**🔙 Backend con Django + DRF**

**⚛️ Frontend con React**

**🛢 PostgreSQL como Base de Datos**

**🔗 APIs externas (como la del SRI)**

**🐳 Contenedores con Docker**

**📦 Buenas prácticas de desarrollo**

**🛠 Librerías Clave del Proyecto TELTEC**

**🐍 Backend (Python / Django)**

|  |  |
| --- | --- |
| **Librería** | **Función** |
| Django | Framework principal para el desarrollo del backend. Permite la creación de modelos, vistas, controladores y administración integrada. |
| djangorestframework (DRF) | Extensión de Django para construir APIs RESTful fácilmente. Permite exponer endpoints seguros, escalables y estructurados. |
| django-cors-headers | Permite habilitar **CORS** (Cross-Origin Resource Sharing), necesario para que el frontend (React) pueda consumir la API del backend sin restricciones. |
| psycopg2 | Librería que permite conectar Django con la base de datos **PostgreSQL**. Es el adaptador oficial de PostgreSQL para Python. |
| drf-yasg | Generador automático de documentación **Swagger** para APIs construidas con DRF. Ideal para documentar y probar endpoints de forma visual. |
| PyJWT / SimpleJWT | Implementa **JSON Web Tokens (JWT)** para autenticar usuarios en el backend y permitir sesiones seguras sin necesidad de cookies. |
| pgcrypto (extensión PostgreSQL) | Usado para cifrar datos sensibles directamente en la base de datos (como contraseñas, tarjetas, etc.). Garantiza protección en nivel de almacenamiento. |
| django-environ | Permite manejar variables de entorno fácilmente, útil para configuración segura y desacoplada (por ejemplo, claves secretas, puertos, credenciales). |

**⚛️ Frontend (React)**

|  |  |
| --- | --- |
| **Librería** | **Función** |
| React | Librería principal para construir la **interfaz de usuario** del sistema de forma declarativa y basada en componentes. |
| Axios | Cliente HTTP para hacer peticiones a la API REST del backend. Más potente y limpio que fetch. |
| React Router DOM | Permite crear rutas y navegación en el frontend de manera dinámica (SPA: Single Page Application). |
| Bootstrap / TailwindCSS | Frameworks de estilos que permiten diseñar interfaces responsivas y modernas rápidamente. |
| Chart.js | Librería para mostrar **gráficas interactivas** y estadísticas de recaudación, clientes, y morosidad. |
| React Icons | Paquete de íconos listos para usar en botones, formularios y navegación. |
| React Toastify | Muestra **notificaciones tipo toast** (éxito, error, advertencia) cuando ocurren eventos como guardar, eliminar, errores, etc. |

**📦 DevOps, Infraestructura y Otros**

|  |  |
| --- | --- |
| **Herramienta** | **Función** |
| Docker + Docker Compose | Permite crear contenedores independientes para el backend, frontend y base de datos. Facilita despliegue, pruebas y versionado del entorno. |
| PostgreSQL 14+ | Base de datos relacional robusta y escalable que almacena toda la información de clientes, pagos, usuarios, etc. |
| Swagger / OpenAPI | Protocolo estándar para documentar y consumir APIs. Usado junto con drf-yasg para la documentación visual de los endpoints. |
| Git + GitHub / GitLab | Control de versiones y colaboración en equipo. Permite mantener versiones, ramas, y revisión de código. |
| Nginx (opcional) | Servidor web que puede actuar como **proxy reverso** entre el frontend, backend y el público. Mejora el rendimiento y la seguridad en producción. |

**🔗 Integración con API del SRI (Servicio de Rentas Internas – Ecuador)**

|  |  |
| --- | --- |
| **Uso de la API del SRI** | **Descripción** |
| **Facturación electrónica** | El sistema TELTEC se conectará con la API oficial del SRI para emitir facturas electrónicas válidas, con firma digital y autorización tributaria. |
| **Firmado de XML** | Se generará un archivo XML con los datos de la factura y se firmará con un certificado digital autorizado (token o archivo .p12). |
| **Autorización SRI** | Luego del firmado, se enviará el XML a la API del SRI para su validación. El sistema almacenará la respuesta con el código de autorización. |
| **Consulta del RUC/Cédula** | Es posible validar la existencia del RUC del cliente usando la API pública del SRI. |
| **Reenvío y cancelación** | Se manejarán procesos de reenvío de facturas rechazadas o cancelación por error administrativo, según los lineamientos del SRI. |