

Armando José Salgado Rosa

Ingeniero Roberto Salguero

Arquitectura y Diseño de Software

Miércoles, 28 de enero de 2026

Laboratorio 1: Resultados de la práctica

Pruebas con Pytest

Ilustración 1

Ejecución de las pruebas desde consola en Visual Studio Code

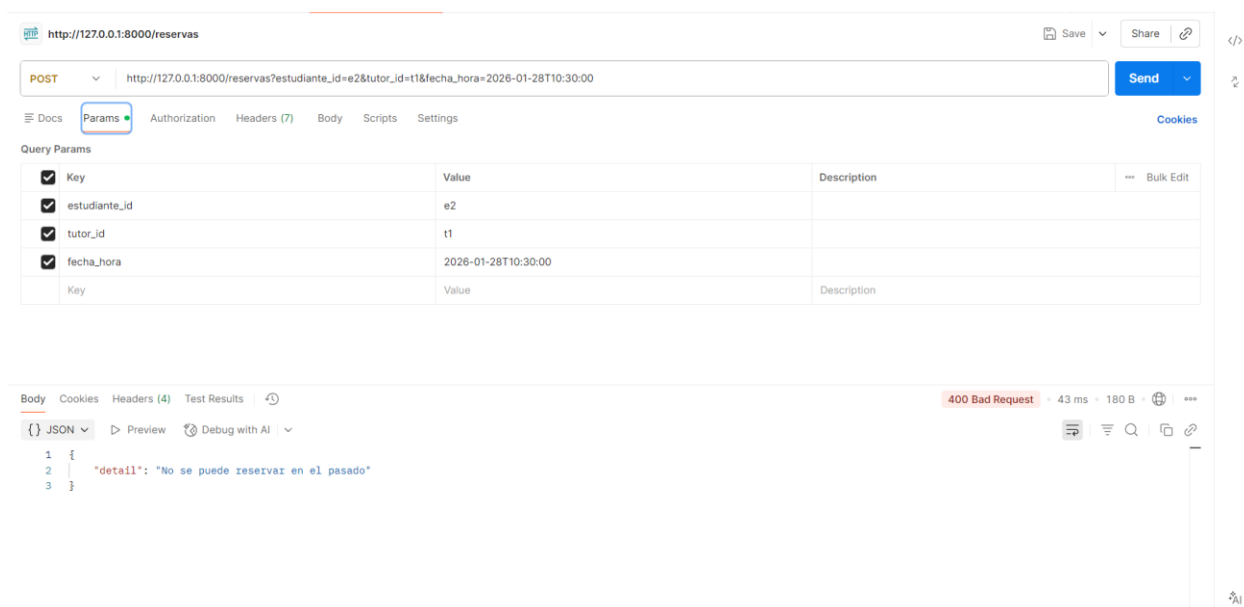
```
PS C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1> .venv\Scripts\activate
(.venv) PS C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1> cd tutorias
(.venv) PS C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1\tutorias> $env:PYTHONPATH="."
(.venv) PS C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1\tutorias> pytest -q
..
2 passed in 0.03s
(.venv) PS C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1\tutorias>
```

Fuente. Elaboración propia.

Prueba ejemplo con un request

Ilustración 2

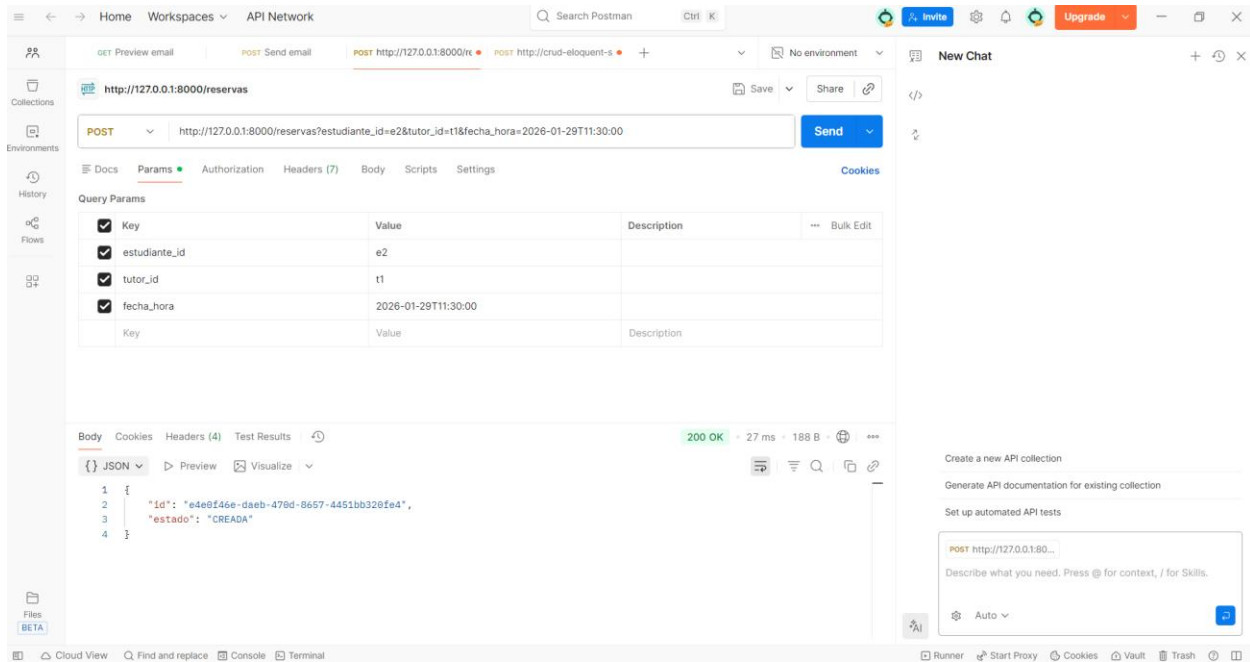
Prueba en Postman – caso reserva fallida por fecha pasada



Fuente. Elaboración propia en la herramienta de Postman.

Ilustración 3

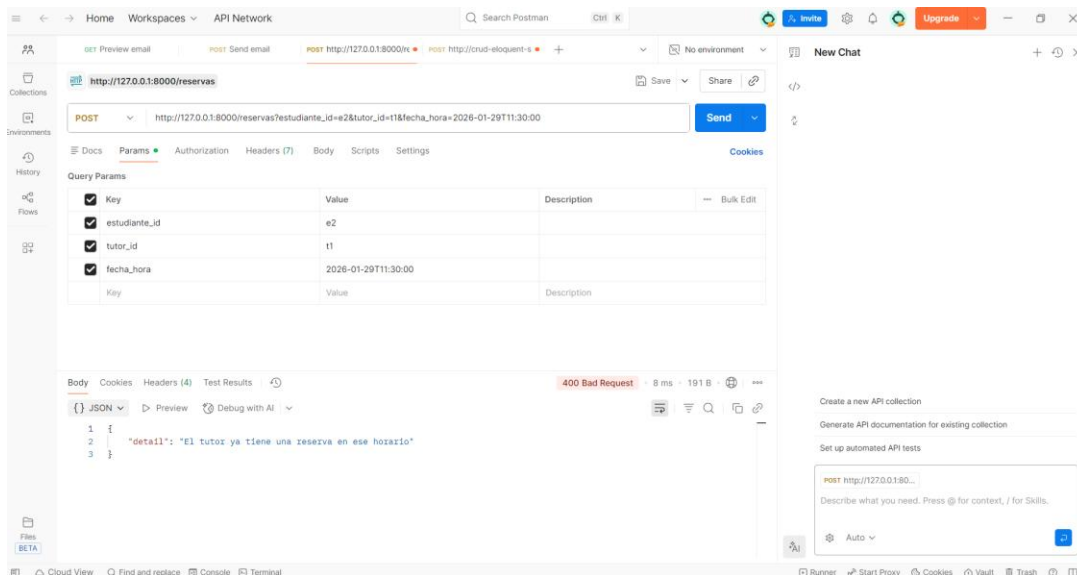
Prueba en Postman – caso exitoso de creación de reserva



Fuente. Elaboración propia en la herramienta de Postman.

Ilustración 4

Prueba en Postman – caso fallido porque ya existe una tutoría en ese horario



Fuente. Elaboración propia con la herramienta de Postman.

Ilustración 5

Levantamiento del servidor y resultados en consola a las request

The screenshot shows a VS Code editor with a Python FastAPI application. The file explorer on the right shows the project structure, including folders like 'tutorias', 'app', 'api', 'domain', 'infrastructure', and 'models'. The main editor displays the 'main.py' file, which defines a FastAPI application with a single endpoint for creating reservations. The terminal at the bottom shows the output of running the application using uvicorn, including startup logs and two HTTP requests: a successful 200 OK response and a 400 Bad Request response.

```

1 from fastapi import FastAPI, HTTPException
2 from datetime import datetime
3 from app.application.services import ReservaService
4 from app.infrastructure.repositories import ReservaRepoMemoria
5 from app.domain.errors import ReglaNegocioError
6
7 app = FastAPI(title="Tutorías API")
8
9 repo = ReservaRepoMemoria()
10 service = ReservaService(repo)
11
12 @app.post("/reservas")
13 def crear_reserva(estudiante_id:str, tutor_id:str, fecha_hora:str):
14     try:
15         dt = datetime.fromisoformat(fecha_hora)
16         r = service.crear_reserva(estudiante_id, tutor_id, dt)
17         return {"id":r.id, "estado":r.estado}
18     except ReglaNegocioError as e:
19         raise HTTPException(status_code=400, detail=str(e))

```

```

PS C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1\tutorias> uvicorn app.api.main:app --reload
INFO: Will watch for changes in these directories: ['C:\Docs\ESEN 2026\Arquitectura\Laboratorios\Laboratorio1\tutorias']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [26468] using WatchFiles
INFO: Started server process [21572]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:64528 - "POST /reservas?estudiante_id=e2&tutor_id=t1&fecha_hora=2026-01-29T11:30:00 HTTP/1.1" 200 OK
INFO: 127.0.0.1:49243 - "POST /reservas?estudiante_id=e2&tutor_id=t1&fecha_hora=2026-01-29T11:30:00 HTTP/1.1" 400 Bad Request

```

Fuente. Elaboración propia.