

Guía de Instalación

Sistema de Gestión Médica Hospitalaria

Grupo 04

13 de noviembre de 2025

Índice

1	Introducción	2
2	Instalación Local (Modo Desarrollo)	2
2.1	Paso 1: Requisitos Previos	2
2.2	Paso 2: Clonar el Repositorio	2
2.3	Paso 3: Configuración de la Base de Datos MySQL/XAMPP	3
2.4	Paso 4: Configurar el Backend (Python / FastAPI)	3
2.5	Paso 5: Configurar el Frontend (React/Vite)	4
2.6	Paso 6: Iniciar el Sistema	4
2.7	Acceso y Usuarios Predeterminados	4
3	Instalación con Docker (Recomendada para Producción)	5
3.1	Paso 1: Requisitos Previos	5
3.2	Paso 2: Clonar el repositorio	5
3.3	Paso 3: Desplegar los contenedores	5
3.4	Paso 4: Acceder al sistema	5
3.5	Comandos útiles de Docker	5

1. Introducción

Esta guía detalla los pasos necesarios para instalar y configurar el **Sistema de Gestión Médica Hospitalaria**. Se ofrecen dos métodos principales:

- **Instalación Local (Modo Desarrollo):** Instalación manual y personalizada ideal para entornos de desarrollo.
- **Instalación con Docker (Recomendado para Producción):** Instalación automatizada que facilita el despliegue en entornos estables y reproducibles.

Siga el procedimiento adecuado según su entorno.

2. Instalación Local (Modo Desarrollo)

2.1. Paso 1: Requisitos Previos

Asegúrese de tener instaladas las siguientes herramientas:

- Python 3.10 o superior
- Node.js 18 o superior (incluye npm)
- MySQL 8.0 o superior o bien, puede instalar XAMPP, que incluye MySQL y un panel sencillo para administrar la base de datos.
- Git

Nota: Verifique las versiones ejecutando en terminal los comandos `python -version`, `node -version`, `mysql -version`. Con XAMPP, basta con iniciar los servicios desde el panel de control.

Alternativa MySQL/XAMPP:

Si desea evitar la configuración por línea de comandos, puede usar XAMPP para administrar MySQL fácilmente. Solo inicie los servicios Apache y MySQL desde el panel de XAMPP, luego acceda a <http://localhost/phpmyadmin> desde su navegador. Allí podrá crear la base de datos y usuario de forma gráfica.

2.2. Paso 2: Clonar el Repositorio

Abra una terminal y ejecute:

```
git clone https://github.com/YormanOna/GRUPO04_SGM_27897.git  
cd GRUPO04_SGM_27897
```

2.3. Paso 3: Configuración de la Base de Datos MySQL/XAMPP

Opción A: Usando MySQL vía terminal

Inicie sesión en MySQL y cree la base de datos y usuario:

```
mysql -u root -p
CREATE DATABASE GestionMedicaDB;
CREATE USER 'hospital_user'@'localhost' IDENTIFIED BY 'hospital_pass';
GRANT ALL PRIVILEGES ON GestionMedicaDB.* TO 'hospital_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Opción B: Usando XAMPP y phpMyAdmin

1. Inicie los servicios Apache y MySQL desde el panel de XAMPP.
2. Abra su navegador y diríjase a <http://localhost/phpmyadmin>.
3. Cree la base de datos GestionMedicaDB mediante el menú de phpMyAdmin.
4. En la pestaña de Privilegios agregue un usuario hospital_user y asigne la contraseña hospital_pass, otorgando todos los permisos a la nueva base de datos.

(Optional) Para cargar datos de ejemplo:

```
mysql -u hospital_user -p GestionMedicaDB < DB/InsertDatos.sql
```

2.4. Paso 4: Configurar el Backend (Python / FastAPI)

Crear y activar entorno virtual e instalar dependencias

Linux / Mac:

```
cd Backend
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Windows:

```
cd Backend
python -m venv venv
venv\Scripts\activate
pip install -r requirements.txt
```

Configurar variables de entorno

Cree el archivo .env en la carpeta Backend/ con el siguiente contenido:

```
DB_USER=hospital_user
DB_PASSWORD=hospital_pass
DB_HOST=localhost
DB_PORT=3306
DB_NAME=GestionMedicaDB
```

```
SECRET_KEY=tu_clave_secreta_muy_segura_cambiar_en_produccion  
ALGORITHM=HS256  
ACCESS_TOKEN_EXPIRE_MINUTES=43200
```

Importante:

Reemplace DB_PASSWORD y SECRET_KEY por valores seguros antes de usar en producción.

2.5. Paso 5: Configurar el Frontend (React/Vite)

```
cd ../Frontend  
npm install
```

Cree un archivo .env en la carpeta Frontend/ con:

```
VITE_API_URL=http://localhost:8000  
VITE_WS_URL=ws://localhost:8000/ws
```

2.6. Paso 6: Iniciar el Sistema

En dos terminales separadas, ejecute:

Terminal 1 - Backend:

```
cd Backend  
source venv/bin/activate # O venv\Scripts\activate en Windows  
uvicorn app.main:app --reload
```

Terminal 2 - Frontend:

```
cd Frontend  
npm run dev
```

El sistema estará disponible en <http://localhost:5173>.

2.7. Acceso y Usuarios Predeterminados

Rol	Email / Contraseña
Super Administrador	superadmin@hospital.com / superadmin123
Administrador	admin@hospital.com / admin123
Médico	medico@hospital.com / medico123
Enfermera	enfermera@hospital.com / enfer123
Farmacéutico	farmacia@hospital.com / farma123

3. Instalación con Docker (Recomendada para Producción)

3.1. Paso 1: Requisitos Previos

Instale:

- [Docker Desktop](#)
- [Git](#)

3.2. Paso 2: Clonar el repositorio

```
git clone https://github.com/Yorman0na/GRUPO04_SGM_27897.git  
cd GRUPO04_SGM_27897
```

3.3. Paso 3: Desplegar los contenedores

```
cd "Docker Compose"  
docker-compose up -d
```

Este comando levantará 3 contenedores: MySQL, Backend (FastAPI) y Frontend (React).

3.4. Paso 4: Acceder al sistema

- Frontend (Aplicación Web): <http://localhost:5173>
- Backend API: <http://localhost:8000>
- Documentación Swagger: <http://localhost:8000/docs>
- MySQL (cliente externo):
Host: localhost
Puerto: 3307
Usuario: hospital_{user}
Contraseña : hospital_{pass}

3.5. Comandos útiles de Docker

<code>docker-compose logs -f</code>	<i># Ver logs en tiempo real</i>
<code>docker-compose ps</code>	<i># Ver estado de contenedores</i>
<code>docker-compose restart</code>	<i># Reiniciar contenedores</i>
<code>docker-compose down</code>	<i># Detener y eliminar contenedores</i>
<code>docker-compose down -v</code>	<i># Eliminar volúmenes (bdd)</i>
<code>docker-compose up -d --build</code>	<i># Reconstruir imágenes</i>
<code>docker stats</code>	<i># Ver uso de recursos</i>

Notas Finales

- Los usuarios por defecto se crean automáticamente la primera vez que se levanta el sistema. Es recomendable cambiar las contraseñas en producción.
- Para mejorar el rendimiento en producción, utilice el archivo `docker-compose.prod.yml`, que incluye Nginx, Gunicorn con múltiples workers y configuraciones optimizadas de MySQL.