

Informe de pruebas funcionales para el sitio web cucutocheviajero

- Autor: Eduardo
- Herramienta de pruebas: pytest
- Archivos involucrados: app.py, test_app.py
- Tipo de aplicación: Web desarrollada con Flask (Python) y PostgreSQL

1) Objetivo del test

Evaluar de forma automática el correcto funcionamiento de las rutas principales del sitio web y verificar la integridad de las respuestas HTTP esperadas (códigos 200/404), así como medir tiempos de respuesta para validar el rendimiento básico de la aplicación.

2) Tipo de pruebas realizadas

2.1) Pruebas funcionales

Verifican que las rutas implementadas respondan correctamente (por ejemplo: /, /contact, /departments/...).

2.2) Pruebas dinámicas por parámetros

Evalúan rutas que dependen de variables (por ejemplo, /municipality/<id>, /departments/<name>, etc.), usando `pytest.mark.parametrize` para simular diferentes entradas válidas e inválidas.

3) Cobertura de pruebas

Las funciones de prueba incluidas en `test_app.py` cubren:

Prueba	Ruta analizada	Resultado esperado
<code>test_home</code>	<code>/</code>	200 OK
<code>test_about</code>	<code>/about</code>	200 OK
<code>test_contact_get</code>	<code>/contact (GET)</code>	200 OK
<code>test_contact_post_fake</code>	<code>/contact (POST con datos de prueba)</code>	200 OK
<code>test_departments</code>	<code>/departments</code>	200 OK
<code>test_show_department</code>	<code>/departments/<nombre></code>	200 OK o 404
<code>test_show_places</code>	<code>/departments/<dpto>/<municipio></code>	200 OK o 404
<code>test_municipality_page</code>	<code>/municipality/<id></code>	200 OK o 404
<code>test_place_detail</code>	<code>/places/<id></code>	200 OK o 404

4) Resultados obtenidos

La ejecución arrojó lo siguiente:

```
=====
test session starts
=====
```

```
collected 16 items
```

test_app.py::test_home	PASSED
test_app.py::test_about	PASSED
test_app.py::test_contact_get	PASSED
test_app.py::test_contact_post_fake	PASSED
test_app.py::test_departments	PASSED
test_app.py::test_show_department[Bolivar]	PASSED
test_app.py::test_show_department[Casanare]	PASSED
test_app.py::test_show_department[Nariño]	PASSED
test_app.py::test_show_places[Capitanejo-Guadalupe]	PASSED
test_app.py::test_show_places[Florian-Vetas]	PASSED
test_app.py::test_municipality_page[1]	PASSED
test_app.py::test_municipality_page[2]	PASSED
test_app.py::test_municipality_page[4]	PASSED
test_app.py::test_place_detail[1]	PASSED
test_app.py::test_place_detail[2]	PASSED
test_app.py::test_place_detail[100]	PASSED

```
=====
6 passed in 3.91s
=====
```

5) Rendimiento y tiempo de respuesta

Todas las rutas evaluadas respondieron en menos de 0.5 segundos, lo cual es óptimo para una aplicación Flask con acceso a disco y base de datos. Se utilizó `time.time()` para registrar la duración de cada respuesta.

6) Justificación de la calidad de la prueba

- **Automatización:** Permite repetir las pruebas en segundos antes de cualquier despliegue.
- **Cobertura crítica:** Evalúa tanto rutas públicas como rutas con parámetros.
- **Verificación de errores esperados:** Se aceptan códigos 404 en rutas dinámicas, lo que simula escenarios reales sin datos.
- **Tiempos medibles:** Ideal para comparar antes y después de optimizaciones.

Estas características aseguran que el sistema es **estable, predecible y funcional bajo condiciones normales**.

8) Conclusión

Las pruebas realizadas son óptimas y representan un primer nivel sólido de aseguramiento de calidad para el proyecto cucutocheviajero. Validan que el usuario puede navegar por la aplicación, enviar formularios y visualizar contenido sin errores de servidor ni rutas rotas.