

**UNIVERSIDAD MAYOR REAL Y PONTIFICIA
DE SAN FRANCISCO XAVIER DE CHUQUISACA
FACULTAD DE TECNOLOGÍA**

DIPLOMADO EN TESTING FUNCIONAL DE SOFTWARE



Módulo 4: Pruebas API

Testing WEB API Spotify

Integrantes: Mamani Janco Ismael
Ortiz Mendez William
Ortega Vargas Yessica
Puyal Oliva Faviola

Grupo: 12

Docente: Ing. Marcelo Ferrufino Murillo

Sucre, 2025

1. Objetivo del proyecto

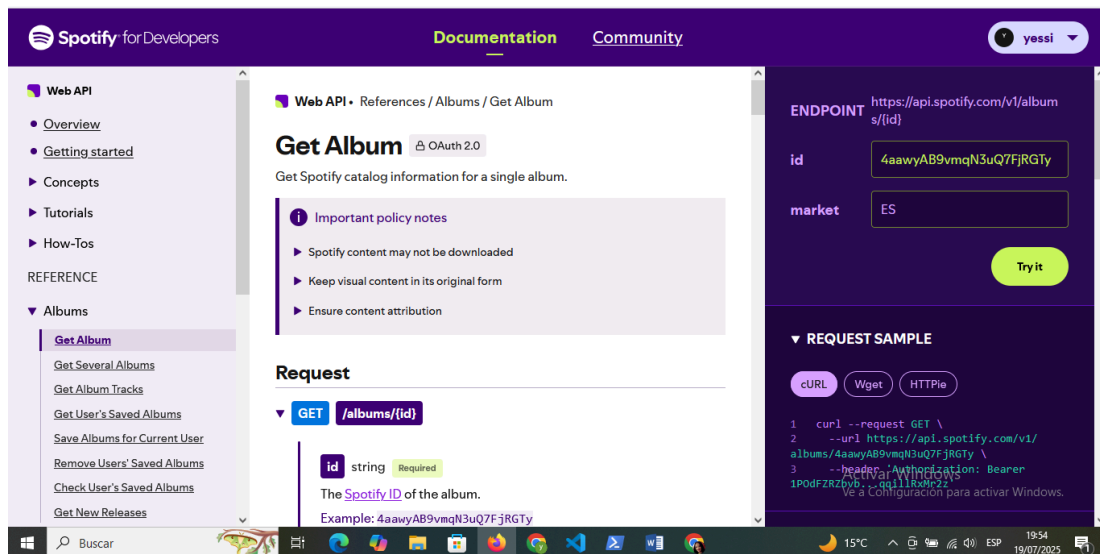
Aplicar los conocimientos adquiridos sobre pruebas APIs en este caso con la web api de spotify, como un análisis, diseño de pruebas exploratorias y automatización básica.

2. API a utilizada para el testing

<https://developer.spotify.com/documentation/web-api>.

3. Elaboración del Proyecto

3.1. Exploratory Testing



❖ Bug y comportamientos inesperados encontrados

BUG
<p>BUG-01 La API permite guardar Albums duplicadas</p> <p>Endpoint: PUT https://api.spotify.com/v1/me/tracks?</p> <p>Descripción: Si el mismo ID de álbumes se guarda varias veces, no hay advertencia ni prevención.</p> <p>Resultado esperado: Mensaje informativo o estado 409 Conflict</p> <p>Resultado actual: 200 OK sin cambios</p> <p>Severidad: Baja</p> <p>Prioridad: Baja</p> <p>Tipo: Lógica de negocio</p>
<p>BUG-02 – API permite eliminar Albums no guardadas</p> <p>Endpoint: DELETE https://api.spotify.com/v1/me/tracks?</p> <p>Descripción:</p> <p>Si el usuario elimina un álbum que no esta guardado, la API devuelve 200 OK igual, lo que puede provocar falsa sensación de éxito.</p> <p>Resultado esperado:</p>

- Código 404 o mensaje: "Track not found in library"

Resultado actual:

- Código 200 OK sin validar si la pista existía

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Media

Clasificación: Integridad de datos

BUG-03 – Api no permite guardar episodios

Endpoint: <https://api.spotify.com/v1/me/episodes>

Descripción:

El usuario desea guardar episodios y no se puede guardar muestra un mensaje de campos vacíos, aún introduciendo las ids correctas o incorrectas

Resultado esperado:

- Código 200 OK episodio guardado con éxito

Resultado actual:

- Muestra un mensaje de "respuesta vacía"

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Media

Clasificación: Integridad de datos

BUG-04 – Api no muestra los episodios guardados

Endpoint: <https://api.spotify.com/v1/me/episodes/contiene>

Descripción:

El usuario no puede verificar que episodios se guardaron muestra cómo guardados episodios que no fueron guardados

Resultado esperado:

Código 200, muestre el resultado como un true de episodio guardado

Resultado actual:

Código 200, muestra el resultado como un false de un episodio guardado y true de un episodio no guardado

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Media

Clasificación: Integridad de datos

BUG-05 La API permite guardar Pistas duplicadas

Endpoint: PUT <https://api.spotify.com/v1/me/tracks?>

Descripción: Si el mismo ID de pistas se guarda varias veces, no hay advertencia ni prevención.

Resultado esperado: Mensaje informativo o estado 409 Conflict

Resultado actual: 200 OK sin cambios

Severidad: Baja

Prioridad: Baja

Tipo: Lógica de negocio

BUG-06 – API permite eliminar canciones no guardadas**Endpoint:** DELETE <https://api.spotify.com/v1/me/tracks?>**Descripción:**

Si el usuario elimina una canción que no está guardada, la API devuelve 200 OK igual, lo que puede provocar falsa sensación de éxito.

Resultado esperado:

- Código 404 o mensaje: "Track not found in library"

Resultado actual:

- Código 200 OK sin validar si la pista existía

Tipo de prueba: Negativa**Severidad:** Media**Prioridad:** Media**Clasificación:** Integridad de datos

3.2. Casos de prueba como escenarios positivos y negativos de cada endpoint

SEARCH
<p>Caso de prueba 1 ID: SPOT-01 Título: Búsqueda exitosa de canción por nombre Precondiciones: Usuario autenticado con token válido Pasos a seguir:</p> <ol style="list-style-type: none">1. Realizar una petición GET al endpoint https://api.spotify.com/v1/search2. Enviar los parámetros q=Shape & type=track3. Incluir: Authorization: OAuth 2.0 <token>4. Presionar botón Send <p>Resultado esperado: Se devuelve un código 200 OK y un listado de canciones que contienen "Shape of You" en el nombre Tipo de prueba: Positiva Severidad: Baja Prioridad: Alta Clasificación: Smoke / Acceptance ¿Candidato a automatización?: Sí</p>
<p>Caso de prueba 2 ID: SPOT-02 Título: Búsqueda de artista específico Precondiciones: Usuario autenticado con token válido Pasos a seguir:</p> <ol style="list-style-type: none">1. Realizar una petición GET al endpoint https://api.spotify.com/v1/search2. Enviar los parámetros q=Beyonce & type=artist3. Incluir: Authorization: OAuth 2.0 <token>4. Presionar botón Send <p>Resultado esperado: Se devuelve código 200 OK y los datos del artista Shakira Tipo de prueba: Positiva Severidad: Baja Prioridad: Alta</p>

Clasificación: Smoke / Acceptance
¿Candidato a automatización?: Sí

Caso de prueba 3

ID: SPOT-03

Título: Solicitud sin parámetro q

Precondiciones: Usuario autenticado con token válido

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/search>
2. Enviar solo el parámetro type=track, sin incluir q
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Se devuelve un código 400 Bad Request con mensaje de error que indique la ausencia del parámetro obligatorio

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Alta

Clasificación: Regression

¿Candidato a automatización?: Sí

Caso de prueba 4

ID: SPOT-04

Título: Solicitud con token inválido

Precondiciones: Usuario con token caducado o erróneo

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/search>
2. Enviar parámetros q=Shakira&type=track
3. Usar en el header un token inválido
4. Presionar botón Send

Resultado esperado: Se devuelve un código 401 Unauthorized con mensaje "The access token expired" o similar

Tipo de prueba: Negativa

Severidad: Alta

Prioridad: Alta

Clasificación: Security / Regression

¿Candidato a automatización?: Sí

TRACKS

Caso de prueba 5

ID: SPOT-TR-01

Título: Obtener detalles de una pista válida

Precondiciones: Usuario autenticado con token válido, se conoce el ID de una pista válida

Pasos a seguir:

1. Realizar una petición GET a <https://api.spotify.com/v1/tracks/{id}>
2. Reemplazar {id} con el ID de una pista válida (ejemplo:
11dFghVXANMIKmJXsNCbNI)
3. Incluir: Authorization: OAuth 2.0 <token>

4. Presionar botón Send

Resultado esperado: Código 200 OK y detalles de la pista: nombre, artista, duración, tipo, popularidad, etc.

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de prueba 6

ID: SPOT-TR-02

Título: Consultar una pista con ID inválido

Precondiciones: Usuario autenticado con token válido

Pasos a seguir:

1. Realizar una petición GET a https://api.spotify.com/v1/tracks/ID_INVALIDO
2. Usar un ID de pista incorrecto o inventado ejemplo: 12345asdfDSAC32
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Código 400 Bad Request o 404 Not Found y mensaje de error descriptivo

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Alta

Clasificación: Regression

¿Candidato a automatización?: Sí

Caso de Prueba 7

ID: SPOT-TR-03

Título: Guardar una canción en la biblioteca del usuario

Precondiciones: Usuario autenticado con token válido con scope user-library-modify

Pasos a seguir:

1. Realizar una petición PUT al endpoint:
<https://api.spotify.com/v1/me/tracks?ids=3KkXRkHbMCARz0aVfEt68P>
2. Incluir: Authorization: OAuth 2.0 <token>
3. No enviar body (el ID está en la URL)
4. Presionar botón Send

Resultado esperado: Código 200 OK y la canción se agrega exitosamente a la biblioteca del usuario

Tipo de prueba: Positiva

Severidad: Media

Prioridad: Media

Clasificación: Functional

¿Candidato a automatización?: Sí

Caso de Prueba 8

ID: SPOT-TR-04

Título: Eliminar una canción de la biblioteca del usuario

Precondiciones: Usuario autenticado con token válido con scope user-library-modify

Pasos a seguir:

1. Realizar una petición **DELETE** al endpoint:

<p>https://api.spotify.com/v1/me/tracks?ids=3KkXRkHbMCARz0aVfEt68P</p> <ol style="list-style-type: none"> Incluir: Authorization: OAuth 2.0 <token> No enviar body (el ID está en la URL) Presionar botón Send <p>Resultado esperado: Código 200 OK y la canción es eliminada correctamente de la biblioteca del usuario</p> <p>Tipo de prueba: Positiva</p> <p>Severidad: Media</p> <p>Prioridad: Media</p> <p>Clasificación: Functional</p> <p>¿Candidato a automatización?: Sí</p>
--

ALBUMS
<p>Caso de Prueba 9</p> <p>ID: SPOT-A-01</p> <p>Título: Obtener información de múltiples álbumes por ID</p> <p>Precondiciones: Usuario autenticado, múltiples IDs válidos</p> <p>Pasos a seguir:</p> <ol style="list-style-type: none"> Realizar GET a https://api.spotify.com/v1/albums?ids={id1},{id2} Enviar 2 IDs válidos separados por coma ejemplo: 382ObEPsp2rxGrnsizN5TX,1ATL5GLyefJaxhQzSPVrLX Incluir: Authorization: OAuth 2.0 <token> Presionar botón Send <p>Resultado esperado: Código 200 OK, albums[] con info de cada álbum</p> <p>Tipo de prueba: Positiva</p> <p>Severidad: Baja</p> <p>Prioridad: Media</p> <p>Clasificación: Functional</p> <p>¿Candidato a automatización?: Sí</p>
<p>Caso de Prueba 10</p> <p>ID: SPOT-A-02</p> <p>Título: Guardar un álbum en favoritos</p> <p>Precondiciones: Usuario autenticado con permisos user-library-modify</p> <p>Pasos a seguir:</p> <ol style="list-style-type: none"> Hacer PUT a https://api.spotify.com/v1/me/albums Enviar en el body JSON { "ids": ["4aawyAB9vmqN3uQ7FjRGTy"] } Incluir: Authorization: OAuth 2.0 <token> Presionar botón Send <p>Resultado esperado: Código 200 OK o 204 No Content</p> <p>Tipo de prueba: Positiva</p> <p>Severidad: Media</p>

Prioridad: Alta
Clasificación: Functional
¿Candidato a automatización?: Sí

Caso de Prueba 11

ID: SPOT-03

Título: Verificar si el usuario tiene un álbum guardado

Precondiciones: Usuario autenticado

Pasos a seguir:

1. Hacer GET a <https://api.spotify.com/v1/me/albums/contains?>
2. Enviar parametro ids ejemplo: 4aawyAB9vmqN3uQ7FjRGTy
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Código 200 OK, respuesta [true] o [false]

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Media

Clasificación: Exploratoria / Regression

¿Candidato a automatización?: Sí

Caso de Prueba 12

ID: SPOT-04

Título: Eliminar un álbum guardado por el usuario

Precondiciones: Usuario autenticado, álbum guardado previamente

Pasos a seguir:

1. Hacer DELETE a <https://api.spotify.com/v1/me/albums>
2. Enviar JSON body:
{ "ids": ["4aawyAB9vmqN3uQ7FjRGTy"] }
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Sen

Resultado esperado: Código 200 OK o 204 No Content

Tipo de prueba: Positiva

Severidad: Media

Prioridad: Alta

Clasificación: Regression / Functional

¿Candidato a automatización?: Sí

PLAYLISTS

Caso de Prueba 13

ID: SPOT-PL-01

Título: Creación de playlist nueva

Precondiciones: Creación de Variable "user_id" valor: 3x7l902l8j17g5pqtj6jce1h

Pasos a seguir:

1. Realizar POST a https://api.spotify.com/v1/users/{{user_id}}/playlists
2. Incluir header: Content-Type: application/json
3. Incluir: Authorization: OAuth 2.0 <token>
4. En el Body, seleccionar Raw - JSON
5. Escribir el código:


```
{
  "name": "Música para estudiar",
  "description": "Canciones para concentración",
  "public": false
}
```

6. Presionar botón Send

Resultado esperado: Código **201 Created** y respuesta JSON con los datos y el ID de la nueva playlist

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Media

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de Prueba 14

ID: SPOT-PL-02

Título: Agregar canciones a una playlist existente

Precondiciones: Playlist creada, crear variable playlist_id valor: 2BeLknXsk2gZh0mEp3sJqZ

Pasos a seguir:

1. Hacer POST a https://api.spotify.com/v1/playlists/{{playlist_id}}/tracks
2. Incluir header: Content-Type: application/json
3. Incluir: Authorization: OAuth 2.0 <token>
4. En el Body, seleccionar Raw - JSON
5. Escribir el código:

```
{
  "uris": [
    "spotify:track:6rqhFgbbKwnb9MLmUQDhG6",
    "spotify:track:4iV5W9uYEdYUVa79Axb7Rh"
  ]
}
```

6. Presionar botón Send

Resultado esperado: Código **201 Created** y snapshot_id en la respuesta

Tipo de prueba: Positiva

Severidad: Media

Prioridad: Alta

Clasificación: Regression / Functional

¿Candidato a automatización?: Sí

Caso de Prueba 15

ID: SPOT-PL-03

Título: Intento de crear playlist sin nombre

Precondiciones: Creación de Variable "user_id" valor: 3x7l902l8j17g5pqt6j6jce1h

Pasos a seguir:

1. Hacer POST a https://api.spotify.com/v1/users/{{user_id}}/playlists
2. Incluir header: Content-Type: application/json
3. Incluir: Authorization: OAuth 2.0 <token>
4. En el Body, seleccionar Raw - JSON
5. Escribir el código:
 - a. {}
6. Presionar botón Send

Resultado esperado: Código **400 Bad Request** y mensaje de error sobre el campo **name** obligatorio

Tipo de prueba: Negativa

Severidad: Alta

Prioridad: Media

Clasificación: Validation

¿Candidato a automatización?: Sí

Caso de Prueba 16

ID: SPOT-PL-04

Título: Intento de eliminar (unfollow) una playlist con ID inválido

Precondiciones: Creación de una variable `invalid_playlistID` valor:
1234567890abcdefABCDEF

Pasos a seguir:

1. Hacer DELETE a
`https://api.spotify.com/v1/playlists/{{invalid_playlistID}}/followers`
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Código **404 Not Found** y Mensaje de error con descripción:
"Playlist not found" o similar

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Media

Clasificación: Functional

¿Candidato a automatización?: Sí

CATEGORIES

Caso de Prueba 17

ID: SPOT-CAT-01

Título: Obtener lista de categorías de contenido disponibles

Precondiciones:

Pasos a seguir:

1. Realizar GET a <https://api.spotify.com/v1/browse/categories>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Código **200 OK** y un objeto JSON con el listado de categorías
(`categories.items`)

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Functional

¿Candidato a automatización?: Sí

Caso de Prueba 18

ID: SPOT-CAT-02

Título: Obtener detalles de una categoría específica

Precondiciones: Variable `category_id` configurada valor: pop

Pasos a seguir:

1. Hacer POST a `https://api.spotify.com/v1/browse/categories/{{category_id}}`

2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Código **200 OK** y detalles de la categoría (name, icons, etc.)

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Media

Clasificación: Functional

¿Candidato a automatización?: Sí

Caso de Prueba 19

ID: SPOT-CAT-06

Título: Solicitar categorías con parámetro **limit** fuera de rango

Precondiciones: Usuario autenticado con token válido

Pasos a seguir:

1. Hacer GET a <https://api.spotify.com/v1/browse/categories?limit=100>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Código **400 Bad Request** y Mensaje indicando que el valor del parámetro **limit** es inválido

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Media

Clasificación: Boundary / Input Validation

¿Candidato a automatización?: Sí

Caso de Prueba 20

ID: SPOT-CAT-07

Título: Usar parámetro de paginación **offset** negativo al listar categorías

Precondiciones: Usuario autenticado con token válido

Pasos a seguir:

1. Hacer GET a <https://api.spotify.com/v1/browse/categories?offset=-10>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Código **400 Bad Request** y Mensaje de error indicando que el valor de **offset** debe ser ≥ 0

Tipo de prueba: Negativa

Severidad: Baja

Prioridad: Media

Clasificación: Input Validation / Pagination

¿Candidato a automatización?: Sí

EPISODIOS

Caso de prueba 21

ID: SPOT-EP-01

Título: Obtención de información del episodio válido

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint
<https://api.spotify.com/v1/episodes/-id>.
2. Enviar los parámetros id con el ID de un episodio valido ejemplo:
5XvKimawmFliJfe93PTd1u
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 200 OK, y muestre la información completa del episodio como descripción, URLs, espectáculos.

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de prueba 22

ID: SPOT-EP-02

Título: Obtención de información del episodio no válido .

Precondiciones: Usuario autenticado con token válido.

Pasos a seguir:

1. Realizar una petición GET al endpoint
<https://api.spotify.com/v1/episodes/{id}>
2. Enviar los parámetros id con el ID de un episodio no valido ejemplo:
512ojhOuo1ktJprKbVcKyQ
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 404 **Resource not found** Mensaje de error que el recurso no fue encontrado

Tipo de prueba: Negativa

Severidad: Baja

Prioridad: Media

Clasificación: Regresión

¿Candidato a automatización?: Sí

Caso de prueba 23

ID: SPOT-EP-03

Título: Obtención de episodios guardados del usuario con datos fuera del rango permitido.

Precondiciones: Usuario autenticado con token válido

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/me/episodes>
2. Enviar los parámetros como ejemplo: market: ES, Limit:51 o 0 y offset:0
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 400, y muestra un mensaje de error de límite invalido

Tipo de prueba: Negativa

Severidad: Baja

Prioridad: Alta

Clasificación: Regresión

¿Candidato a automatización?: SI

Caso de prueba 24

ID: SPOT-EP-04

Título: obtención de episodios guardados del usuario con datos considerados dentro del rango permitido.

Precondiciones: Usuario autenticado con token válido

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/me/episodes>
2. Enviar los parámetros como ejemplo: market: ES, Limit: 20 y offset:5
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 200 OK, muestra un enlace de punto de conexión, devolviendo un resultado completo de la solicitud.

Tipo de prueba: Positivo

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de prueba 25

ID: SPOT-EP-05

Título: Obtención de varios episodios con ids válidos

Precondiciones: Usuario autenticado con token válido e lds válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/episodes>
2. Enviar los parámetros como lds Válidos como ejemplo: "5Xt5DXGzch68nYYamXrNxZ", "0jliC3UEA0vTscepW47d1G"y market:ES
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 200, y muestra la información y contenido de los episodios

Tipo de prueba:Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

SHOW

Caso de prueba 26

ID: SPOT-SH-01

Título: Obtención de Shows con el ID con el campo del Market vacío

Precondiciones: Usuario autenticado con token válido e lds válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/shows/{id}>
2. Enviar el campo del Market vacío
3. Enviar los parámetros como lds Válidos como ejemplo:
"6eGKCVP58uScjBx1QthjmS" y market:ES
4. Incluir: Authorization: OAuth 2.0 <token>
5. Presionar botón Send

Resultado esperado: Devuelve un código 200 de OK, y muestra el detalle de todos los mercados disponibles o habilitados para mostrar el show, además muestra la información detallada de Show

Tipo de prueba:Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de prueba 27

ID: SPOT-SH-02

Título: Guardar Show para el usuario

Precondiciones: Usuario autenticado con token válido e lds válidos

Pasos a seguir:

1. Realizar una petición PUT al endpoint <https://api.spotify.com/v1/me/shows>
2. Enviar los parámetros como lds Válidos como ejemplo:
5CfCWKI5pZ28U0uOzXkDHe,5as3aKmN2k11yfDDDSrvaZ
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 200 oK, y muestra un mensaje de show guardado

Tipo de prueba:Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance
¿Candidato a automatización?: Sí

Caso de prueba 28

ID: SPOT-SH-03

Título: Eliminar shows guardados.

Precondiciones: Usuario autenticado con token válido e lds válidos

Pasos a seguir:

1. Realizar una petición DELETE al endpoint
`https://api.spotify.com/v1/me/shows`
2. Enviar los parámetros como lds Válidos y market valido ejemplo:
`5CfCWKI5pZ28U0uOzXkDHe,5as3aKmN2k11yfDDDSrvaZ`
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 200, y muestra un mensaje Show removed
datos eliminados

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de prueba 29

ID: SPOT-SH-04

Título: Obtención de Shows con el ID errónea

Precondiciones: Usuario autenticado con token válido e lds válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint
<https://api.spotify.com/v1/shows/{id}>
2. Enviar el campo del Market vacío
3. Enviar los parámetros como lds errónea como ejemplo:
`"6eGKCVP58uScjBx1Qthjmk"`
4. Incluir: Authorization: OAuth 2.0 <token>
5. Presionar botón Send

Resultado esperado: Devuelve un código 404, con un mensaje indicando recurso no encontrado ("Resource not found")

Tipo de prueba: Negativo

Severidad: Baja

Prioridad: Alta

Clasificación: Regresión

¿Candidato a automatización?: Sí

ARTISTS

Caso de prueba 30

ID: SPOT-AT-01

Título: Obtención de las mejores canciones del artista con la ID, sin introducir el campo de market

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

- Realizar una petición GET al endpoint <https://api.spotify.com/v1/artists/-id-/top-tracks>
- Enviar el parámetros como ser ID Válido y market vacio para esta prueba ejemplo: 3TVXtAsR1Inumwj472S9r4
- Incluir: Authorization: OAuth 2.0 <token>
- Presionar botón Send

Resultado esperado: Devuelve un código 200, y muestra información del tipo álbum, cantidad de piezas o tracks y muestra información de mercados habilitados para cada track

Tipo de prueba:Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Caso de prueba 31

ID: SPOT-AR-02

Título: Obtener álbumes de un artista específico

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/artists/.id./albums>
2. Enviar el parámetros como ser ID Válido ejemplo: 5JMjt80HJWR1e4hMw63Z5t, Include_group:single, Market:ES, Limit: 10, offset: 5.
3. Incluir: Authorization: OAuth 2.0 <token>
4. Presionar botón Send

Resultado esperado: Devuelve un código 200 OK, y muestra información y páginas de los álbumes seleccionados.

Tipo de prueba:Positiva

Severidad: Baja

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Sí

Play

Caso de prueba 32

ID: SPOT-P-01

Título: Saltar a la siguiente pista de la cola del usuario

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición POST al endpoint <https://api.spotify.com/v1/me/player/next>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Devuelve un código 403 y muestra un mensaje de error "Player command failed: Premium required"

Tipo de prueba: Negativa

Severidad: Media

Prioridad: Media

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: No

Caso de prueba 33

ID: SPOT-P-02

Título: Obtener el estado de reproducción.

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/me/player>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Devuelve un código 204, muestra como resultado 1 Reproducción no disponible

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Media

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: No

Caso de prueba 34

ID: SPOT-P-03

Título: Obtener el estado de reproducción.

Precondiciones:**Pasos a seguir:**

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/me/player>
2. Presionar botón Send

Resultado esperado: Devuelve un código 401, muestra el error "Invalid access token"

Tipo de prueba: Negativa

Severidad: Baja

Prioridad: Media

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: No

Caso de prueba 35

ID: SPOT-P-04

Título: Transferir la reproducción a un nuevo dispositivo

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición PUT al endpoint <https://api.spotify.com/v1/me/player>
2. Incluir: Authorization: OAuth 2.0 <token>
3. requiere "device_ids" "74ASZWbe4IXaubB36ztrGX" en Body
4. Presionar botón Send

Resultado esperado: Devuelve un código 403 y muestra un mensaje de error "Player command failed: Premium required"

Tipo de prueba: Positiva

Severidad: Baja

Prioridad: Baja

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: No

Caso de prueba 36

ID: SPOT-P-05

Título: Pausar la reproducción en la cuenta del usuario.

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición PUT al endpoint <https://api.spotify.com/v1/me/player>
2. Incluir: Authorization: OAuth 2.0 <token>
3. requiere "device_ids" "0d1841b0976bae2a3a310dd74c0f3df354899bc8" en Body
4. Presionar botón Send

Resultado esperado: Devuelve un código 403 y muestra un mensaje de error "Player command failed: Premium required"

Tipo de prueba:Positiva
Severidad: Alta
Prioridad: Alta
Clasificación: Smoke / Acceptance
¿Candidato a automatización?: No

User

Caso de prueba 37

ID: SPOT-U-01

Título: Obtener el perfil del usuario actual.

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/me>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Presionar botón Send

Resultado esperado: Devuelve un código 200 y muestra los datos del usuario actual en formato JSON.

Tipo de prueba:Negativa

Severidad: Media

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: SI

Caso de prueba 38

ID: SPOT-U-02

Título: Obtener el perfil del usuario actual.

Precondiciones:

Pasos a seguir:

1. Realizar una petición GET al endpoint <https://api.spotify.com/v1/me>
2. Presionar botón Send

Resultado esperado: Devuelve un código 401 y muestra un mensaje de error "Invalid access Token".

Tipo de prueba:Negativa

Severidad: Media

Prioridad: Alta

Clasificación: Smoke / Acceptance
¿Candidato a automatización?: SI

Caso de prueba 39

ID: SPOT-U-03

Título: Agregar al usuario actual como seguidor de un artista.

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición PUT al endpoint
<https://api.spotify.com/v1/me/following?type=artist&ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxijzplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Agregar los parámetros type=artist y ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxijzplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6
4. Presionar botón Send

Resultado esperado: Devuelve un código 204, artista seguido.

Tipo de prueba:Positivo

Severidad: Media

Prioridad: Media

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: Si

Caso de prueba 40

ID: SPOT-U-03

Título: Agregar al usuario actual como seguidor de un artista.

Precondiciones:

Pasos a seguir:

1. Realizar una petición PUT al endpoint
<https://api.spotify.com/v1/me/following?type=artist&ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxijzplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6>
2. Agregar los parámetros type=artist y ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxijzplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6
3. Presionar botón Send

Resultado esperado: Devuelve un código 401 y muestra un mensaje de error "Invalid access Token".

Tipo de prueba:Negativo

Severidad: Media

Prioridad: Media

Clasificación: Smoke / Acceptance
¿Candidato a automatización?: Si

Caso de prueba 41

ID: SPOT-U-05

Título: Eliminar al usuario actual como seguidor de un artista.

Precondiciones: Usuario autenticado con token válido e Ids válidos

Pasos a seguir:

1. Realizar una petición DEL al endpoint
<https://api.spotify.com/v1/me/following?type=artist&ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxizplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6>
2. Incluir: Authorization: OAuth 2.0 <token>
3. Agregar los parámetros type=artist y ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxizplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6
4. Presionar botón Send

Resultado esperado: Devuelve un código 204 artista no seguido.

Tipo de prueba:Positiva

Severidad: media

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: si

Caso de prueba 42

ID: SPOT-U-06

Título: Eliminar al usuario actual como seguidor de un artista.

Precondiciones:

Pasos a seguir:

1. Realizar una petición DEL al endpoint
<https://api.spotify.com/v1/me/following?type=artist&ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxizplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6>
2. Agregar los parámetros type=artist y ids=2CIMQHirSU0MQqyYHq0eOx%2C57dN52uHvrHOxizplgu3E%2C1vCWHaC5f2uS3yhpwWbIA6
3. Presionar botón Send

Resultado esperado: Devuelve un código 401 y muestra un mensaje de error "Invalid access Token".

Tipo de prueba:Negativa

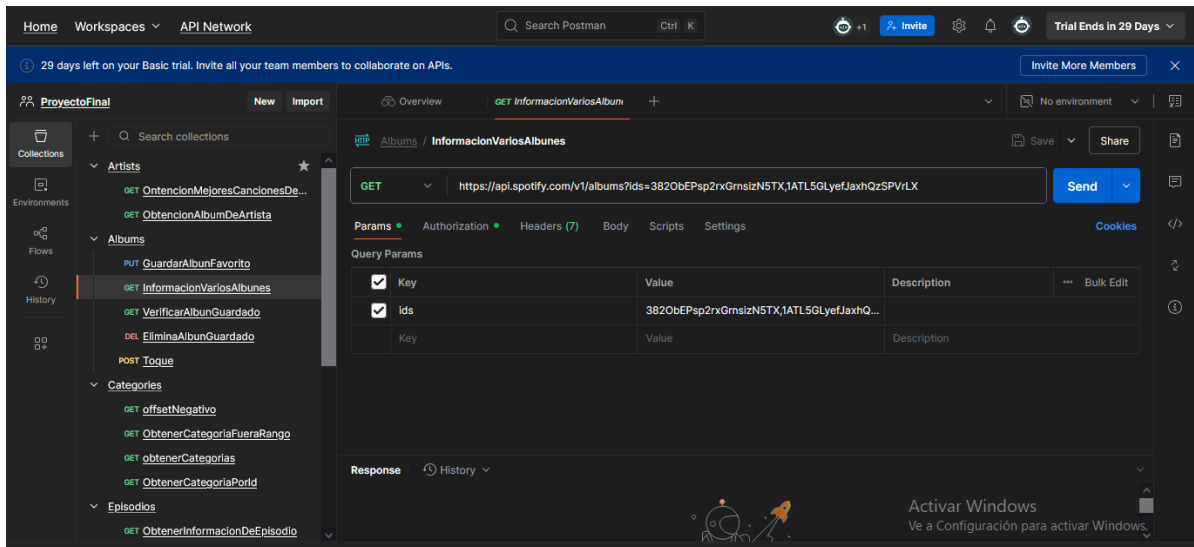
Severidad: media

Prioridad: Alta

Clasificación: Smoke / Acceptance

¿Candidato a automatización?: si

3.3. Creación de colecciones en la herramienta postman con las pruebas diseñadas como GET, POST, PUT Y DELETE.



3.4. Automatización básica de los casos de prueba en karate

A. Descripción del proyecto, es decir cual es el propósito de este

Este proyecto tiene como objetivo realizar pruebas automatizadas a la API pública de Spotify. Se testean distintos endpoints relacionados a la gestión de álbumes, pistas y artistas, como búsqueda de canciones, guardar álbumes o eliminar contenido del usuario autenticado

B. Detallar el Stack de tecnología, es decir que librerías o frameworks se están usando

- Lenguaje: Java
- Framework de pruebas:
[Karate](<https://github.com/karatelabs/karate>)
- Build Tool: Maven
- Editor usado: Visual Studio Code
- Gestión de versiones: Git & GitHub

C. Qué pasos se deben seguir hasta tener corriendo las pruebas automatizadas

- Clonar el repositorio
- git clone
- token

Caso de Prueba #1: Búsqueda exitosa de canción por nombre

Cucumber Report

Features Tags Steps Failures

Project	Date
apitests	19 jul. 2025, 21:18

Features Statistics

The following graphs show passing and failing statistics for features

Steps

A donut chart representing the progress of steps. The chart is a bright green circle with a small gap at the top, indicating 100% completion. Below the chart are three small circles: two white and one black.

Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
src/test/java/apitests/Spotify/endpoints_GET.feature	6	0	0	0	0	6	1	0	1	2.829	Passed
	6	0	0	0	0	6	1	0	1	2.829	1
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%

Tags: @Spotify

Feature src/test/java/apitests/Spotify/endpoints_GET.feature

Buscar canción por nombre

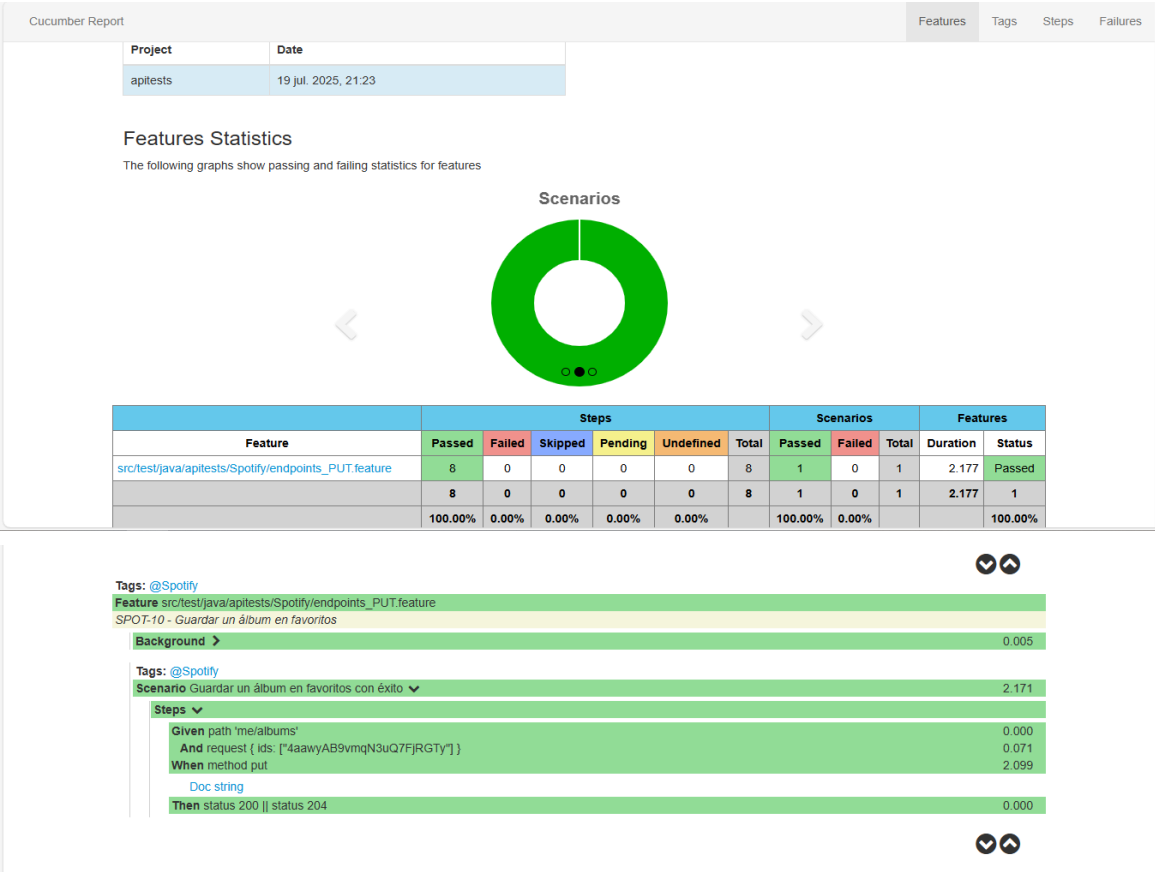
Tags: @Spotify

Scenario Búsqueda con token válido 2.829

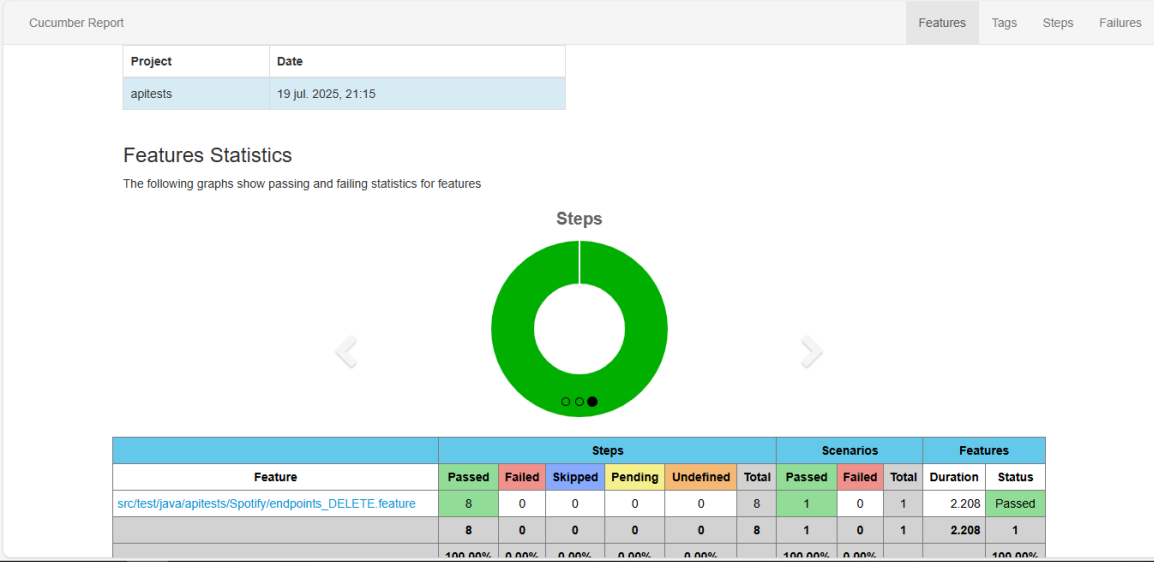
Steps

Given url 'https://api.spotify.com/v1/search'	0.001
Doc string	
And header Authorization = "Bearer BQBDS4K3S-KTAdrf324n4IEZ8FYWsUPJLJmew6MXzUz4QcwnFAX1_4zWzRwLdDw_wz6O7IaiqlWSXxyTZosY07ouJlBf9jrY2Md-SY1990uUa6AMggqTF6jCJRwVLUQDijyT7Q0WbmSD2e4ib2gJ0j8DA5QVexJs5ILuJlwKzGinLerzq-Ly8Ku_d_s0osmedRB8Ny8CVhAH0d5Ka3uRNPDx9KiZegr4E7Pj0zpZULv7XeZSFZWU44E-0ecrUoTT82Rqh86A5SZnXw0Qn9-t4tyRXUm5E8OgvqbQMuz_EGS5tqA94gRdUx2Ke3ldnDqMhwVPNEb5MLt2g2__p1NjqOM1ITqSC8-Ns5Qi8cTYBg"	0.001
And param q = 'Shape of You'	0.001
And param type = 'track'	0.001
When method GET	2.823
Doc string	
Then status 200	0.000

Caso de Prueba #10: Guardar un álbum en favoritos



Caso de Prueba #12:Eliminar un álbum guardado por el usuario



Tags: @Spotify

Feature src/test/java/apitests/Spotify/endpoints_DELETE.feature

SPOT-12 - Eliminar un álbum guardado por el usuario

Background >

0.012

Tags: @Spotify

Scenario Eliminar álbum de favoritos con éxito

4.257

Steps

Given path 'me/albums'

0.001

And request { ids: ["4aawyAB9vmqN3uQ7FJRGTy"] }

0.172

When method delete

4.083

Doc string

Then status 200 || status 204

0.000