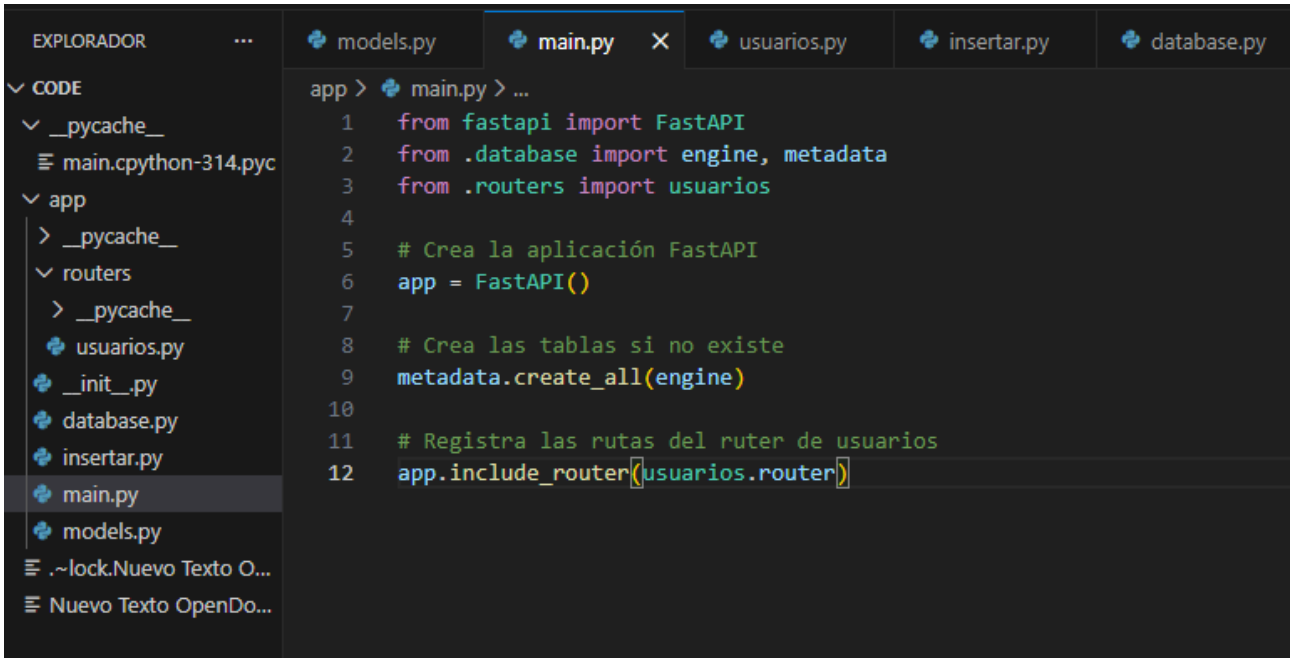
Crea la aplicación y la tabla si no existiera




Defino la tabla crear

Creo un router para agrupar las rutas relacionadas con los usuarios.

Abro las dependencias de sesión

Llamo a la clase de insertar usuarios

+ el de listar usuarios en -JSON

models.py | main.py | usuarios.py | insertar.py | database.py

app > routers > usuarios.py > listar_usuarios

```python
from fastapi import APIRouter, Depends
from sqlalchemy.orm import Session
from ..database import SessionLocal
from ..models import usuarios

# Crea un router para agrupar las rutas relacionadas con usuarios
router = APIRouter(prefix="/usuarios", tags=["Usuarios"])

# Dependencia que crea y cierra una sesión en la base de datos por petición
def get_db():
    db = SessionLocal()     #Abre la sesion
    try:
        yield db            #Devuelve la ruta solicitada
    finally:
        db.close()          #Se cierra al terminar la petición

#Creación de usuario
@router.post("/")
def crear_usuario(data: dict, db: Session= Depends(get_db)):
    # Construye una sentencia SQL INSERT con los datos recibidos
    query = usuarios.insert().values(data)
    # Ejecuta la sentencia en la base de datos
    db.execute(query)
    # Guarda los cambios
    db.commit()
    # Mensaje tranqulizador de todo OK
    return{"mensaje":"Usuario creado correctamente"}

# Listar usuarios
@router.get("/")
def listar_usuarios (db: Session= Depends(get_db)):
    # Construye una sentencia SELECT * FROM usuaraios (obtiene todo los datos de la tabla 'usuarios')
    query = usuarios.select()
    # Ejecuta la consulta y obtiene todos los registros
    resultado = db.execute(query).fetchall()
    # Convierto el resultado que esta en ROW a JSON
    usuarios_lista = [dict(row._mapping) for row in resultado]
    # Devuelve los datos ahora en JSON
    return usuarios_lista
```

La clase del Insert

models.py | main.py | usuarios.py | insertar.py | database.py

app > insertar.py > ...

```python
from sqlalchemy import insert, MetaData, Table

# Importo la conexión y la tabla
from models import engine, usuarios

# Lista de usuarios a insertar
nuevos_usuarios = [
    {"nombre":"Abc","apellidos":"periodico","año_nacimiento":1990},
    {"nombre":"Manolo","apellidos":"prejubilado","año_nacimiento":1960},
    {"nombre":"Austero","apellidos":"Bermejales","año_nacimiento":1900},
]

# insertar en la base de datos
with engine.connect() as conn:
    conn.execute(usuarios.insert(), nuevos_usuarios)
    conn.commit()

# Mensaje tranquilizador
print("Usuarios insertados correctamente")
```

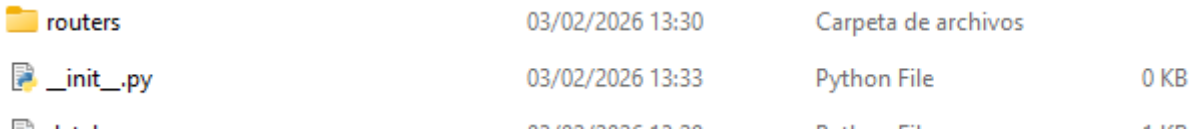
Conexión con la base de datos

models.py | main.py | usuarios.py | insertar.py | database.py

app > database.py > ...

```python
from sqlalchemy import create_engine, MetaData
from sqlalchemy.orm import sessionmaker

# Datos de la conexión
usuario = "root"
contraseña = ""
host = "localhost"
nombre_bd = "prueba"

# Cadena de conexión para SQLAlchemy usando el driver PyMySQL
DATABASE_URL = f"mysql+pymysql://{usuario}:{contraseña}@{host}/{nombre_bd}"

# Crea el motor de conexión (engine) que gestiona la comunicación con MySQL
engine = create_engine(DATABASE_URL)
# Objeto que almacena la estructura de las tablas
metadata = MetaData()

# Crea una fábrica de sesiones para interactuar con la base de datos
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
```

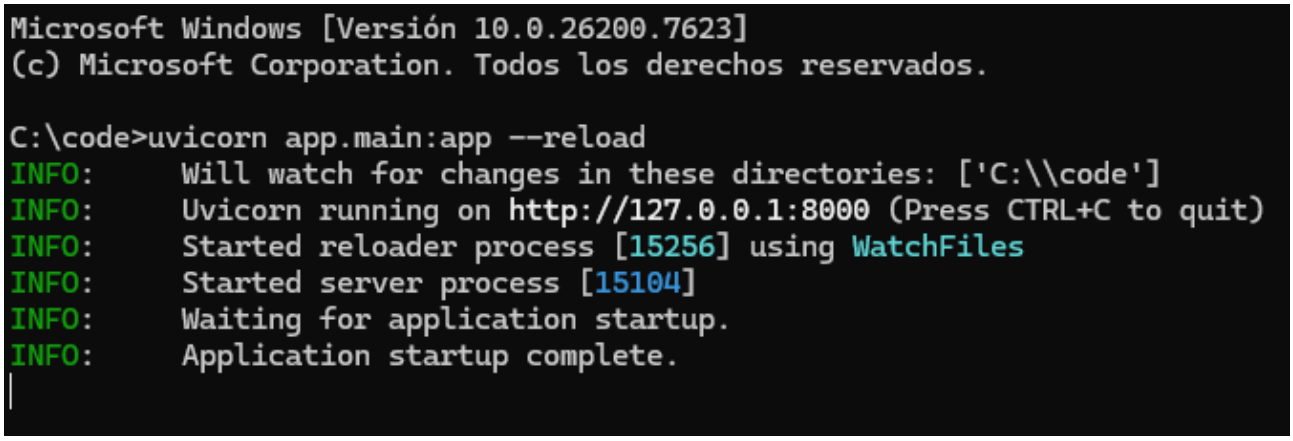
Para poder iniciar la app se necesita la clase –init--.py obligatoriamente aunque este vacía.



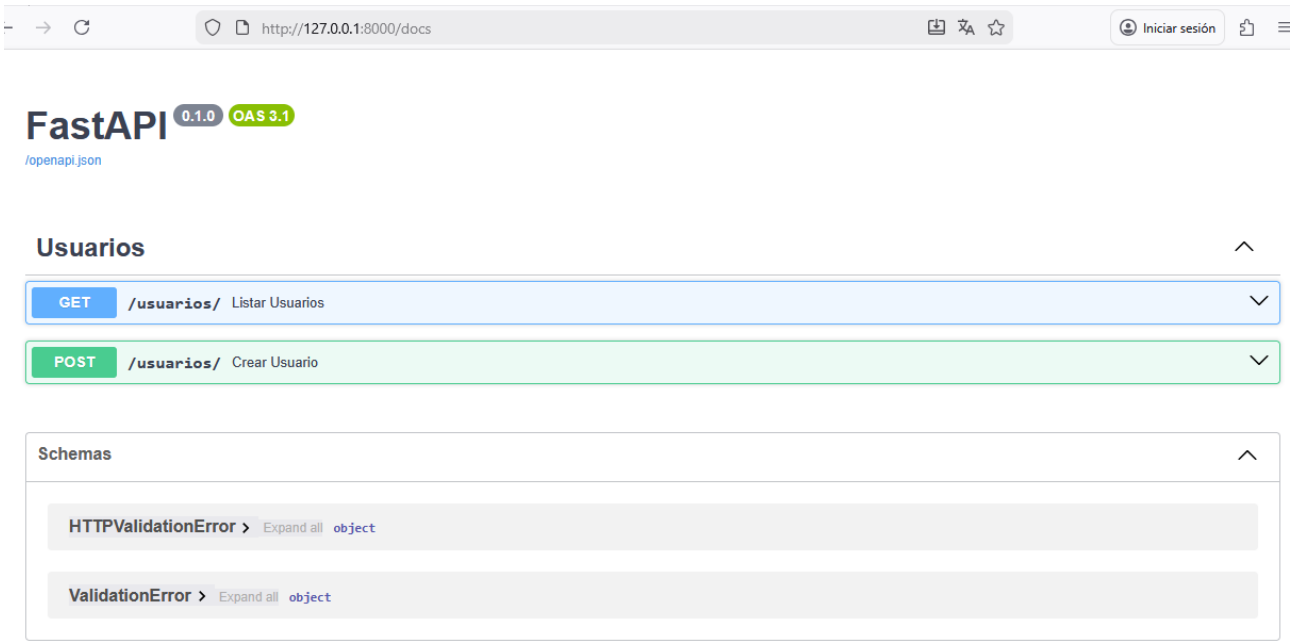

Ejecuto en la consola de comando desde la carpeta del proyecto.

uvicorn app.main:app --reload

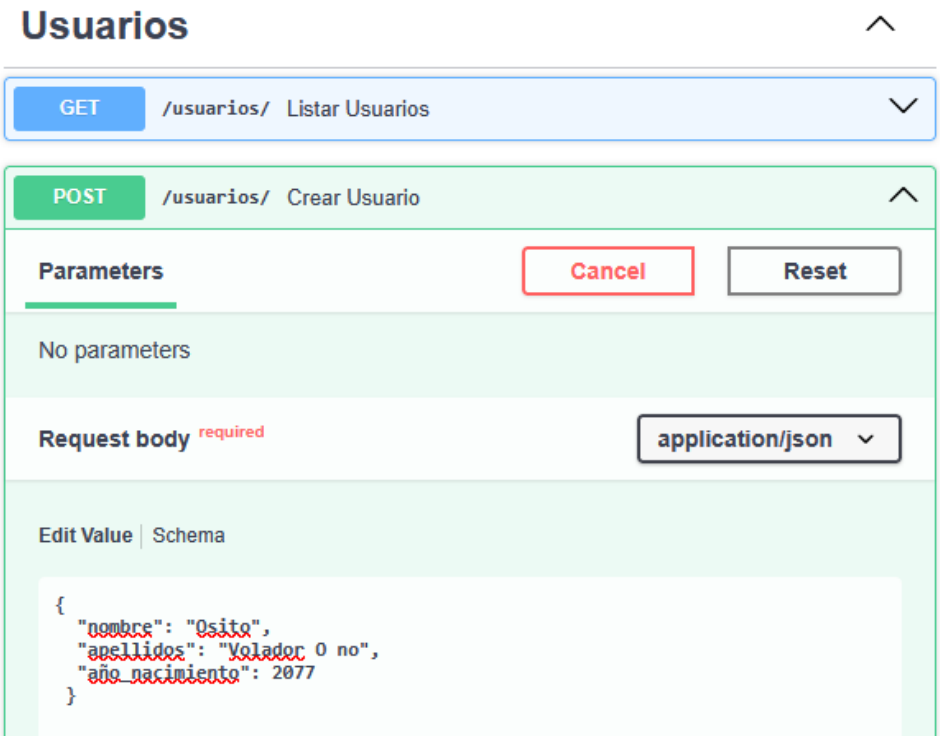
Y se inicia.



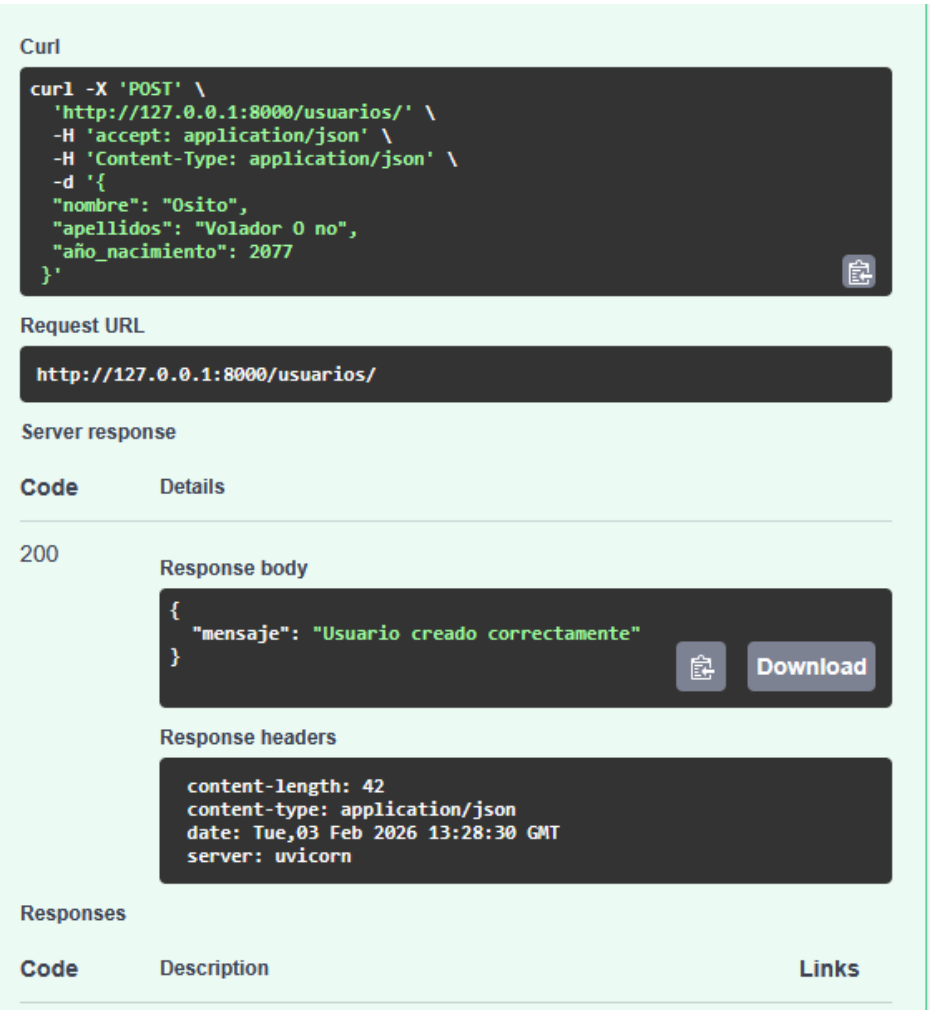

Ahora entro en “http://127.0.0.1:8000/docs”



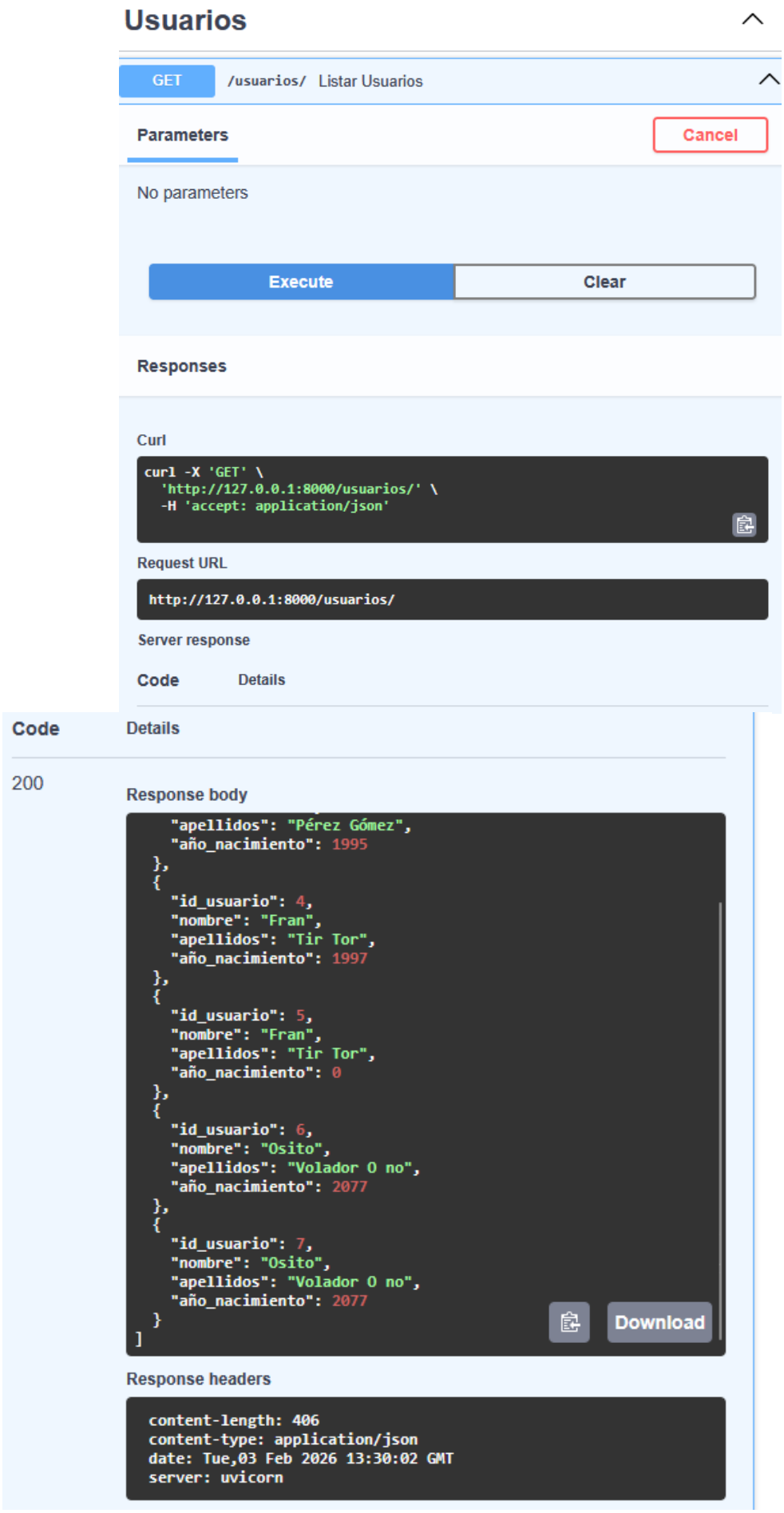
Eejcuto desde Usuarios POST los usuarios a añadir




Muestra el mensaje con el contenido añadido y mas abajo el de confirmación



Ahora en el Usuarios GET obtengo la lista de usuarios en la tabla



Los datos coinciden con la tabla vista desde MYSQL