

Guía de Aprendizaje: Pruebas de Software

Módulos: Crear, Actualizar, Eliminar Canciones

Desarrollado por:

Jeiffer Harley Mosquera Abadía
Yeraldin Olaya Ríos

Septiembre 2025

Instructor: Jonathan Espitia

SENA – Centro de Gestión de Mercados, Logística y TIC (CGMLTI)

Análisis y Desarrollo de Software (ADSO)

Ficha: 3147246 – Bogotá D.C.

Nombre del Módulo: Gestión de Canciones en la plataforma WaveSound

Desarrollador: Jeiffer Harley Mosquera

Cliente: Yeraldin Olaya

Enfoque: Se debe implementar el módulo de **crear, actualizar y eliminar canciones**, con validaciones y almacenamiento correcto en la base de datos mediante APIs de FastAPI.

Requisitos Funcionales

Crear Canción

- RF-01: El sistema debe permitir ingresar el **título** y **descripción** de la canción (campos obligatorios).
- RF-02: Se debe validar que la **duración** sea un número entero positivo.
- RF-03: URLs de **archivo** y **portada** deben ser válidas y obligatorias.
- RF-04: Debe seleccionar **género** y **álbum** de listas predefinidas.
- RF-05: La API debe devolver un objeto con `id_cancion` y `fecha_creacion`.

Actualizar Canción

- RF-06: Se permite modificar cualquier campo de una canción existente.
- RF-07: Se debe verificar que el `id_cancion` exista.
- RF-08: La API debe devolver la canción actualizada correctamente.

Eliminar Canción

- RF-09: Permite eliminar una canción mediante su `id_cancion`.
- RF-10: Devuelve un estado de éxito o error si la canción no existe.

Casos de Uso / Historias de Usuario

HU01 – Crear canción:

Como **artista/productor**, quiero crear una canción ingresando todos los datos requeridos, para que quede registrada en la plataforma y disponible para reproducción.

HU02 – Actualizar canción:

Como **artista/productor**, quiero actualizar la información de una canción existente, para corregir datos o mejorar la información asociada.

HU03 – Eliminar canción:

Como **artista/productor**, quiero eliminar una canción, para retirar de la plataforma contenido que ya no deseo mantener.

Pruebas Realizadas

5.1 Pruebas Funcionales / Manuales:

- Se verificó que los formularios de creación, actualización y eliminación almacenaran los datos correctamente en la base de datos mediante las APIs.
- Se comprobó la validación de campos obligatorios y tipos de datos correctos.

5.2 Pruebas Automatizadas con Pytest (Python):

- Se configuró un **Test Client** para simular peticiones HTTP a la API de FastAPI.
- Se utilizaron **fixtures** para simular la base de datos y evitar tocar la base real.
- Se aplicó **monkeypatch** para reemplazar funciones reales por versiones simuladas y controlar la respuesta de la API.

Endpoints probados:

- `POST /canciones/` → Crear canción
- `PUT /canciones/{id_cancion}` → Actualizar canción
- `DELETE /canciones/{id_cancion}` → Eliminar canción

Nombre de las Pruebas:

Pruebas unitarias y de integración para los módulos de creación, actualización y eliminación de canciones.

Resultados

- Se validó que los datos enviados se almacenan correctamente en la base de datos simulada.
- Se verificó que la API devuelve los códigos HTTP correctos (200 OK para éxito, 404 o 400 en caso de error).
- Se asegura que las funciones de gestión de canciones funcionan de manera confiable según las historias de usuario.

Capturas de las pruebas:

```
MINGW64:/c/Users/USER/WaveSound/WAVESOUND_JY/backend

tests\test_actualizar_cancion.py . [ 33%]
tests\test_crear_cancion.py . [ 66%]
tests\test_eliminar_cancion.py . [100%]

===== warnings summary =====
mi_entorno\Lib\site-packages\pydantic\_internal\_config.py:323
  C:\Users\USER\WaveSound\WAVESOUND_JY\backend\mi_entorno\Lib\site-packages\pydantic\_internal\_config.py:323: PydanticDeprecatedSince20: Support for class-based `config` is deprecated, use ConfigDict instead. Deprecated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.11/migration/
    warnings.warn(DEPRECATION_MESSAGE, DeprecationWarning)

tests/test_crear_cancion.py::test_crear_cancion
  C:\Users\USER\WaveSound\WAVESOUND_JY\backend\tests\test_crear_cancion.py:22: PydanticDeprecatedSince20: The `dict` method is deprecated; use `model_dump` instead. Deprecated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.11/migration/
    return {**cancion.dict(), "id_cancion": 1, "fecha_creacion": "2025-09-24T17:00:00"}

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 3 passed, 2 warnings in 1.29s =====
```

Repositorio:

https://github.com/olaya19/Pruebas_Software