

MANUAL DE USUARIO

Plataforma SafeRide – Backend

Introducción

Este manual describe el uso del backend SafeRide, una plataforma de monitoreo de viajes desarrollada en Django REST Framework.

Está orientado a usuarios evaluadores o técnicos que necesiten interactuar, probar y validar el funcionamiento del sistema.

El sistema expone una API REST segura (HTTPS), documentación Swagger y microservicios independientes para demostrar arquitectura moderna.

Requisitos para el Usuario

Para utilizar SafeRide se requiere:

- Navegador web (Chrome, Edge o Firefox)
- Acceso local al proyecto en ejecución
- No se requiere instalación de herramientas adicionales (Swagger y DRF GUI son web)

Acceso a la Plataforma

Página principal

Desde el navegador, acceder a: <https://localhost:8443/>

La página principal muestra:

- Descripción del sistema
- Arquitectura de microservicios
- Accesos rápidos a:
 - Documentación API (Swagger)
 - Panel de administración Django
 - API Root

El navegador mostrará una advertencia de seguridad debido al certificado autofirmado.

Para efectos de evaluación, se debe aceptar y continuar.

Acceso al Panel de Administración

URL: <https://localhost:8443/admin/>

Credenciales de prueba (entorno evaluación):

- Usuario: jose
- Contraseña: Inacap123

Desde el panel se pueden:

- Crear pasajeros
- Crear conductores
- Registrar vehículos
- Visualizar viajes
- Revisar pagos

Uso de la API REST (Interfaz DRF)

URL: <https://localhost:8443/api/>

Desde esta vista el usuario puede:

- Ver todos los endpoints disponibles
- Ejecutar operaciones CRUD usando formularios HTML
- Probar GET, POST, PUT, PATCH y DELETE sin herramientas externas

Ejemplo:

- Crear un pasajero
- Crear un conductor
- Crear un viaje
- Ver la respuesta JSON del sistema

Documentación Swagger (OpenAPI)

URL: <https://localhost:8443/api/docs/>

Swagger permite:

- Ver todos los endpoints documentados
- Expandir cada endpoint
- Ejecutar solicitudes GET y POST
- Visualizar respuestas JSON en tiempo real

Es la principal herramienta para validar el correcto funcionamiento del backend.

Flujo de Uso Principal

1. Crear un **Passenger**
2. Crear un **Driver**
3. Crear un **Vehicle** asociado al conductor
4. Crear un **Trip** seleccionando pasajero y origen/destino
5. Al crear el viaje:
 - Se guarda en la base de datos
 - Se intenta enviar un evento a Kafka (simulado)
6. Crear o consultar el **Payment** asociado al viaje

Microservicios

El sistema incluye microservicios independientes que simulan servicios externos.

Servicios disponibles:

Servicio	Puerto	Función
Route Service	8001	Rutas y tracking
Notifier Service	8002	Notificaciones
Billing Service	8003	Métricas y facturación

Ejemplo de prueba:

<http://localhost:8001/health>

Cada servicio responde con estado ok.

Manejo de Errores

- Validaciones automáticas al crear registros
- Respuestas HTTP claras (400, 404, 201, 200)
- Mensajes descriptivos en JSON
- Logs visibles en consola del servidor

Cierre

SafeRide permite validar de forma clara:

- Arquitectura backend moderna
- API REST profesional
- Seguridad mediante HTTPS
- Documentación OpenAPI
- Uso de microservicios
- Persistencia de datos y pruebas funcionales

Este sistema cumple los objetivos de la evaluación de Backend.