

# Manual Técnico - PROYECTO\_1PEX

## Sistema de Gestión de Reparaciones

### Arquitectura del Sistema

**Visión General** PROYECTO\_1PEX es una aplicación web full-stack que sigue una arquitectura cliente-servidor con separación clara entre frontend y backend.

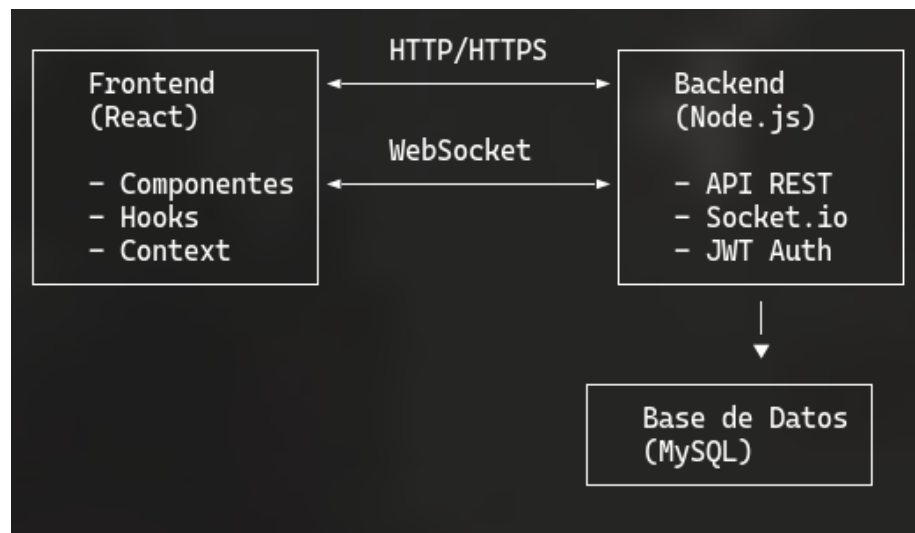


Figure 1: arquitectura

## Tecnologías Utilizadas

### Frontend

- **React 18**: Framework principal para la interfaz de usuario
- **React Router**: Navegación de aplicación de una sola página (SPA)
- **Axios**: Cliente HTTP para llamadas a la API
- **CSS Modules**: Estilos encapsulados por componente
- **Context API**: Gestión de estado global de la aplicación
- **Custom Hooks**: Lógica reutilizable y encapsulada

### Backend

- **Node.js**: Entorno de ejecución JavaScript del lado del servidor
- **Express.js**: Framework web minimalista para Node.js
- **JWT (JSON Web Tokens)**: Autenticación stateless y segura
- **bcrypt**: Librería para hashing de contraseñas
- **Socket.io**: Comunicación en tiempo real vía WebSockets

- **Sequelize:** ORM (Object-Relational Mapping) para MySQL

#### Base de Datos

- **MySQL 8.0+:** Sistema de gestión de base de datos relacional
- **Docker:** Contenedorización para entornos consistentes

#### DevOps y Contenedorización

- **Docker & Docker Compose:** Orquestación de contenedores para desarrollo y despliegue
- **Nginx:** Proxy reverso (para producción)
- **PM2:** Gestor de procesos para aplicaciones Node.js

## Instalación y Configuración para Pruebas con Docker

#### Prerrequisitos

- **Docker** (versión 20.10+)
- **Docker Compose** (versión 1.29+)
- **Git** (para clonar el repositorio)

#### Pasos de Instalación

##### 1. Clonar el repositorio:

```
git clone https://github.com/ElCabaKing/PROYECTO_1PEX.git
cd PROYECTO_1PEX
```

##### 2. Configurar variables de entorno (opcional):

- El proyecto incluye configuraciones por defecto en `docker-compose.yml`
- Si necesitas personalizar, crea archivos `.env` en `backend/` y `frontend/` copiando de `.env.example`

##### 3. Construir e iniciar los contenedores:

```
docker-compose up --build
```

- Este comando construirá las imágenes de Docker para frontend, backend y base de datos
- Iniciará todos los servicios automáticamente
- La base de datos se inicializará con los scripts SQL en `initdb/`

##### 4. Acceder a la aplicación:

- **Frontend:** `http://localhost:3000`
- **Backend API:** `http://localhost:5000`
- La aplicación estará lista para pruebas una vez que todos los contenedores estén ejecutándose

### Verificación de Instalación

- Abre `http://localhost:3000` en tu navegador
- Deberías ver la página de login de PROYECTO\_1PEX
- Credenciales de prueba (según datos iniciales):
  - Usuario: admin
  - Contraseña: admin123 (o según configuración)

### Comandos Útiles para Docker

- **Detener los contenedores:**  
`docker-compose down`
- **Ver logs de un servicio específico:**  
`docker-compose logs backend`  
`docker-compose logs frontend`  
`docker-compose logs db`
- **Reiniciar un servicio:**  
`docker-compose restart backend`
- **Acceder al contenedor de la base de datos:**  
`docker-compose exec db mysql -u root -p proyecto1pex`

### Solución de Problemas Comunes

- **Puerto ocupado:** Asegúrate de que los puertos 3000, 5000 y 3306 estén libres
- **Error de construcción:** Verifica que Docker tenga suficiente memoria asignada
- **Base de datos no inicializada:** Espera a que el contenedor `db` termine de inicializarse (puede tomar 1-2 minutos)

---

\*Manual Técnico v1.0 - Diciembre 2025