ANÁLISE DE DADOS DAS REDES SOCIAS DA PRPG/UFG

Foco : Perfil do Instagram da PRPG

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

PÓS UFG

objetivos :

- Analisar as interações do público **no perfil do Instagram da PRPG** para compreender a **percepção social da pós-graduação.**
- Aprimorar a comunicação para **ampliar a visibilidade institucional** e fortalecer a imagem pública da universidade.
- Contribuir para os objetivos do Plano Nacional de Pós-Graduação, que enfatiza a **transparência**, o **impacto social** e a **conexão entre a academia e a sociedade**.

Coleta dos dados

- Para coletar os dados do Instagram da PRPG, utilizamos a **API Graph da Meta (Facebook/Instagram)**.

- Esta é a interface oficial de dados da Meta, que nos permite acessar as informações **diretamente da fonte**. Isso **assegura a confiabilidade, autenticidade e precisão dos dados**.

- Através desta API, conseguimos coletar dados estruturados sobre as **publicações** (legenda do post, data, tipo de mídia), suas **métricas de engajamento** (como curtidas, comentários e compartilhamentos) e o **conteúdo dos comentários** realizados nas postagens.

Meta

Graph API

ACESSO A API GRAPH

Slides explicando o passo a passo para conseguir ter acesso a API graph

- Instagram API com login do facebook
- conta comercial ou criador

Passo a passo :

1. Criar conta no site da meta (pode ter qualquer email)
2. Criar um aplicativo no site da meta (na conta criada)
3. Configurar os produtos do aplicativo (mostrar os produtos que devem ser adicionados e como devemos configurá-los)
4. Gerar o token meia-vida/duradouro (mostrar o por que devemos gerar um token e como fazemos isso)

tudo feito como um passo a passo , com imagens de exemplo para orientar

Variáveis

- Nos próximos slides, apresentamos algumas das variáveis que podem ser coletadas via **Instagram API** , lembrando , com login via **Facebook** e conta **Business**.

- Essas variáveis incluem dados do usuário, das publicações, métricas de desempenho (insights) e informações sobre comentários.

- Para mais informações e documentação completa, acesse: Meta for Developers.

Dados do usuário

variável	descriçao
id	Identificador único da conta
username	Nome de usuário no Instagram
name	Nome completo do perfil
biography	Texto da biografia do perfil
followers_count	Número total de seguidores
follows_count	Quantidade de contas seguidas

variável	descriçao
media_count	Total de publicações feitas
follower_demographics	Perfil demográfico dos seguidores (idade, gênero, localização)
engaged_audience_demographics	Perfil demográfico das contas que interagiram com o conteúdo
follows_and_unfollows	Número de seguidores ganhos e perdidos no período

Dados das publicações

variável	descrição	variável	descrição
id	Identificador único da mídia	**alt_text**	Texto alternativo para acessibilidade
caption	legenda da publicação	**like_count**	Quantidade de curtidas
media_type	Tipo do conteúdo (imagem, vídeo, carrossel)	**comment_count**	Quantidade de comentários
permalink	Link da mídia	**view_count**	Número de visualizações (para vídeo ou Reel)
media_product_type	Local de publicação (Feed, Story, Reel, Anúncio)	**comments**	Lista de comentários associados
timestamp	Data e hora da publicação	**insights**	Métricas de desempenho da mídia

Insights das publicações

variável	descrição
impressions	Total de vezes que a publicação foi exibida
reach	Total de contas únicas que viram a publicação
saved	Quantidade de vezes que foi salva
shares	Número de compartilhamentos
follows	Novos seguidores originados pela publicação
profile_visits	Visitas ao perfil a partir da mídia

variável	descrição
total_interactions	Soma de curtidas, comentários, shares e salvamentos
views	Visualizações únicas (vídeo/reel)
ig_reels_avg_watch_time	Tempo médio assistido (Reels)
ig_reels_video_view_total_time	Tempo total assistido (Reels)
ig_reels_aggregated_all_plays_count	Total de reproduções (Reels)
clips_replay_count	Replays de clipes (Reels)

Comentários nas publicações

variável	descrição
id	Identificador do comentário
text	Texto do comentário
timestamp	Data/hora
like_count	Curtidas no comentário
replies	Respostas ao comentário

Variáveis escolhidas

Para cumprir os objetivos do nosso trabalho, selecionamos as variáveis que julgamos mais relevantes, listadas na tabela a seguir.

Variáveis escolhidas

Dados das Publicações

variável	descriçao
id	Identificador da publicação
caption	legenda da publicação
timestamp	Data e hora da publicação
comment_count	Quantidade de comentários
like_count	Quantidade de curtidas
media_type	Tipo da mídia (imagem, vídeo, carrossel)
url_midia	URL da mídia

Insights das publicações

variável	descriçao
reach	Total de contas únicas que viram a publicação
saved	Quantidade de vezes que foi salva

Comentários e respostas

variável	descriçao
id	identificador do comentário
text	Texto do comentário
timestamp	Data e hora do comentario
like_count	Quantidade de curtidas no comentário

obs : 'respostas' são as respostas de um comentário , ou seja , um comentário de comentário .

Token de acesso

- Para coletar os dados dessas variáveis, optamos pela criação de um token de longa duração, pois ele mantém o acesso ativo por mais tempo, reduzindo a necessidade de renovação frequente.
- Além disso, para coletar essas variáveis específicas, o token precisa conter as seguintes permissões :

instagram_basic

pages_show_list

pages_read_engagement

instagram_manage_insights

instagram_manage_comments

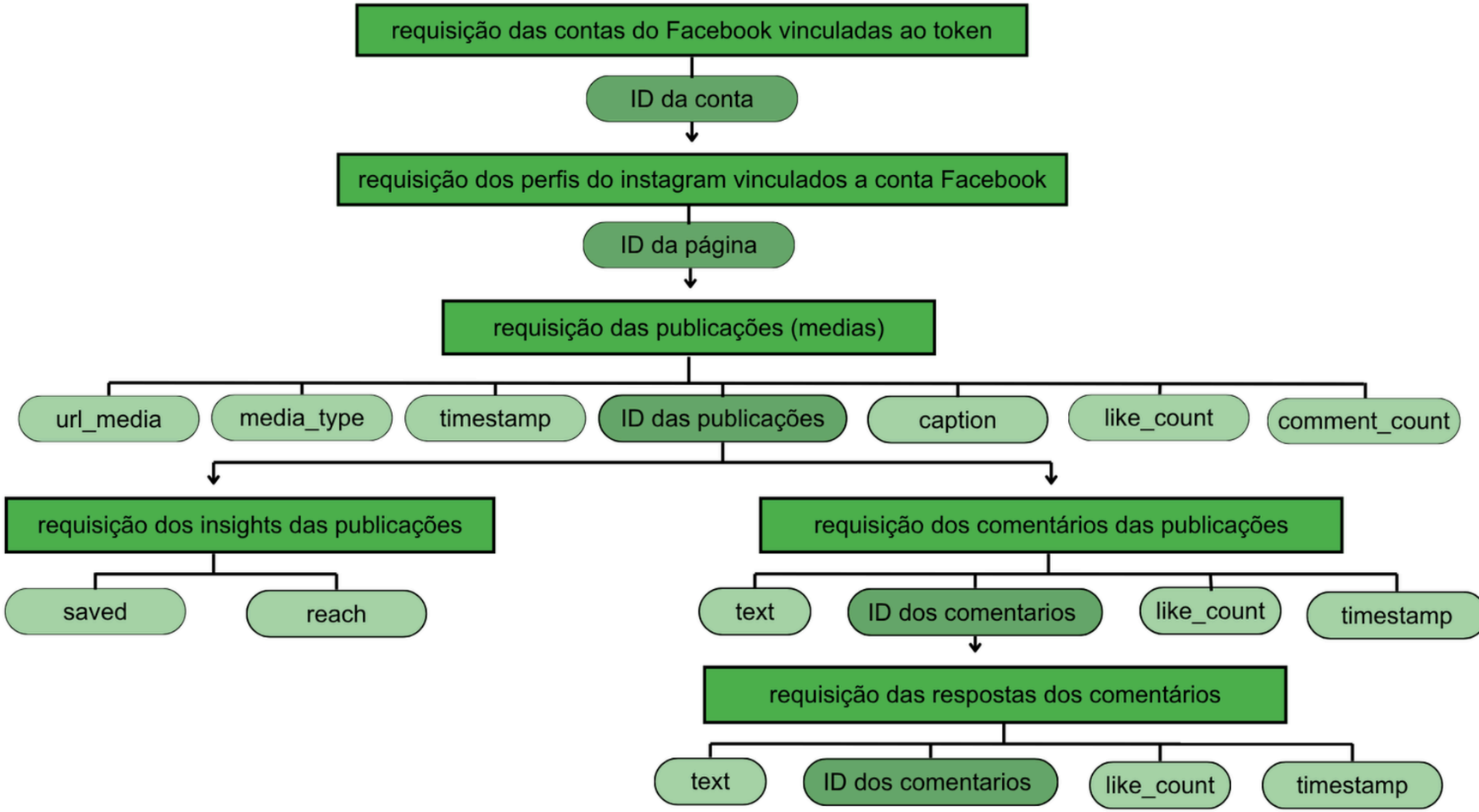
Requisições

A seguir, apresentamos como são feitas as requisições à **API Graph** para coletar os dados das variáveis selecionadas anteriormente. As requisições são demonstradas tanto pela **URL padrão** da API quanto utilizando a biblioteca **httr2** do **R**, que foi a **linguagem utilizada para realizar as requisições nesse trabalho** .

Antes de apresentar o código das requisições, mostramos a seguir um diagrama representando o fluxo geral de coleta de dados. Esse esquema ilustra a sequência de chamadas feitas à API Graph, desde a obtenção das páginas vinculadas ao token até a coleta dos comentários e respostas.

Requisições

Primeiro, é feita a requisição das páginas associadas ao token de acesso, de onde obtemos o ID da conta. Em seguida, com esse ID, são coletados os dados das publicações (media). A partir do ID de cada publicação, realizam-se as requisições dos insights e dos comentários . Por fim, com o ID de cada comentário, é possível coletar as respostas de cada comentário.



Requisições

Antes de apresentar os exemplos das requisições, separei algumas observações importantes sobre o funcionamento da API

- As respostas retornadas pela API Graph vêm no formato **JSON**, padrão utilizado para a troca de dados entre cliente e servidor.
  No R, a função **resp_body_json()** da biblioteca httr2 é utilizada para **converter a resposta JSON em um objeto R estruturado (listas)**, o que facilita o acesso e o tratamento das informações retornadas.

- Quando uma conta possui muitas publicações, comentários ou respostas, a API não retorna todos os dados de uma vez. Ela divide os resultados em **páginas** para evitar sobrecarga. Cada resposta contém um campo chamado **"paging" → "next"**, que já traz a **URL completa da próxima página de dados**, permitindo repetir a requisição até que não existam mais páginas disponíveis.

Requisição das contas do Facebook vinculadas ao token

Primeira etapa: obter as contas do Facebook associadas ao token de acesso.
Cada conta retorna um ID, que será usado para requisitar os perfis do Instagram.

URL padrão:

```
https://graph.facebook.com/v23.0/me/accounts?access_token={seu_token}
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_conta <- request("https://graph.facebook.com/v23.0/me/accounts") %>%
  req_url_query(
  access_token = {seu_token}) %>%
  req_perform() %>%
  resp_body_json()
```

Requisição dos perfis do instagram vinculados a conta Facebook

Com o **ID da conta do Facebook**, obtemos os perfis do Instagram conectados.
Cada perfil fornece o **ID** que será usado para coletar as publicações .

URL padrão:

```
https://graph.facebook.com/v23.0/{id_da_conta}?
fields=instagram_business_account&access_token={seu_token}
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_paginas <- request("https://graph.facebook.com/v23.0/{id_da_conta}") %>%
  req_url_query(
    fields = "instagram_business_account" ,
    access_token = {seu_token}
  ) %>%
  req_perform() %>%
  resp_body_json()
```

Requisição das publicações (medias)

A partir do **ID do perfil do Instagram**, coletamos as publicações (media) associadas.
Alem dos dados solicitados , cada publicação fornece o **ID exclusivo** que será usado para coletar os **insights** e **comentarios**

URL padrão:

```
"https://graph.facebook.com/v23.0/{id_da_pagina_ig}/media?
fields=id,caption,timestamp,comments_count,like_count,media_type,media_url
&access_token={seu_token}"
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_publicacoes ←
request("https://graph.facebook.com/v23.0/{id_da_pagina_ig}/media") %>%
    req_url_query(
      fields = "id,caption,timestamp,comments_count,like_count,media_type,media_url",
      access_token = {seu_token}
    ) %>%
    req_perform() %>%
    resp_body_json()
```

Requisição dos insights das publicações

Para cada **ID da publicação**, coletamos os insights.

URL padrão:

```
"https://graph.facebook.com/v23.0/{id_publicacao}/insights?
metric=reach,saved&access_token={seu_token}
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_insights <-
request("https://graph.facebook.com/v23.0/{id_publicacao}/insights") %>%
  req_url_query(
    metric = "reach,saved" ,
    access_token = {seu_token}
  ) %>%
  req_perform() %>%
  resp_body_json()
```

Requisição dos comentários das publicações

Para cada **ID da publicação**, coletamos os comentários.

URL padrão:

```
"https://graph.facebook.com/v23.0/{id_publicacao}/comments?
fields=text,timestamp,like_count&access_token={seu_token}
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_comentarios <-
request("https://graph.facebook.com/v23.0/{id_publicacao}/comments") %>%
  req_url_query(
    fields = "text,timestamp,like_count",
    access_token = {seu_token}
  ) %>%
  req_perform() %>%
  resp_body_json()
```

Requisição das respostas dos comentários

Para cada **ID de comentário**, coletamos as respostas do comentarios.

URL padrão:

```
"https://graph.facebook.com/v23.0/{id_comentario}/replies?
fields=text,timestamp,like_count&access_token={seu_token}
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_respostas <-
request("https://graph.facebook.com/v23.0/{id_comentario}/replies") %>%
  req_url_query(
    fields = "text,timestamp,like_count",
    access_token = {seu_token}
  ) %>%
  req_perform() %>%
  resp_body_json()
```

Requisição das respostas dos comentários

Para cada **ID de comentário**, coletamos as respostas do comentarios.

URL padrão:

```
"https://graph.facebook.com/v23.0/{id_comentario}/replies?
fields=text,timestamp,like_count&access_token={seu_token}
```

Exemplo em R com httr2:

```
library(httr2)
library(magrittr)

resposta_respostas <-
request("https://graph.facebook.com/v23.0/{id_comentario}/replies") %>%
  req_url_query(
    fields = "text,timestamp,like_count",
    access_token = {seu_token}
  ) %>%
  req_perform() %>%
  resp_body_json()
```

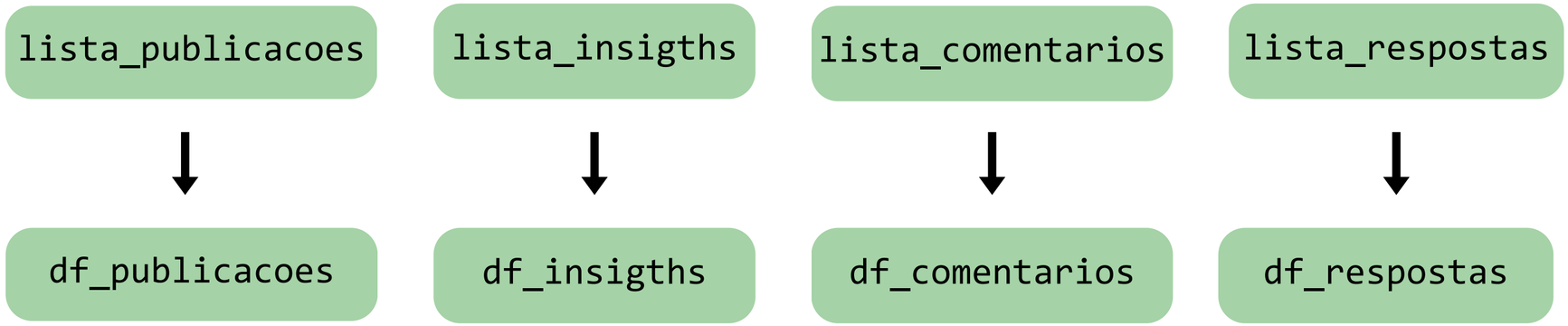
Organização dos dados

- Em todo processo de organização dos dados utilizamos a linguagem R

- As respostas das requisições (incluindo todas as páginas retornadas pela paginação) foram guardadas em listas :

`lista_publicacoes` `lista_insigths` `lista_comentarios` `lista_respostas`

- Utilizando os pacotes **purr** , **magrittr e tibble** tranformamos cada lista em um data frame com suas respectivas variáveis .



Organização dos dados

As colunas de cada data frame ficaram organizadas da seguinte maneira:

df_publicacoes

id_publicacao	caption	timestamp	comments_count	like_count	media_type	media_url

df_insights :

id_publicacao	reach	saved

df_comentarios :

id_publicacao	id_comentario	text	like_count	timestamp

df_respostas :

id_comentario	id_resposta	text	like_count	timestamp

Organização dos dados

Depois, juntamos os data frames df_publicacoes e df_insights pela coluna **id_publicacao** , formando o df dados_publicacoes

```
library(dplyr)

dados_publicacoes <- full_join(df_publicacoes, df_insights, by = "id_publicacao")
```

Organização dos dados

- A partir do df_respostas , agrupamos os **id_resposta** que pertencem ao mesmo **id_comentario** em listas. Ou seja, juntamos todos os **id_resposta** de cada comentário .
- Em seguida, unimos esse resultado ao df_comentarios pelo **id_comentario**, formando o dados_comentarios , onde cada linha representa um comentário e os ids das respostas associadas ficam listadas na coluna **ids_respostas**.

```
library(dplyr)

respostas_ids <- df_respostas %>% group_by(id_comentario) %>%
summarise(ids_respostas = list(id_resposta))

dados_comentarios <- df_comentarios %>% left_join(respostas_ids, by =
"id_comentario")
```

Organização dos dados

- As colunas de cada data frame ficaram organizadas da seguinte maneira:

dados_publicacoes

id_publicacao	caption	timestamp	comments_count	like_count	media_type	media_url	reach	saved

dados_comentarios

id_publicacao	id_comentario	text	like_count	timestamp	ids_respostas

Verificaçao e limitações

- Após a coleta dos dados, foi realizada uma verificação geral para avaliar a consistência das informações obtidas pela API, com foco principal em verificar se as publicações e os comentários foram coletados corretamente. Durante essa etapa, foram observadas algumas diferenças entre os valores retornados e os exibidos diretamente no perfil, o que levou à identificação de certas limitações da API Graph.

Publicações

- Primeiro, observamos a quantidade de publicações coletadas pela API e a quantidade total de publicações exibidas no perfil.

- Foi identificada uma diferença: a API havia coletado menos publicações do que o número total mostrado no perfil.

- Ao analisar as publicações não coletadas, verificamos que se tratavam de posts em que o perfil da PRPG havia sido convidado a colaborar, e não o autor principal da publicação.

Comentários

- Durante a verificação dos comentários, observamos uma diferença entre o total de comentários coletados pela API (considerando tanto os comentários principais quanto as respostas) e o total indicado pela variável **qtd_comentarios** presente no **df_publicações**.

- Para investigar essa diferença, identificamos as publicações em que a API não retornou nenhum comentário, comparando todos os **id_publicacao** com aqueles presentes em **df_comentarios**.

- Em seguida, filtramos as publicações com qtd_comentarios == 0 para comparar com os casos em que não houve retorno de comentários pela API.

Comentários

- Essa análise revelou que existiam publicações nas quais a variável qtd_comentarios indicava comentários, mas a API não retornava nenhum.

- Ao inspecionar esses casos manualmente, observou-se que o Instagram contabilizava o comentário, mas nenhum aparecia ao abrir a seção de comentários, indicando uma possível inconsistência entre os dados exibidos e os retornados pela API.