

API de Gestión de Clientes para Agencia de Viajes

Resumen del proyecto

Este proyecto consiste en una **API REST desarrollada con FastAPI** que permite gestionar los clientes de una agencia de viajes. La aplicación se conecta a una base de datos **MySQL/MariaDB** y permite realizar las siguientes operaciones:

- Dar de alta nuevos clientes.
- Consultar la lista completa de clientes.
- Buscar por nacionalidad.
- Ver qué clientes tienen seguro de viaje.
- Actualizar datos de contacto.
- Eliminar clientes.

La API se puede desplegar tanto en un entorno local como en **una red local**, lo que permite su uso compartido desde otros equipos conectados a la misma red.

[Enlace](#) al repositorio.

Tecnologías utilizadas

- **Python 3.12**
 - **FastAPI** → Framework principal de la API.
 - **Uvicorn** → Servidor ASGI que ejecuta la API.
 - **SQLAlchemy** → ORM para manejar la base de datos MySQL/MariaDB.
 - **Pydantic** → Validación de datos.
 - **Passlib[bcrypt]** → Encriptación de contraseñas.
-

Instalación y configuración

1. Crear un entorno virtual

```
python -m venv venv
```

Activar el entorno virtual:

```
# Windows
.\venv\Scripts\activate
# Linux/Mac
source venv/bin/activate
```

Si da error, abrir PowerShell como administrador y ejecutar:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

2. Instalar dependencias

```
pip install fastapi uvicorn[standard] pymysql sqlalchemy  
"passlib[bcrypt]==1.7.4" "bcrypt==4.0.1"
```

3. Configurar la base de datos

Editar database.py con tus datos de conexión:

```
DB_USER = "root"  
DB_PASSWORD = ""  
DB_HOST = "127.0.0.1"  
DB_NAME = "agencia_viajes"
```

Ejemplo de URL:

```
SQLALCHEMY_DATABASE_URL =  
f"mysql+pymysql://{DB_USER}:{DB_PASSWORD}@{DB_HOST}/{DB_NAME}"
```

4. Crear la base de datos

En tu entorno MySQL/MariaDB:

```
CREATE DATABASE agencia_viajes;
```

5. Crear las tablas

El proyecto ya incluye el modelo de datos en models.py. Solo hay que ejecutar:

```
from database import Base, engine  
Base.metadata.create_all(bind=engine)
```

6. Ejecutar la API

```
uvicorn main:app --reload
```

Para permitir el acceso desde otros equipos de la red local:

```
uvicorn main:app --host 0.0.0.0 --port 8000 --reload
```

Luego, desde otro dispositivo conectado a la misma red, se puede acceder mediante:

```
http://<ip_local>:8000/docs
```

Ejemplo:

`http://192.168.1.45:8000/docs`

Estructura del proyecto

`agencia_viajes_backend/`

```
|  
├── main.py  
├── database.py  
├── models.py  
├── schemas.py  
├── crud.py  
└── auth.py
```

Explicación de los archivos: - `main.py`: punto de entrada de la aplicación, define las rutas. - `database.py`: conexión a la base de datos y configuración del motor SQLAlchemy. - `models.py`: define las tablas en la base de datos. - `schemas.py`: modelos de entrada/salida (Pydantic). - `crud.py`: funciones básicas de creación, lectura, actualización y borrado.

Endpoints principales

Método	Ruta	Descripción
POST	<code>/register</code>	Registrar un nuevo usuario/cliente
POST	<code>/login</code>	Iniciar sesión y obtener token JWT
POST	<code>/mensajes</code>	Enviar un mensaje (ejemplo de relación entre usuarios)
GET	<code>/mensajes/{usuario_id}</code>	Obtener los mensajes de un usuario

(Para la versión de la agencia de viajes, estas rutas pueden adaptarse a la gestión de clientes.)

Solución de problemas

Si aparecen errores con `bcrypt` o `passlib`:

```
pip uninstall bcrypt passlib  
pip install "passlib[bcrypt]==1.7.4" "bcrypt==4.0.1"
```

Si el entorno virtual se corrompe:

```
deactivate
rm -r -fo .\env\
python -m venv env
.\env\Scripts\activate
pip install -r requirements.txt
```

Despliegue en red local

Para compartir la API con otros equipos: 1. Ver la IP del servidor: `bash ipconfig` 2. Copiar la dirección IPv4 (ej. 192.168.1.45). 3. Ejecutar la API con: `bash uvicorn main:app --host 0.0.0.0 --port 8000 --reload` 4. Acceder desde otro equipo en la misma red a: `http://192.168.1.45:8000/docs`

Manual para el comprador / reinstalación

1. Instalar Python 3.12 o superior.
2. Clonar o copiar el proyecto `agencia_viajes_backend`.
3. Crear y activar el entorno virtual.
4. Instalar las dependencias indicadas.
5. Configurar la base de datos con tus credenciales.
6. Crear la base de datos `agencia_viajes` y las tablas.
7. Ejecutar el servidor con `uvicorn main:app --reload`.

Con estos pasos, el comprador podrá reinstalar la aplicación en cualquier momento sin depender del desarrollador.

Conclusión

La **API de Gestión de Clientes para Agencia de Viajes** ofrece una solución completa, funcional y documentada para el control de clientes, con un diseño modular, fácil despliegue y compatibilidad en red local. Esta documentación garantiza que el cliente pueda mantener o reinstalar el sistema sin dificultades.