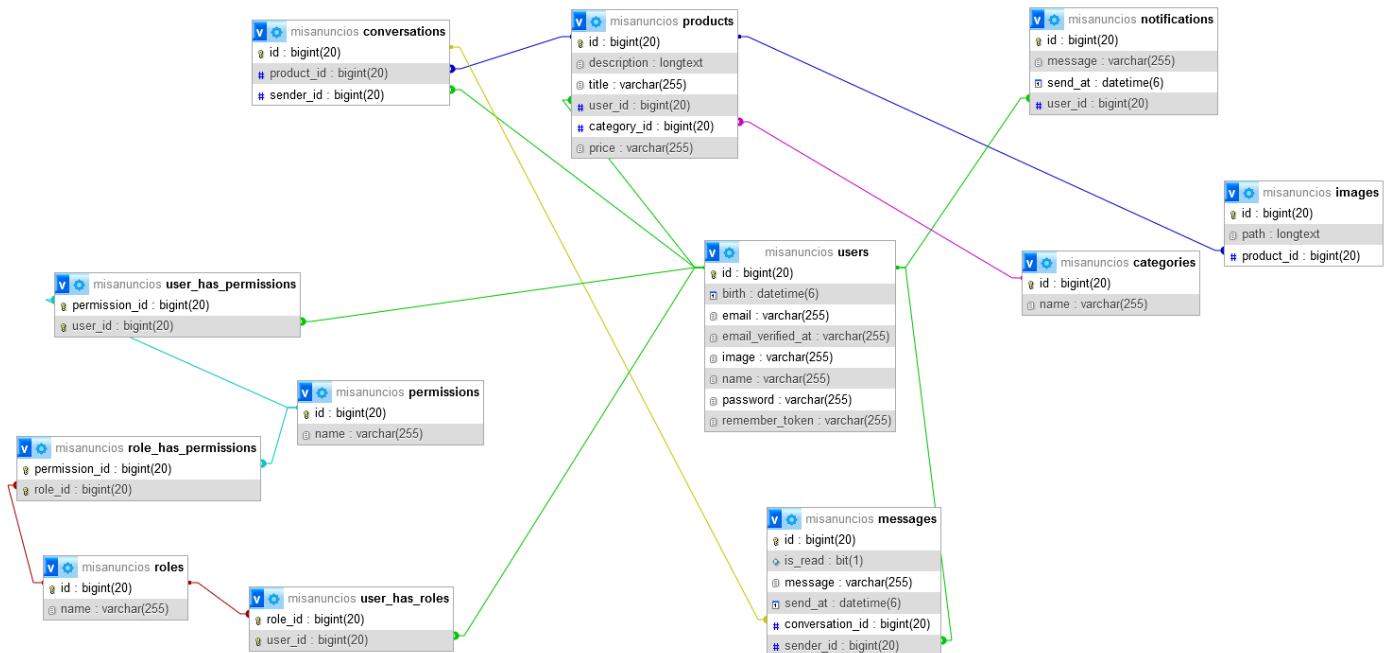
Tienes nuevo Mensaje!: ok 🔥

NUEVO

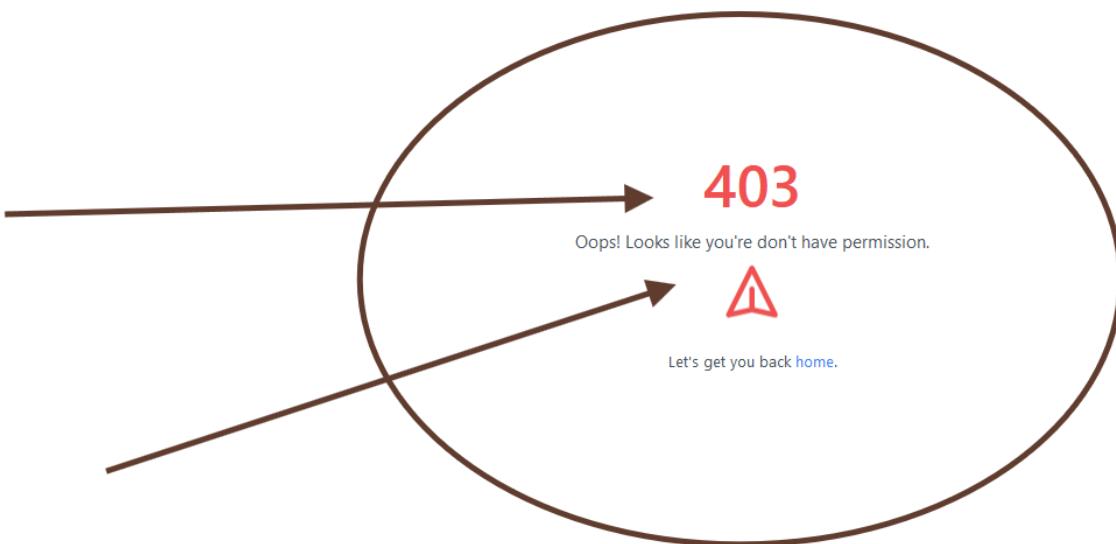
2024-06-01 11:29:41

- Configuración de la base de datos: MySQL, creación de esquemas y tablas.

la base de datos se crea automáticamente con JPA



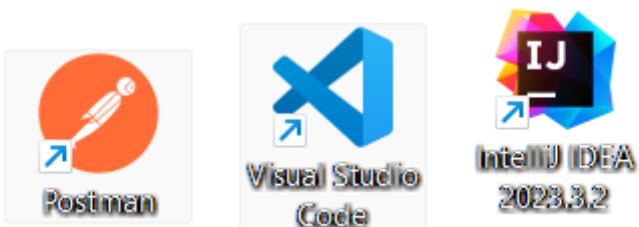
- Gestión de seguridad: Configuración de Spring Security, autenticación y autorización.



VI. Mantenimiento y evolución

a. Plan de mantenimiento y soporte

- Monitoreo del sistema: Herramientas y procedimientos.



XAMPP Control Panel v3.3.0

b. Identificación de posibles mejoras y evolución del proyecto

1. Sistema de notificaciones en tiempo real:

- Se ha identificado la necesidad de un sistema de notificaciones en tiempo real para mejorar la comunicación con los usuarios. Esta funcionalidad permitirá a los usuarios recibir alertas instantáneas sobre nuevos mensajes, actualizaciones de anuncios y otras actividades relevantes.
- **Implementación:** Se integrará una solución basada en WebSockets para permitir la transmisión de datos en tiempo real entre el servidor y el cliente.

2. Optimización del algoritmo de búsqueda:

- Se trabajará en la optimización del algoritmo de búsqueda utilizando técnicas de aprendizaje automático para ofrecer resultados más personalizados. implementando tecnología artificial

3. Mejora de la interfaz de usuario:

- Basado en la retroalimentación, se rediseñará la interfaz de usuario para hacerla más intuitiva. Esto incluirá la simplificación de menús, mejor organización de la información y un diseño más moderno y atractivo.
- **Implementación:** Colaboración con diseñadores UX para desarrollar prototipos, realizar pruebas de usabilidad y aplicar los cambios necesarios.

c. Actualizaciones y mejoras futuras

Nuevas Características: Añadir funcionalidades adicionales para mejorar la usabilidad y experiencia del usuario.

Optimización del Código: Mejorar la velocidad y eficiencia revisando y optimizando el código existente.

Seguridad: Implementar medidas adicionales para proteger los datos de los usuarios y cumplir con normativas de privacidad.

Interfaz y Experiencia de Usuario (UI/UX): Rediseñar la interfaz para hacerla más intuitiva y accesible para todos los usuarios.

Integración y Compatibilidad: Asegurar la compatibilidad con múltiples dispositivos y sistemas, e integrar con otros servicios para ampliar la funcionalidad.

Recopilación de Feedback: Implementar mecanismos para recolectar y analizar feedback de los usuarios, identificando áreas de mejora.

Actualización de Tecnologías: Mantener el proyecto actualizado con las últimas versiones de tecnologías y frameworks.

Documentación y Soporte: Mejorar la documentación y crear tutoriales para facilitar el uso y resolución de problemas por parte de los usuarios.

VII. Conclusiones

a. Evaluación del proyecto

Objetivos Cumplidos:

1. **Plataforma fácil de usar:** Accesible y amigable, facilita publicación y búsqueda de anuncios.
2. **Interacción comprador-vendedor:** Sistema de mensajería integrado que promueve la comunicación.
3. **Monetización:** Inclusión de espacios publicitarios para generar ingresos.
4. **Registro y gestión de cuentas:** Sistema seguro y eficiente para usuarios.
5. **Publicación y gestión de anuncios:** Los usuarios pueden crear, editar y eliminar anuncios.
6. **Búsqueda avanzada:** Sistema de filtros eficientes para encontrar anuncios específicos.
7. **Documentación completa:** Guías de usuario y documentación técnica detallada.

Objetivos No Cumplidos:

1. **Notificaciones en tiempo real:** No está implementado por limitaciones de tiempo.
2. **Optimización de búsqueda:** Técnicas de aprendizaje automático no implementadas.
3. **Mejoras de UI:** No se completaron todas las recomendaciones de UX.

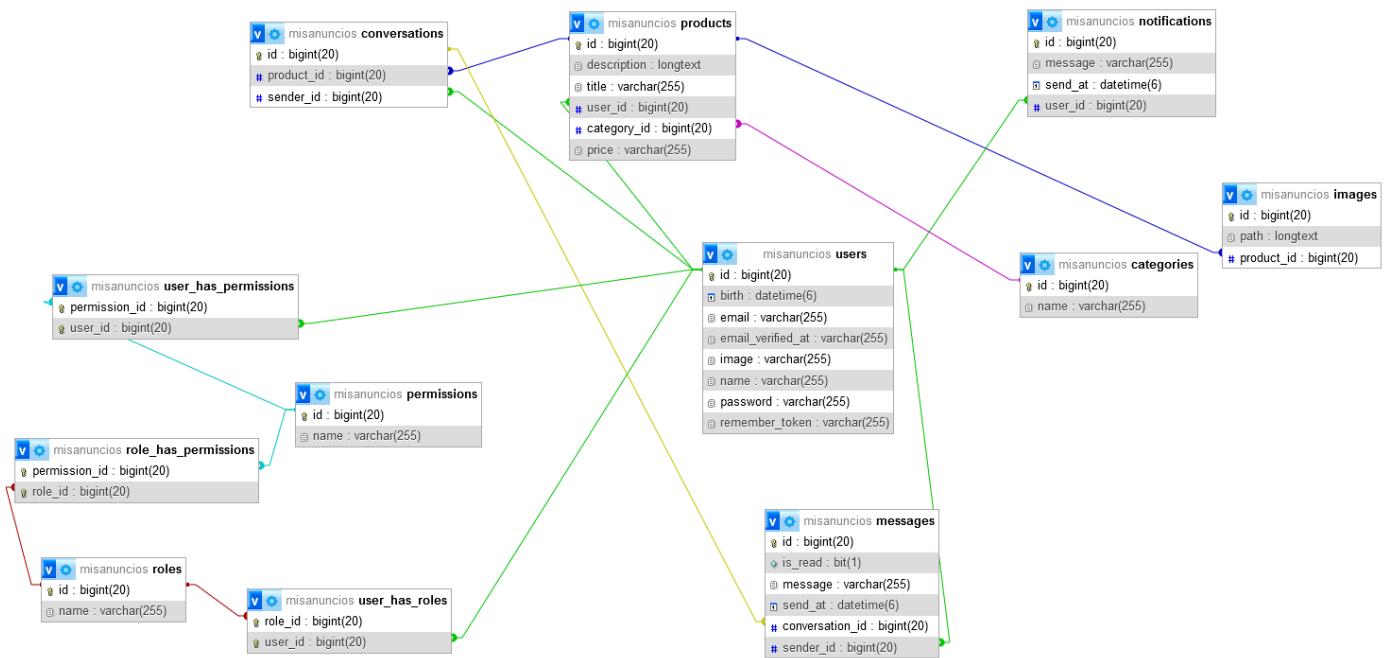
Calidad del Producto:

- **Funcionalidad:** Cumple con las funcionalidades principales.
- **Usabilidad:** Interfaz intuitiva y diseño responsivo.
- **Rendimiento:** Optimizada para alto volumen, pero con áreas de mejora.

b. Cumplimiento de objetivos y requisitos

base de datos : **Tiene**

tiene 12 tablas con columnas de fechas y identificador único como clave primaria , hecho com mysql , pero funciona en cualquier tipo de base de datos (oracle , postgres) porque esta hecho con JPA



Backend : Tiene

está hecho con spring boot es un microservicio ,

1) contraseña encriptada

```

1 usage  ± OYaagoub
public User signup(UserDto input) {
    User user = new User();
    user.setName(input.getName());
    user.setEmail(input.getEmail());
    user.setPassword(passwordEncoder.encode(input.getPassword()));

    return userService.save(user);
}
  
```

2) rutas y controladores protegidos,

```

20
21     @Slf4j
22     @RestController
23     @RequestMapping("/api/v3/content")
24     @CrossOrigin("*")
25     @AllArgsConstructor
26     public class CategoryController {
27         private CategoryService categoryService;
28         private CategoryDtoMapper categoryDtoMapper;
29
30         1 usage
31         private final CycleAvoidingMappingContext context= new CycleAvoidingMappingContext();
32
33         @JsonView({Views.CategorySample.class})
34         @GetMapping("/categories")
35         public ResponseEntity<Object> getCategories() throws JsonProcessingException {
36             var response = categoryService.getAllCategories().stream().map(category -> {
37                 return categoryDtoMapper.toDto(category,context);
38             })
39             .collect(Collectors.toList());
40             return ResponseEntity.ok().body(response);
41         }
42
43         @GetMapping("/categories/name")
44         public ResponseEntity<Object> getCategoriesByName() throws JsonProcessingException {
45             var response = categoryService.getAllCategoriesByName();
46             return ResponseEntity.ok().body(response);
47         }

```

3) uso de sesiones o Token

Nuevo Producto

productos en lo sin meter

ia para que pueda anunciar
o desee respetando a los usuarios de

Estado	Método	URL
200	POST	http://192.168.1.133:8080/api/v3/auth/login
200	GET	http://localhost:4200/assets/images/34dc40x-34dc54sx-4c3cx3-4c5...
200	OPTIONS	http://192.168.1.133:8080/api/v3/auth/user
200	OPTIONS	http://192.168.1.133:8080/api/v3/content/categories
200	GET	http://192.168.1.133:8080/api/v3/auth/user
200	GET	http://192.168.1.133:8080/api/v3/content/categories

Cabeceras Cookies Solicitud Respuesta Tiempos Traza de la pila

Filtrar propiedades

JSON Sin procesar

```

token: "eyJhbGciOiJIUzI1NiJ9eyJzdWliOjkZXZlbG9wZXJAZS5jb2oiLCJpYXQiOjE3MTc0NDgslmV4cCl6MTcxNzQ1MDE4OH0
        .pgql-hSAIFbY6VjfxU58N_av2uz46DHKzC6ZM5UK2iA"
expiresIn: 3600000

```

comprobaciones previas a la inserción,

```
@PostMapping("/conversations/save/{idProduct}")
private ResponseEntity<Object> save(@PathVariable long idProduct){
    if(getAuthentication() instanceof User user){
        Product product=productService.find(idProduct);
        if(product!=null){
            System.out.println("user.product.id: "+product.getUser().getId() +" user.id"+user.getId());
            if(product.getUser().getId()==user.getId()){
                return ResponseEntity.ok().body(true);
            }
            Conversation con = conversationService.findByUserProduct(product.getId(),user.getId());
            if(con==null && product.getUser().getId()!=user.getId()){
                con= new Conversation();
                con.setSender(user);
                con.setProduct(product);
                conversationService.save(con);
            }
            return ResponseEntity.ok().body(true);
        }
    }
    return ResponseEntity.ok().body(false);
}
```

4) error que se produzcan en cualquier petición

```

    ↳ OYaagoub
    @RestControllerAdvice
    public class GlobalExceptionHandler {
        ↳ OYaagoub
        @ExceptionHandler(Exception.class)
        public ProblemDetail handleSecurityException(Exception exception) {
            ProblemDetail errorDetail = null;

            // TODO send this stack trace to an observability tool
            exception.printStackTrace();

            if (exception instanceof BadCredentialsException) {
                errorDetail = ProblemDetail.forStatusAndDetail(HttpStatusCode.valueOf(code: 401), exception.getMessage());
                errorDetail.setProperty("description", "The username or password is incorrect");

                return errorDetail;
            }

            if (exception instanceof AccountStatusException) {
                errorDetail = ProblemDetail.forStatusAndDetail(HttpStatusCode.valueOf(code: 403), exception.getMessage());
                errorDetail.setProperty("description", "The account is locked");
            }

            if (exception instanceof AccessDeniedException) {
                errorDetail = ProblemDetail.forStatusAndDetail(HttpStatusCode.valueOf(code: 403), exception.getMessage());
                errorDetail.setProperty("description", "You are not authorized to access this resource");
            }
        }
    }

```

Frontend: **Tiene**

HTML, CSS y JavaScript **eso lo Tiene** estoy usando angular con framework de dar estilos el que es tailwind y para peticiones como sabemos que angular es un framework de front-end está separado complementamente desde back-end , el que está comunicando con back a través de apis y es responsive

Misanuncios

Inicio Productos Marketplace Sobre me

Nuevo Producto 



Anunciar tus productos en nuestro mundo sin meter prisa.

Proporcionamos una plataforma para que pueda anunciar libremente sus productos como desee respetando a los usuarios de nuestra plataforma. .

Get started →

Misanuncios

Nuevo Producto



Anunciar tus productos en nuestro mundo sin meter prisa.

Proporcionamos una plataforma para que pueda anunciar libremente sus productos como desee respetando a los usuarios de nuestra plataforma. .

Get started →

Durante la ejecución del proyecto, aprendimos varias lecciones importantes. La gestión del tiempo es fundamental, y es clave ser flexible pero riguroso en la planificación. La integración de nuevas tecnologías puede ser desafiante, por lo que es esencial contar con una base sólida de conocimientos. Usar herramientas de colaboración. La retroalimentación constante de usuarios y partes interesadas mejora la funcionalidad y usabilidad del producto. Las pruebas y validaciones continuas aseguran un producto final de alta calidad.

Para futuros proyectos, recomiendo centrarse en la reactividad en tiempo real, implementando sistemas que respondan rápidamente a eventos y cambios, como notificaciones en tiempo real y actualizaciones automáticas. Dedicar más tiempo a la fase de planificación para identificar características esenciales y posibles riesgos es vital.

VIII. Bibliografía y referencias

<https://www.youtube.com/>

<https://www.w3schools.com/>

<https://tailwindcss.com/>

<https://spring.io/projects/spring-boot>

<https://angular.dev/>

<https://kubernetes.io>