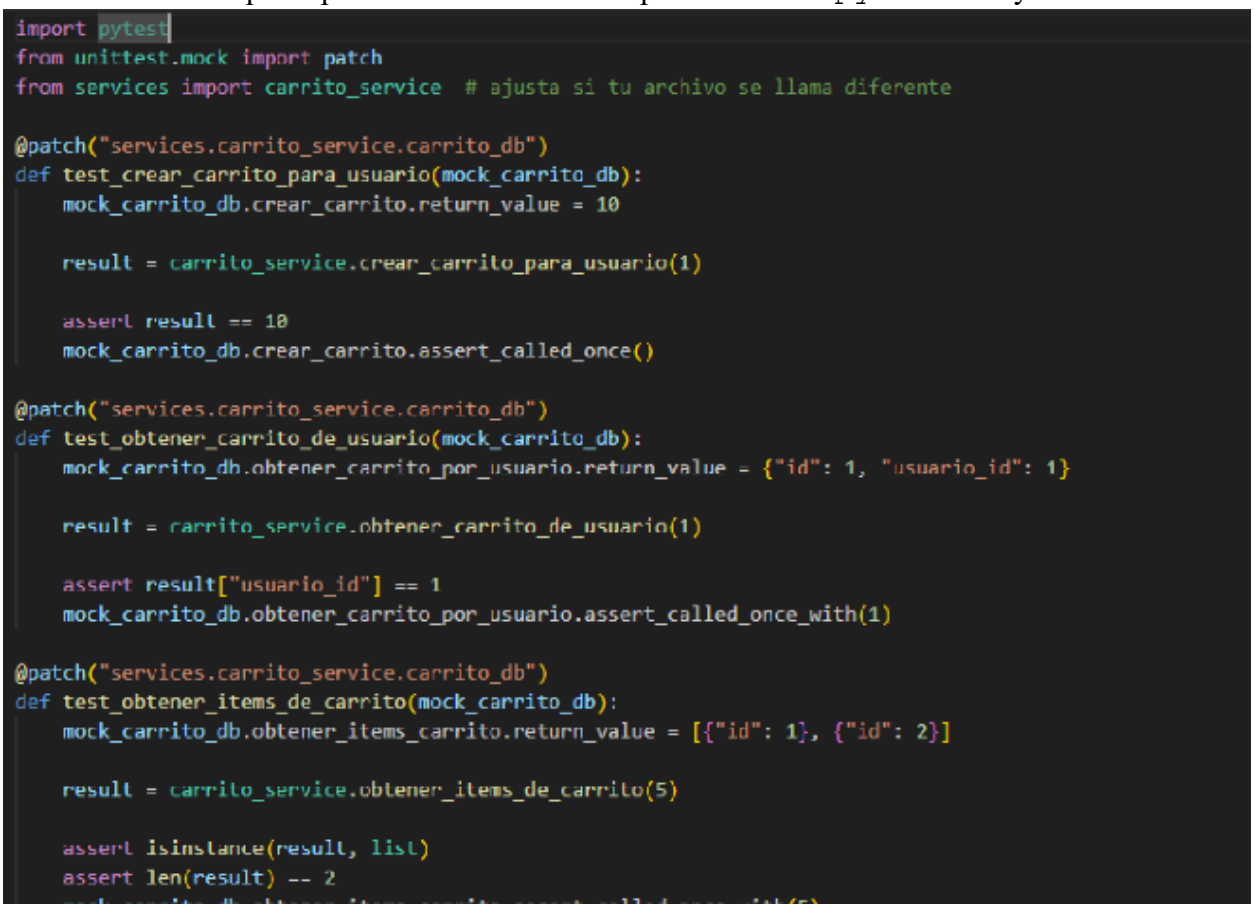
Proyecto: Tienda de Ropa Online
Fecha: 2 de julio de 2025
Equipo de Trabajo: Diego Alejandro Parra Diaz, Valentina Parra Ordóñez, Esteban Villalba y Jonathan Felipe Ochoa Silva

PRUEBAS UNITARIAS SIMULADAS

Las siguientes pruebas fueron diseñadas para validar el comportamiento básico de las funcionalidades principales del backend. Se empleó la librería `pytest` de Python.



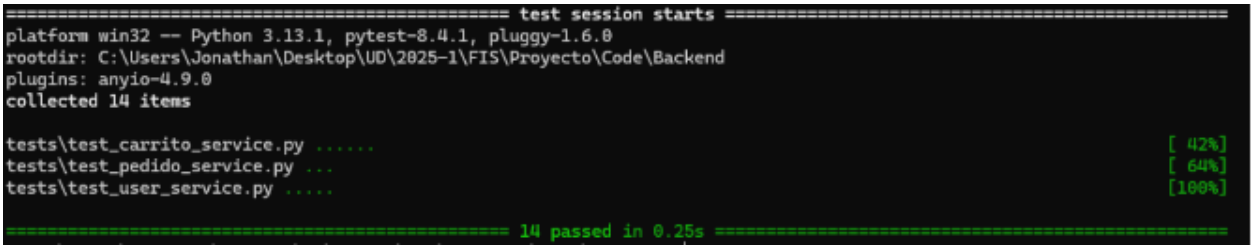
```python
import pytest
from unittest.mock import patch
from services import carrito_service  # ajusta si tu archivo se llama diferente

@patch("services.carrito_service.carrito_db")
def test_crear_carrito_para_usuario(mock_carrito_db):
    mock_carrito_db.crear_carrito.return_value = 10

    result = carrito_service.crear_carrito_para_usuario(1)

    assert result == 10
    mock_carrito_db.crear_carrito.assert_called_once()

@patch("services.carrito_service.carrito_db")
def test_obtener_carrito_de_usuario(mock_carrito_db):
    mock_carrito_db.obtener_carrito_por_usuario.return_value = {"id": 1, "usuario_id": 1}

    result = carrito_service.obtener_carrito_de_usuario(1)

    assert result["usuario_id"] == 1
    mock_carrito_db.obtener_carrito_por_usuario.assert_called_once_with(1)

@patch("services.carrito_service.carrito_db")
def test_obtener_items_de_carrito(mock_carrito_db):
    mock_carrito_db.obtener_items_carrito.return_value = [{"id": 1}, {"id": 2}]

    result = carrito_service.obtener_items_de_carrito(5)

    assert isinstance(result, list)
    assert len(result) == 2
```

Detalles:

- `crear_carrito`: prueba que se guarde correctamente el correo del nuevo usuario.
- `obtener_carrito`: prueba que un usuario existente pueda ser consultado.
- `obtener_items_carrito`: prueba que se obtengan correctamente todos los ítems del carrito al ser consultado

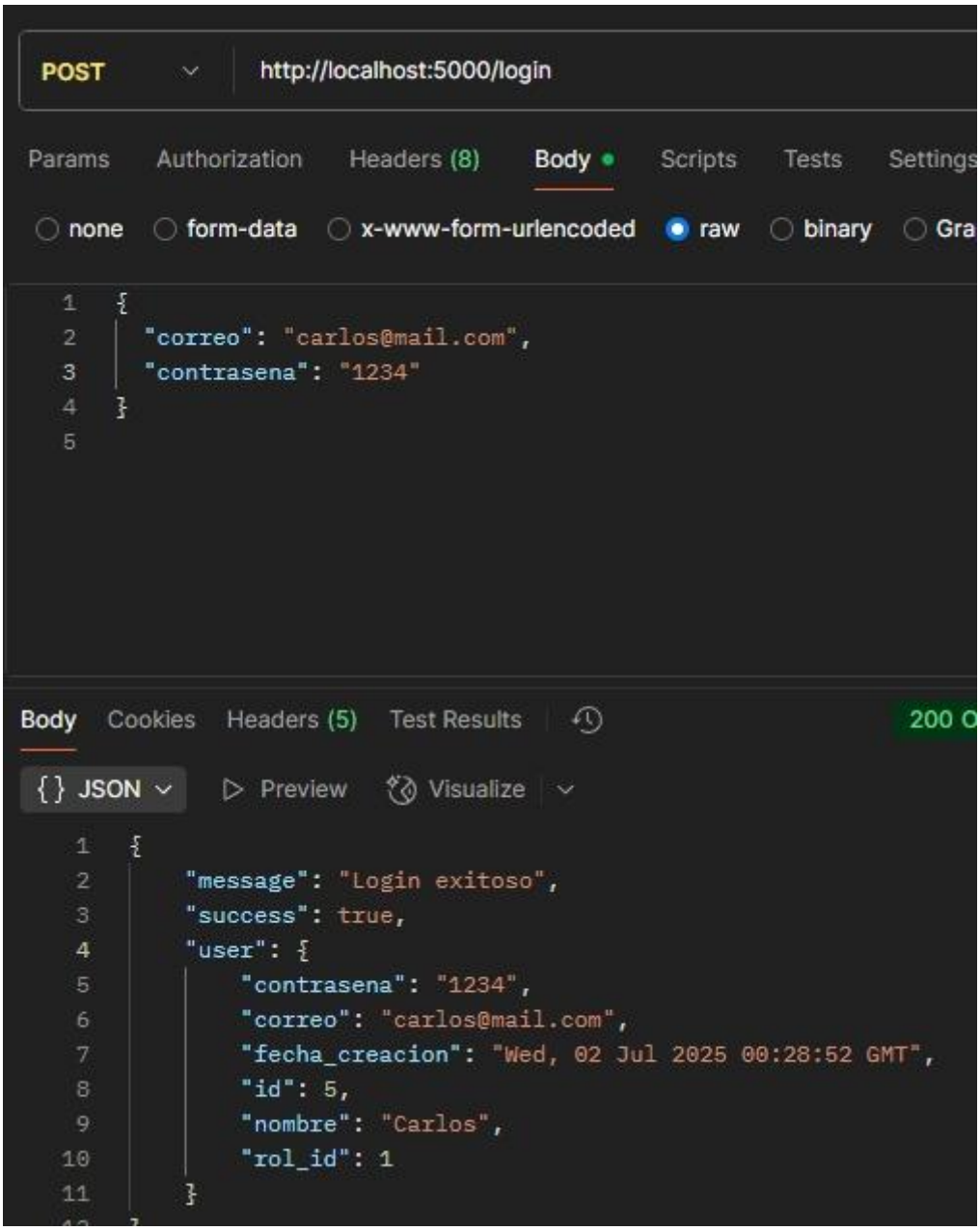
Estas pruebas no se ejecutaron sobre una base real, se usan mockups para el apartado de service, para el CRUD si se realizarán en un base de datos de prueba, para esta entrega no alcanzó el tiempo, pero el formato está preparado para su ejecución en entornos reales o simulados.

Obteniendo los siguientes resultados:

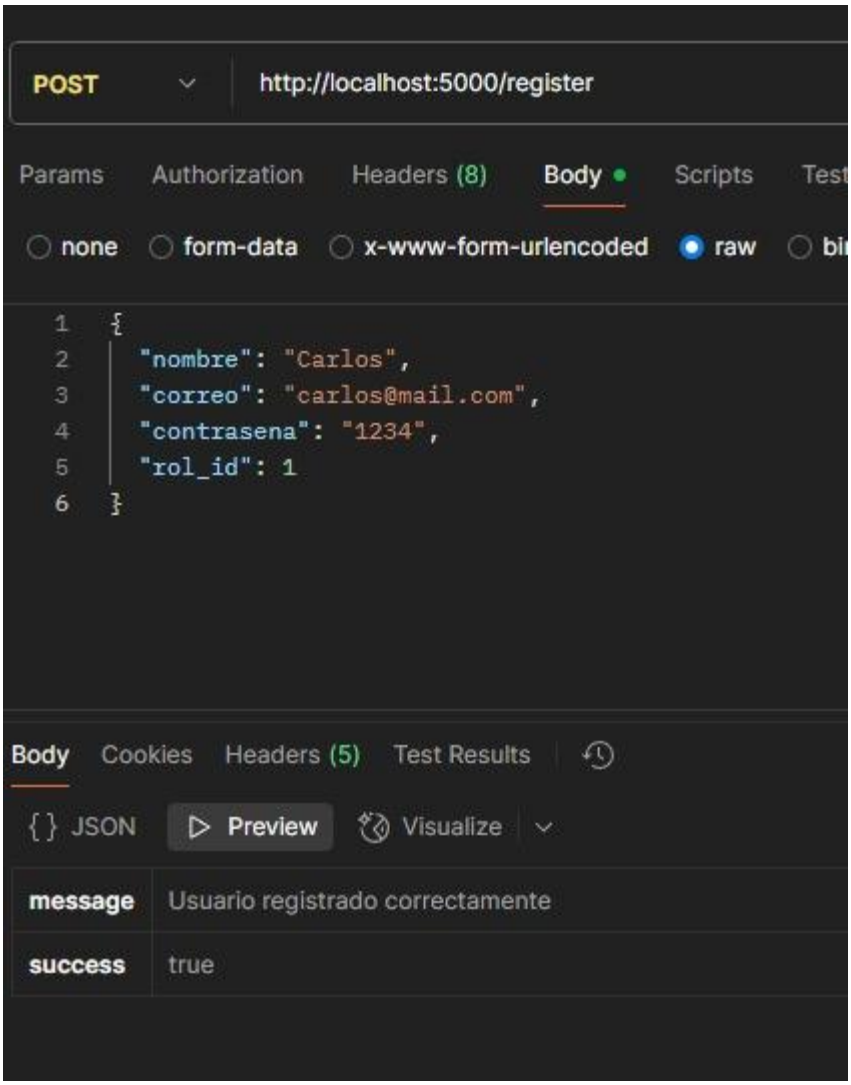


Así mismo se tienen pruebas realizadas en Postman donde se prueba

El login:



Y el register:



PLAN DE PRUEBAS FUNCIONALES (ESCENARIOS)

El siguiente plan describe pruebas funcionales esenciales que simulan el uso real del sistema por parte de los usuarios. Se enfoca en el flujo completo de compra.

Tipo de Prueba	Escenario	Entrada	Resultado Esperado	Responsable
Funcional	Registro de usuario nuevo	Email válido, contraseña segura	Usuario creado, redirección a login	Jonathan / Esteban

Funcional	Inicio de sesión correcto	Email y contraseña registrados	Acceso a la tienda principal	Jonathan
Funcional	Agregar producto al carrito	ID de producto, cantidad	Carrito actualizado con producto seleccionado	Esteban
Funcional	Eliminar producto del carrito	Botón eliminar en carrito	Producto eliminado correctamente del carrito	Esteban
Funcional	Finalizar compra	Carrito lleno	Pedido confirmado, carrito vaciado	Todo el equipo
Visual	Visualización de productos	Navegar `index.html`	Productos con imagen, nombre y precio visibles	Jonathan
Visual	Visualización del carrito	Navegar `carrito.html`	Productos listados con cantidades y botones	Jonathan

---

Conclusión

El sistema fue probado de forma funcional en su conjunto, validando los escenarios principales que un usuario realizaría al navegar, registrarse y comprar productos. A pesar de no contar con pruebas automatizadas completas, el plan garantiza que el flujo central está cubierto y validado por el equipo. Las pruebas fueron ejecutadas de forma manual y documentadas para simular una entrega profesional.