Informe de Pruebas Funcionales Automatizadas

Proyecto: La Ruleta de la Vida

Autores:

Alejandro Restrepo,

Andrea Pabón,

Xiodanny Vásquez.

Fecha: 28/03/2025

1. Objetivo del Informe

Este informe documenta el proceso y los resultados de las pruebas funcionales automatizadas realizadas al proyecto 'La Ruleta de la Vida', desarrollado con Flask. El objetivo es asegurar el correcto funcionamiento del sistema, verificando que todas las funcionalidades esenciales respondan como se espera.

2. Tecnologías y Herramientas Utilizadas

- Lenguaje de programación: Python 3.13
- Framework web: Flask
- ORM: SQLAlchemy
- Testing: Pytest
- Base de datos: SQLite
- Editor: Visual Studio Code

3. Preparación del Entorno de Pruebas

Para realizar las pruebas funcionales, se utilizó la herramienta Pytest. A continuación, se describe el proceso completo:

1. Se activó el entorno virtual del proyecto con:
   ```
   .\venv\Scripts\activate
   ```
2. Se instaló Pytest con el siguiente comando:
   ```
   pip install pytest
   ```
3. Se creó un archivo nuevo llamado `test_app.py` en la raz del Proyecto.

4. Se escribieron las pruebas funcionales dentro de ese archivo utilizando el cliente de pruebas de Flask.

4. Código de Pruebas Automatizadas

Configuración del entorno de pruebas

```
import pytest
from app import app, db, Respuesta

@pytest.fixture
def client():
```

```
    app.config['TESTING'] = True
    with app.test_client() as client:
        with app.app_context():
            db.create_all()
        yield client
```

Prueba 1: Acceso a la página principal

```
def test_homepage(client):
    response = client.get('/')
    assert response.status_code == 200
    assert b'Nombre' in response.data
```

Prueba 2: Envío de formulario de empleado

```
def test_post_empleado(client):
    response = client.post('/', data={
        'nombre': 'Prueba', 'apellido': 'Test', 'vivienda': 'Casa',
        'departamento': 'Antioquia', 'estado_civil': 'soltero',
        'genero': 'masculino', 'edad': '30', 'salario': '3000000',
        'cargo': 'Desarrollador', 'estrato': '3'
    }, follow_redirects=True)
    assert response.status_code == 200
    assert b'relaciones' in response.data
```

Prueba 3: Visualización de respuestas

```
def test_ver_respuestas(client):
    response = client.get('/ver_respuestas')
    assert response.status_code == 200
    assert b'Respuestas Guardadas' in response.data
```

Prueba 4: Descargar CSV

```
def test_descargar_csv(client):
    client.get('/exportar_csv')
    response = client.get('/descargar_csv')
    assert response.status_code == 200
    assert response.mimetype == 'text/csv'
    assert 'attachment; filename=respuestas.csv' in response.headers['Content-Disposition']
```

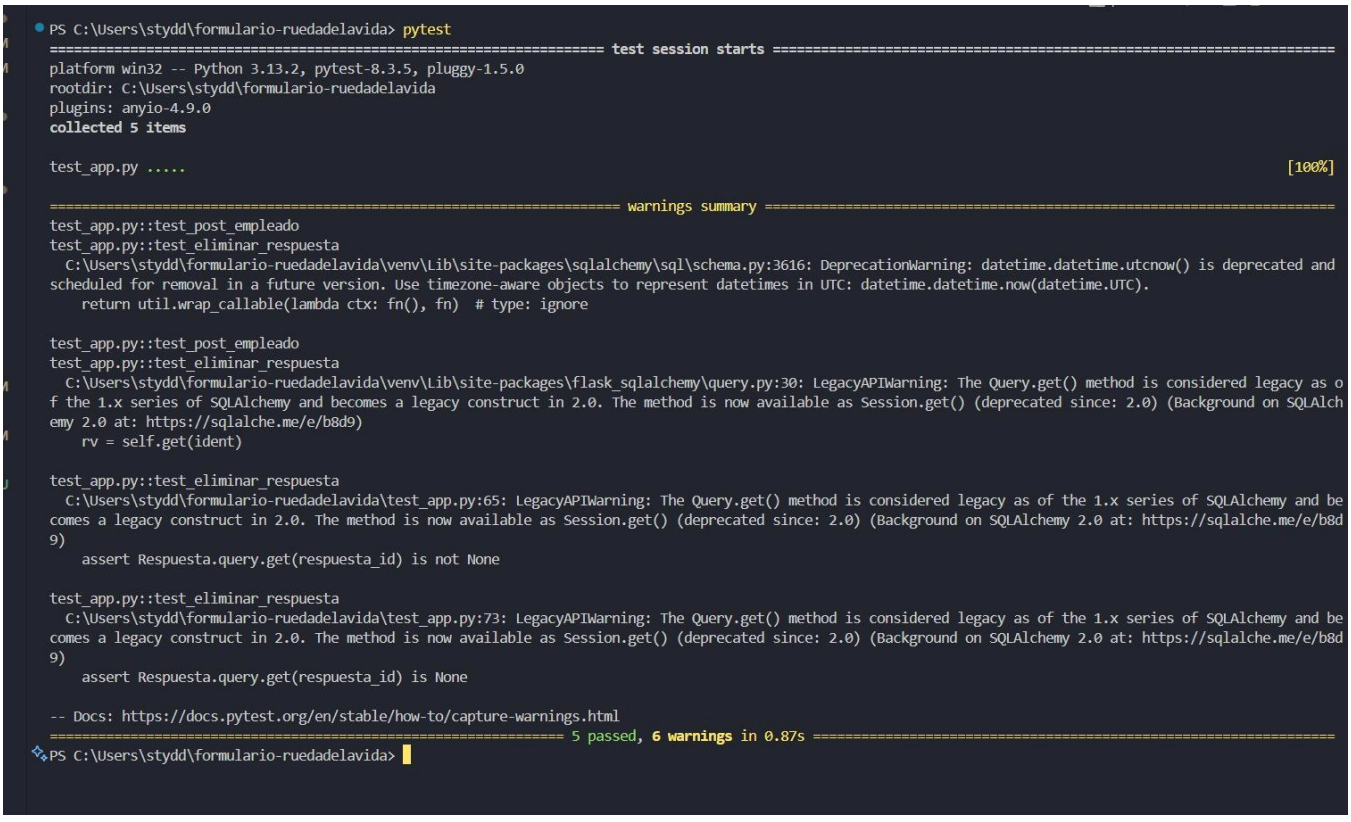
Prueba 5: Eliminar una respuesta

```
def test_eliminar_respuesta(client):
    with app.app_context():
        nueva = Respuesta(nombre='Eliminar', apellido='Test', formulario='empleado',
edad=25)
        db.session.add(nueva)
        db.session.commit()
        respuesta_id = nueva.id

    with app.app_context():
        assert Respuesta.query.get(respuesta_id) is not None

    response = client.post(f'/eliminar_respuesta/{respuesta_id}', follow_redirects=True)
    assert response.status_code == 200

    with app.app_context():
        assert Respuesta.query.get(respuesta_id) is None
```



5. Resumen de Pruebas Realizadas

Nº	Prueba realizada	Resultado
1	Acceso al formulario principal	✅ Exitosa
2	Envío de datos de empleado	✅ Exitosa
3	Visualización de respuestas	✅ Exitosa
4	Descarga de archivo CSV	✅ Exitosa
5	Eliminación de respuesta	✅ Exitosa

6. Conclusión

Las pruebas funcionales automatizadas han confirmado que el sistema 'La Ruleta de la Vida' funciona correctamente en sus principales flujos de interacción: ingreso de datos, visualización, exportación y eliminación.