Proposta do Projeto:

**Ingestão de dados na AWS da API sobre filmes/séries TMDB afim de criar um painel contendo quais longas-metragens recém-lançados tem melhor avaliação.**

**Objetivo do projeto:**

**Esse projeto tem como objetivo testar os conhecimentos adquiridos no Bootcamp da How sobre Engenharia de Dados. Para isso realização da ingestão, tratativa, consumo e apresentação das informações da API utilizarei ferramentas vistas durante todo o Bootcamp, consolidando meu aprendizado. Além dessa consolidação, espero que ao final do projeto consiga ter um painel com informações relevantes sobre os filmes me auxiliando na tomada de decisão sobre qual o próximo longa-metragem que devo assistir. 😊**

**Construção do projeto:**

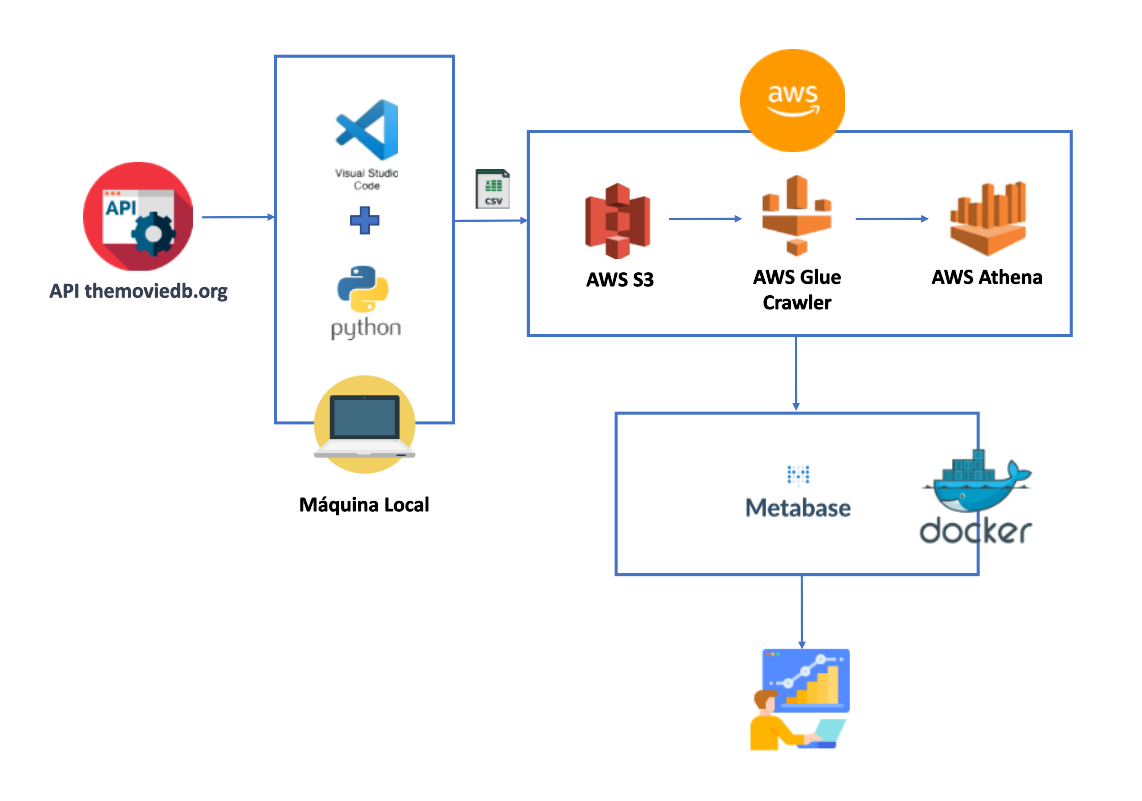
**PASSOS:**

* **Leitura de dados da API** [**https://www.themoviedb.org/**](https://www.themoviedb.org/)
  + **API aberta contendo diversas informações de filmes/séries (Ex.: Classificação, data lançamento, receita, etc)**
  + **A leitura será com Python na máquina local e toda a transformação de dados iniciais para carregar a informação estruturada em um dataframe pandas vai ocorrer nessa etapa.**
  + **Esse dataframe pandas será enviada para o AWS S3 em formato .csv via código Python**
* **Com o dataframe no S3, utilizarei AWS Crawler para reconhecer os metadados e carregar essa tabela em um database denominado filmes**
* **Com a tabela carregado, farei a leitura utilizando AWS Athena afim de ter acesso rápido e eficiente a essas informações**
* **Utilizando docker, conectarei o Metabase na AWS Athena e carregarei meus dados de filmes afim de criar o painel com tudo consolidado, atingindo meu objetivo que é ter essas visualizar de forma eficaz quais filmes recém-lançados tem melhor avaliação**

**PRODUTO/SERVIÇOS/SISTEMAS UTILIZADOS**

* **Python**
* **AWS Crawler**
* **AWS Athena**
* **Docker**
* **Metabase**

**Escopo da arquitetura do projeto:**



**Evolução do projeto:**

* **Criação de pipeline mais robusto e automatizado, tirando o desenvolvimento da máquina local**
* Adição de tratativas de erro e log em toda a etapa de leitura da API
* Versionamento dos códigos

**Fontes:**

<https://how-bootcamps.memberkit.com.br/>

<https://aws.amazon.com/pt/premiumsupport/knowledge-center/?nc2=h_m_ma>

<https://www.themoviedb.org/documentation/api>

<https://dev.to/m0nica/how-to-use-the-tmdb-api-to-find-films-with-the-highest-revenue-82p>

<https://fabianolira.com.br/consultas-sql-no-seu-data-lake-com-aws-athena-e-glue/>

<https://muriloalvesdev.medium.com/conectando-metabase-ao-athena-aws-utilizando-docker-8cef92980597>

<https://github.com/dacort/metabase-athena-driver>