

# Spotify API

 Por Darlan Filho 

 Fique atento a todas as instruções que este documento oferece, a continuidade do processo dependerá disso!

Você está na etapa do desafio técnico, parabéns por ter chegado até aqui! Neste desafio, queremos conhecer suas habilidades técnicas em foco prático e aplicado na resolução de um problema. Aqui conheceremos seu estilo de código, aptidões técnicas e, sobretudo, a sua capacidade de resolução de problemas.

## O desafio... ☺

O nosso time gosta bastante de ouvir música, e com isso gostaríamos de ter um aplicativo que se comunicasse com o Spotify e nos mostrasse algumas informações de forma mais prática.

## Objetivo ☺

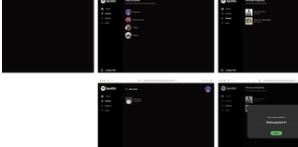
Você deve criar uma aplicação que autentique com o Spotify e cumpra os seguintes casos de uso:

- Listar os artistas que mais ouvimos;
- Listar os álbuns desses artistas;
- Listar as playlists do usuário;
- Criar uma nova playlist;
- Exibir os dados do usuário.

## UI/Protótipo ☺

Para solucionar esse desafio, o seu aplicativo deve também seguir a interface proposta. [Aqui](#) você poderá acessar o link da interface disponível no Figma, bem como seu protótipo funcional.

 **Desafio Front-end - Luizalabs**  
Atualizado em 26 de dez. de 2024  
Created with Figma  
 [Abrir pré-visualização](#)



## Instruções ☺

A seguir estão os requisitos (obrigatórios e bônus) que serão levados em consideração para a análise da sua solução. Ao submeter o projeto, **inclua o texto abaixo em seu README**, marcando tudo aquilo que realmente foi feito.

```
1 # Requisitos
2 ## Requisitos obrigatórios
3 - [ ] Seguimentação de commits
4 - [ ] Lint
5 - [ ] Autenticação via Spotify
6 - [ ] Listar artistas
7 - [ ] Listar albuns de um artista
8 - [ ] Utilizar paginação (scroll infinito ou não)
9 - [ ] Funcionamento offline
```

```
10 - [ ] Testes unitários
11 - [ ] Deploy da aplicação
12
13 ## Bônus
14 - [ ] Testes E2E
15 - [ ] Integração com Sentry
16 - [ ] CI/CD
17 - [ ] Responsividade (celular e tablet)
18 - [ ] Qualidade de código (Sonarqube)
19 - [ ] PWA
```

A solução do desafio deve seguir a interface proposta acima (podendo também haver melhorias por parte do candidato), e isso também será critério de avaliação.

## Spotify API

O desafio deverá obrigatoriamente utilizar a [API do Spotify](#). Para facilitar o processo de implementação, deixaremos abaixo as rotas que serão utilizadas nos casos de uso:

- >Listar os artistas que mais ouvimos



### [Web API Reference | Spotify for Developers](#)

Get the current user's top artists or tracks based on calculated affinity.

 developer.spotify.com

- Listar os álbuns desses artistas



### [Web API Reference | Spotify for Developers](#)

You may not facilitate downloads of Spotify content or enable "stream ripping"

 developer.spotify.com

- Listar as playlists do usuário



### [Web API Reference | Spotify for Developers](#)

Get a list of the playlists owned or followed by the current Spotify user.

 developer.spotify.com

- Criar uma nova playlist



### [Web API Reference | Spotify for Developers](#)

Create a playlist for a Spotify user. (The playlist will be empty until you add tracks.) Each user is generally limited to a maximum of 11000 playlists.

 developer.spotify.com

- Exibir os dados do usuário



### [Web API Reference | Spotify for Developers](#)

Get detailed profile information about the current user (including the current user's username).

 developer.spotify.com

## TL;DR

1. A arquitetura fica a seu critério, mas serão avaliados os fatores: manutenibilidade, escalabilidade e desempenho.

2. É obrigatório o uso do [React](#) para a resolução desse problema.
3. Escreva um README descrevendo o passo a passo de como executar a aplicação e o que é necessário para tal.
4. Deixe claro na documentação (README) as escolhas utilizadas, ao que tange tecnologia e padrões arquiteturais aplicados na resolução.