Cristina Margaret Calvario Serrano

Módulo 7. Automatización de pruebas con IA.

Caso práctico 7. Crear las pruebas unitarias para todo los servicios REST creados en el backend

Texto

El contenido generado por IA puede ser incorrecto.

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

* pytest: El framework de testing que estamos usando
* app: La instancia de tu aplicación Flask (probablemente definida en app.py)
* db: El objeto de SQLAlchemy para interactuar con la base de datos
* User: El modelo de datos para usuarios (definido en models.py)

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Un fixture en pytest es una función que provee datos, configuraciones o recursos para los tests

* app.config['TESTING'] = True: Habilita el modo testing de Flask (desactiva algunos features para mejor testing)
* app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///:memory:': Configura una base de datos SQLite en memoria (aislada y temporal)
* app.test\_client(): Crea un cliente de pruebas para simular peticiones HTTP
* db.create\_all(): Crea todas las tablas en la base de datos temporal
* yield client: Proporciona el cliente a los tests que usen este fixture
* db.drop\_all(): Limpia la base de datos después de cada test (se ejecuta después del yield)

Este test verifica que la ruta POST '/users' funciona correctamente:

* Simula una petición HTTP POST al endpoint '/users' con datos JSON de un usuario
* Verifica:
* Que el código de estado HTTP es 201 (CREATED)
* Que el mensaje en la respuesta JSON es el esperado

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Cuando las pruebas pasan correctamente, verás algo como:

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.