Exercício 2 Desafio:

```
import requests
import boto3
import json
from datetime import datetime

api_key = "da9f57cfdff115b5e3676c7b983f5304"

def lambda_handler(event, context):
    # Obter a lista de filmes do TMDB
    filmes = get_movies()

    # Dividir em lotes de 100 filmes
    arquivos_divididos = dividir_arquivo(filmes)

    # Formatar e enviar os dados para o S3
    formatando_e_enviando_dados(arquivos_divididos)

def get_movies():
    movies = []
    for page in range(1, 1301):
        url = f"https://api.themoviedb.org/3/discover/movie?api_key={api_key}&page={page}"
        response = requests.get(url)
        data = response.json()
        movies.extend(data.get('results', []))
    return movies

def dividir_arquivo(ids_filmes):
    limite = 100
    arquivos = [ids_filmes[i:i + limite] for i in range(0, len(ids_filmes), limite)]
    return arquivos
```

```
def formatando_e_enviando_dados(arquivos):
    for indice, ids in enumerate(arquivos, 1):
        filmes = []
        for movie in ids:
            df = {
                "titulo": movie.get("title", ""),
                "titulo_original": movie.get("original_title", ""),
                "visao_geral": movie.get("overview", ""),
                "data_lancamento": movie.get("release_date", ""),
                "generos": [genero.get("name", "") for genero in movie.get("genres", [])],
                "tempo_duracao(minutos)": movie.get("runtime", ""),
                "id_filme": movie.get("id", ""),
                "media_de_votos": movie.get("vote_average", ""),
                "total_votos": movie.get("vote_count", ""),
                "popularidade": movie.get("popularity", ""),
                "receita": movie.get("revenue", ""),
                "produtoras": [produtora.get("name", "") for produtora in
movie.get("production_companies", [])],
            }
            filmes.append(df)



        filmes = sorted(filmes, key=lambda x: x['popularidade'])
        nome_arquivo_s3 =
f'Raw/tmdb/json/{datetime.utcnow().strftime("%Y/%m/%d")}/filmes({indice}).json'
        arquivo_json = json.dumps(filmes, ensure_ascii=False, indent=2)

        # Enviar dados para o S3 com credenciais
        enviar_dados(arquivo_json, nome_arquivo_s3)


def enviar_dados(arquivo, nome_arquivo):
```

Configurar cliente S3 com credenciais

s3 = boto3.client("s3", region_name="us-east-1")

Enviar arquivo para o S3

s3.put_object(Body=arquivo, Bucket="data-lake-kesley", Key=nome_arquivo)

print(f"Arquivo {nome_arquivo} enviado com sucesso para o S3.")