**Documento Funcional: SEMINARIO\_FINAL**

**1. Descripción General**

SEMINARIO\_FINAL es una aplicación web para gestionar afectaciones y reclamos. Incluye un backend (Python/Flask) que maneja la lógica y la base de datos, y un frontend (React) que ofrece la interfaz de usuario. El sistema permite autenticación, creación de bases de datos, afectaciones y reclamos.

**2. Requisitos**

* **Hardware**: PC con 4GB RAM mínima.
* **Software**:
  + Node.js (v18+)
  + Python (3.11+)
  + Git
  + Render y Vercel (opcional para despliegue)

**3. Estructura del Proyecto**

text

ContraerAjusteCopiar

SEMINARIO\_FINAL/

├── backend/

│ ├── app.py

│ ├── requirements.txt

│ └── controladores/

└── frontend/

├── src/

├── .env

└── package.json

**4. Funcionalidades**

* **Autenticación**: Login con username y password (via /SGE/Login).
* **Panel de Control (Admin)**:
  + Crear nueva base de datos (/API/BD/CreaTablas).
  + Crear afectaciones (/API/AF/NuevaAfectacionAT/0).
  + Generar reclamos (/API/AF/GeneraReclamos).
  + Normalizar afectaciones (/API/AF/NormalizarElementosAleatorios).
* **Acceso**: Solo admins (detectado por sessionStorage.getItem('rol')).

**5. Instalación y Ejecución (Localhost)**

* **Clonar**

bash

ContraerAjusteCopiar

git clone https://github.com/robertosl77/SEMINARIO\_FINAL.git

cd SEMINARIO\_FINAL

* **Backend**
  + cd backend
  + pip install -r requirements.txt
  + python app.py (corre en http://localhost:5000)
* **Frontend**
  + cd frontend
  + npm install
  + Edita .env:

text

ContraerAjusteCopiar

REACT\_APP\_API\_URL=http://localhost:5000

* + npm run start (corre en http://localhost:3000)
* **Prueba**
  + Abre http://localhost:3000, ingresa test/Test1234.

**6. Despliegue**

* **Backend (Render)**
  + Vincula backend/ en Render.
  + Configura:
    - Root: backend
    - Build: pip install -r requirements.txt
    - Start: python app.py
  + URL: https://seminario-final.onrender.com
* **Frontend (Vercel)**
  + cd frontend
  + vercel
  + Configura:
    - Scope: Roberto's projects
    - Name: seminario-final
    - Branch: dev\_host
    - Env Var: REACT\_APP\_API\_URL=https://seminario-final.onrender.com
  + vercel --prod
  + URL: ej. https://seminario-final.vercel.app

**7. Flujo de Login (Resumido)**

1. Usuario abre http://localhost:3000.
2. Ingresa credenciales.
3. Frontend envía POST a REACT\_APP\_API\_URL/SGE/Login.
4. Backend valida y responde (200 o error).
5. Si ok, redirige al dashboard.

**8. Notas**

* Usa master para localhost (presentación).
* Ajusta CORS en Controlador.py si hay errores:

python

ContraerAjusteCopiar

from flask\_cors import CORS

CORS(app, resources={r"/SGE/\*": {"origins": "\*"}})