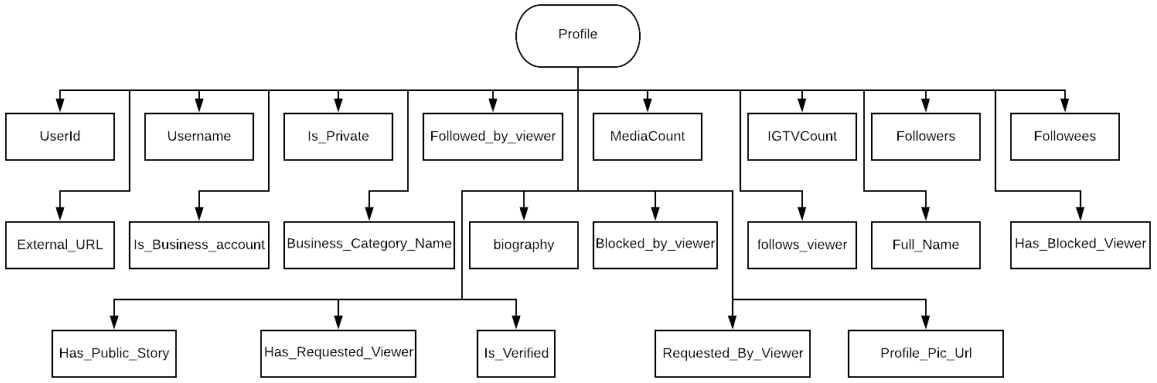
Documentação do Coletor de dados do Instagram utilizando Python e Instaloader

1- Sumário de Atributos

Nessa seção apresentamos os principais objetos do Instagram e quais são seus atributos possíveis de extrair utilizando o Instaloader.

1.1 Profile

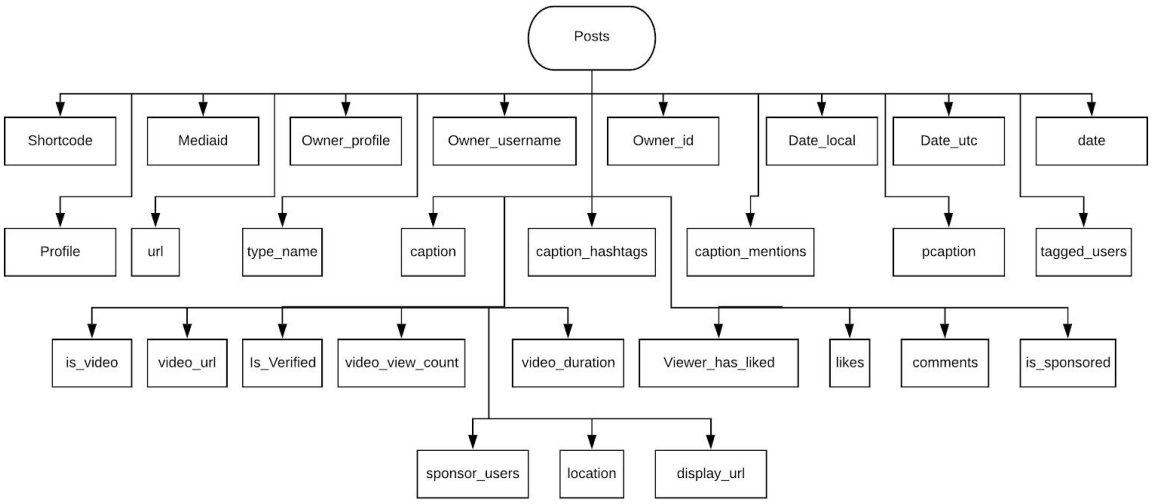


- UserId (int): Retorna o id do profile.
- Username (str): Retorna o nome de usuário da conta analisada.
- IsPrivate (bool): Retorna “True” caso a conta analisada seja privada.
- Followed_by_viewer (bool): Retorna “True” caso a conta pesquisada seja seguida pelo usuário que utiliza o coletor.
- Mediacount (int): Retorna o número de posts que a conta pesquisada possui no total.
- IgtvCount (int): Retorna o número de vídeos IGTV que a conta possui no total.
- Followers (int): Retorna o número de seguidores que a conta pesquisada possui.
- Followees (int): Retorna o número de usuários que a conta pesquisada segue.
- External_URL (str): Caso a conta possua um link externo de acesso retorna esse URL.
- Is_Business_Account (bool): Retorna “True” caso a conta pesquisada seja uma conta comercial.
- Business_Category_Name (str): Retorna a qual categoria comercial a conta pertence, caso o parâmetro acima tenha sido “True”.
- Biography (str): Retorna a biografia do perfil.

- Blocked_by_viewer (bool): Retorna “True” caso a conta pesquisada tenha sido bloqueada pelo usuário do coletor.
- Follows_Viewer (bool): Retorna “True” caso a conta pesquisada siga o usuário do coletor.
- Full_Name (str): Retorna o nome completo do perfil.
- Has_Blocked_Viewer (bool): Retorna “True” caso a conta pesquisada tenha bloqueado o usuário do coletor.
- Has_Public_Story (bool): Retorna “True” caso a conta pesquisada possua story’s públicos.
- Has_Requested_Viewer (bool): Retorna “True” caso a conta pesquisada tenha requisitado seguir o perfil do usuário do coletor.
- Is_Verified (bool): Retorna “True” caso o perfil seja verificado.
- Requested_by_viewer (bool): Retorna “True” caso a conta pesquisada tenha sido requisitada a autorização para seguir pelo usuário do coletor.
- Profile_Pic_URL (str): Retorna a URL da foto de perfil da conta.

Os parâmetros relacionados à “Profile” com os quais trabalhamos no nosso coletor foram: **Username**; **Is_Business_Account;** e **Business_Category_Name**. Todos esses são relacionados às definições gerais do arquivo csv que o script cria. **Username** é um parâmetro extremamente necessário porque serve de identificação para o usuário que estamos tratando e, os outros dois parâmetros relacionados a **Business** são utilizados para definição de quais perfis comerciais seriam mais convenientes de serem pesquisados, já que esses podem conter comentários de pessoas que se interessam pelo assunto e auxiliar no aprofundamento de nossas pesquisas.

1.2 Posts



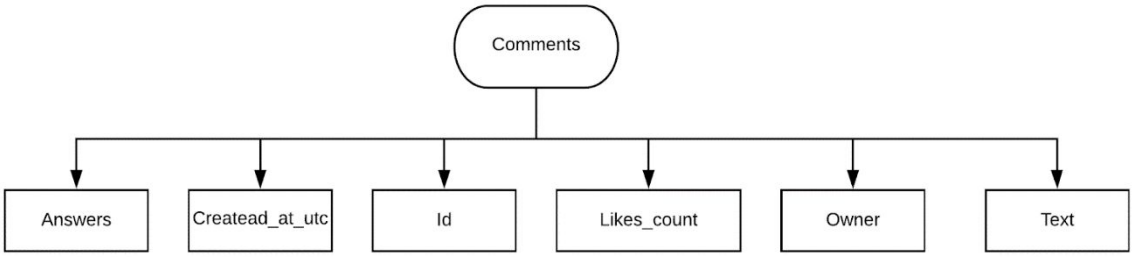
- Shortcode (str): Média shortcode do url que será utilizado para acessar o post em si, de maneira simplificada é a parcela em código do link que direciona ao post.
- Mediaid (int): Representação decimal da média shortcode.

- Owner_profile (profile): Retorna a instância para o profile do usuário que enviou o post.
- Owner_username (str): Retorna em letras minúsculas o nome de usuário do dono do post.
- OwnerId (int): Retorna o id do profile.
- Date_local (datetime): Retorna a data e o horário em que o post foi criado (horário local).
- Date_utc (datetime): Retorna a data e o horário em que o post foi criado (UTC).
- Date (datetime): Sinônimo a date_utc.
- Profile (datetime): Sinônimo ao owner_username.
- Url (str): Retorna a URL da foto ou vídeo do post.
- Typename (str): Retorna o tipo de mídia do post.
- Caption (str): Retorna o texto do post.
- Caption_hashtags (list[str]): Retorna uma lista de todas as hashtags utilizadas no post.
- Caption_mentions (list[str]): Retorna uma lista com todas as menções presentes no post.
- Pcaption (str): Retorna uma versão imprimível da caption.
- Tagged_users (list[str]): Retorna uma lista com todos os usuários que receberam “tag” no post. (Pessoas que o usuário marca como presentes na foto)
- Is_video (bool): Retorna “True” caso o post contenha vídeo.
- Video_url (str): Retorna caso exista, a url do vídeo do post
- Video_view_count (int): Retorna o número de visualizações que o vídeo teve.
- Video_duration (int): Retorna à duração do vídeo.
- Viewer_has_liked (bool): Retorna “True” caso o usuário do coletor tenha curtido o post.
- Likes (int): Retorna o número de likes do post.
- Comments (int): Retorna o número de comentários do post.
- Is_sponsored (bool): Retorna “True” caso o post seja patrocinado.
- Sponsor_users (list[profiles]): Caso o post seja patrocinado retorna uma lista com todos usuários que patrocinam esse post.
- Location (PostLocation): Retorna caso o dono do post tenha colocado a localização.
- Display_url (str): Retorna a url da thumbnail do vídeo ou imagem do post.

Os parâmetros utilizados no nosso coletor relacionados ao post são novamente utilizados de forma direta para preenchimento do CSV, são eles: **Owner_username; Date**; **Caption**; **Caption_hashtags**; **Likes**; **Comments**; **Tagged_users** e; **Location**. Todas essas propriedades são utilizadas para auxiliar na análise dos dados coletados, **Date** e **Location**, por exemplo, são utilizadas para se ter uma análise temporal e regional desses posts relacionados ao assunto, assim de acordo com certos marcos temporais e diferenças regionais podemos retirar quais consequências tiveram na rede social. Já **Owner_username** é utilizado para a coleta do sexo do usuário e, o

restante das propriedades são armazenados no arquivo CSV para análise posterior da sua relevância.

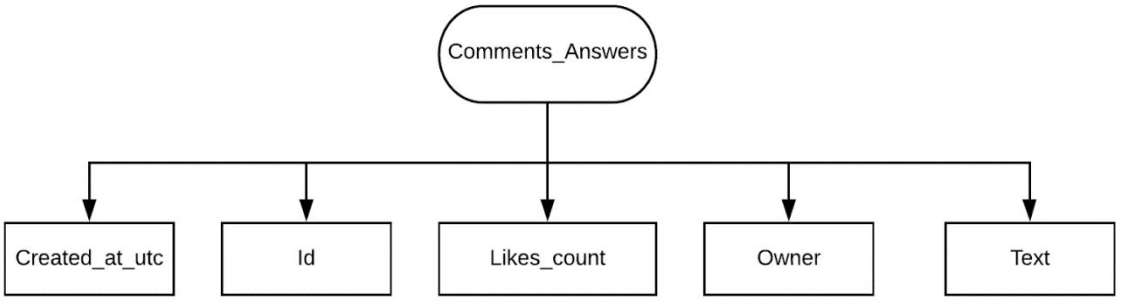
1.3 Comentários



- Answers (PostCommentAnswers): Retorna a instância com todas as respostas a esse comentário.
- Created_at_utc (datetime): Retorna a data e o horário em que o comentário foi postado.
- Id (int): Retorna o id do comentário.
- Likes_count (int): Retorna o número de likes que o comentário recebeu.
- Owner (profile): Retorna a instância com o profile do usuário que fez o comentário.
- Text (str): Retorna o texto do comentário.

O conteúdo utilizado dos comentários serve somente para fim de filtragem mais aprofundada nos posts, o **Text** dos comentários do post que coletamos são analisados buscando palavras-chave e caso alguma dessas palavras-chave seja encontrada, o post é tratado com maior relevância. Por exemplo, se a palavra emprego é encontrada junto de corona em um comentário de um post, então o post terá uma flat que indicará isso e desta forma será tratado com maior importância.

1.4 Respostas de Comentários



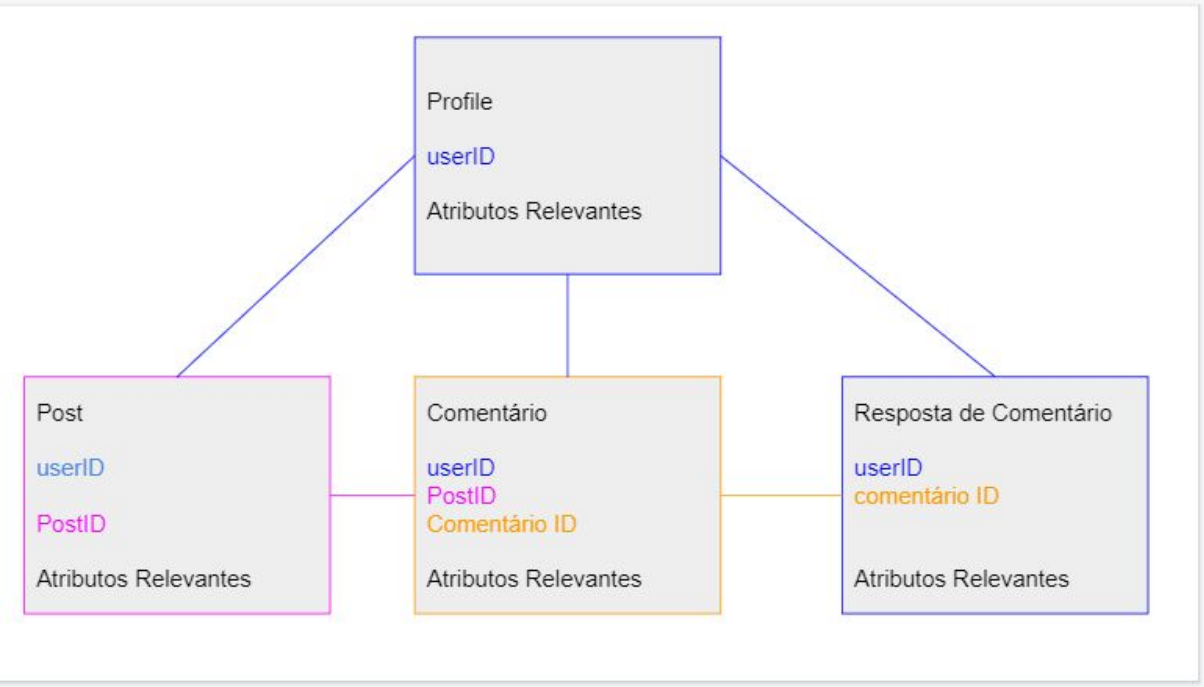
- Created_at_utc (datetime): Retorna a data e o horário em que o comentário foi postado.
- Id (int): Retorna o id do comentário.

- Likes_count (int): Retorna o número de likes que o comentário recebeu.
- Owner (profile): Retorna a instância com o profile do usuário que fez o comentário.
- Text (str): Retorna o texto do comentário.

No atual momento não é utilizada nenhuma propriedade de resposta de comentários no script.

2- Coleta de dados

Como descrito acima temos 4 instâncias principais no instagram e essas instâncias se relacionam de muitos para 1, por exemplo, 1 post tem apenas um profile de origem mas um profile possui muitos posts. Dessa forma, para otimizar a coleta de dados, propomos a coleta de cada instância em um arquivo CSV distinto que irá conter os atributos considerados relevantes para as análises posteriores. A forma é apresentada abaixo:



Os atributos relevantes de cada instância ainda não foram definidos.

3- Instaloader - Funcionamento do Coletor

O coletor tem suas funções divididas em quatro scripts, são eles:

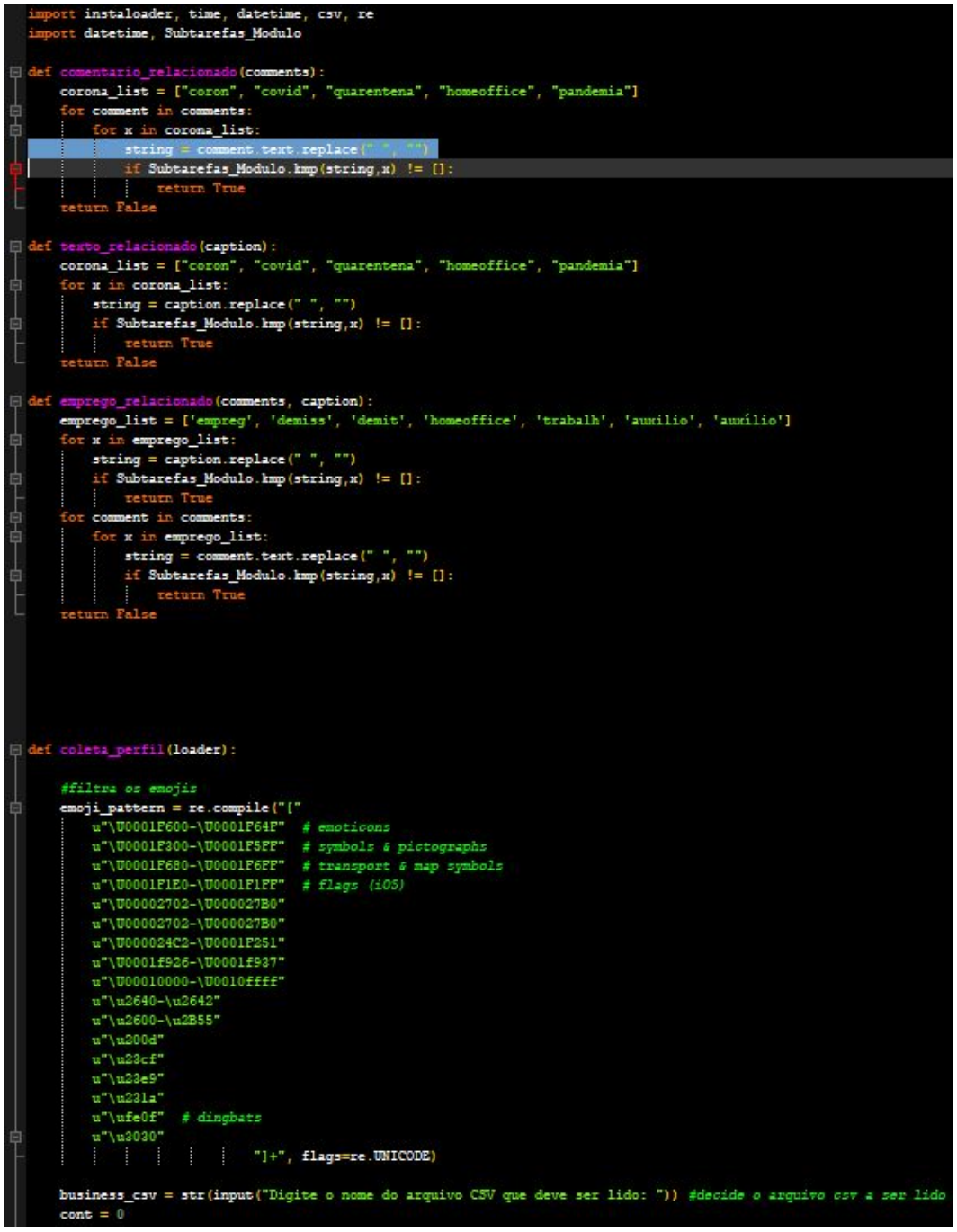
Baixar_Hashtags - Faz a função central para o restante do coletor, é nele onde o login do usuário é solicitado e onde é escolhido qual modo do coletor será utilizado (Hashtag “h” ou Perfil “p”).

```
import instaloader, time, datetime, csv
import Hashtag_Modulo, Perfil_Modulo
from datetime import date

if __name__ == '__main__':

    # incialização e login---------
    loader = instaloader.Instaloader()
    user_login = str(input("Instagram login: "))
    loader.interactive_login(user_login)
    #-----------------------------
    modo = str(input("Qual modo você deseja utilizar (p ou h): "))
    if modo == "h":
        Hashtag_Modulo.coleta_hashtag(loader) #função de coleta da tag
    if modo == "p":
        Perfil_Modulo.coleta_perfil(loader) #função de coleta do perfil
```

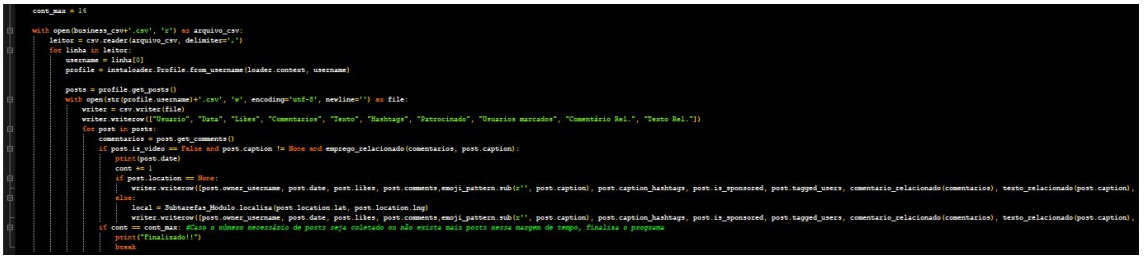
Subtarefas_Modulo - Tem em seu conteúdo duas tarefas, utilizar do processo KMD, que faz a procura de uma palavra em uma string, processo utilizado tanto em hashtag quanto em perfis para procurar palavras-chaves e também tem como tarefa definir os parâmetros de localidade do perfil caso haja os dados necessários.

```
import re
from geopy.geocoders import Nominatim

def kmp(t, p):
    """return all matching positions of p in t"""
    next = [0]
    j = 0
    for i in range(1, len(p)):
        while j > 0 and p[j] != p[i]:
            j = next[j - 1]
        if p[j] == p[i]:
            j += 1
        next.append(j)
    # the search part and build part is almost identical.
    ans = []
    j = 0
    for i in range(len(t)):
        while j > 0 and t[i] != p[j]:
            j = next[j - 1]
        if t[i] == p[j]:
            j += 1
        if j == len(p):
            ans.append(i - (j - 1))
            j = next[j - 1]
    return ans

def localiza(lat, long):
    geolocator = Nominatim(user_agent="CDA UFMG")
    loc = str(lat)+", "+str(long)
    location = geolocator.reverse(loc)
    dados = []
    if "state" in location.raw["address"].keys():
        dados.append(location.raw["address"]["state"])
    else:
        dados.append("None")
    if "city" in location.raw["address"].keys():
        dados.append(location.raw["address"]["city"])
    else:
        dados.append("None")
    if "region" in location.raw["address"].keys():
        dados.append(location.raw["address"]["region"])
    else:
        dados.append("None")
    return dados
```

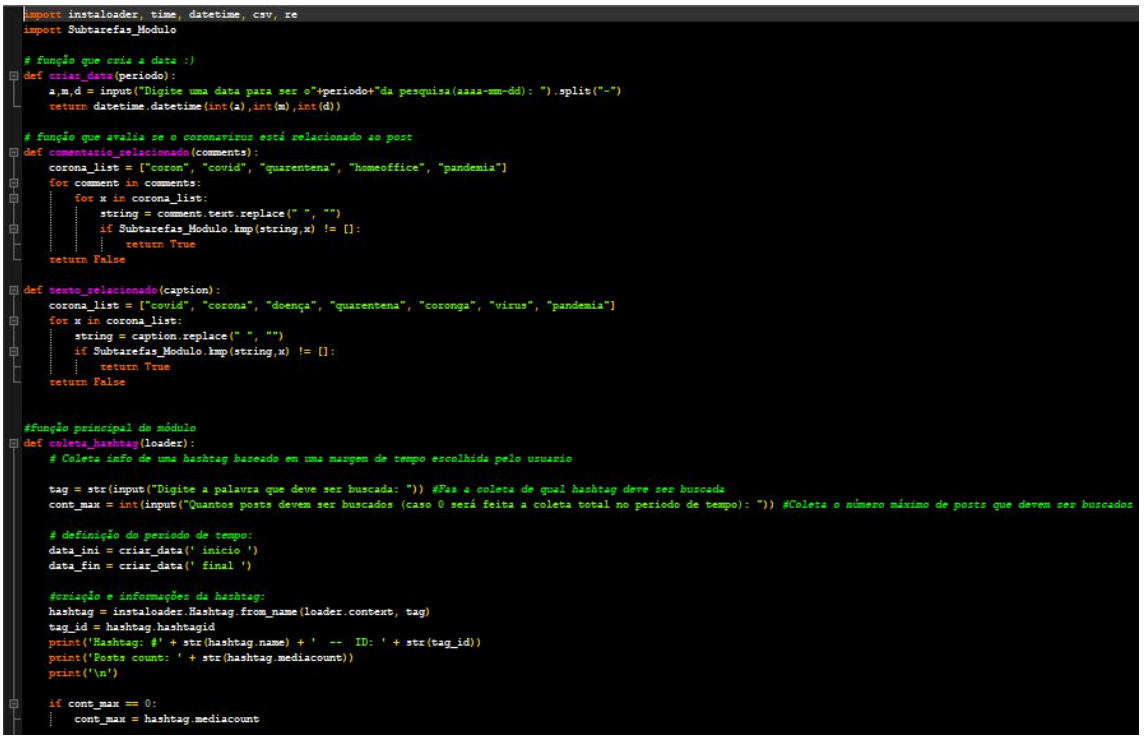
Perfil_Modulo - Faz a coleta dos dados do csv a partir dos perfis coletados como perfis comerciais anteriormente pelo Hashtag_Modulo, buscando posts e comentários que sejam relacionados ao tema de pesquisa, economia e covid.

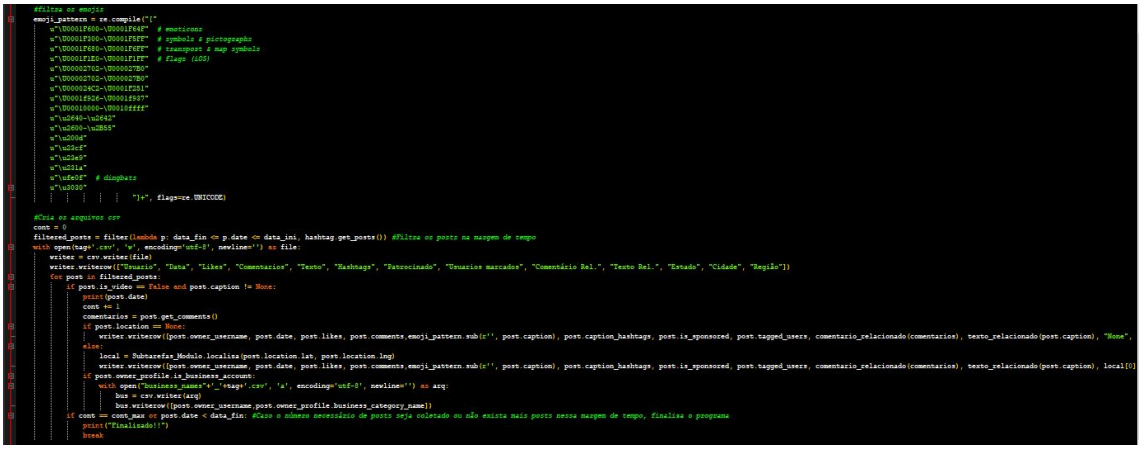


```
import instaloader, time, datetime, csv, re
import datetime, Subtarefas_Modulo

def comentario_relacionado(comments):
    corona_list = ["coron", "covid", "quarentena", "homeoffice", "pandemia"]
    for comment in comments:
        for x in corona_list:
            string = comment.text.replace(" ", "")
            if Subtarefas_Modulo.kmp(string,x) != []:
                return True
    return False

def texto_relacionado(caption):
    corona_list = ["coron", "covid", "quarentena", "homeoffice", "pandemia"]
    for x in corona_list:
        string = caption.replace(" ", "")
        if Subtarefas_Modulo.kmp(string,x) != []:
            return True
    return False

def emprego_relacionado(comments, caption):
    emprego_list = ['empreg', 'demiss', 'demit', 'homeoffice', 'trabalh', 'auxilio', 'auxílio']
    for x in emprego_list:
        string = caption.replace(" ", "")
        if Subtarefas_Modulo.kmp(string,x) != []:
            return True
    for comment in comments:
        for x in emprego_list:
            string = comment.text.replace(" ", "")
            if Subtarefas_Modulo.kmp(string,x) != []:
                return True
    return False




def coleta_perfil(loader):

    #filtra os emojis
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U00002702-\U000027B0"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"  # dingbats
        u"\u3030"
                          "]+", flags=re.UNICODE)

    business_csv = str(input("Digite o nome do arquivo CSV que deve ser lido: ")) #decide o arquivo csv a ser lido
    cont = 0
```



Hashtag_Modulo - Faz a coleta dos dados do csv a partir de uma busca feita por palavras chaves, dando maior enfoque nos posts que tenham relação com o tema da pesquisa, economia e covid.





Tanto para perfil quanto para hashtag são coletados os seguintes atributos:

```
["Usuario", "Data", "Likes", "Comentarios", "Texto", "Hashtags", "Patrocinado", "Usuarios marcados", "Comentário Rel.", "Texto Rel.", "Estado", "Cidade", "Região"]
```

O fluxo da coleta percorre as seguintes etapas:

1- O usuário do coletor deve digitar seu login.

```
Instagram login:
```

2- Digitar sua senha.

```
Enter Instagram password for teste_123412:
```

3- Escolher qual modo do script será utilizado, “p” para perfil e “h” para hashtag.

```
Qual modo você deseja utilizar (p ou h):
```

Caso seja escolhido perfil:

4- Escolher qual dos arquivos de perfis comerciais já coletados por hashtag deve ser lido.

```
Digite o nome do arquivo CSV que deve ser lido:
```

5- Digitar qual o número de posts que deve ser coletado dos perfis.

```
Digite o número de posts que devem ser coletados do perfil:
```

6 - Dessa forma será coletada a quantidade de posts dos perfis.

Caso seja escolhido hashtag:

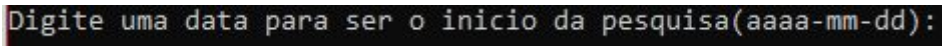
4 - Digitar qual palavra-chave será buscada.

```
Digite a palavra que deve ser buscada:
```

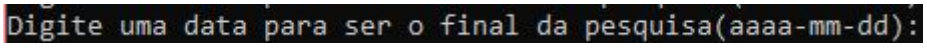
5 - Digitar quantos posts devem ser coletados, caso seja colocado 0 todos os posts dentro da margem de tempo serão buscados.

```
Quantos posts devem ser buscados (caso 0 será feita a coleta total no periodo de tempo):
```

6 - Digitar a data na qual os posts devem começar a ser coletados, essa deve ser a data mais próxima a atual.



7 - Digitar a data final para a coleta, deve ser a data mais antiga.



8 - Desta forma serão coletados os posts e seus respectivos atributos dentro da margem de tempo e da quantidade solicitada pelo usuário do coletor.

4- Instaloader – Análise de Limitações

4.1 Coleta 1:

Detalhes e informações a respeito da coleta:

A análise de estresse demonstra muito bem como as paradas que são realizadas pelo próprio algoritmo representam um grande limitador para o projeto, visto que fazem com que o tempo demandado para a coleta se alongue de maneira significativa:

- · Hashtag coletada: “desemprego”
- · Funcionalidades: filtragem por tempo, busca de palavras chave relacionadas ao coronavírus nos comentários e legendas.
- · Quantidade de posts coletados: 1.478
- · Horário de início da coleta: 15hr 03min 46seg
- · Horário do final da coleta: 17hr 50min 43seg
- · Tempo total de coleta: 10.017 segundos (2hr 46min 57seg)
- · Tempo médio de coleta por post: 6.78 segundos
- · Quantidade de vezes que o instaloader parou: 30
- · Quantidade de tempo que o instaloader ficou parado: 6204 segundos (1hr 43min 24seg)
- · Média de tempo necessário para ser possível coletar a base de dados inteira dessa hashtag (120383 posts): 816196 segundos (9 dias 10hr 43min 16seg)

Essa coleta, entretanto, é mais antiga e os resultados atuais são diferentes e, em relação ao tempo de demora da coleta ainda mais desanimadores.

Resultados interessantes:

Esses resultados demonstram que a coleta acima, que demorou cerca de 2 horas e 46 minutos, conseguiu coletar uma quantidade pequena de posts realmente ligados à pesquisa comparado ao tempo que se estendeu:

- · Quantidade de posts com comentários relacionados ao covid: 47/1478
- · Quantidade de posts com a legenda relacionada ao covid: 362/1478

4.2 Coleta Atualmente:

É importante salientar que a forma de coleta atual vem demonstrando um desafio um pouco maior para o coletor.

Na coleta, há um erro recorrente, o qual representa uma das exceções do coletor (descritas na documentação do instaloader).

Para resolver essa questão basta realizar uma atualização. Porém, após feito, o coletor fica ainda mais limitado, parando de 20 em 20 posts (antes parava de 100 em 100) e com um tempo médio de parada de 500 segundos (antes era em média 400/450), o que atrapalha ainda mais o tempo.

- Hashtag coletada: coronavirusbrazil
- Funcionalidades: filtragem por tempo, busca de palavras chave relacionadas ao coronavírus nos comentários e legendas (o que seria meio desnecessário nesse caso, visto que a própria hashtag já é relacionada ao corona) e busca e identificação da localidade dos posts.
- Quantidade de posts coletados: 312
- Horário de início da coleta: 14:05:31
- Horário do final da coleta: 17:51:51
- Tempo total de coleta: 13.580 segundos (3:46:20)
- Tempo médio de coleta por post: 43.52 segundos
- Quantidade de vezes que o instaloader parou: 29
- Quantidade de tempo que o instaloader ficou parado: 11569 segundos (3hr 12min 49seg) -> esse tempo, porém, é maior do que poderia ser, pois estima-se que alguns computadores recorrentemente retornam um erro que só é solucionado quando aperta-se uma tecla aleatória do teclado. Nesse cenário, gerou-se uma pausa de 63 minutos a mais do que o observado em situações nas quais não ocorre o erro (cerca de 3780 segundos). Assim, o tempo de parada do Instaloader em si foi de 7789 segundos (2:9:49)
- Média de tempo necessário para na situação em que ocorre o erro, seja possível coletar a base de dados inteira dessa hashtag (574.570 posts): 17.914.798 segundos (207 dias 8hr 19min 58seg) -> para esse cálculo foi desconsiderado os 63 minutos que o programa ficou parado no computador somente por não ter sido apertado nenhuma tecla.

De fato esses resultados são desanimadores em relação ao tempo de coleta do Instaloader. Uma ressalva, entretanto, é que a realidade de erro encontrada em alguns computadores pode não ser a mesma em todos os outros, e que todas essas análises acima foram feitas somente em uma máquina. Dessa forma, talvez seja possível que, a partir do momento em que for disponibilizado servidores, esse tempo seja reduzido de maneira massiva, tornando possível a coleta.

4.3 Usabilidade:

Tendo em vista as limitações apresentadas pelo coletor, atualmente vemos que o mesmo pode ser utilizado de duas formas diferentes:

- Coletor de curto prazo: Identifica-se que o coletor é muito eficiente para coletas diárias e/ou de curto prazo, o que permite que seja construído um banco de dados com as informações atuais e que, com o tempo e com a junção dessas coletas diárias, chega-se no objetivo de se construir uma base robusta com dados que percorram todos os meses de interesse.

- Coletor de longo prazo: É possível realizar coletas de meses anteriores, mas seria necessário mapear os possíveis erros que o coletor possa encontrar e criar rotinas de correção. Seriam necessários coletores ininterruptos para que o tempo seja razoável e pensar em formas de otimização com relação a número de requisições por post. E uma coleta paralela relacionada a contas, hashtags, temporal e outros fatores que influenciam em requisições e a política de banimento do Instagram e, por consequência, do Instaloader.

6- Análise Palavras-chave

Uma das formas de coletarmos dados do Instagram é por meio das Hashtags, ou palavras-chave. De maneira resumida, definimos uma palavra-chave de interesse (todas foram selecionadas dentro do contexto do nosso tema) e definimos também o intervalo de tempo da coleta. Dessa forma, o instaloader vai percorrer todos os posts com essa Hashtag existentes nesse intervalo de tempo selecionado, e para cada post coletado, vai nos fornecer atributos como: quantidade de comentários, owner do post, texto do post, número de curtidas, todas as contas business que fizeram posts com essa Hashtag, dentre outros.

Primeiro constatamos a necessidade de analisar a viabilidade de realizar a coleta dessa forma, pois, como já citado em outros momentos dessa documentação, tempo é um fator comprometedor, visto que a coleta funciona da seguinte maneira:

- O instaloader percorre os posts mais recentes até os mais antigos, e salva somente aqueles que entram no intervalo de tempo definido por nós. Dessa forma, o instaloader precisa fazer muitas consultas seguidas ao Instagram, e para que não tenhamos a conta banida, ele para de fazer requisições a cada N posts por um intervalo de tempo (em média 400 segundos). No contexto geral da coleta, isso resulta em um grande tempo disposto.

Com o que foi citado acima e com a presente necessidade de realizarmos a coleta das palavras-chave para fins de análise de conteúdo, ou seja, análise do que essa coleta nos retorna de informações, foi decidido pelos membros do squad que a tarefa seria feita sob as seguintes condições:

- Definir um intervalo de tempo de 1 dia, com início sendo no dia anterior ao dia da coleta e fim sendo 2 dias anteriores. Dessa forma o instaloader não precisaria percorrer vários dias de posts que não seriam coletados até chegar no intervalo de tempo definido. Para fins de contextualização, a maioria dos membros do nosso squad fez a coleta no dia 16/07, com intervalo de tempo começando no dia 16/07 e terminando no dia 15/07;
- Fazer a análise baseado nos atributos: Total de posts, quantidade de posts presentes nesse intervalo de tempo e quantidade de contas business.

Dessa forma, segue abaixo os resultados, em números, da coleta de cada palavra-chave (as tabelas estão separadas pois cada membro ficou responsável por um conjunto de palavras):

Hashtag	**Total de posts**	**Total de posts (15/07 - 16/07)**	**Total de contas business**
Demissão	17.830	75	50
Demitido	3.764	23	20
Demitida	767	3	3
Salario	90.005	170	103
Queda nas vendas	48	0	0
Corte salarial	29	0	0
Suspensão de contrato	605	17	15

Hashtag	**Total de posts**	**Total de posts coletados**	**Total de contas business**	**Legendas relacionadas**	**Comentários relacionados**
Desempregada	14.637	23	9	4	1
Desempregado	66.227	90	29	12	1
Desemprego	122.282	200	82	64	4
Emprego	944.868	593	272	45	7

Home Office	4.338.796	970	514	970	116
Redução de jornada	1004	6	6	4	0
Trabalho em casa	-	-	153	-	-

Hashtag	**Total de posts**	**Total de posts (15/07 - 16/07)**	**Total de contas business**
Auxilio emergencial	86.477	488	267
Bolsa Família	6.845	48	33
Doação	263.046	66	33
Doação	363.173	78	40
Vaquinha	79.700	134	55
Cesta Básica	**37.646**	70	40
Cesta Básica	6.542	10	3
Trabalho informal	**1.836**	5	0
MEI	**1.174.846**	18	4
Nos por Nos	**47.156**	11	6

Hashtag	**Total de posts**	**Total de posts (15/07 - 16/07)**	**Total de contas business**
Serviços essenciais	4.939	17	15
Serviços essenciais	1.520	5	4
Trabalho Doméstico	789	1	0
Trabalho Domestico	2.081	2	1
Diarista	40.082	97	63
Camelô	3.098	2	1
Camelo	39.819	40	28
Falir	1.566	0	0
Falida	6.323	1	0

Falência	8.465	39	17
Falido	5.478	3	0
Falência	5.933	40	22

Hashtag	**Total de posts**	**Total de posts (15/07 - 16/07)**	**Total de contas business**
quebrar	5.497	1	0
quebrada	383.301	3	0
quebrado	34.942	10	5
capitaldegiro	25.701	82	62
credito	406.001	+402	+353
caixafraco	0	0	0
fluxodecaixa	34.125	115	91

Podemos tirar alguns insumos de análise dessa coleta, dentre eles:

- Muitas palavras-chave retornaram uma quantidade de posts tão pequena que possivelmente não tenham impacto na nossa análise e, portanto, podem ser desconsideradas;
- Em contrapartida, por mais que algumas palavras-chave resultem em poucos posts, talvez valha a pena fazer a coleta deles, na medida em que a maioria dos posts com essa hashtag contenham conteúdos diretamente relacionados ao nosso tema (seria necessário fazer uma busca, talvez manual, para verificar isso);
- Muitas palavras-chave acentuadas retornam um maior número de posts quando estão sem os acentos (precisamos considerar que os usuários não acentuam as palavras), mas, algumas delas têm sentido ambíguo, o que nos leva a necessidade de limpar os posts que contenham a hashtag com o sentido que não nos interessa. Observe o exemplo Camelô, que quando não acentuada, vira Camelo (animal).
- Ao longo do processo, observamos que algumas palavras-chave interessantes para o nosso tema não foram selecionadas para a coleta. Dessa forma, é necessário que haja uma reavaliação dessas palavras, de forma a excluir aquelas que não forem necessárias, e adicionar as que podem trazer insumos para nossa análise.

Resultados interessantes:

- Quantidade de posts em regiões consideradas metropolitanas ou em desenvolvimento:

  Metropolitano: 81

  Não metropolitano: 20

  Sem informação: 210

- Quantidade de posts distribuídos por região:

1	Sem Informação	204
2	Região Sudeste	52
3	Região Nordeste	21
4	Região Norte	5
5	Região Centro-Oeste	13
6	Região Geográfica Intermediária de Criciúma	3
7	Região Sul	10
8	Região Geográfica Intermediária de Joinville	1
9	Região Geográfica Intermediária de Blumenau	2

5- Viabilidade da coleta por perfis

O Instagram for Business, ou Instagram para empresas são perfis no Instagram que possuem uma conta comercial voltada para um nicho de informação específico, ou seja, é um conjunto de ferramentas que permite que negócios tenham um perfil comercial, façam anúncios e acessem dados dos seguidores na rede social.

A coleta desses perfis ocorre da seguinte forma: colocamos como entrada o nome do perfil comercial no qual desejamos coletar os dados, e o instaloader percorre todos os posts desse determinado perfil pegando atributos como: quantidade de Likes, texto do post, quantidade de comentários, etc.

Para sabermos quais perfis coletar, primeiro usamos as Hashtags relacionadas à emprego (tema principal do nosso squad), fazendo uma busca por todos os posts existentes num intervalo de tempo especificado. Tal coleta nos traz várias informações, dentre elas, todos os nomes de contas de perfis comerciais, e com isso, conseguimos realizar o processo citado acima.

Em determinado momento, nos surgiu um questionamento: O que era mais viável, coletar posts por meio de uma Hashtag específica, ou coletar todos os posts de um perfil específico (fazer a coleta por perfis)? Para responder essa pergunta, é preciso levar em consideração alguns pontos importantes:

-Queremos fazer a coleta da maior quantidade possível de posts que realmente tenham relação com o nosso tema e não apenas apresentem conteúdos supérfluos, e;

-Queremos que a coleta seja viável (quantidade de posts e, principalmente, em termos de tempo);

Dessa forma, esclarecemos aqui os prós e os contras de se fazer a coleta por perfis:

Pontos negativos

-A análise de coleta de perfil pode não ser satisfatória, caso um perfil coletado tenha apenas 5% dos posts abrangendo o tema tratado, visto que, de todo modo, teríamos que processar os outros 95%, analisá-los e perceber a necessidade de limpá-los, acrescentando tempo e complexidade à nossa análise;

-Contas business podem apresentar muita propaganda e/ou não demonstrar o impacto negativo do coronavírus no empreendimento deles, visto que o intuito dessas contas tende a ser mais para usar a pandemia como um impulsionador para o seu negócio;

Pontos positivos

-O tempo de coleta total de um perfil é menor que o tempo de coleta total de uma Hashtag;

-A coleta total de um perfil cobre posts em vários intervalos temporais, enquanto o mesmo só seria possível com as Hashtags caso todos os posts associados à ela fossem coletados, o que é temporalmente inviável;

-As contas business podem tender a direcionar mais seus posts para o assunto que queremos analisar;

-Os posts das contas business que se enquadram no nosso tema tendem a ser posts mais informativos e com maior teor de seriedade;

Desse modo, concluímos que seja necessário uma reavaliação do objetivo do nosso squad, para que assim possamos decidir qual forma de coleta atende melhor o nosso propósito.

6- Ferramenta para analisar o sexo dos perfis coletados

O identificador utiliza uma base do Brasil IO que é uma iniciativa sem fins lucrativos de democratização de dados. A base contém dados do Censo de 2010 com os nomes dos entrevistados, grupo, Gênero provável, frequência masculino, Frequência Feminino, Probabilidade do Gênero.

Nome	Grupo	Gênero	Freq. Feminino	Freq. Masculino	Freq. Total	Freq. do Grupo	Probabilidade	Nomes alternativos
AABRAO	ABRAAO	M		26	26	32296	1.00	ABRAAO|ABRAHAO|ABRAO|ABRHAO|ABRRAO|ADR
AADRIANA	ADRIANA	F	94		94	568459	1.00	ABRIANA|ADRAINA|ADRIANA|ADRIANNA|ADRRIANA
AADRIANO	ADRIANO	M		53	53	338554	1.00	ABRIANO|ADRIANNO|ADRIANO|ADRYANO
AAILTON	AILTON	M		23	23	246915	1.00	AELTOM|AELTON|AHILTON|AILTHON|AILTOM|AILTOI
AALAN	ALAN	M		27	27	221601	1.00	AHLAM|AILAM|AILAN|ALAAN|ALAM|ALAN|ALANN|AYI
AALESSANDRA	ALESSANDRA	F	44		44	341637	1.00	ALECHANDRA|ALESSANDRA|ALEXANDRA|ALEXSS
AALESSANDRO	ALESSANDRO	M		25	25	238113	1.00	ALECHANDRO|ALESSANDRO|ALEXANDRO|ALEXSS
AALEX	ALEX	M		36	36	312401	1.00	ALESS|ALEX|ELEX|HALEX|LEX|UALEX
AALEXANDRE	ALEXANDRE	M		70	70	448972	1.00	ALECHANDRE|ALESSANDRE|ALEXANDRE|ALEXANI
AALINE	ALINE	F	66		66	530550	1.00	AILINE|ALEINE|ALIINE|ALINE|ALINER|ALINHE|ALINN
AALIYAH	LIA	F	28		28	68767	1.00	AELIA|AILIA|ALAI|ALAIR|ALAUIR|ALAY|ALAYR|ALIA|A
AAMANDA	AMANDA	F	89		89	467319	1.00	AMANDA|AMANDAH|AMANDHA|AMANNDA|AMMANI

O código que um integrante do Squad 4 desenvolveu compara o primeiro nome do perfil que é extraído da função Profille.full_name do Instaloader com a lista de nomes no Brasil IO e retorna a probabilidade de este nome ser de um sexo específico. O código em geral é simples e de fácil entendimento.

Características da Classificação do Brasil.IO

O Brasil.io classifica um nome em feminino ou masculino baseado na quantidade de pessoas que possuem aquele nome dado o seu sexo. Também disponibiliza um conjunto de nomes semelhantes, 30 mil nomes não possuem nomes semelhantes. Considerando apenas aqueles nomes que possuem nomes semelhantes, a estratégia é olhar para esses nomes semelhantes e contabilizar a quantidade de homens e mulheres que possuem esses nomes, e criar uma classificação alternativa, baseado no "voto de sexo" de cada nome. Os resultados não são muito animadores: se um nome "certamente" se refere ao sexo feminino, ele pode ser classificado como masculino, dependendo dos

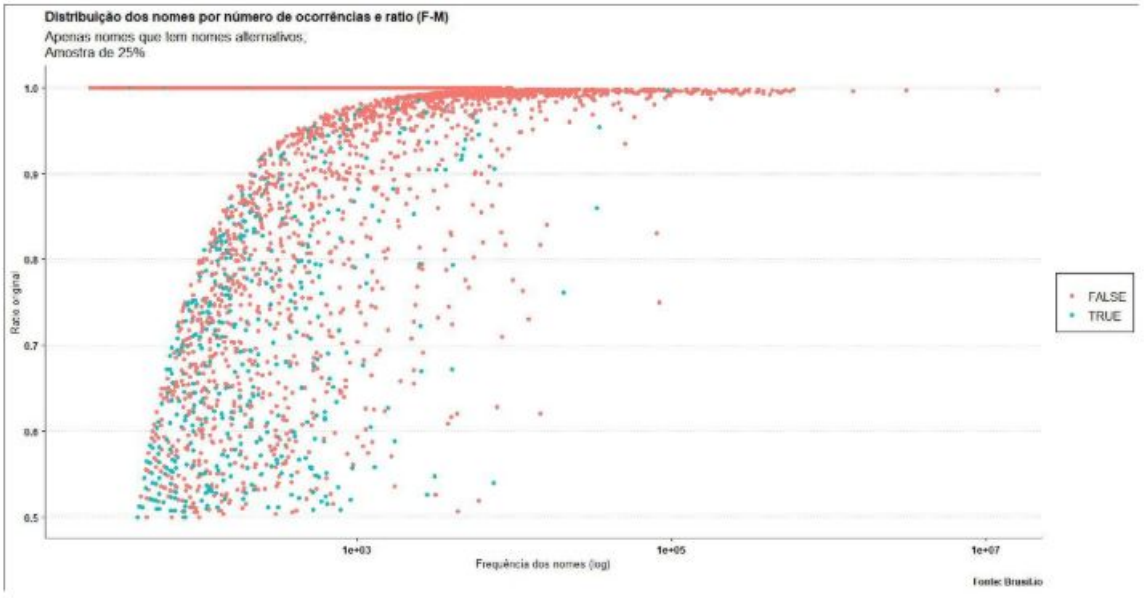
nomes que são similares; há distinção sutil entre a quantidade de pessoas que possuem esse nome na base de dados. O gráfico abaixo procura ilustrar isso. TRUE remete a casos em que a classificação foi diferente.

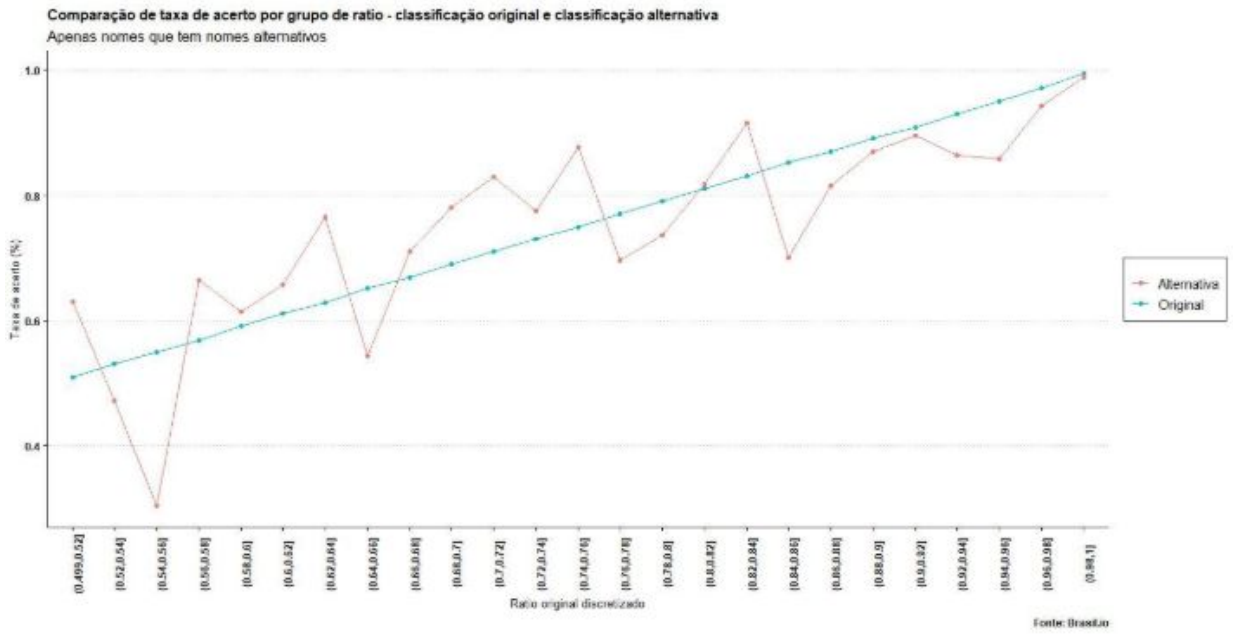
Queremos fazer uso de uma classificação alternativa quando a classificação original não é certeira (próximo dos 50%) ou quando temos poucos nomes na base de dados. A próxima tarefa é tentar obter esse(s) limiar/parâmetros que permitam ter melhores resultados de classificação.

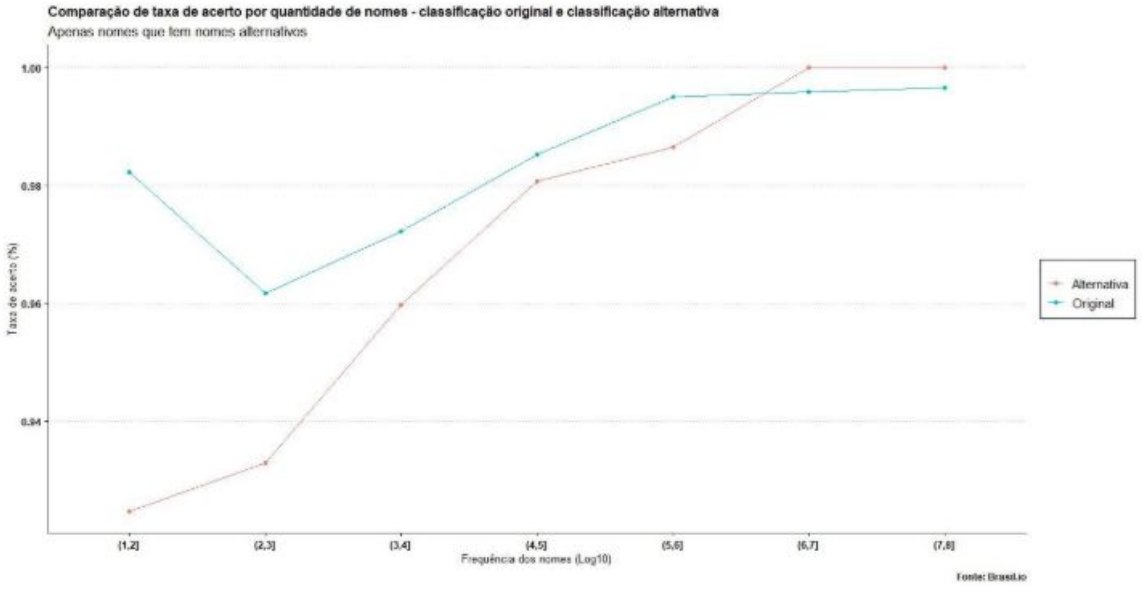
Os dois ao final procuram ilustrar a comparação entre a taxa de acerto original e a taxa de acerto usando uma classificação alternativa: usando os nomes alternativos como forma de classificação. Os gráficos mostram que a forma alternativa de classificação não é sistematicamente melhor que a forma original, nem considerando nomes como poucos registros ou que a proporção homem e mulher é próxima de 0.5.

Por esses motivos, não há diferença significativa entre a forma original de classificação e a forma alternativa.

Em relação à implementação da funcionalidade, o repositório do GitHub do Squad 4 faz essa implementação.







Algumas observações:

-Seria interessante comparar a frequência do nome selecionado para análise (censo 2010) com a realidade relatada no IBGE para que seja possível gerar uma taxa de erro.

-Possibilidade Extra para complementar critérios da análise:

-Determinação de sexo e idade por foto: Face ++.

-É interessante combinar essa base com critérios de nomes brasileiros para os nomes que são divididos. Como por exemplo: em geral nomes terminados em ‘a’ são femininos. Caso um nome tenha uma baixa frequência ou seja ambíguo pode-se fazer a análise por grupos. Existem APIs construídas em outros países que fazem isso, mas, em geral, utilizam Machine Learning.