

Universidad de las Fuerzas Armadas ESPE

Departamento de Ciencias de la Computación

Aplicaciones Distribuidas

Freddy Leonel Pachacama

Carlos Pogo

Erika Changoluisa

NRC: 3882

Miercoles, 5 de marzo del 2025  
Sangolquí – Ecuador

**Índice**

[1 Resumen ejecutivo 3](#_Toc192252106)

[2 Introducción 3](#_Toc192252107)

[2.1 Contexto y justificación 3](#_Toc192252108)

[2.2 Objetivos 3](#_Toc192252109)

[2.3 Alcance del Sistema 4](#_Toc192252110)

[3 Análisis del Problema y Requerimientos 4](#_Toc192252111)

[3.1 Descripción del problema 4](#_Toc192252112)

[3.2 Usuario y Roles 4](#_Toc192252113)

[3.3 Requerimientos Funcionales 5](#_Toc192252114)

[3.4 Requerimientos No Funcionales 5](#_Toc192252115)

[4 Diseño del Sistema 5](#_Toc192252116)

[4.1 Arquitectura del Sistema 5](#_Toc192252117)

[4.2 Base de Datos 6](#_Toc192252118)

[5 Implementación 7](#_Toc192252119)

[5.1 Backend 7](#_Toc192252120)

[5.2 Endpoints Principales 7](#_Toc192252121)

[5.3 Seguridad y Autenticación 9](#_Toc192252122)

[5.4 WebSockets 10](#_Toc192252123)

[5.5 Frontend 11](#_Toc192252124)

[6 Pruebas y Validaciones: 12](#_Toc192252125)

[7 Desafíos y Aprendizajes 15](#_Toc192252126)

[7.1 Problemas encontrados y cómo se resolvieron 15](#_Toc192252127)

[7.2 Posibles mejoras futuras 15](#_Toc192252128)

[8 Conclusiones y Recomendaciones 16](#_Toc192252129)

[8.1 Cumplimiento de los objetivos 16](#_Toc192252130)

[8.2 Impacto del sistema 16](#_Toc192252131)

[8.3 Recomendaciones 17](#_Toc192252132)

[9 Anexos 17](#_Toc192252133)

# **Resumen ejecutivo**

Actualmente, las subastas en línea de automóviles han ido creciendo debido a su eficiencia para automatizar la venta y compra de automóviles. Sin embargo, la automatización de estos procesos requiere una infraestructura robusta que pueda garantizar la seguridad y fluidez de las transacciones.

En este proyecto se desarrolló un sistema de subastas de autos en tiempo real, donde vendedores tienen la capacidad de registrar vehículos y los compradores pueden participar mediante la realización de pujas, esto se logró gracias a la utilización de APIs y WebSockets. En cuanto a la seguridad, el sistema implementa seguridad que se basa en roles utilizando JWT, además se realizó varias validaciones en la capa de lógica de negocio para así garantizar la integridad de las subastas.

Dando como resultado una aplicación funcional que garantiza una gestión segura de subastas, con una interfaz amigable e intuitiva para compradores y vendedores, garantizando transparencia y eficiencia en la compra y venta de vehículos.

# **Introducción**

## **Contexto y justificación**

Las subastas de automóviles en línea han permitido un acceso más ágil y competitivo al mercado automotor. Sin embargo, la falta de plataformas seguras y en tiempo real puede causar problemas como el intento de fraudes, errores en las pujas y dificultad en la administración de subastas.

Este proyecto busca solucionar estas problemáticas mediante el desarrollo de un sistema en línea que haga uso de tecnologías como WebSockets para comunicación en tiempo real y autenticación basada en JWT para garantizar la seguridad de los usuarios.

## **Objetivos**

**Objetivo General**

Desarrollar una plataforma de subastas de automóviles en tiempo real, segura y eficiente, que permita una interacción entre vendedores y compradores mediante un sistema robusto de gestión de subastas y pujas.

**Objetivos Específicos**

* Implementar un sistema de autenticación y gestión de usuarios basado en roles.
* Desarrollar una API RESTful para la gestión de subastas y usuarios.
* Implementar WebSockets para la comunicación en tiempo real durante las subastas.
* Diseñar una interfaz de usuario intuitiva que facilite la participación de los compradores y vendedores.
* Garantizar la seguridad y la integridad de los datos mediante validaciones y control de acceso.

## **Alcance del Sistema**

* Registro y autenticación de usuarios con roles de comprador, vendedor y administrador.
* Creación y gestión de subastas por parte de los vendedores.
* Participación en subastas en tiempo real por parte de los compradores.
* Determinación automática de ganadores según la puja más alta.
* Implementación de reglas de negocio para garantizar la validez de las pujas y la seguridad de los datos.

# **Análisis del Problema y Requerimientos**

## **Descripción del problema**

El mercado de subastas online trae consigo varios desafíos como podría ser la falta de plataformas en tiempo real que permitan una ágil interacción entre compradores y vendedores, posibles fraudes o manipulación de pujas sin un sistema seguro de autenticación, hasta la necesidad de gestionar múltiples subastas simultáneamente con reglas pre establecidas.

Este sistema trata de dar solución a todos los desafíos antes mencionados, para lo que se ha implementado una infraestructura segura y eficiente capaz de gestionar confiablemente las subastas en tiempo real.

## **Usuario y Roles**

El sistema tiene la necesidad de tres tipos de usuarios:  
**Vendedores:** Tienen la capacidad de ingresar vehículos y crear subastas.

**Compradores:** Tienen la posibilidad de participar en subastas y participar en pujas.

**Administradores:** Son los encargados de gestionar usuarios y supervisan la integridad del sistema.

## **Requerimientos Funcionales**

* Los usuarios deben registrarse con credenciales seguras.
* Un vendedor puede publicar autos con detalles como marca, modelo, año y precio base.
* Una subasta tiene un tiempo de duración definido y puede incluir múltiples autos.
* Los compradores pueden realizar pujas en tiempo real y el sistema determinará el ganador al finalizar la subasta.
* No se permite que un comprador puje por autos publicados por el mismo.

## **Requerimientos No Funcionales**

* En cuanto a seguridad, la autenticación se basa en JWT, encriptación de contraseñas y eliminación lógica de registros.
* Permitir escalabilidad, definiendo una arquitectura que permita el manejo de múltiples subastas simultáneamente.
* El sistema debe ser resistente a fallos, para lo que se utilizó tecnologías que aseguran una respuesta rápida y una comunicación en tiempo real escalable.

# **Diseño del Sistema**

## **Arquitectura del Sistema**

El presente proyecto se basa en una arquitectura cliente-servidor, utilizando las siguientes tecnologías.

* **Backend**

**Capa API:** Contiene los controladores REST que gestionan las peticiones de los usuarios.

**Capa de Servicio:** Es donde se implementa la lógica del negocio del sistema.

**Capa de Repositorio:** Se maneja la persistencia de datos utilizando Spring Data JPA y MySQL.

**Seguridad:** Implementa autenticación y autorización con JWT.

* **Frontend**

Interfaz de usuario moderna y responsiva.

Consumo de API REST para la interacción con el backend.

Uso de WebSockets para comunicación en tiempo real en las subastas.

Esta arquitectura garantiza un sistema modular, escalable y seguro, permitiendo la integración eficiente de las diferentes capas del sistema.

## **Base de Datos**

El sistema de subastas de autos está respaldado por una base de datos relacional diseñada en MySQL. La estructura del BD está compuesta por varias tablas que permiten la correcta gestión de usuarios, autos, subastas y pujas.

**Usuarios**

* Contiene información de los usuarios registrados.
* Se diferencian tres roles: Vendedor, Comprador y Administrador.
* Se almacena la contraseña de forma segura y los usuarios tienen un estado activo/inactivo.

**Autos**

* Se registran los autos a subastar con datos como marca, modelo, año y precio base.
* Relacionado con la tabla de usuarios para identificar al vendedor.
* Tiene un estado que cambia según el proceso de subasta, como son Disponible, Subastado, No vendido o Vendido.

**Subastas**

* Cada subasta tiene un vendedor asociado y una duración determinada.
* Puede incluir varios autos dentro de una misma subasta.
* Cambia de estado según su progreso como puede ser Activa, Finalizada o Cancelada.

**Autos en Subasta**

* Tabla intermedia que se encarga de asociar los autos con una subasta especifica.

**Pujas**

* Registra las ofertas realizadas por los compradores sobre los autos en subasta.
* Cada puja asocia a un comprador y un auto en subasta.
* Se aplica una restricción para evitar pujas de valor negativo o nulo.

**Ganadores**

* Registra el comprador ganador de cada subasta y el precio final de compra.
* Se relaciona con la tabla de usuario y autos en subasta.

# **Implementación**

## **Backend**

* Se utilizó el lenguaje Java con el Framework de Spring Boot.
* Otras tecnologías que se utilizó en el presente proyecto son Hibernate para ORM, Spring Security para la autorización y autenticación con JWT.

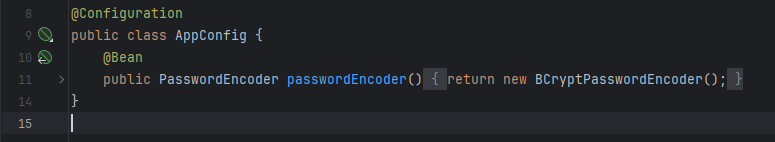
## **Endpoints Principales**

* **Autenticación y Autorización**
  + Path principal: /api/authorization-server
    - POST /login: Permite que los usuarios se autentifiquen y reciban un token JWT.
* **Gestión de Usuarios**
  + Path principal: /api/user
    - GET /findAll: Obtiene una lista de todos los usuarios registrados, solo habilitado para usuarios con rol de administrador.
    - POST /créate: Registra un nuevo usuario en el sistema.
    - PATCH /update: Actualiza la información de un usuario, opción válida únicamente para usuarios con el rol de administrador.
    - DELETE /delete?usuarioId={id}: Elimina un usuario del sistema, opción válida únicamente para usuarios con el rol de administrador.
* **Gestión de subastas de autos**
  + Path principal: /api/car
    - POST /créate: Crea una nueva subasta de autos, opción válida para administradores y vendedores.
    - DELETE /delete?autosSubastaId={id}: Elimina una subasta en específico, opción válida para administradores y vendedores.
* **Gestión de autos**
  + Path principal: /api/car
    - GET /findAll: Obtiene una lista de todos los autos disponibles, opción válida para todos los usuarios.
    - POST /create/{id}: Crea un nuevo auto que se asocia a un usuario, opción válida para administradores y vendedores.
    - PATCH /update?autoId={id}: Modifica la información de un auto, opción válida para administradores y vendedores.
    - DELETE /delete?autoId={id}: Elimina un auto en específico, opción válida para administradores y vendedores.
* **Gestión de pujas**
  + Path principal: /api/bid
    - POST /créate: Crea una nueva puja en una subasta, opción válida para administradores y compradores.
    - PATCH /update/{id}: Modifica la información de una puja especifica.
    - DELETE /delete/{id}: Elimina una puja específica, opción válida para administradores y compradores.
* **Gestión de ganadores**
  + Path principal: /api/winner
    - POST /créate: Registra un ganador de una subasta, opción válida solo para administradores.

## **Seguridad y Autenticación**

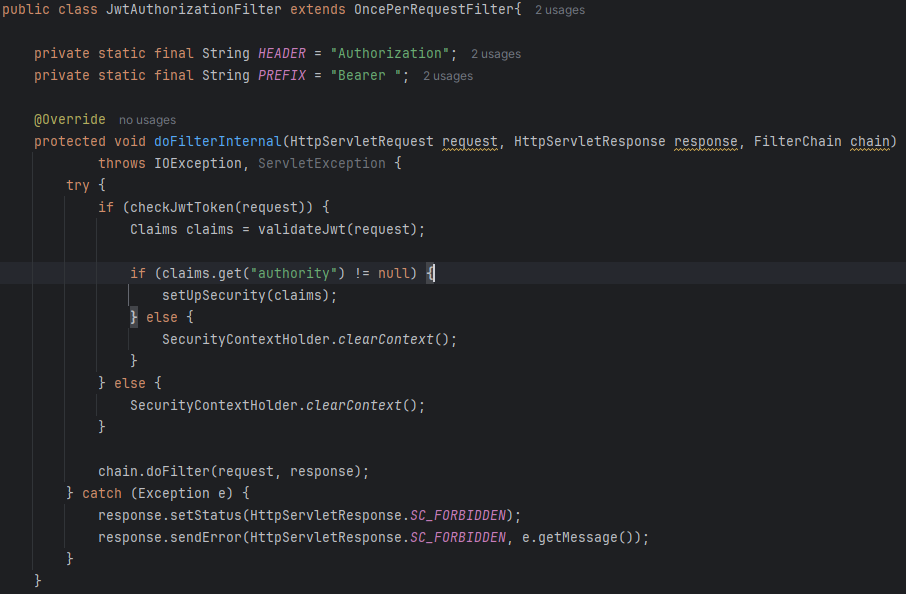
La seguridad y autenticación en el backend se maneja mediante JWT para la validación de usuarios. Se utilizan las siguientes configuraciones:

* **Codificación de contraseñas:** Se utiliza BCryptPasswordEncoder para almacenar contraseñas de forma segura.



*Imagen 1: Clase AppConfig.*

* **Filtro de autorización:** Se utiliza JwtAuthorizationFilter, que valida los tokens en cada petición.



*Imagen 2: Clase JwtAuthorizationFilter*

* **Interceptor de WebSockets:** Se utiliza JwtWebSocketInterceptor para garantizar que solo usuarios autenticados puedan acceder a las subastas en tiempo real.
* **Configuración de Seguridad:** SecurityConfig maneja la protección de rutas y la integración del filtro de autorización JWT.



*Imagen 3: Clase SecurityConfig*

## **WebSockets**

* **Manejo de Pujas:** Para permitir la interacción en tiempo real en las subastas, se utiliza Spring WbSockets con seguridad basada en JWT
* **Configuración:** La configuración de WebSockets está definida en WebSocketConfig



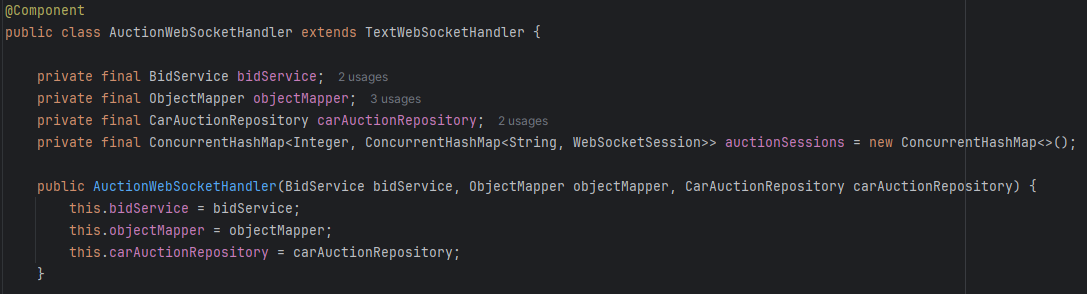
*Imagen 4: Clase WebSocketConfig*

* **Interceptor JWT para WebSockets:** El interceptor JwtWebSocketInterceptor autentica las conexiones WebSocket.



*Imagen 5: Clase JwtWebSocketInterceptor*

* **Manejo de mensajes**: La clase AuctionWebSocketHandler gestiona la interacción entre los usuarios y las subastas.



*Imagen 6 y 7: Clase AuctionWebSocketHandler*

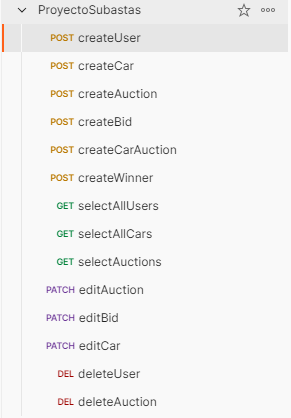
## **Frontend**

* Tecnología utilizada (React, Angular, etc.)
* Pantallas principales y su funcionalidad

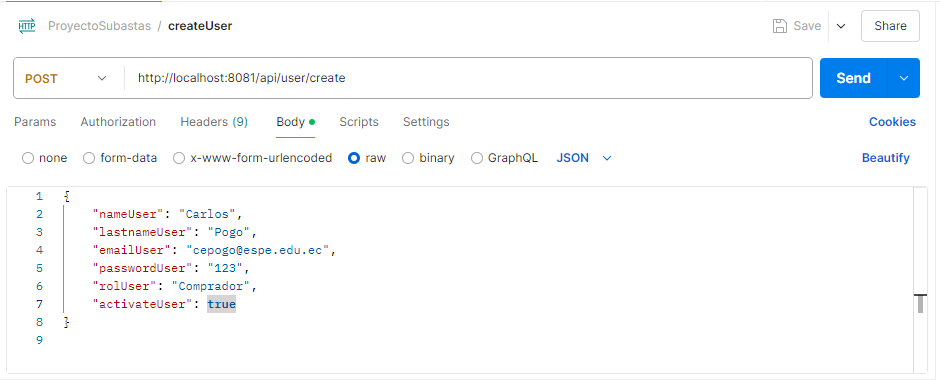
(LLENAR ESTA PARTE LISH Y POGOSH)

# **Pruebas y Validaciones:**

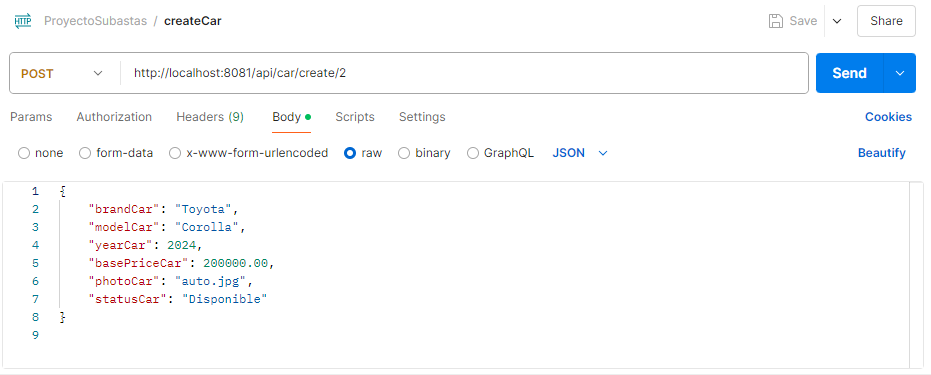
Se realizó las pruebas y comprobación de validaciones en el backend utilizando la herramienta Postman.



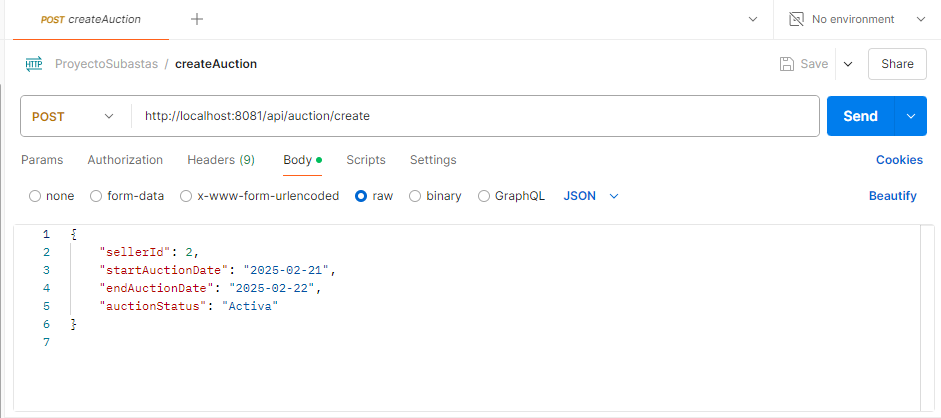
*Imagen 8: Directorio de pruebas realizadas*



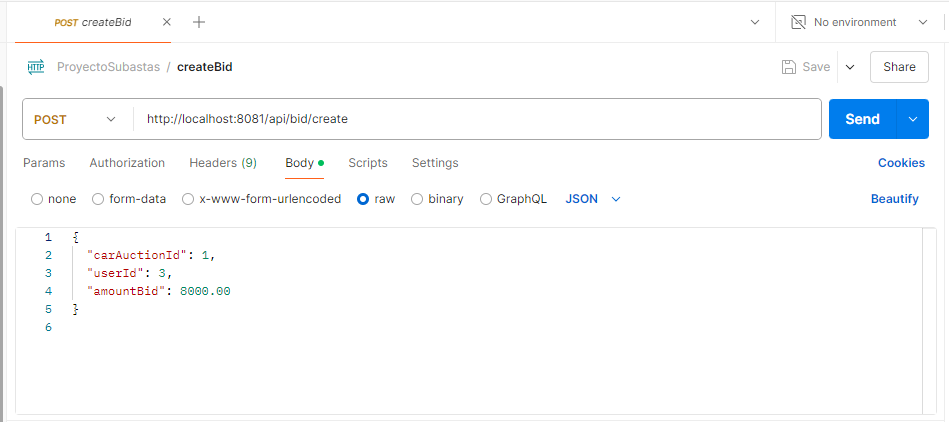
*Imagen 9: Creación de usuario*



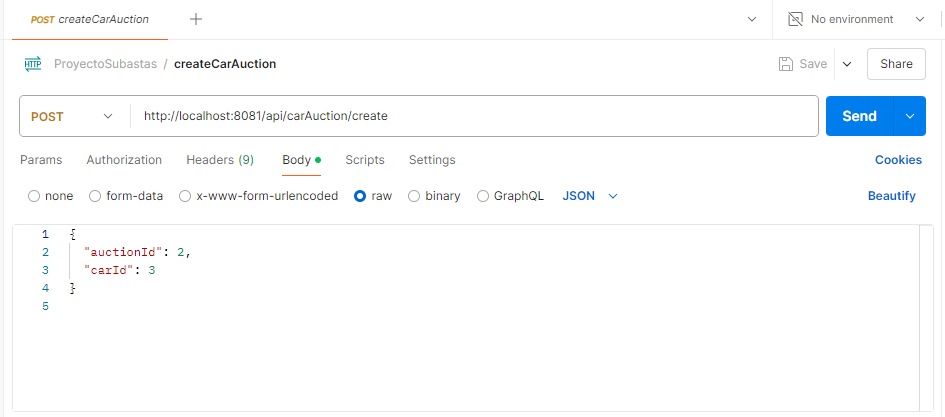
*Imagen 10: Creación de auto*



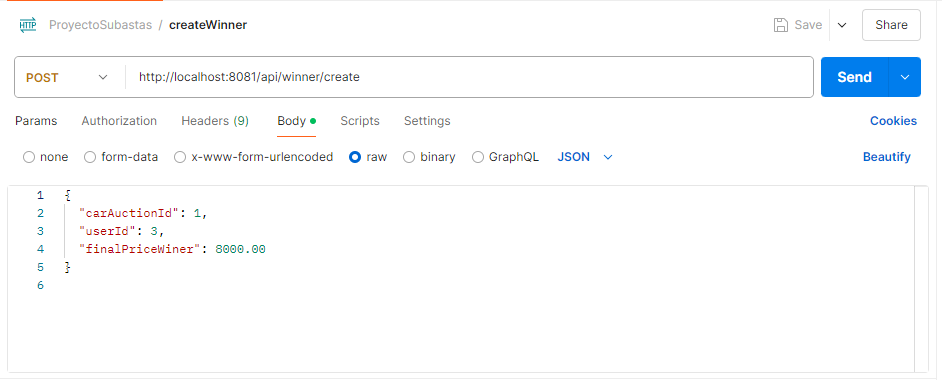
*Imagen 11: Creación de subasta*



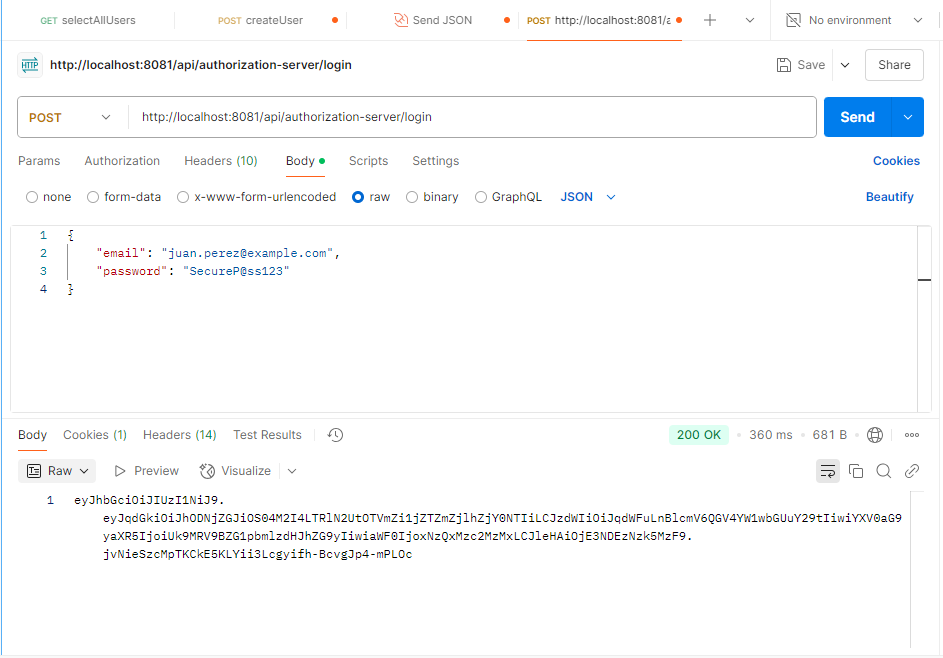
*Imagen 12: Creación de puja*



*Imagen 13: Unión de auto con subasta*



*Imagen 14: Creación de un ganador*



*Imagen 15: Autenticación de un usuario*

# **Desafíos y Aprendizajes**

## **Problemas encontrados y cómo se resolvieron**

Durante el desarrollo del sistema de subastas de autos, se presentaron varios desafíos técnicos y organizativos.

* **Seguridad en la autenticación y autorización**
  + **Problema:** La gestión de autenticación con JWT requería una validación segura de los tokens para evitar accesos no autorizados.
  + **Solución:** Se implementó un JwtAuthorizationFilter en Spring Security para interceptar y validar tokens antes de permitir el acceso a recursos protegidos.
* **Manejo de WebSockets y autenticación**
  + **Problema:** Se requirió asegurar que solo usuarios autenticados pudieran unirse a las salas de subasta y realizar pujas en tiempo real.
  + **Solución:** Se desarrolló un JwtWebSocketInterceptor que verifica los tokens JWT en las solicitudes de conexión WebSocket, permitiendo solo el acceso de usuarios con el rol adecuado.
* **Persistencia y gestión de pujas**
  + **Problema:** Evitar inconsistencias en la base de datos cuando varias pujas se realizaban simultáneamente.
  + **Solución:** Se aplicó el uso de transacciones y sincronización a nivel de servicio (BidService) para garantizar la integridad de las pujas.
* **Comunicación en tiempo real**
  + **Problema:** La transmisión de actualizaciones de pujas en tiempo real a todos los participantes de la subasta debía ser eficiente y sin demoras.
  + **Solución:** Se utilizó AuctionWebSocketHandler para gestionar conexiones WebSocket y un sistema de difusión de mensajes en cada sala de subasta.

## **Posibles mejoras futuras**

* **Mejora en la seguridad del sistema**
  + Implementar renovación automática de tokens JWT con refresh tokens para mejorar la gestión de sesiones.
  + Usar claves más seguras y almacenarlas en un gestor de secretos en lugar de definirlas directamente en el código.
* **Optimización del rendimiento de WebSockets**
  + Reducir el uso de recursos optimizando la estructura de datos de las salas de subasta y los mensajes enviados.
  + Implementar balanceo de carga en la gestión de conexiones WebSocket para manejar un alto número de usuarios concurrentes.
* **Escalabilidad del sistema**
  + Implementar un mecanismo de almacenamiento en caché para reducir la carga en la base de datos.
  + Permitir integración con otros medios de pago para ampliar la funcionalidad del sistema.

# **Conclusiones y Recomendaciones**

## **Cumplimiento de los objetivos**

El desarrollo del sistema de subastas de autos cumplió con los objetivos establecidos:

* Se creó una plataforma segura con autenticación y autorización robusta mediante JWT y filtros de seguridad.
* Se implementó un mecanismo en tiempo real para la gestión de pujas, garantizando transparencia y rapidez en las subastas.
* Se desarrolló una estructura escalable que permite futuras mejoras e integraciones.

## **Impacto del sistema**

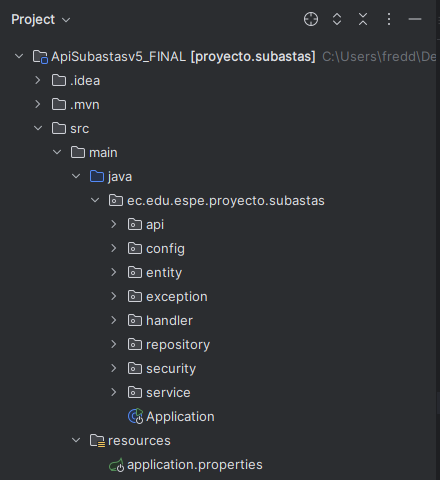
El sistema de subastas de autos ha tenido un impacto positivo en la gestión de este tipo de transacciones, destacándose en los siguientes aspectos:

* **Accesibilidad:** Permite a los usuarios participar en subastas desde cualquier ubicación de manera segura.
* **Transparencia:** La implementación de WebSockets garantiza que todas las pujas sean visibles en tiempo real, reduciendo fraudes y malas prácticas.
* **Automatización:** La digitalización de la subasta reduce la carga administrativa y mejora la eficiencia operativa.

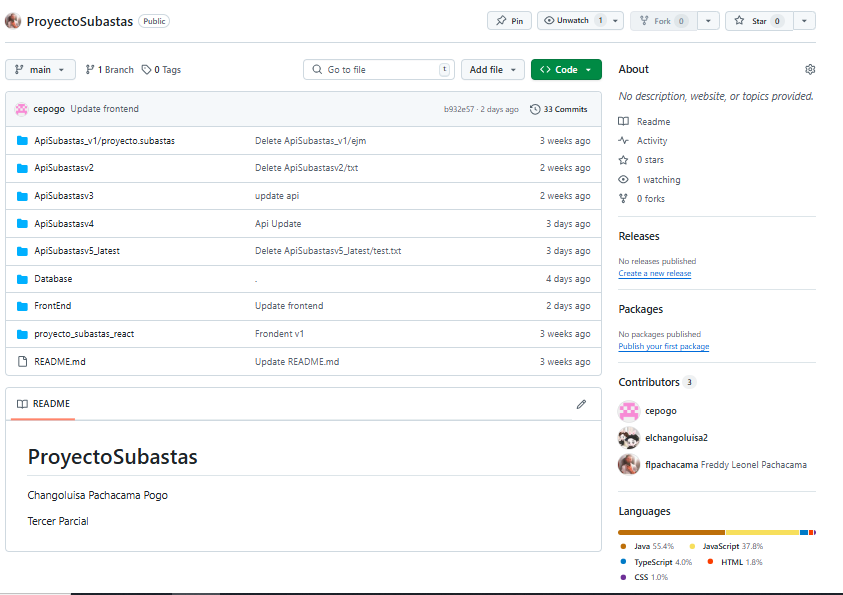
## **Recomendaciones**

* Implementar herramientas de monitoreo para identificar problemas de rendimiento o seguridad en tiempo real.
* Optimizar la interfaz y flujo de la aplicación para facilitar la participación de los usuarios en las subastas.
* Incluir notificaciones en tiempo real vía correo o SMS sobre actualizaciones importantes en las subastas.
* Permitir la personalización de filtros para que los usuarios puedan encontrar subastas según sus intereses.

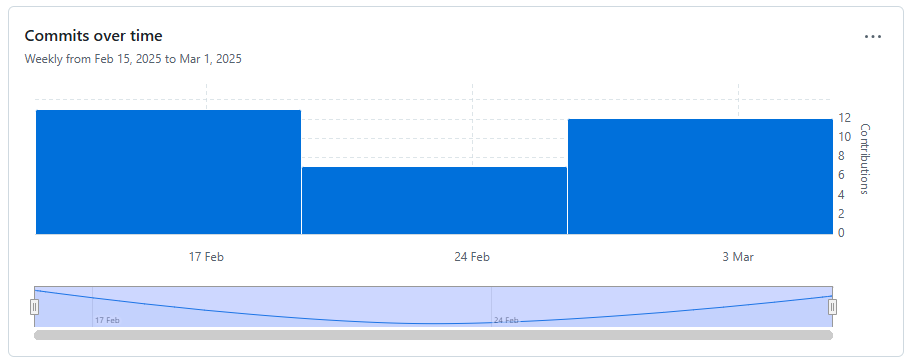
# **Anexos**



*Anexo 1: Directorio final del backend del proyecto*



*Anexo 2: Repositorio en GitHub del proyecto*



*Anexo 3: Commits totales*