Logotipo

Descrição gerada automaticamente

***Documentação do Código: Integração de APIs e Geração de Banco de Dados***

Kevelin Suellen de Souza Oliveira

Turma: 58875

Data de criação: 22/07/2024

Última modificação: 05/09/2024

Sumário

[Objetivo 3](#_Toc175040630)

[Público alvo 3](#_Toc175040631)

[Saída 3](#_Toc175040632)

[Nível de privacidade 3](#_Toc175040633)

[Pré requisitos 3](#_Toc175040634)

[Ambientes 3](#_Toc175040635)

[Bibliotecas 3](#_Toc175040636)

[Arquivos 4](#_Toc175040637)

[API’s utilizadas 4](#_Toc175040638)

[Funções criadas 4](#_Toc175040639)

[Tratamentos aplicados 4](#_Toc175040640)

[ **Tratamento de datas** 4](#_Toc175040641)

[ **Tratamento de Valores Ausentes** 4](#_Toc175040642)

[ **Conversão de Tipos** 5](#_Toc175040643)

[ **Agregação** 5](#_Toc175040644)

[Método de saída 5](#_Toc175040645)

[Exemplo de consulta 6](#_Toc175040646)

[Versionamento 6](#_Toc175040647)

[Referências 6](#_Toc175040648)

# Objetivo

Este projeto tem por finalidade extrair dados da API do YouTube para analisar e comparar canais na plataforma. O objetivo é reunir informações relevantes sobre o desempenho dos canais e dos vídeos, possibilitando uma análise detalhada e comparativa.

## Público alvo

Este código é voltado para analistas de dados e profissionais de marketing digital que desejam fornecer insights valiosos para criadores de conteúdo. Ele permite coletar métricas importantes para acompanhar o crescimento de canais e identificar tópicos que possam aumentar o engajamento do público.

## Saída

O código gerado nos fornece um arquivo Análise\_Youtube.db.

## Nível de privacidade

Os dados coletados são públicos, pois podem ser acessados por qualquer usuário que navegue pelo Youtube.

Entretando, para conseguir acesso a API, é necessário solicitar uma chave. O acesso a chave é feito através do google clound. A seguir segue um tutorial do Google de como obter a chave:

https://cloud.google.com/docs/authentication/api-keys?hl=pt-br

# Pré requisitos

## Ambientes

O código foi executado em Python 3.7 utilizando o Visual Code como editor.

## Bibliotecas

As bibliotecas utilizadas são:

* googleapiclient.discovery: v Interação com APIs do Google
* Pandas: Manipulação e análise de dados.
* Requests: Realização de requisições HTTP.
* Isodate: Manipulação de datas e horários em formato ISO 8601.
* Sqlite3: Interação com bancos de dados SQLite.

## Arquivos

**Chave da API:** O código requer que a chave da API do YouTube seja inserida diretamente no código, substituindo 'YOUR\_API\_KEY' pela chave válida que você obtém no Google Cloud Console.

## API’s utilizadas

A API utilizada foi a YouTube Data API v3.Ela permite o acesso e interação com os dados do YouTube, como informações sobre vídeos, canais, playlists, comentários, etc.

Para acessar a API, é necessário criar um projeto no Google Cloud Console, habilitar a YouTube Data API v3 e gerar credenciais de API.

## Funções criadas

* **get\_channel\_id(youtube, custom\_url):**

**youtube**: Este é o objeto da API que você usa para se conectar ao YouTube. Ele contém as credenciais e métodos para fazer as requisições à API.

**custom\_url**: Este é o URL personalizado do canal do YouTube. O parâmetro serve como entrada para que a API identifique e busque o ID do canal associado a essa URL.

* **get\_channel\_stats(youtube, channel\_ids):**

**youtube**: O objeto da API do YouTube, necessário para realizar a conexão e buscar as informações dos canais.

**channel\_ids**: Este é o ID ou lista de IDs dos canais de YouTube cujas estatísticas você quer obter. Esses IDs são únicos e identificam os canais na plataforma.

* **get\_video\_ids(youtube, playlist\_id):**

**youtube**: O objeto da API do YouTube usado para buscar os vídeos da lista de reprodução (playlist).

**playlist\_id**: O ID da playlist de onde você deseja buscar os vídeos. Cada canal tem uma playlist de uploads, que pode ser usada para coletar os vídeos do canal.

* **get\_video\_details(youtube, video\_ids, channel\_name):**

**youtube**: O objeto da API do YouTube usado para buscar detalhes sobre os vídeos.

**video\_ids**: Uma lista contendo os IDs dos vídeos que você quer analisar. Cada vídeo no YouTube tem um ID único.

**channel\_name**: O nome do canal do YouTube. Este parâmetro pode ser usado para organizar ou identificar os vídeos por canal ao coletar os detalhes.

* **get\_channel\_activity(youtube, playlist\_id):**

**youtube**: O objeto da API que faz a conexão e coleta dados da playlist do canal.

**playlist\_id**: O ID da playlist de vídeos enviados pelo canal. A API usa este ID para encontrar os vídeos e calcular métricas como frequência de uploads e engajamento.

## Tratamentos aplicados

* **Tratamento de datas**

Formatação**:** A parte da A data é convertida para um objeto datetime do Pandas e formatada para o padrão dia/mês/ano. E como as datas são separadas usando o separador 'T' elas são tratadas para que venham separadas por barras (/).

* **Tratamento de Valores Ausentes**

Uso de get: O método get é usado para obter um valor de um dicionário, se a chave 'publishedAt' não existir no dicionário, o valor 'N/A' será atribuído.

* **Conversão de Tipos**

O número de inscritos, que é uma string no JSON, é convertido para um número inteiro usando a função int(). Se a chave 'subscriberCount' não existir, o valor padrão será 0.

* **Agregação**

Os dados são agrupados por canal e mês usando o método groupby.

* **Erro de conexão**

Verificação de falhas na comunicação com a API.

## Método de saída

Os dados extraídos da API, irão gerar um arquivo .XLSX com 4 abas.

Primeira aba - Estatísticas dos Canais

channel\_id - object

channel\_name - object

published\_date - object

published\_time - object

country - object

subscribers - int64

views - int64

total\_videos - int64

playlist\_id – object

Segunda aba – Destalhes dos vídeos

channel\_name - object

video\_id - object

title - object

published\_date - object

published\_time - object

duration - object

category\_id - object

views - int64

likes - int64

comments - int64

Terceira aba – Uploads mensais

channel\_name - object

month - object

uploads - int64

Quarta aba – Média de enganjamento

Metric - object

average - float64

channel\_name – object

## Exemplo de consulta

conn = sqlite3.connect('Análise\_Youtube.db')

channel\_stats.to\_sql('Estatísticas dos Canais', conn, if\_exists='replace', index=False)

all\_video\_details\_df.to\_sql('Detalhes dos Vídeos', conn, if\_exists='replace', index=False)

monthly\_uploads\_df.to\_sql('Uploads Mensais', conn, if\_exists='replace', index=False)

avg\_engagement\_df.to\_sql('Média de Engajamento', conn, if\_exists='replace', index=False)

query = "SELECT \* FROM 'Estatísticas dos Canais'"

df = pd.read\_sql(query, conn)

print(df)

channel\_id channel\_name published\_date published\_time \

0 UCNWlx\_0XxpUkci6PnCwH\_dQ Bruna Vieira 15/11/2011 01:47:55

1 UCy9xDEfMD8jrXU3NatFLLtw Taciele Alcolea 08/09/2009 15:51:09

2 UC5qJINQ80-zvtGQ4fq3kSRQ Fabi Santina 02/10/2010 01:53:56

country subscribers views total\_videos playlist\_id

0 BR 1330000 95343145 826 UUNWlx\_0XxpUkci6PnCwH\_dQ

1 BR 5860000 888883572 2275 UUy9xDEfMD8jrXU3NatFLLtw

2 BR 1820000 330503095 2371 UU5qJINQ80-zvtGQ4fq3kSRQ

## Versionamento

Um arquivo requiriment.txt foi gerado e será fornecido juntamente com a documentação deste projeto.

## Referências

Foi utilizado como referência o tutorial do canal Programação dinâmica do Youtube

https://www.youtube.com/watch?v=olDCJ1w3FLM&t=538s