

Validación de la Conectividad Básica del Backend

Danna Yisel Segura Contreras, Erick Sebastián Ladino

23 de septiembre de 2025

1 Enfoque

Verificación de la operatividad del backend para el registro de nuevos usuarios y la correcta inserción de datos en la base de datos.

2 Casos de Uso: Módulo de Prueba de Humo

2.1 Descripción

Se ha desarrollado un script de prueba de humo para garantizar que el componente crítico del backend (la API Flask) funcione correctamente después de cada despliegue. Esta prueba se enfoca en la funcionalidad de registro de usuarios.

Este módulo de prueba, denominado `test_registro_usuario_exitoso()`, está diseñado para validar la conectividad del backend a través del endpoint `/registro`. Su propósito es confirmar que la API puede recibir una solicitud de tipo POST, procesar los datos y registrar un nuevo artista en la base de datos, devolviendo una respuesta exitosa.

3 Módulo de Pruebas: Registro de Usuarios (Endpoint `/registro`)

3.1 Descripción General

Este módulo se centra en el endpoint `/registro`, que no requiere autenticación. La prueba verifica que el servidor responde correctamente a las peticiones POST y que los datos se procesan e insertan de manera exitosa. Esto constituye una validación fundamental del estado del servicio.

3.2 Funcionalidad de la Prueba (`test_registro_usuario_exitoso`)

- **Acción:** La prueba realiza una solicitud HTTP POST al endpoint `/registro` con un payload (cuerpo de datos) JSON que contiene la información de un nuevo usuario

(nombre, contacto, documento, contraseña). Esta acción simula la forma en que un cliente de la aplicación registraría una nueva cuenta.

- **Validación:**

1. **Código de Estado (201 Created):** La primera validación verifica que el código de estado de la respuesta sea 201. Un código 201 indica que el servidor ha creado el recurso de manera exitosa. Si la respuesta fuera un 500 Internal Server Error, la prueba fallaría, indicando un problema con el servicio de base de datos o la lógica de registro.
2. **Formato de los Datos:** La segunda validación asegura que la respuesta JSON del servidor contenga los campos esperados: “success”: True y “message”: “Artista registrado con éxito”. Esto confirma que el backend no solo está activo, sino que también puede procesar y validar la información de entrada, interactuar con la base de datos y retornar una respuesta en un formato válido.

3.3 Ejecución y Resultado Esperado

- **Comando:** Para ejecutar solo esta prueba, puedes usar el siguiente comando en la terminal (asumiendo que estás usando pytest):

```
pytest <nombre_del_archivo.py>::test_registro_usuario_exitoso
```

- **Resultado Esperado:** Si la prueba es exitosa, verás un resultado similar a este:

```
===== test session starts =====
collected 1 item

backend\test_api_aona.py .
[100 %]

===== 1 passed in 0.20s =====
```

El punto (.) indica que la prueba pasó sin errores. Si fallara, verías una F junto con un informe de error detallado.