ThreeLogics

Manual del Desarrollador

ThreeLogics

ÍNDICE :

1. Introducción

Este manual proporciona una guía técnica completa para desarrollar, mantener y desplegar el sistema **ThreeLogics**, una plataforma web de gestión de almacenes diseñada para pequeñas y medianas empresas.

El documento está orientado a desarrolladores que se integren al equipo o quieran colaborar en el proyecto, sin necesidad de haber trabajado previamente con el stack.

2. Stack Tecnológico

Frontend

- React con Vite
- Tailwind CSS

Librerías y dependencias como:

- ShadCN UI (componentes accesibles)
- FullCalendar (visualización de eventos)
- Recharts (gráficos en dashboard)

Backend

- Node.js con Express

Base de datos y autenticación

- Supabase (PostgreSQL)
- Supabase Auth (JWT + RLS)
- Supabase Storage (imágenes)

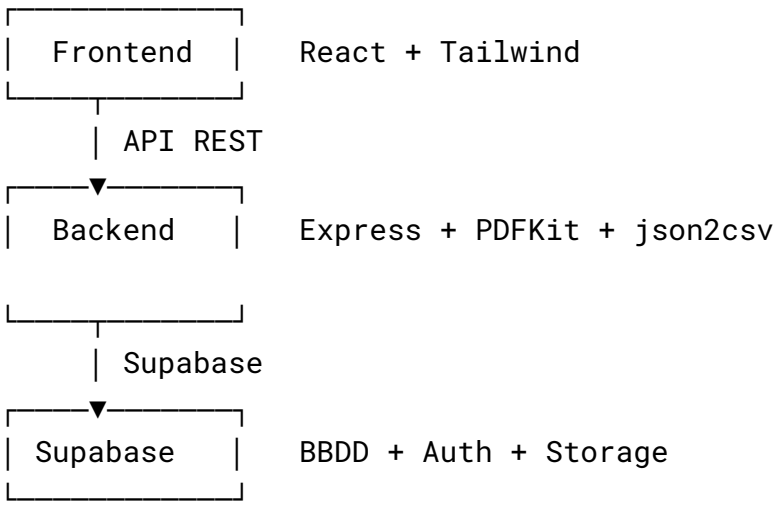
Infraestructura y herramientas

- Vercel (frontend)
- Render(backend)

- GitHub (repositorio + control de versiones)
- Trello + Notion (gestión de tareas y documentación)
- Visual Studio Code (IDE recomendado)

3. Arquitectura del Proyecto

El sistema sigue una arquitectura basada en **capas separadas** y **modularidad**:



- Frontend accede a Supabase directamente (CRUD) y al backend Express (lógica avanzada).
- Backend centraliza operaciones complejas, seguridad adicional y endpoints propios.

4. Estructura del Repositorio

```
/threelogics-app
  ├── src/components
  ├── src/pages
  ├── src/context
  ├── src/assets
  ├── src/utils
  ├── src/supabaseClient.js
  └── App.jsx
```

```
/backend
  ├── routes/
  ├── controllers/
  ├── middlewares/
  ├── utils/
  ├── supabaseClient.js
  └── index.js

.env
README.md
index.html
```

5. Configuración del Entorno

Requisitos

- Node.js v18 o superior
- pnpm (recomendado) o npm
- Supabase Project con claves
- Vercel y Render

Variables de entorno

Frontend – `.env`

```
VITE_SUPABASE_URL=
VITE_SUPABASE_ANON_KEY=
VITE_API_URL=
```

Backend – `.env`

```
SUPABASE_URL=
SUPABASE_SERVICE_ROLE_KEY=
JWT_SECRET=
PORT=5000
FRONTEND_URL=
```

Instalación

```
# Frontend
```

```
pnpm install
pnpm run dev

# Backend
cd backend
pnpm install
pnpm nodemon index.js
```

6. Supabase – Base de Datos

Tablas principales

- auth.user
- productos
- movimientos
- pedidos
- ubicaciones
- categorías

Tablas secundarias

- usuario_ubicaciones
- detalles_pedidos
- conocimientos
- questions
- answers
- estado_sistema
- historial_recuperacion
- suscriptors

Seguridad

- **RLS (Row Level Security)** activado en todas las tablas.
- Políticas SQL según rol (**`admin`**, **`usuario`**).
- Clave **`user_id`** usada para asociar datos al usuario autenticado.

Almacenamiento

- Uso de Supabase Storage para imágenes.

7. Backend – API y Funcionalidades

Principales endpoints:

Método	Ruta	Descripción
GET	**`/dashboard/report e-pdf`**	Genera un PDF con estadísticas y filtros
PUT	**`/pedidos/:id/esta do`**	Actualiza el estado de un pedido
POST	**`/productos`**	Crea un nuevo producto
DELETE	**`/movimientos/:id`**	Elimina un movimiento (según permisos)

Middlewares

- **`authMiddleware.js`:** Valida el token JWT.

PDFKit

- Se usa para crear reportes descargables de productos y movimientos.
- PDFs incluyen columnas dinámicas según el rol.

Json2csv

- Se usa para crear reportes descargables de movimientos.
- Los CSV incluyen columnas dinámicas según el rol.

8. Frontend – Interacción

La interfaz de usuario de ThreeLogics está desarrollada en **React** utilizando **Vite** como entorno de desarrollo rápido, y se estiliza completamente con **Tailwind CSS**, lo que permite una maquetación ágil y responsiva sin necesidad de CSS tradicional.

Estructura del frontend

- Componentes distribuidos por módulo (**`Productos`**, **`Movimientos`**, **`Pedidos`**, etc.)
- **`supabaseClient.js`** ubicado en **`src/`** como punto único de conexión con Supabase.

Integración con Supabase

- Se utiliza el SDK oficial **`@supabase/supabase-js`.**
- El archivo **`supabaseClient.js`** expone una instancia cliente configurada con la URL y la **`anon_key`.**
- El frontend realiza operaciones CRUD directamente contra la base de datos a través del cliente de Supabase.

`Ejemplo de uso en React:`

```
const { data, error } = await supabase.from('productos').select('*')
```

Funcionalidades clave implementadas

- **Sistema de notificaciones** al crear, actualizar o eliminar registros.
- **Visualización condicional** de campos y botones según el rol (**`admin`** o **`usuario`**).
- **Paginación dinámica** basada en la altura del viewport (por ejemplo, en la vista de pedidos).
- **Filtros avanzados y ordenación** en tablas (nombre, categoría, fecha, etc.).
- **Calendario de movimientos** implementado con FullCalendar e integrado con Tailwind.
- **Dashboard analítico** con Recharts, mostrando métricas adaptadas según el rol.
- **Validación de cuenta y recuperación de contraseña** por correo.

Librerías clave

- **`@supabase/supabase-js`:** conexión con la base de datos y autenticación.
- **`react-router-dom`:** control de rutas de la aplicación.
- **`fullcalendar`**: calendario de movimientos.
- **`recharts`:** gráficos circulares y de barras en el panel de estadísticas.

9. Despliegue

Frontend

- Desplegado automáticamente desde GitHub a Vercel.
- Variables **`.env`** gestionadas desde el dashboard de Vercel.

Backend

- Render / Google Cloud Run (futuro).

Base de Datos

- Supabase gestionado desde dashboard web.
- RLS, triggers y funciones SQL se actualizan manualmente o con scripts de migración.

10. Buenas Prácticas

- Pull Requests revisadas siempre por al menos 1 miembro del equipo.
- Documentación viva en Notion.
- Validaciones en frontend y backend.
- No modificar políticas RLS sin test en entorno de staging.

11. Recursos Adicionales

- Repositorio GitHub: https://github.com/threeLogics/
- Notion (decisiones técnicas y arquitectura)
- Trello (sprints y bugs)
- Documentación de Supabase: https://supabase.com/docs
- Manual de Usuario y Admin en carpeta **`/documentacion`**
- google analytics para monitoreo en tiempo real

12. Contacto del equipo

- Adrián Vaquero – Frontend & UI
- Iker Domínguez – Backend & Supabase
- Daniel Ramiro –Testing & Arquitectura
- Email de contacto general: **soporte@threelogics.com**

Nota final:
Este proyecto está pensado para crecer. Cada desarrollador nuevo que se una debería poder ponerse en marcha en menos de 1 hora si este manual se sigue correctamente.