

Descripción proyecto integrado



Alumno: Nicolás Álvarez Zapata
Curso: CFGS Desarrollo de aplicaciones web

Descripción de la idea

Desarrollar una aplicación que esté diseñada para que los aficionados a las motos puedan organizar y unirse a quedadas de forma sencilla. Facilitando así la conexión entre usuarios, permitiendo la planificación de eventos, la gestión de participantes y la creación de una comunidad motera amigable, permitiendo así a los usuarios informar cual es su motocicleta y las modificaciones que esta tiene, comentar en las quedadas y poder dar like a los comentarios.

La aplicación busca simplificar la experiencia de las quedadas moteras, haciendo que sea fácil para los usuarios disfrutar de su pasión por las motos y conectar con otros con el mismo hobby de manera accesible.

Tecnologías, dependencias y versiones

El backend del proyecto será desarrollado en **Java en su versión 21**, concretamente con el framework **Spring Boot en su versión 3.2.1**, este usa para la persistencia de los datos la especificación **JPA**, concretamente hace uso de la implementación **Hibernate en su versión 6.4.1**. La base de datos del proyecto es **PostgreSQL JDBC en su versión 42.6.0**.

La estructura y el maquetado del frontend se basa en **HTML5** junto con **Bootstrap en su versión 5.3.2** haciendo también que sea en menor medida responsive y adaptable a distintas resoluciones. Las vistas se basan en el motor de plantillas **Thymeleaf en su versión 3.2.1** que se integra a la perfección con Spring para hacer dinámicas dichas vistas.

Para los estilos se usa **CSS3**.

Se hace uso en menor medida el lenguaje de programación **Javascript** para cuestiones sobre la validación en el lado cliente.

Se hace uso de la biblioteca **SweetAlert2 en su versión 11.10.5** para los diálogos modales que se ofrecen para dar información al usuario y la biblioteca de código abierto **Bootstrap Icons en su versión 1.11.3** para dar un aspecto más robusto al sitio web incluyendo algunos de sus iconos.

En cuanto a la seguridad del sitio web se usa el framework **Spring Security en su versión 3.2.1**.

Cada aplicación web con Spring boot incluye embebido en cada proyecto una instancia del servidor web **Tomcat**, abstrayendo la necesidad de configurarlo manualmente al desplegar la aplicación en el servidor en el que se lleve a cabo el despliegue.

Para los registros Logs y llevar un control de lo que ocurre en la aplicación,, se usa la dependencia a la API **SLF4J en su versión 2.21.1**. Estos logs son almacenados en el mismo servidor donde está desplegado el proyecto.

Para el acceso a los datos generando un fichero en formato PDF de estos, se usa la **biblioteca JavaScript html2pdf.js** que proporciona una solución simple para convertir contenido HTML en archivos PDF directamente desde el navegador web del usuario. Esta herramienta, permite generar documentos PDF fácilmente a partir de elementos HTML, lo que resulta útil para crear informes, facturas, certificados y otros tipos de documentos digitales de forma dinámica y sin la necesidad de utilizar software externo.

Servidor de despliegue

El proyecto está desplegado en una **instancia de AWS** sobre **servidor web Nginx** con **proxy inverso** escuchando en el puerto 8080 que es donde se despliega el servidor tomcat embebido del proyecto, todo esto bajo el **protocolo HTTPS** de transferencia de datos segura y encriptada.

AWS proporciona capacidad de computación escalable bajo demanda en la nube. El **sistema operativo en mi caso es Amazon Linux 2 de 64 bits**, que es una distribución de Linux propia de Amazon Web Services. Brinda un entorno de ejecución enfocado en la seguridad, estable y de alto rendimiento para desarrollar y ejecutar aplicaciones en la nube. Concretamente la instancia de AWS es la que cubre la capa gratuita: **T3 Micro con 2 vCPU y 1 GB de memoria RAM y 20 GB en disco SSD de uso general**.

Detalles principales tecnologías

Spring Boot:

Spring Boot es un framework de desarrollo de aplicaciones Java que simplifica la creación de aplicaciones autónomas y listas para producción. Proporciona un enfoque basado en convenciones para la configuración, eliminando la necesidad de configuraciones manuales y permitiendo un rápido desarrollo.

Ventajas:

- **Configuración mínima:** Spring Boot proporciona una configuración predeterminada que puede ser modificada según sea necesario.
- **Integración fácil:** Permite la integración sencilla con otras tecnologías y frameworks de Spring.
- **Despliegue sencillo:** Las aplicaciones Spring Boot pueden ser empaquetadas como JAR o WAR y desplegadas fácilmente en servidores de aplicaciones.

Puntos a destacar: La facilidad de uso y la rapidez de desarrollo que me ofreció Spring Boot me permitió acelerar el proceso de desarrollo del proyecto, reduciendo la carga de trabajo relacionada con la configuración y de esta manera, poder centrarme en lo importante, el desarrollo en sí mismo de la aplicación.

Spring Security:

Es un framework de seguridad incorporado en Spring Boot que proporciona funciones de autenticación y autorización para aplicaciones Java. Permite proteger las aplicaciones contra vulnerabilidades comunes de seguridad, como la autenticación no autorizada y los ataques de inyección de código.

Ventajas:

- **Configuración flexible:** Spring Security ofrece una amplia gama de opciones de configuración para adaptarse a los requisitos específicos de seguridad de cada aplicación.
- **Integración con Spring:** Se integra perfectamente con otros componentes de Spring, lo que facilita su uso en aplicaciones Spring existentes.
- **Gestión de sesiones:** Proporciona funciones para la gestión segura de sesiones de usuario, incluida la protección contra ataques de secuestro de sesión.

Puntos a destacar: La integración de Spring Security garantiza la seguridad de la plataforma, protegiendo los datos, el acceso no autorizado y la información confidencial de los usuarios.

Aspect-Oriented Programming (AOP):

AOP es un paradigma de programación que permite modularizar aspectos transversales de una aplicación, separando la funcionalidad del registro del log de la lógica de negocio del proyecto. De esta manera se evita repetición de código en todas las clases y permite un código más limpio y mantenible. En mi caso me baso en este paradigma para gestionar los logs del aplicativo.

Ventajas:

Separación de preocupaciones: Permite separar la lógica de aspectos transversales de la lógica de negocio, lo que facilita el mantenimiento y la escalabilidad de la aplicación.

Reutilización de código: Permite reutilizar aspectos comunes en diferentes partes de la aplicación, reduciendo la duplicación de código.

Thymeleaf:

Es un motor de plantillas para aplicaciones web Java que permite la creación de vistas HTML fácilmente legibles y mantenibles. Proporciona una sintaxis natural e intuitiva que facilita la integración de datos dinámicos en las páginas web.

Ventajas:

- **Integración con Spring:** Thymeleaf se integra perfectamente con Spring MVC, lo que facilita la creación de vistas dinámicas en aplicaciones web basadas en Spring.
- **Sintaxis intuitiva:** La sintaxis de Thymeleaf es similar a HTML, lo que facilita su aprendizaje y su uso por parte de los desarrolladores.

PostgreSQL

Sistema de gestión de bases de datos relacional de código abierto y altamente escalable. Ofrece una amplia gama de características avanzadas, incluyendo soporte para tipos de datos complejos, transacciones ACID y replicación.

Ventajas:

- **Fiabilidad:** PostgreSQL es conocido por su estabilidad y fiabilidad, lo que lo convierte en una opción popular para aplicaciones críticas.
- **Rendimiento:** Ofrece un rendimiento sólido incluso en entornos de alta carga, gracias a su arquitectura optimizada y sus capacidades de indexación avanzadas.

Puntos a destacar: La elección de PostgreSQL como base de datos en el proyecto proporcionó una solución robusta y escalable para el almacenamiento de datos, garantizando la integridad y la disponibilidad de la información.

SweetAlert:

Biblioteca de JavaScript que permite crear fácilmente ventanas modales personalizadas y atractivas para mostrar mensajes al usuario. Proporciona una experiencia de usuario mejorada mediante la presentación de mensajes claros y visualmente atractivos.

Ventajas:

- **Personalización:** SweetAlert ofrece una amplia gama de opciones de personalización, como estilos, animaciones y tipos de botones, que permiten adaptar las ventanas modales según las necesidades del proyecto.
- **Facilidad de uso:** Es fácil de integrar en aplicaciones web y requiere un mínimo de configuración para empezar a utilizarlo.

Puntos a destacar: La incorporación de SweetAlert en el proyecto mejoró significativamente la experiencia del usuario al proporcionar mensajes claros y atractivos, lo que facilitó la comunicación de información importante y la interacción con la aplicación.

Diagrama Entidad-Relación BikerConnect

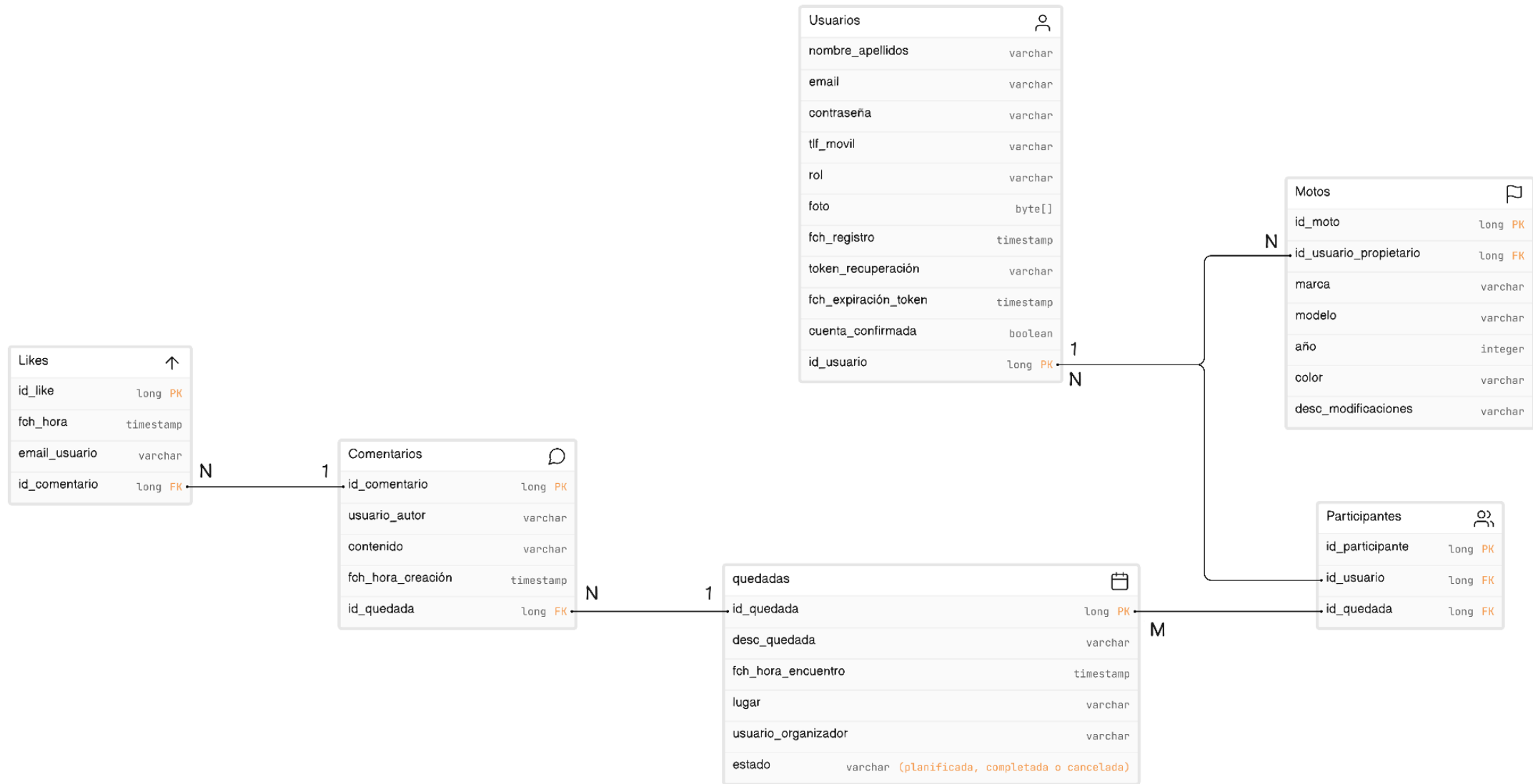


Diagrama de flujo funcionalidades BikerConnect

