

# Archivos Principales y Funcionalidades

## 1. webpayService.js

- Contiene la configuración y conexión con el SDK oficial de Transbank.
- Define las funciones para crear y confirmar transacciones con WebPay.
- Se configuró para trabajar en modo Integration (test) por defecto, y en modo Production si se cargan las variables de entorno correspondientes.
- Se añadió una función de mapeo de estados para traducir las respuestas de WebPay (AUTHORIZED, FAILED, ABORTED) a los estados internos del sistema (ACCEPTED, REJECTED, ERROR).

## 2. Endpoint Post /properties/buy

- Inicia el proceso de compra al recibir la solicitud del usuario desde el frontend.
- Valida el token del usuario y obtiene la información de la propiedad seleccionada.
- Calcula el costo de la reserva y crea la transacción en WebPay usando el servicio anterior.
- Guarda un nuevo registro en la base de datos con estado PENDING y disminuye la disponibilidad de visitas de la propiedad.
- Devuelve al frontend la URL y token de WebPay para redirigir al usuario al portal de pago.
- Publica la solicitud en el broker MQTT para mantener trazabilidad del proceso.

## 3. Endpoint Post /payments/webpay/return

- Es el callback principal que recibe WebPay después del pago.
- Toma el token\_ws que envía Transbank y confirma la transacción en los servidores de WebPay.
- Actualiza el estado de la solicitud en la base de datos según el resultado:
  - ACCEPTED: pago exitoso
  - REJECTED / ERROR: pago fallido o cancelado
- Si el pago es exitoso:
  - Se genera automáticamente la boleta en AWS Lambda.
  - Se envía un correo de confirmación al usuario con el resultado del pago.
  - Se publica el resultado por MQTT para actualizar el estado en los demás servicios.
- Finalmente, redirige al usuario al frontend con el resultado visible.

## 4. Endpoint Get /payments/webpay/return

- Atiende casos especiales como cancelaciones o expiraciones de sesión.

- Si el usuario cancela, se marca la solicitud como REJECTED y se restaura la visita en la propiedad.
- En caso de errores o timeouts, intenta confirmar nuevamente la transacción y dejar el estado actualizado.
- Garantiza que todos los flujos terminen con un estado definido y sincronizado con los demás sistemas.