

# **INFORME TÉCNICO FINAL**

Polideportivo Torrelodones

Informe de Pruebas Automatizadas

Versión: 1.0

Fecha: Diciembre 2025

Autor: Alejandro Diaz Bravo

## Índice

- 1. Introducción
- 2. Alcance y Objetivos
- 3. Herramientas y Entorno
- 4. Estrategia y Diseño de Pruebas
- 5. Ejecución y Resultados
- 6. Estado E2E y Recomendaciones
- 7. Archivos de tests
- 8. Conversaciones con la IA (espacio)
- 9. Salida de ejecución de los tests (espacio)
- 10. Anexos

## 1. Introducción

Este documento detalla la generación, ejecución y resultados de las pruebas automáticas desarrolladas para la aplicación "Polideportivo Torreloones". Incluye tests unitarios, de rutas y un test E2E diseñado para Playwright.

## 2. Alcance y Objetivos

Objetivo: validar las funcionalidades críticas: creación de reservas, validación de solapamientos, cancelación mediante token y APIs relacionadas (/api/crear\_reserva, /api/mis\_reservas, /api/cancel\_reserva).

## 3. Herramientas y Entorno

- Lenguaje: Python 3.x
- Web: Flask
- BD: SQLite
- Test: pytest (unitarios y rutas)
- E2E: Playwright (test incluido, requiere instalación de Playwright y navegadores)

## 4. Estrategia y Diseño de Pruebas

Se han generado tests mediante interacción con un asistente de IA. Tipos de pruebas:

- Unitarias: lógica de negocio y funciones de BD
- Integración ligera: endpoints y APIs con Flask
- E2E (opcional): flujo reserva + cancelación en navegador con Playwright

## 5. Ejecución y Resultados

A continuación se muestra el resultado de ejecutar la batería de pruebas con `pytest`

```
PS C:\Users\adbra\OneDrive\Documentos\MASTER DEVOPS\PRIMER AÑO\PSU IA 2\PolideportivoTorreloones> python -m pytest -q
..... [100%]
7 passed, 1 skipped in 0.93s
```

## 6. Estado E2E y Recomendaciones

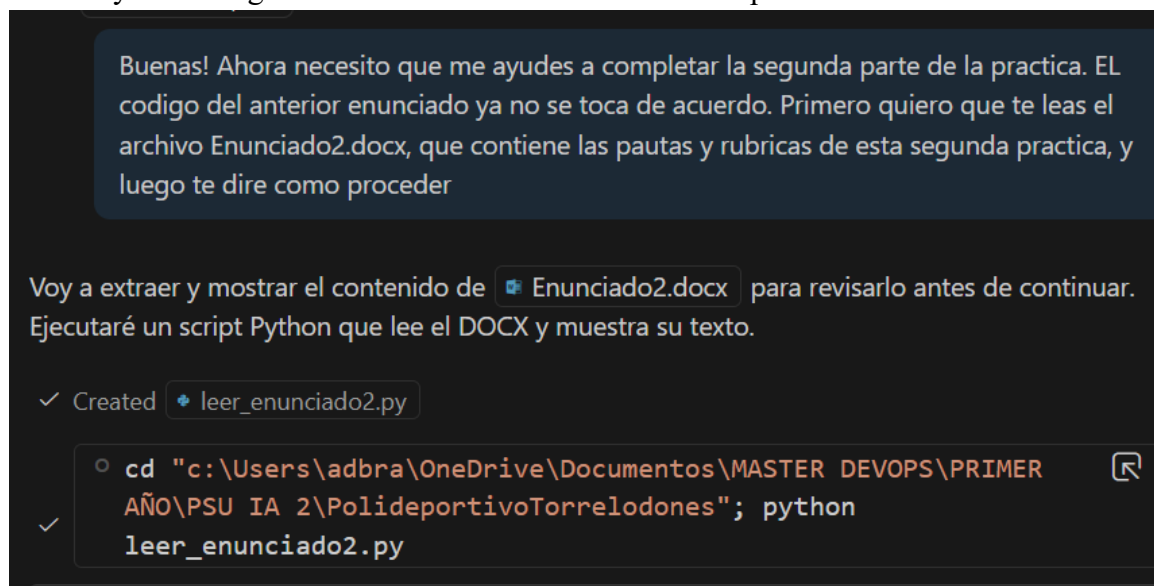
El test E2E `tests/test\_e2e\_playwright.py` está presente. Está configurado para `skip` cuando Playwright no está instalado en el entorno de ejecución.

## 7. Archivos de tests

- tests/test\_database.py
- tests/test\_app.py
- tests/test\_e2e\_playwright.py

## 8. Conversaciones relevantes con la IA

Se incluyen los fragmentos más relevantes con Github Copilot:



Quiero que generes tests que cubran todo mi código de aplicación, luego prepares el entorno y generes y corras los scripts de los tests por favor

✓ Planned test implementation and created test files

✓  
◦ `cd "c:\Users\adbra\OneDrive\Documentos\MASTER DEVOPS\PRIMER AÑO\PSU IA 2\PolideportivoTorrelodones"; python -m pip install -r requirements-tests.txt`

① Enable [shell integration](#) to improve command detection

✓  
◦ `python -m pytest -q`

① Enable [shell integration](#) to improve command detection

Quiero pruebas de UI por favor, generalas y ejecutalas

✓ Planned and created end-to-end UI tests

✓ Edited `requirements-tests.txt` +1 -0

✓  
◦ `cd "c:\Users\adbra\OneDrive\Documentos\MASTER DEVOPS\PRIMER AÑO\PSU IA 2\PolideportivoTorrelodones"; python -m pip install -r requirements-tests.txt`

① Enable [shell integration](#) to improve command detection

✓  
◦ `python - << 'PY'`  
`try:`  
`import selenium`  
`print('selenium OK', selenium.__version__)`