

Guía Técnica

1. Descripción General del Sistema

Esta guía técnica documenta los componentes internos de la aplicación web desarrollada para resolver problemas de cálculo numérico. La lógica numérica se implementa en Octave, mientras que la interfaz y la API se desarrollaron en tecnologías web modernas, siguiendo una estructura de proyecto basada en dos componentes principales: frontend y backend.

2. Tecnologías Utilizadas

- **Frontend:** HTML, CSS, JavaScript
 - **Backend:** Python + Flask
 - **Lógica Numérica:** Octave (ejecutado desde Python con subprocess)
 - **Servidor Local:** Flask en modo desarrollo
 - **Control de versiones:** Git + GitHub
-

3. Arquitectura del Sistema

El sistema sigue una arquitectura cliente-servidor:

- **Cliente (Frontend):** Interfaz gráfica donde el usuario ingresa datos (index.html con style.css y script.js).
- **Servidor (Backend):** Recibe los datos del frontend, ejecuta los cálculos en Octave a través de app.py y devuelve los resultados procesados.
- **Octave:** Ejecuta el cálculo numérico usando el script dosisóptima.m y almacena los resultados en resultado.txt.

[Usuario] \leftrightarrow [Frontend (HTML/JS)] \leftrightarrow [Backend (Flask)] \leftrightarrow [Octave (dosisóptima.m) \rightarrow resultado.txt]

4. Estructura del Proyecto

```
/
├── frontend/
│   ├── index.html    # Interfaz de usuario
│   ├── style.css     # Página principal
│   ├── script.js     # Estilos de la interfaz
│   └── medicamentos.json # Lógica del cliente
│                       # Datos de prueba / ejemplo
├── backend/
│   ├── app.py        # Lógica de servidor y cálculo
│   ├── dosisóptima.m # Servidor Flask y lógica de integración
│   └── resultado.txt  # Script Octave con los cálculos numéricos
│                       # Archivo de salida de resultados
└── docs/             # Documentación del proyecto
```

5. Instalación y Ejecución

5.1 Requisitos Previos

- Python 3.8 o superior
- Octave instalado y accesible desde terminal
- pip (gestor de paquetes de Python)

5.2 Instalación

git clone <https://github.com/usuario/proyecto-final.git>

cd proyecto-final

pip install -r requirements.txt

5.3 Ejecución en modo local

cd backend

python app.py

Luego abrir index.html en la carpeta frontend usando un navegador.

6. Mantenimiento y Actualizaciones

- Para modificar los métodos numéricos, editar dosisóptima.m.
- Para cambiar la lógica del servidor o rutas, modificar app.py.
- Para cambiar estilos o estructura de interfaz, editar archivos en frontend/.
- Para actualizar dependencias, agregar/modificar en requirements.txt si existe.

7. Seguridad y Buenas Prácticas

- Validar todos los datos del usuario en app.py.
- Controlar errores al ejecutar Octave y manejo del archivo resultado.txt.
- Mantener el código bien comentado y estructurado.
- Usar control de versiones para gestionar cambios.

8. Créditos

Desarrollado por: JUAN DAVID PERTUZ HOYOS

Asignatura: Modelamiento en Ingeniería de Sistemas

Profesor: José Leonardo Simancas García
Universidad de la Costa