## **Pruebas de Bases de Datos**

Las pruebas de bases de datos verifican que la interacción de tu aplicación con la base de datos sea correcta. Esto incluye insertar, actualizar, eliminar y consultar datos.

**Herramientas:**

* **unittest:** El módulo estándar de Python.
* **pytest:** Un marco de pruebas más flexible y potente.
* **SQLAlchemy:** ORM popular para interactuar con bases de datos relacionales.

**Ejemplo con SQLAlchemy y pytest:**

Python

import pytest from sqlalchemy import create\_engine from sqlalchemy.orm import sessionmaker from .models import User # Suponiendo que tienes un modelo de usuario@pytest.fixture(scope="module")def db\_session(): engine = create\_engine("sqlite:///:memory:") Session = sessionmaker(bind=engine) session = Session() yield session session.close() def test\_create\_user(db\_session): user = User(name="Alice", email="alice@example.com") db\_session.add(user) db\_session.commit() user\_from\_db = db\_session.query(User).filter\_by(email="alice@example.com").first() assert user\_from\_db is not None assert user\_from\_db.name == "Alice"

Use code [with caution.](file:///C:/faq#coding)

**Puntos clave:**

* **Fixture:** Un fixture como db\_session crea una sesión de base de datos para cada prueba y la cierra al final.
* **Verificación:** Se verifica que el usuario se haya creado correctamente en la base de datos.