

TRANSPUBLI CALI

Proyecto Integrador – Desarrollo de Software I

Aplicación móvil para facilitar el acceso al transporte público en Cali para personas con discapacidad visual.

1. Estructura del Proyecto

```
proyecto-integrador/
├── frontend/           -> Aplicación móvil en React Native
│   └── App.js          -> Pantalla principal
├── backend/           -> Servidor con Node.js y Express
│   ├── server.js      -> API REST básica
│   └── database.sql    -> Script de base de datos en PostgreSQL
├── documentacion/     -> Documentos de referencia
│   ├── API_REST_TranspubliCali.txt
│   └── ERD_TRANSPUBLI_CALI.txt
├── Plan_Pruebas_*.docx -> Plan de pruebas automatizadas
├── Seguridad_*.docx    -> Análisis de seguridad y autenticación
├── Pruebas_Usuarios_*.docx -> Informe de pruebas con usuarios reales
├── README.md          -> Guía del proyecto
└── TRANSPUBLI_CALI_ENTREGABLE.zip -> Carpeta comprimida con todo el proyecto
```

2. Instalación desde el archivo ZIP

1. Descargar el archivo TRANSPUBLI_CALI_ENTREGABLE.zip.
2. Extraer el contenido en su computador local.
3. Ingresar a la carpeta descomprimida:

```
cd TRANSPUBLI_CALI
```

3. Configuración del Backend

1. Acceder a la carpeta backend:

```
cd backend
```

2. Instalar las dependencias del proyecto:

```
npm install
```

3. Crear la base de datos usando PostgreSQL:

```
psql -f database.sql
```

4. Ejecutar el servidor:

```
node server.js
```

5. Verificar funcionamiento accediendo a:

```
http://localhost:3000/
```

4. Configuración del Frontend

1. En otra terminal, ingresar a la carpeta:

```
cd ../frontend
```

2. Instalar las dependencias:

```
npm install
```

3. Ejecutar la app utilizando Expo:

```
npx expo start
```

4. Escanear el código QR con la aplicación Expo en un dispositivo móvil o abrir el simulador.

5. Documentación Técnica

- API_REST_TranspubliCali.txt: Endpoints disponibles (GET, POST, PUT, DELETE).

- ERD_TRANSPUBLI_CALI.txt: Diagrama entidad-relación de la base de datos.
- Plan_Pruebas: Casos de prueba unitarios e integrados.
- Análisis de Seguridad: Riesgos mitigados según OWASP.
- Pruebas con Usuarios Reales: Resultados de accesibilidad y ajustes.

6. Validación del Proyecto

- Pruebas unitarias y de integración realizadas con Jest y Mocha.
- Autenticación segura con JWT y contraseñas encriptadas con bcrypt.
- Pruebas reales con usuarios con discapacidad visual.
- Simulaciones de carga realizadas con Artillery.

7. Despliegue y Recomendaciones

- El backend puede ser desplegado en plataformas como Heroku o Render.
- El frontend puede ser alojado mediante Expo.
- Se recomienda el uso de variables de entorno para:
 - Clave secreta de JWT
 - Cadena de conexión a la base de datos

8. Observaciones Finales

Este proyecto representa una solución tecnológica accesible que combina funcionalidades móviles, seguridad en el backend, estructura modular y validación real con usuarios. Todos los archivos están organizados en carpetas independientes y se encuentran incluidos dentro del archivo comprimido entregable.