

CASO DE TESTE - API SPOTIFY

Carlos E. Aguiar¹, Evellyn R. Oliveira¹, Marcelo V. Carvalho¹, Stefany F. Taveira¹, Victor V. Costa¹

Engenharia de Software - Faculdade Metropolitana de Manaus (FAMETRO)

eduaguaiar1006@gmail.com, evellyn.kainos@gmail.com,
marcelo123br35@gmail.com, Stefany9490@gmail.com, victorvahid76@gmail.com

1. Introdução

Este estudo foca na API do Spotify, uma interface fundamental que permite o acesso programático ao vasto catálogo de músicas, artistas e playlists do serviço. Este trabalho foi criado para que desenvolvedores e analistas interessados possam compreender a estrutura e o funcionamento dos diversos *endpoints* que compõem a API. Inspirado na necessidade de integração com plataformas que dependem de dados musicais, este estudo explora as funcionalidades do sistema, que incluem o acesso a dados de Artistas, Músicas (Faixas), Álbuns, Gêneros e Playlists.

A arquitetura da API do Spotify é baseada em *endpoints* RESTful que interagem com o banco de dados e sistemas internos do Spotify. O sucesso e a confiabilidade de qualquer aplicação que se integre a esta plataforma dependem fundamentalmente da robustez e da precisão dessa API.

Este Artigo apresenta uma série de Casos de Teste focados em Endpoints específicos da API do Spotify (consultável no GitHub - link: <https://github.com/VictorVahid/Testes-Academicos-API-Spotify>). O objetivo é validar tanto os cenários de sucesso (*Happy Path*), garantindo que os dados (como nome do artista, lista de faixas ou *link de streaming*) sejam retornados corretamente, quanto os cenários negativos, assegurando que a API trate erros de entrada (como *IDs* inexistentes, formato de requisição inválido ou falta de autenticação) com códigos de *status HTTP* e mensagens de erro apropriados. Este rigor no teste é essencial para manter a disponibilidade, o desempenho e a facilidade de integração do sistema.

2. Casos de Teste

Nesse tópico será detalhado os testes para os 30 Endpoints da API do Spotify.

2.1 Endpoint: /v1/search

Descrição: Pesquisar artistas, músicas, álbuns ou playlists.

ID	Cenário	Métodos	Entradas	Resultados esperados

1	Positivo – Buscar artista existente	GET	q=queen&type=artist	200 OK e retorno contendo "Queen"
2	Negativo – Campo 'q' vazio	GET	q=&type=artist	400 BAD REQUEST
3	Negativo – Tipo inválido	GET	q=queen&type=qualquercoisa	400 BAD REQUEST

2.2 Endpoint: /v1/artists/{id}

Descrição: Buscar informações de um artista específico.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Artista válido	GET	/v1/artists/1vCWHaC5f2uS3yhpwWbIA6 (Avicii)	200 OK – Dados do artista
2	Negativo – ID inexistente	GET	/v1/artists/xyz123	404 NOT FOUND
3	Negativo – ID em formato inválido	GET	/v1/artists/123	400 BAD REQUEST

2.3 Endpoint: /v1/artists/{id}/top-tracks

Descrição: Buscar as faixas mais populares de um artista.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Artista existente	GET	/v1/artists/{id}/top-tracks?market=BR	200 OK – Lista de músicas
2	Negativo – Falta do parâmetro market	GET	/v1/artists/{id}/top-tracks	400 BAD REQUEST

3	Negativo – Artista inexistente	GET	/v1/artists/00/top-tracks	404 NOT FOUND
---	--------------------------------	-----	---------------------------	---------------

2.4 Endpoint: /v1/albums/{id}

Descrição: Buscar informações completas de um álbum.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – ID válido	GET	/v1/albums/4aawyAB9vmqN3uQ7FjRGTy	200 OK – Dados do álbum
2	Negativo – Álbum inexistente	GET	/v1/albums/invalido123	404 NOT FOUND
3	Negativo – ID com formato incorreto	GET	/v1/albums/123	400 BAD REQUEST

2.5 Endpoint: /v1/albums/{id}/tracks

Descrição: Lista todas as músicas de um álbum.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Álbum válido	GET	/v1/albums/{id}/tracks	200 OK – Lista de faixas
2	Negativo – Álbum inexistente	GET	/v1/albums/000/tracks	404 NOT FOUND

2.6 Endpoint: /v1/tracks/{id}

Descrição: Obter detalhes de uma faixa específica.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Track válida	GET	/v1/tracks/{id}	200 OK – Dados da música
2	Negativo – Track	GET	/v1/tracks/123	404 NOT FOUND

	inexistente			
3	Negativo – ID inválido	GET	/v1/tracks/a bc	400 BAD REQUEST

2.7 Endpoint: /v1/me/player/currently-playing

Descrição: Retorna a música que está tocando no momento (com token válido).

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Token válido	GET	/v1/me/player/currently-playing	200 OK – Música atual
2	Negativo – Token ausente	GET	-	401 UNAUTHORIZED
3	Negativo – Nada tocando	GET	-	204 NO CONTENT

2.8 Endpoint: /v1/me/player/recently-played

Descrição: Lista as últimas músicas ouvidas pelo usuário.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Token válido	GET	/v1/me/player/recently-played?limit=20	200 OK – Lista de faixas
2	Negativo – Token inválido	GET	-	401 UNAUTHORIZED

2.9 Endpoint: /v1/playlists/{id}

Descrição: Buscar informações sobre uma playlist.

ID	Cenário	Métodos	Entradas	Resultados esperados
1	Positivo – Playlist válida	GET	/v1/playlists/{id}	200 OK – Dados da playlist

2	Negativo – Playlist inexistente	GET	/v1/playlists /xyz123	404 NOT FOUND
---	------------------------------------	-----	--------------------------	------------------