**Manual de Operación**

**Estructura del Manual de Operación**

Un buen Manual de Operación debe incluir:

1. **Introducción** – Breve descripción del sistema y su propósito.
2. **Requisitos Previos** – Qué necesita el usuario antes de operar el sistema.
3. **Acceso a la Aplicación** – Cómo iniciar sesión o abrir el sistema.
4. **Funciones Principales** – Explicación detallada de cada función del sistema.
5. **Cierre de Sesión y Apagado** – Cómo salir correctamente del sistema.
6. **Resolución de Problemas Comunes** – Errores frecuentes y soluciones.

**1. Introducción**

Este sistema permite gestionar y registrar visitas a instituciones educativas, almacenando la información en una base de datos **MySQL** administrada dentro de un entorno **Docker**. La aplicación está desarrollada en **Python con Flask** y se accede a través de un **navegador web**.

**2. Requisitos Previos**

Antes de operar el sistema, asegúrate de que:

✅ Docker está instalado y funcionando correctamente.  
✅ Los contenedores de **MySQL, Flask y Nginx** están corriendo.  
✅ La base de datos **Visitas** está creada y configurada.  
✅ Python y las dependencias están instaladas en el contenedor de Flask.

Si aún no has iniciado los servicios, puedes hacerlo con:

docker-compose up -d

Para verificar los contenedores en ejecución:

docker ps

**3. Acceso a la Aplicación**

**3.1 Iniciar el Servidor**

Para ejecutar el sistema dentro de Docker, usa el siguiente comando:

docker-compose up -d

Esto iniciará los contenedores de MySQL, Flask y Nginx en segundo plano.

**3.2 Abrir en el Navegador**

Una vez iniciado el servidor, abre tu navegador y accede a:

http://localhost:8081

Si el servidor está en una máquina remota, usa la **IP del servidor** en lugar de localhost:

http://(IP\_SERVIDOR):8081

**4. Funciones Principales**

La aplicación permite:

✅ Registrar visitas a instituciones educativas.  
✅ Consultar visitas previas almacenadas en la base de datos.  
✅ Administrar usuarios y permisos de acceso.

Para ingresar datos en la aplicación, sigue el instructivo detallado dentro del sistema.

**5. Cierre de Sesión y Apagado**

**5.1 Detener el Servidor**

Si necesitas detener el sistema, ejecuta en la terminal:

docker-compose down

Esto apagará todos los contenedores asociados al sistema.

**6. Resolución de Problemas Comunes**

| **Error** | **Posible Solución** |
| --- | --- |
| mysql.connector.errors.ProgrammingError: 1045 | Verifica el usuario y contraseña en las variables de entorno o en config.py. |
| No carga la página en el navegador | Asegúrate de que los contenedores están corriendo (docker ps). Reinicia el contenedor de Flask si es necesario. |
| No conecta a MySQL | Verifica que el contenedor de MySQL está activo con docker ps y reinícialo si es necesario: docker restart mysql-server. |