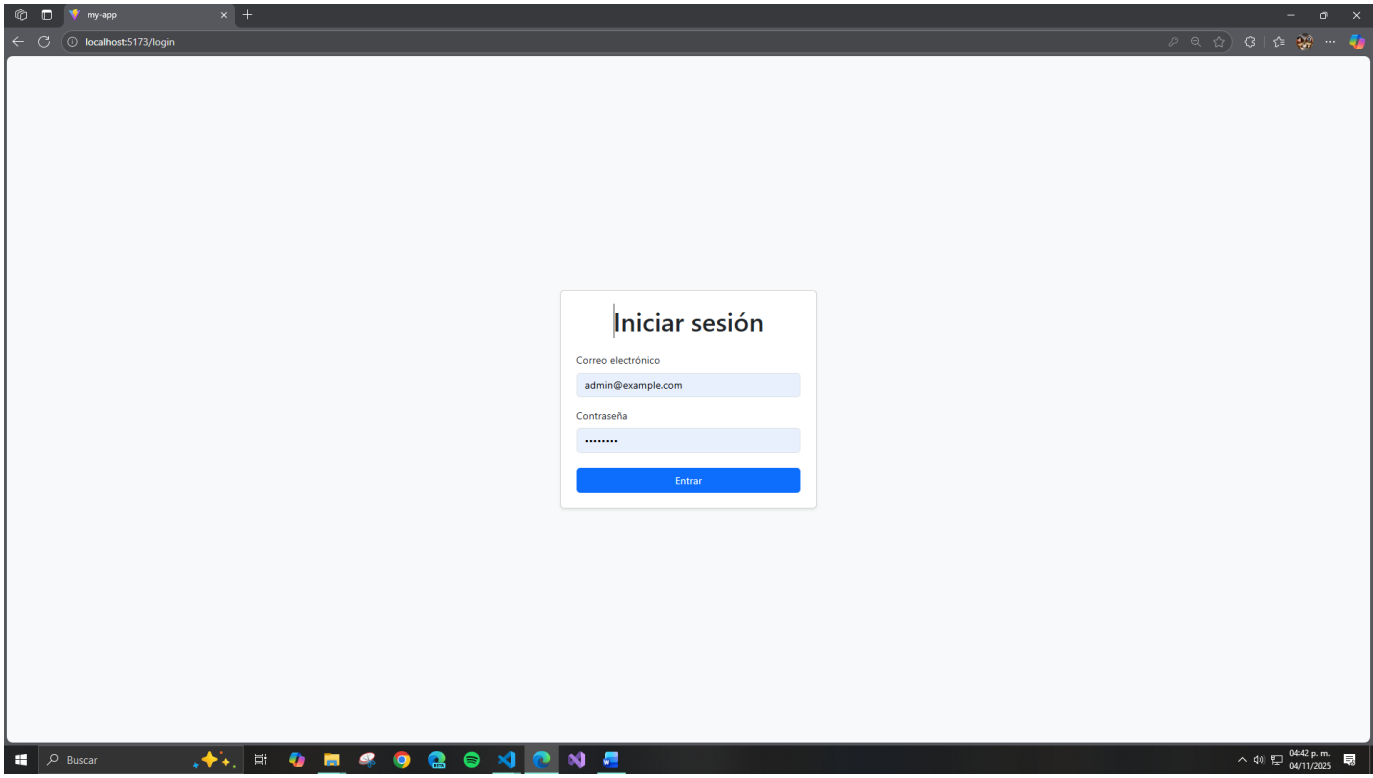
UNIVERSIDAD DON BOSCO

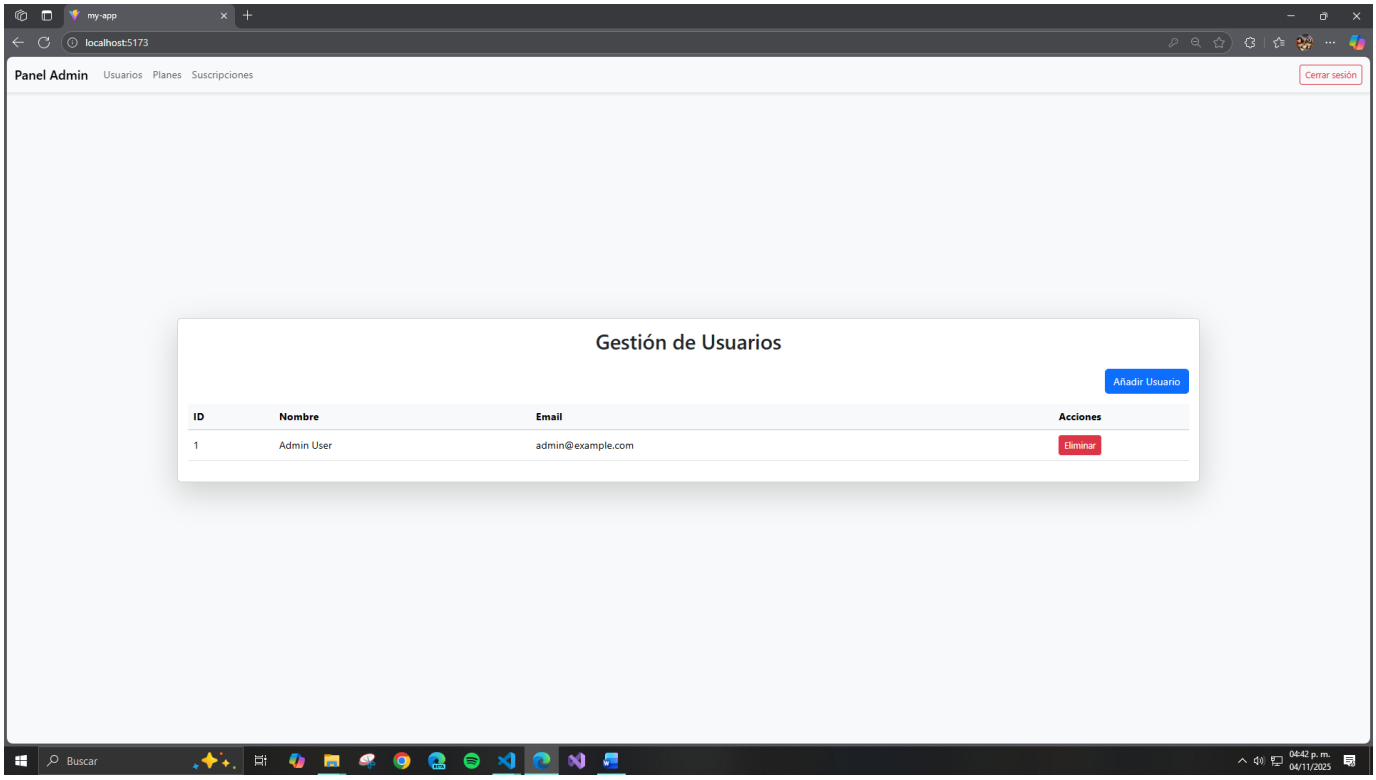
Facultad de Ingeniería
Escuela de Computación

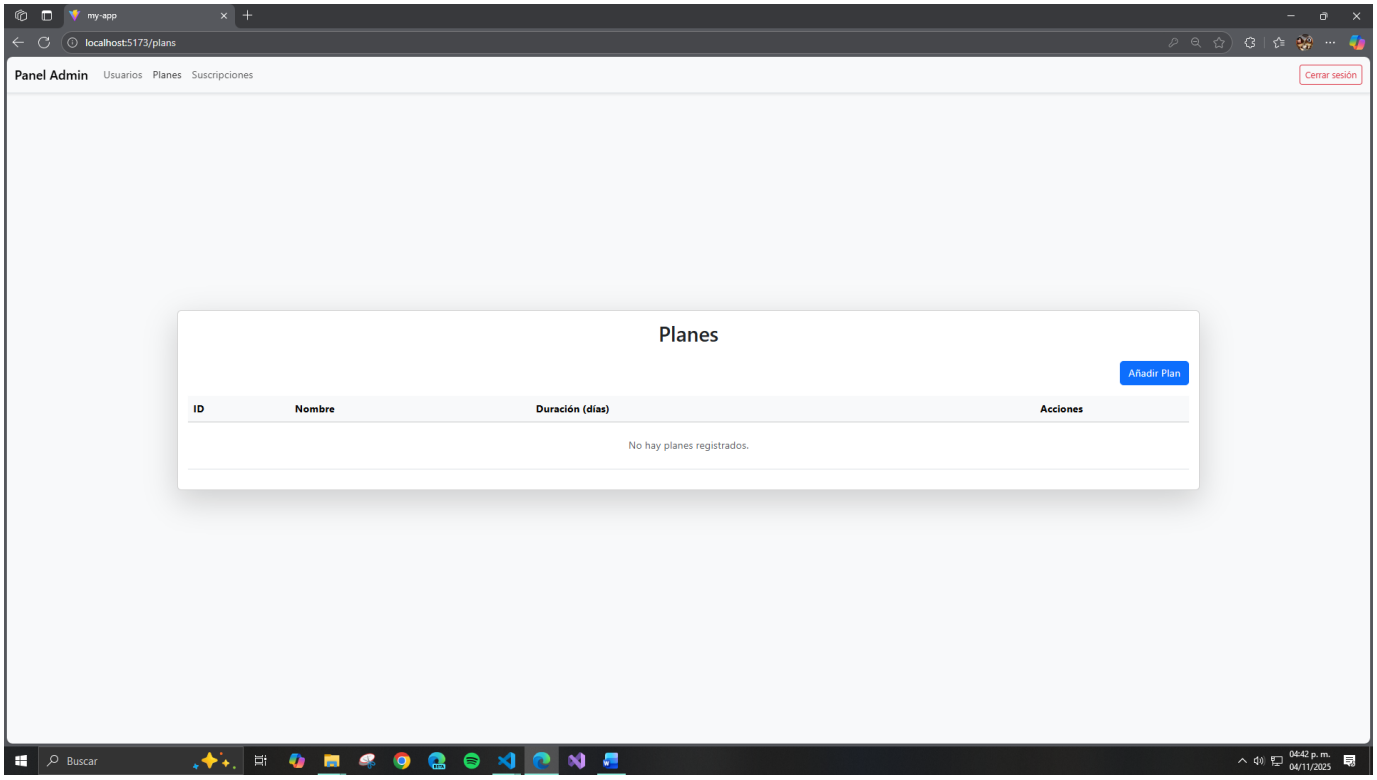
Desafío 1
Grupo: G03L

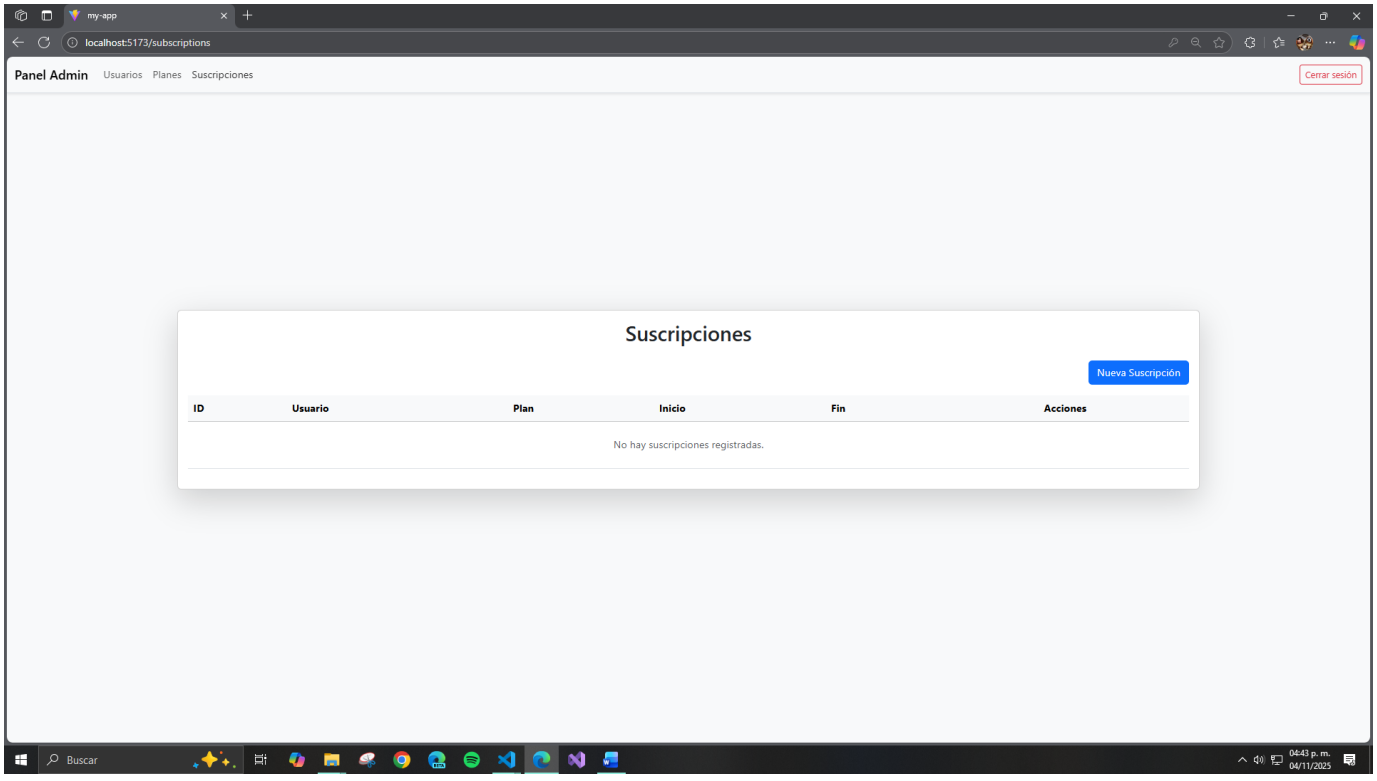
William Ernesto Ramos Valladares RV200068

Repositorio


my-app
localhost:5173/login
Iniciar sesión
Correo electrónico
admin@example.com
Contraseña
Entrar
Buscar
04:42 p. m.
04/11/2025


my-app
localhost:5173
Panel Admin
Usuarios
Planes
Suscripciones
Cerrar sesión
Gestión de Usuarios
Añadir Usuario
ID
Nombre
Email
Acciones
1
Admin User
admin@example.com
Eliminar
Buscar
04:42 p. m.
04/11/2025





El flujo **JWT (JSON Web Token)** implementado sigue el siguiente proceso técnico:

1. **Autenticación inicial:**
   El usuario envía sus credenciales (por ejemplo, email y contraseña) al endpoint de login.

2. **Validación:**
   El backend valida las credenciales contra la base de datos. Si son correctas, genera un **token JWT** firmado con una **clave secreta** (o par de claves en caso de RS256).
3. **Generación del token:**
   El JWT incluye tres partes codificadas en Base64:
   - **Header:** tipo de token y algoritmo de firma.
   - **Payload:** información del usuario (claims) como user_id, role, o exp (fecha de expiración).
   - **Signature:** firma criptográfica que garantiza la integridad del token.
4. **Entrega del token:**
   El servidor devuelve el JWT al cliente, que lo almacena localmente (por ejemplo, en localStorage o en una cookie segura).
5. **Autorización en peticiones posteriores:**
   En cada solicitud a endpoints protegidos, el cliente envía el token en el encabezado Authorization: Bearer <token>.
6. **Verificación del token:**
   El backend intercepta la petición, valida la firma y revisa que el token no haya expirado.
   - Si es válido, se extrae la información del usuario y se concede acceso.
   - Si no, se devuelve un error 401 Unauthorized.
7. **Renovación :**
   Algunos sistemas usan un **refresh token** para emitir nuevos tokens de acceso sin requerir reautenticación completa.