

## **PRUEBA TÉCNICA – FULL STACK DEVELOPER ACCIÓN CLIMÁTICA**

### **V. SISTEMA DE INFORMACION**

#### **Flujo de la Solución**

##### **1. Registro y Autenticación del Usuario**

- El usuario se registra a través del frontend.
- El frontend envía una solicitud API al backend.
- El backend valida los datos y los guarda en la base de datos.
- El usuario puede iniciar sesión y acceder a sus datos.

##### **2. Gestión de Parcelas**

- El usuario registra nuevas parcelas mediante un formulario en el frontend.
- El frontend envía los datos al backend mediante una solicitud API.
- El backend guarda los datos en la base de datos y los asocia con el usuario correspondiente.
- El usuario puede ver, editar y eliminar parcelas desde el panel de control del frontend.

##### **3. Registro y Monitoreo de Actividades**

- El usuario registra actividades diarias mediante un formulario en el frontend.
- El frontend envía los datos al backend.

- El backend guarda los datos en la base de datos y los asocia con la parcela correspondiente.
- El usuario puede ver el historial de actividades, así como gráficos y reportes en el frontend.

#### **4. Visualización y Análisis de Datos**

- El usuario puede visualizar datos históricos y actuales de actividades y parcelas.
- El frontend solicita datos al backend para generar gráficos y reportes.
- El backend recupera los datos de la base de datos y los envía al frontend.
- El frontend presenta los datos de manera visual e intuitiva.

### **Tecnologías Recomendadas**

#### **1. Frontend:**

- React con Material-UI para componentes de interfaz de usuario.
- Axios para las solicitudes HTTP al backend.

#### **2. Backend:**

- Flask (Python) con SQLAlchemy para la interacción con la base de datos.
- Flask-JWT-Extended para la gestión de tokens JWT.

#### **3. Base de Datos:**

- PostgreSQL o MySQL para el almacenamiento de datos relacionales.

#### **4. Docker:**

- Docker para contenerización de los servicios.
- Docker Compose para la orquestación y despliegue.

## **Implementación y Despliegue**

### **1. Desarrollo:**

- Desarrollar los componentes del frontend y backend de manera modular.
- Implementar la base de datos y definir los esquemas.
- Contenerizar cada componente utilizando Docker.

### **2. Pruebas:**

- Realizar pruebas unitarias y de integración para asegurar la funcionalidad de cada componente.
- Realizar pruebas de rendimiento para asegurar que el sistema puede manejar múltiples usuarios.

### **3. Despliegue:**

- Utilizar Docker Compose para desplegar los contenedores en un entorno de prueba.
- Documentar el proceso de despliegue para facilitar la implementación en entornos de producción.