

Guía de Configuración Local de Serena MVP

Esta guía proporciona instrucciones detalladas para configurar un entorno de desarrollo local para el proyecto Serena MVP.

Requisitos Previos

Antes de comenzar, asegúrate de tener instalado:

- **Python 3.9+**

```
bash

python --version
# o en algunos sistemas
python3 --version
```

- **Node.js 16+**

```
bash

node --version
npm --version
```

- **Git**

```
bash

git --version
```

- **Editor de código** (recomendado: Visual Studio Code, PyCharm)

Paso 1: Clonar el Repositorio

1. Abre una terminal o línea de comandos
2. Clona el repositorio:

```
bash

git clone https://github.com/tu-usuario/serena-project.git
cd serena-project
```

Paso 2: Configurar el Backend

2.1. Crear y activar entorno virtual

```
bash
```

```
cd backend
```

```
# En Windows
```

```
python -m venv venv
```

```
venv\Scripts\activate
```

```
# En macOS/Linux
```

```
python3 -m venv venv
```

```
source venv/bin/activate
```

2.2. Instalar dependencias

```
bash
```

```
pip install -r requirements.txt
```

2.3. Configurar variables de entorno

```
bash
```

```
# Copiar el archivo de ejemplo
```

```
cp .env.example .env
```

```
# Editar el archivo .env con tu editor preferido
```

```
# Por ejemplo:
```

```
# DATABASE_URL=sqlite:///instance/app.db
```

```
# GOOGLE_API_KEY=tu_clave_api_aquí
```

```
# VERIFY_TOKEN=serena-secret-key
```

```
# JWT_SECRET_KEY=tu_clave_jwt_secreta
```

2.4. Inicializar la base de datos

```
bash
```

```
# Crear directorio para SQLite si no existe
```

```
mkdir -p instance
```

```
# Inicializar la base de datos y aplicar migraciones
```

```
flask db upgrade
```

```
# Opcionalmente, crear datos de prueba
```

```
flask seed # Si se ha implementado el comando seed
```

Paso 3: Configurar el Frontend

```
bash
```

```
cd ../frontend
```

```
# Instalar dependencias
```

```
npm install
```

```
# Crear archivo de variables de entorno
```

```
echo "VITE_API_URL=http://localhost:5000/api" > .env
```

Paso 4: Iniciar los Servicios

4.1. Iniciar el Backend

En una terminal (mantén el entorno virtual activado):

```
bash
```

```
cd backend
```

```
flask run
```

El backend estará disponible en: <http://localhost:5000>

4.2. Iniciar el Frontend

En otra terminal:

```
bash
```

```
cd frontend
```

```
npm run dev
```

El frontend estará disponible en: <http://localhost:3000>

Paso 5: Verificar la Instalación

5.1. Verificar el Backend

Abre un navegador o usa curl para verificar el backend:

```
bash
```

```
curl http://localhost:5000/api/health
```

```
# Deberías recibir una respuesta similar a:
```

```
# {"status": "ok", "version": "0.1.0"}
```

5.2. Verificar el Frontend

Abre <http://localhost:3000> en tu navegador. Deberías ver la página de inicio de sesión de Serena.

Paso 6: Configurar el Webhook para Desarrollo

Para probar la funcionalidad de WhatsApp localmente, puedes usar ngrok o una herramienta similar para exponer tu servidor local a internet.

6.1. Instalar y configurar ngrok

1. Descargar ngrok desde ngrok.com
2. Extraer y seguir las instrucciones de configuración
3. Exponer tu servidor Flask:

```
bash
```

```
ngrok http 5000
```

4. Tomar nota de la URL generada (por ejemplo, <https://a1b2c3d4.ngrok.io>)

6.2. Configurar Twilio o MessageBird

Usar la URL de ngrok para configurar el webhook en Twilio o MessageBird:

- URL: <https://a1b2c3d4.ngrok.io/api/webhook/whatsapp>
- Método: POST
- Token de verificación: El mismo que configuraste en tu archivo .env

Paso 7: Datos de Prueba y Cuentas

7.1. Cuenta de Administrador por Defecto

El sistema viene preconfigurado con una cuenta de administrador:

- Email: `admin@serena.com`
- Contraseña: `admin123`

IMPORTANTE: Cambiar esta contraseña inmediatamente en un entorno real.

7.2. Crear Datos de Prueba Manualmente

Puedes crear datos de prueba a través de la interfaz de administrador:

1. Inicia sesión con la cuenta de administrador
2. Ve a la sección "Administración"
3. Crea pacientes, cuidadores y usuarios de prueba

Solución de Problemas Comunes

Error al iniciar el backend

Problema: Error "Address already in use"

Solución: El puerto 5000 ya está en uso. Cambia el puerto:

```
bash  
flask run --port=5001
```

Problema: Error de módulo no encontrado

Solución: Asegúrate de que estás en el directorio correcto y el entorno virtual está activado:

```
bash  
cd backend  
source venv/bin/activate # o venv\Scripts\activate en Windows  
pip install -r requirements.txt
```

Error al iniciar el frontend

Problema: Puerto 3000 en uso

Solución: Normalmente Vite ofrecerá automáticamente usar otro puerto. Acepta la sugerencia presionando 'y'.

Problema: Error "Module not found"

Solución: Reinstala las dependencias:

```
bash  
  
rm -rf node_modules  
npm install
```

Error de conexión entre frontend y backend

Problema: El frontend no puede conectar con el backend

Solución: Verifica que:

1. El backend está ejecutándose
2. Las variables de entorno están configuradas correctamente
3. No hay problemas de CORS (cors.init_app está configurado en el backend)

Desarrollo y Pruebas

Ejecutar Pruebas del Backend

```
bash  
  
cd backend  
pytest
```

Ejecutar Pruebas del Frontend

```
bash  
  
cd frontend  
npm test
```

Formato de Código y Linting

```
bash
```

```
# Backend (si se ha configurado flake8 o black)
```

```
cd backend
```

```
black .
```

```
flake8
```

```
# Frontend
```

```
cd frontend
```

```
npm run lint
```

```
npm run format
```

Siguientes Pasos

Una vez que tengas el entorno local funcionando, puedes:

1. Explorar y familiarizarte con la estructura del código
2. Implementar nuevas características
3. Corregir problemas en el [tracker de issues](#)
4. Ejecutar las pruebas antes de enviar cambios

Para contribuir al proyecto, consulta el archivo [CONTRIBUTING.md](#).