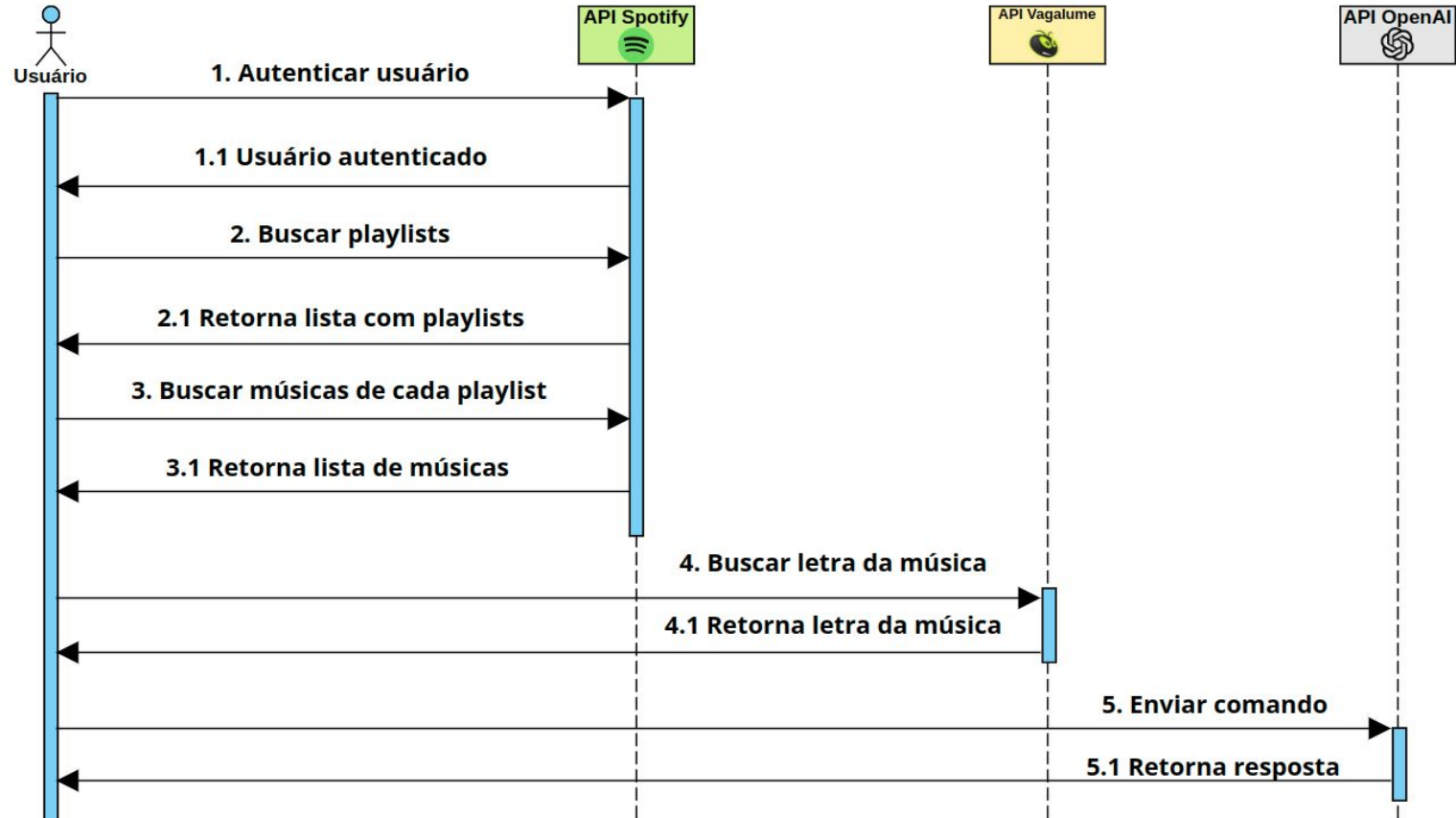


Python, OpenAI, Spotify e Maizena: Uma receita rápida para fazer análise de sentimento de músicas

Overview da aplicação



Libs utilizadas

requests: para fazermos as requisições HTTP nas APIs

spotipy: uma biblioteca python para a API do Spotify

openai: para acessarmos os modelos de Inteligência Artificial



```
pip install requests spotipy openai
```

1º Passo: pegar credenciais das APIs e configurar as variáveis de ambiente

Spotify: criar um app no dashboard do Spotify (<https://developer.spotify.com/dashboard/create>)

Vagalume: criar um app (<https://auth.vagalume.com.br/settings/api/>)

OpenAI: criar uma secret key (<https://platform.openai.com/account/api-keys>)



```
# settings.py
```

```
import os
```

```
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")  
SPOTIPY_CLIENT_ID = os.getenv("SPOTIPY_CLIENT_ID")  
SPOTIPY_CLIENT_SECRET = os.getenv("SPOTIPY_CLIENT_SECRET")  
SPOTIPY_REDIRECT_URI = os.getenv("SPOTIPY_REDIRECT_URI")  
VAGALUME_API_KEY = os.getenv("VAGALUME_API_KEY")
```

2º Passo: criar um client para acessar a API do Spotify

```
# spotify_client.py
import spotify
from spotify.oauth2 import SpotifyOAuth

from settings import (SPOTIPY_CLIENT_ID, SPOTIPY_CLIENT_SECRET,
                      SPOTIPY_REDIRECT_URI)

class SpotifyClient:
    def __init__(self):
        scope = "user-read-private,playlist-read-private"
        auth_manager = SpotifyOAuth(
            client_id=SPOTIPY_CLIENT_ID,
            client_secret=SPOTIPY_CLIENT_SECRET,
            redirect_uri=SPOTIPY_REDIRECT_URI,
            scope=scope
        )
        self.sp = spotify.Spotify(auth_manager=auth_manager)
```

```
@property
def username(self) -> str:
    current_user = self.sp.current_user()
    return current_user["id"]

def get_playlists(self) -> list:
    return self.sp.user_playlists(self.username)["items"]

def get_tracks_from_playlist(self, playlist_id: int) -> list:
    response = self.sp.playlist_tracks(playlist_id=playlist_id)
    return response["items"]
```

3º Passo: criar um client para acessar a API do Vagalume

```
# vagalume_client.py

import requests

from settings import VAGALUME_API_KEY

class VagalumeClient:
    def __init__(self):
        self.url = "https://api.vagalume.com.br/search.php?"

    def get_lyrics(self, track: str, artist: str) -> str:
        query = f"art={artist}&mus={track}&apikey={VAGALUME_API_KEY}"
        url = self.url + query
        response = requests.get(url)

        try:
            lyrics = response.json()["mus"][0]["text"]
            return lyrics.replace("\n", " ").strip()
        except KeyError:
            return ""
```

4º passo: criar um client para acessar a API da OpenAI

```
# openai_client.py

import openai

from settings import OPENAI_API_KEY

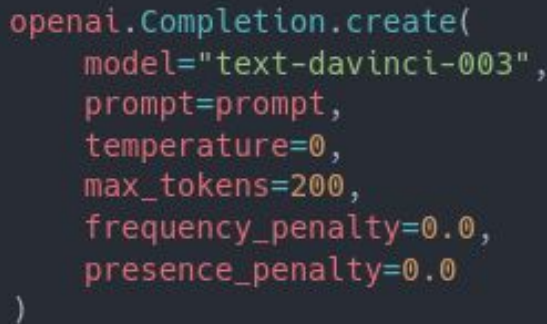
class OpenAIClient:
    def __init__(self):
        openai.api_key = OPENAI_API_KEY

    def classify_sentiment(self, lyrics: str) -> str:
        prompt = f"Classify the sentiment in this lyrics:{lyrics}"

        response = openai.Completion.create(
            model="text-davinci-003",
            prompt=prompt,
            temperature=0,
            max_tokens=200,
            frequency_penalty=0.0,
            presence_penalty=0.0
        )
        return response["choices"][0]["text"]
```


Entendendo alguns conceitos...

Token: Cada token pode ser uma palavra, um número, uma pontuação, um caractere especial ou até mesmo parte de uma palavra, como um fragmento. Por exemplo, a frase "Olá, como você está?" pode ser dividida em seis tokens: ["Olá", ",", "como", "você", "está", "?"].



```
openai.Completion.create(  
    model="text-davinci-003",  
    prompt=prompt,  
    temperature=0,  
    max_tokens=200,  
    frequency_penalty=0.0,  
    presence_penalty=0.0  
)
```

```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQ0eXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "text-davinci-003",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```

**HORA DE COLOCAR A MÃO NA
MASSA!**





That's all Folks!



Yuri Sales

Engenheiro de software na Imperium Markets

repositório: <https://github.com/yurisalesc/spotify-openai>

linkedin: <https://www.linkedin.com/in/yuriscosta/>

