UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

HÉBER DOS SANTOS SALES
SULLYVAN DE MOURA FIRMINO DA SILVA

Crawler para Coleta de Dados do Twitter

Prof. Filipe Braida do Carmo, D.Sc.
Orientador

Nova Iguaçu, Maio de 2024

Crawler para Coleta de Dados do Twitter

HÉBER DOS SANTOS SALES
SULLYVAN DE MOURA FIRMINO DA SILVA

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

HÉBER DOS SANTOS SALES

SULLYVAN DE MOURA FIRMINO DA SILVA

Aprovado por:

Prof. Filipe Braida do Carmo, D.Sc.

Prof. Leandro Guimaraes Marques Alvim, D.Sc.

Prof. Ubiratam Carvalho De Paula Junior, D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Maio de 2024

DOCUMENTOS COMPROBATÓRIOS Nº 9152/2024 - CoordCGCC (12.28.01.00.00.98)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 03/07/2024 19:56)
FILIPE BRAIDA DO CARMO
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###295#4

(Assinado digitalmente em 03/07/2024 16:46)
LEANDRO GUIMARAES MARQUES ALVIM
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###008#2

(Assinado digitalmente em 04/07/2024 08:16)
UBIRATAM CARVALHO DE PAULA JUNIOR
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###426#4

(Assinado digitalmente em 03/07/2024 20:44)
HEBER DOS SANTOS SALES
DISCENTE
Matrícula: 2020######2

(Assinado digitalmente em 03/07/2024 22:29)
SULLYVAN DE MOURA FIRMINO DA SILVA
DISCENTE
Matrícula: 2020######7

*Visualize o documento original em https://sipac.ufrrj.br/documentos/ informando seu número: **9152**, ano: **2024**, tipo: **DOCUMENTOS COMPROBATÓRIOS**, data de emissão: **03/07/2024** e o código de verificação: **c4a6059169***

Agradecimentos

Héber dos Santos Sales

Gostaria de expressar minha mais profunda gratidão aos meus pais, Washington e Maria das Dôres, onde o amor incondicional e apoio incansável foram os pilares que sustentaram minha jornada até este momento. Sempre me incentivaram a perseguir meus sonhos e nunca hesitaram em oferecer seu apoio e orientação, mesmo diante de tantos desafios. Sou imensamente grato por todo o sacrifício que fizeram para me proporcionar as melhores oportunidades e por serem fonte constante de inspiração e motivação. Sei que o amor e apoio serão sempre o alicerce do meu sucesso. Obrigado, do fundo do meu coração.

À minha querida irmã, Hellem, meu sincero agradecimento. Sua constante presença e apoio foram fundamentais para mim. Obrigado por todas as palavras de encorajamento e por tornar minha jornada mais leve e significativa.

Agradeço também à minha tia Tânia e minha prima Tamar, cujo apoio e estímulo desde o início foram fundamentais para moldar meu compromisso com a educação e minha determinação em alcançar meus objetivos. A confiança em mim foi um verdadeiro presente.

Quero expressar minha profunda gratidão ao meu parceiro de TCC, Sullyvan. Sua colaboração foi verdadeiramente indispensável para o êxito deste trabalho. Juntos, enfrentamos desafios com determinação e alcançamos nossos objetivos com sucesso.

Aos meus colegas e amigos do curso de Ciência da Computação, que caminharam comigo desde o início da graduação, compartilhando risadas, preocupações pré e

pós-provas, e enfrentando as madrugadas de estudos juntos. Um agradecimento especial ao Sullyvan, Marcos e Kevyn, que foram minha equipe durante minha iniciação científica e cujo apoio foi inestimável.

Expresso meu sincero agradecimento a todos os professores e membros do Departamento de Ciência da Computação da Universidade Federal Rural do Rio de Janeiro (UFRRJ-IM). Em especial ao meu orientador, Filipe Braida, pela valiosa oportunidade concedida ao me integrar neste projeto que agora apresento neste trabalho. Essa oportunidade não apenas enriqueceu minha experiência acadêmica, mas também foi um ponto crucial para o avanço em minha carreira profissional, abrindo portas para novos horizontes e desafios. Sou profundamente grato pela confiança depositada em mim e em meus companheiros ao longo desta jornada.

Agradeço a todos que estiveram ao meu lado e contribuíram neste processo. Muito obrigado!

Sullyvan de Moura Firmino da Silva

Primeiramente, gostaria de agradecer à minha mãe, Isabelle, pelo suporte durante toda a minha jornada, sempre me incentivando a focar nos estudos e me motivando a continuar. Ela foi a pessoa que possibilitou que eu concluísse a graduação, priorizando minha educação acima de muitas outras coisas. Muito obrigado por todo amor, carinho e apoio.

Gostaria de agradecer à minha avó, Raimunda, que me proporciona todo seu apoio a meus projetos e objetivos, sempre preocupada em saber como estou e procurando formas de me ajudar quando necessito.

Agradeço à minha companheira, Giulia, que a todo momento esteve ao meu lado para me ajudar em cada desafio que enfrentei. Seu apoio e confiança me incentivaram a continuar e chegar até onde cheguei. Muito obrigado.

Também sou grato aos professores do Departamento de Ciência da Computação da Universidade Federal Rural do Rio de Janeiro (UFRRJ-IM). Vocês me proporcionaram conhecimento que levarei para toda minha vida pessoal e profissional.

Agradeço ao meu orientador, Filipe Braida, a quem admiro e que me concedeu a oportunidade de fazer parte deste projeto apresentado neste trabalho. Oportunidade esta que abriu portas em minha carreira profissional. Obrigado.

Gostaria de agradecer à minha dupla neste trabalho, Héber, que esteve comigo durante todo o desenvolvimento deste projeto. Além de uma dupla, é um amigo que levarei para a vida.

Quero agradecer a meus amigos do curso de Ciência da Computação, destacando a equipe da minha iniciação cientifica, Heber, Kevyn e Marcos, que sempre estiveram presentes compartilhando momentos durante minha jornada na faculdade.

Agradeço a todos que participaram e contribuíram durante minha graduação!

RESUMO

Crawler para Coleta de Dados do Twitter

Héber dos Santos Sales e Sullyvan de Moura Firmino da Silva

Maio/2024

Orientador: Filipe Braida do Carmo, D.Sc.

Em um ambiente digital onde redes sociais geram informações de forma massiva e constante, essas plataformas se estabeleceram como ricas fontes de dados para análise e pesquisa. Contudo, a extração e armazenamento manual dessas informações é uma tarefa inviável, ressaltando a importância de ferramentas automatizadas para a coleta e organização desses dados. Dentro deste cenário, o Twitter se destaca como uma plataforma especialmente relevante para estudos, devido à acessibilidade proporcionada por sua API, que facilita a extração desses dados. Este trabalho aborda o desenvolvimento de um *crawler* focado na extração e organização de dados do Twitter, detalhando os desafios técnicos enfrentados e as soluções implementadas para uma operação eficiente do sistema. O objetivo deste trabalho é desenvolver uma ferramenta robusta útil para pesquisadores e profissionais, facilitando o acesso e a organização dos dados do Twitter para análises de mercado e estudos sociais. As funcionalidades incluem busca automatizada de tweets por termos específicos, usuários desta rede social e metadados associados e esses. O projeto se demonstrou eficaz em extrair e guardar dados do Twitter ao armazenar aproximadamente 9.2M de tweets e 10M de perfis de usuários em um período de um ano de testes respeitando as restrições da API do Twitter.

ABSTRACT

Crawler para Coleta de Dados do Twitter

Héber dos Santos Sales and Sullyvan de Moura Firmino da Silva

Maio/2024

Advisor: Filipe Braida do Carmo, D.Sc.

In a digital environment where social networks generate information massively and constantly, these platforms have established themselves as rich data sources for analysis and research. However, the manual extraction and storage of this information is an unfeasible task, highlighting the importance of automated tools for the collection and organization of this data. Within this scenario, Twitter stands out as an especially relevant platform for studies, due to the accessibility provided by its API, which facilitates the extraction of these data. This work addresses the development of a crawler focused on the extraction and organization of Twitter data, detailing the technical challenges faced and the solutions implemented for efficient system operation. The objective of this work is to develop a robust tool useful for researchers and professionals, facilitating access to and organization of Twitter data for market analysis and social studies. The functionalities include automated tweet searches by specific terms, users of this social network, and associated metadata. The project proved effective in extracting and storing data from Twitter by storing approximately 9.2M tweets and 10M user profiles over a one-year test period while respecting the Twitter API's restrictions.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

As redes sociais são uma fonte abundante de informações, graças à grande quantidade de usuários que compartilham diariamente conteúdos como pequenos textos, fotos, links e muito mais. Segundo o *Datareportal*, constatou-se que em janeiro de 2024, mundialmente, as redes sociais contavam com mais de 5 bilhões de usuários[1]. Diante dessa grande de quantidade de dados, a tarefa de filtragem e análise dessas informações manualmente se torna não apenas impraticável, mas impossível, devido ao imenso volume de dados disponibilizadas na internet a cada momento.

Nesse contexto, surge a necessidade de sistemas de análise capazes de processar e extrair valor dessa vastidão de dados. Isso possibilita a identificação de padrões e tendências nos dados e habilita uma compreensão mais profunda das dinâmicas sociais, comportamentais e de consumo manifestadas através das redes sociais. Assim, a capacidade de organizar, analisar e interpretar esses dados não só tem implicações significativas para o campo da ciência de dados, mas também para empresas, governos e organizações que buscam fundamentar suas decisões com base em informações atualizadas e relevantes extraídas das interações *online*.

Para tornar mais eficaz o processo de análise de dados, torna-se crucial a disponibilidade de conjuntos de dados sobre algum(s) assunto(s) que já tinham sido

[1]DataReportal: Global Social Media Statistics - https://datareportal.com/social-media-users. Data de acesso: 8/04/2024

coletados e estruturados de forma a otimizar essa análise. Dessa forma, surge a demanda por ferramentas especializadas capazes de realizar a coleta desses dados em grande escala e de armazená-los organizadamente de maneira sistemática. Essa organização prévia permite que analistas e cientistas de dados acessem, explorem e interpretem as informações de forma mais eficiente.

O Twitter se destaca como uma das plataformas de redes sociais mais populares, movimentando uma grande quantidade de informações todos os dias. Sua popularidade a torna uma fonte valiosa de dados para análise. A disponibilização de uma Interface de Programação de Aplicações (API) por parte do Twitter aumenta essa utilidade, pois facilita o acesso programático a uma ampla gama de dados publicados na plataforma, incluindo tweets e perfis de usuários.

1.1 Objetivo

O objetivo deste trabalho é desenvolver uma aplicação que realize a busca e armazenamento organizado de dados através da API do Twitter. A aplicação foca na coleta de informações como tweets, usuários e outros metadados relacionados. Este objetivo surge da necessidade de analisar e interpretar a imensa quantidade de dados gerados nas redes sociais, especialmente no Twitter, que se destaca pela sua popularidade e volume de informações compartilhadas diariamente.

A grande quantidade de dados disponibilizados nas redes sociais torna a tarefa de filtragem e análise manual impraticável. Assim, torna-se crucial a utilização de ferramentas de análise capazes de processar e extrair valor desses dados. A organização e estruturação dos dados previamente coletados facilitam o acesso e a interpretação pelos analistas e cientistas de dados, otimizando a identificação de padrões e tendências sociais, comportamentais e de consumo .

Para atingir este objetivo, a aplicação proposta é capaz de realizar buscas automatizadas e manuais por termos específicos, perfis de usuários e seus respectivos metadados. Ela permite o cadastro de termos de interesse, possibilitando tanto buscas instantâneas quanto agendadas em intervalos periódicos, promovendo uma

organização eficiente dos dados coletados.

A proposta envolve o desenvolvimento de uma ferramenta robusta que seja útil para pesquisadores e profissionais, facilitando o acesso e a organização dos dados do Twitter para análises de mercado e estudos sociais. Ao longo do desenvolvimento, foram enfrentados desafios técnicos significativos, como o gerenciamento das limitações da API do Twitter e a implementação de estratégias para a coleta e armazenamento eficientes dos dados.

1.2 Organização do Trabalho

Este trabalho é composto por cinco capítulos, que incluem uma introdução, fundamentação teórica, proposta do trabalho, apresentação e análise dos dados obtidos e as conclusões. A seguir, apresentamos uma visão geral sobre cada capítulo.

• Capítulo 1: Consiste em uma visão geral sobre o contexto associado ao estudo deste trabalho.

• Capítulo 2: Consiste na introdução conceitos essenciais sobre web *crawlers*, destacando a importância da coleta automatizada de dados da web devido ao volume e velocidade com que são gerados. Apresenta a arquitetura de um web *crawler* e aborda questões legais e técnicas associadas à sua utilização.

• Capítulo 3: Consiste na proposta do desenvolvimento de um *crawler* específico para o Twitter, discutindo a motivação por trás da escolha desta plataforma e a relevância dos dados que ela oferece para diversos campos de análise de dados. Detalha a arquitetura proposta, explicando cada componente do sistema e como eles interagem para coletar, tratar e armazenar os dados obtidos.

• Capítulo 4: Consiste em discutir a implementação do sistema, explanando as tecnologias utilizadas e o motivo de sua utilização. Além disso discute decisões cruciais tomadas durante o desenvolvimento e também apresenta uma seção dedicada às estatísticas geradas a partir dos dados coletados do Twitter.

• Capítulo 5: Consiste nas conclusões obtidas com base no trabalho e em possíveis

trabalhos futuros relacionados a ele.

Capítulo 2

Web Crawlers

O avanço e a popularização da tecnologia têm sido fatores importantes na aceleração da geração e consumo de dados na web. De acordo a 11ª versão do infográfico anual *"Data Never Sleeps"*[1], da empresa *Domo*, a internet alcançou 5,2 bilhões de usuários globais em 2023, representando cerca de 65% da população mundial. Esta grande quantidade de usuários foi responsável pela criação e consumo de cerca de 120 *zettabytes* ao longo do ano.

Essa imensa quantidade de dados gerada anualmente evidencia a crescente necessidade da utilização de recursos de software para a análise, interpretação e gerenciamento desses vastos conjuntos de dados. Já que a abordagem manual, além de impraticável devido à vastidão dos dados, não conseguiria acompanhar a velocidade e escala em que as informações são produzidas e modificadas na web.

Dentro desse cenário, surgem conceitos importantes como o de *Data Mining*, descrito por Witten, Frank e Hall (2011) como um método de sondagem em extensas bases de dados com o intuito de descobrir padrões, consolidá-los e prever futuras tendências com base nesses achados. Especificamente sobre a internet, o termo *Web Mining* é introduzido, referindo-se a essa específica abordagem de extração de dados online, cujo processo inicial se realiza por meio de web *crawlers*. (DR.P.PONMUTHURAMALING, 2013)

[1]Data Never Sleeps 11.0 - Quantidade de dados publicados na internet a cada minuto. https://www.domo.com/learn/infographic/data-never-sleeps-11. Data de acesso: 16/04/2024

Na próxima seção será apresentado e discutido o conceito de *crawling* de dados na *web*. Serão abordados temas como a definição de web *crawling* e suas principais funcionalidades.

2.1 Crawling de Dados na Web

Web *Crawlers* constituem ferramentas automatizadas projetadas para a coleta sistemática de dados na internet, operando mediante a navegação por páginas *web* para extrair informações de acordo com critérios pré-estabelecidos. A principal motivação para o uso de *web crawlers* é a necessidade de coletar, de forma automatizada, volumes significativos de dados da *web*, uma necessidade que se torna cada vez maior devido ao crescente volume de informações disponíveis online. *Crawlers* permitem a organização, a indexação e a análise de informações de maneira eficaz, servindo a uma variedade de propósitos como alimentar bancos de dados de motores de busca, realizar análises de mercado, monitorar concorrentes, entre outros.(CLAUSSEN; PEUKERT, 2019)

Os *web crawlers* desempenham um papel crucial na indexação de sites para motores de busca, uma função essencial para a estrutura da internet como a conhecemos hoje. Além disso, eles são indispensáveis em uma gama diversificada de aplicações, como sistemas de agregação de notícias e sites de comparação de preços em plataformas de *e-commerce*. Eles são particularmente valiosos para pesquisas acadêmicas, onde a coleta automatizada de dados pode fornecer *insights* em várias disciplinas, demonstrando a adaptabilidade e a relevância transversal dos *crawlers* na era digital. (CLAUSSEN; PEUKERT, 2019)

Segundo Bastos (2006), um *crawler* inicia o processo de busca de conteúdos a partir de um endereço específico, realizando essa coleta através de requisições direcionadas a uma determinada fonte de dados. Os web *crawlers* comumente utilizam o protocolo de transferência de hipertexto, conhecido pela sigla HTTP, e podem operar através de outros protocolos, como o protocolo de transferência de arquivos, FTP. As duas funções primárias de um *crawler* são a solicitação e a coleta

do conteúdo do endereço fornecido, enquanto as ações de indexar e armazenar os dados, embora importantes, dependem dos propósitos específicos da implementação e, portanto, podem variar, existindo ou não necessidade. (RIBEIRO, 2013)

Os web *crawlers* podem ser classificados em dois tipos principais: os não focados e os focados. Os web *crawlers* não focados têm o objetivo de explorar a internet como um todo, sem realizar distinções ou julgamentos sobre o conteúdo que encontram. Esta abordagem de varredura extensiva demanda um sistema de armazenamento robusto para lidar com os dados coletados, além de exigir significativa capacidade computacional devido à vastidão da web. (ABKENARI; SELAMAT, 2010)

Por outro lado, os web *crawlers* focados operam limitando sua busca a áreas específicas da internet e filtrando apenas os dados que são relevantes para uma finalidade previamente estabelecida. Esta estratégia permite a eliminação de conteúdo irrelevantes, mantendo um banco de dados seletivo e de tamanho gerenciável. Esta abordagem mais direcionada não só otimiza os recursos utilizados na busca e processamento dos dados, como também assegura uma coleta de informações mais alinhada aos interesses específicos da pesquisa, proporcionando um equilíbrio eficaz entre a amplitude da varredura e a precisão dos dados coletados. (ABKENARI; SELAMAT, 2010)

Os web *crawlers* tradicionais iniciam seu processo de busca e coleta de dados a partir de uma URL específica, estabelecendo-a como ponto de partida para sua atividade de navegação na web. A partir dessa URL inicial, os *crawlers* seguem os *hyperlinks* contidos na página, movendo-se sistematicamente para outras páginas web relacionadas (BRIN; PAGE, 1998). A eficácia desse processo depende da implementação do *crawler* em extrair o conteúdo das páginas web e em seguir os links de maneira eficiente, evitando assim problemas como loops infinitos ou a coleta de informações redundantes ou irrelevante.

2.2 Arquitetura de um Web Crawler

Vários modelos de arquitetura para *web crawlers* foram propostos e discutidos em estudos acadêmicos, e muitos adotaram como referência a estrutura proposta por Heydon e Najork (1999) em seu projeto de *crawler* intitulado *Mercator*. Esta arquitetura segmenta o processo de *crawling* em tarefas específicas, separadas em diferentes módulos. A adoção ampla do *Mercator* se deve à sua modularidade e flexibilidade, permitindo a substituição ou adição de componentes conforme necessário A Figura 2.1 é uma visão conceitual simplificada da arquitetura de *Mercator*, que pode ajudar a entender o funcionamento de um *web crawler*.

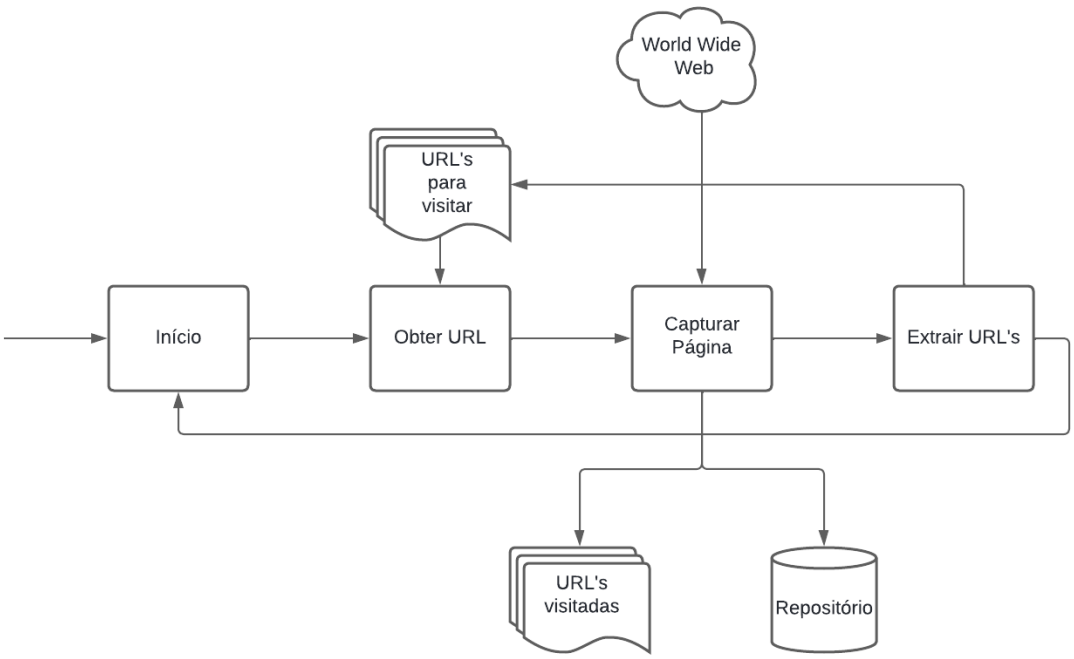

Figura 2.1: Arquitetura Conceitual de um Crawler

A Figura representa os passos seguidos no funcionamento de um *crawler*. Para organizar as URLs, o sistema as separa em URLs que ainda serão exploradas e a as que já foram acessadas. As URLs já acessadas compõem uma lista de páginas já capturadas, evitando a busca de dados repetidos. Já a lista de URLs a serem exploradas possui as páginas que serão visitadas no futuro. A composição inicial desta lista é definida manualmente ou adquirida por uma fonte externa antes da

primeira execução do crawler.

Neste processo, o *crawler* percorre a lista de URLs pendentes, seleciona uma URL, recupera a página web correspondente, salva essa página em um repositório e, em seguida, transfere a URL para a lista de URLs já visitadas. O *crawler* analisa e extrai todos os *hyperlinks* de cada página obtida, transforma os *hyperlinks* relativos em absolutos e verifica se algum dos novos *hyperlinks* já foi processado anteriormente. Se um *hyperlink* já estiver na lista de visitados, ele é descartado, caso contrário, é adicionado à lista de URLs a serem visitadas. Esse procedimento continua até que todas as URLs pendentes sejam processadas. Se houver falha na obtenção de uma URL, ela é recolocada na lista de pendências.

2.3 Considerações Legais sobre Web Crawlers

A utilização de web *crawlers* apresenta uma grande complexidade legal. Há questões legais em torno da violação do Ato de Fraude e Abuso de Computadores (CFAA) — legislação dos Estados Unidos que proíbe o acesso não autorizado ou excessivo a computadores e redes, visando prevenir a fraude e a invasão de sistemas — bem como preocupações com violações de direitos autorais ao coletar e reutilizar informações sem permissão. Além disso, o desrespeito aos termos de serviço de sites pode resultar em ações legais contra os operadores de *crawlers*. Em resposta a esses potenciais problemas legais, surgiram medidas como o *Robots Exclusion Protocol*, conhecido popularmente pelo arquivo *robots.txt*, presente em diversos servidores que sinaliza se um site permite ou não a visita de robôs automatizados (GOLD; LATONERO, 2017).

Outra consideração legal crítica envolve a privacidade dos dados coletados. Leis como o Regulamento Geral de Proteção de Dados (GDPR) da União Europeia obrigam os operadores de *crawlers* a serem cuidadosos ao coletar, armazenar e processar dados pessoais. Isso inclui garantir que o consentimento adequado seja obtido quando necessário, manter os dados seguros e usá-los de maneira ética e legal. Violações da privacidade dos dados podem resultar em consequências significativas,

incluindo multas pesadas e danos à reputação (GOLD; LATONERO, 2017).

Muitos *websites* implementam medidas de proteção *Anti-Crawler* (AnACP). ACP refere-se a um conjunto de técnicas e tecnologias projetadas para impedir ou limitar o acesso dos *crawlers* aos dados do site, garantindo que apenas usuários autorizados possam acessar o conteúdo. Essas medidas podem variar desde testes simples, como CAPTCHA (Figura 2.2), que distinguem entre usuários humanos e *bots*, até soluções mais complexas baseadas em análise de comportamento e bloqueio de IPs suspeitos. O objetivo do ACP é proteger os interesses financeiros e operacionais dos websites, prevenindo a sobrecarga de seus servidores e a perda potencial de propriedade intelectual, ao mesmo tempo que tentam evitar um impacto negativo na experiência do usuário (LIU; FENG; SUN, 2023).

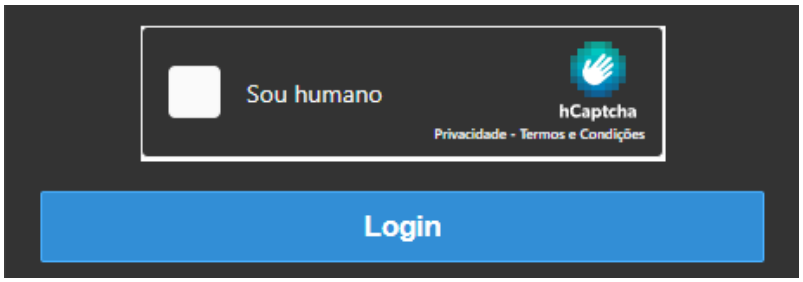

Figura 2.2: Medida de Proteção Anti-Crawler: Captcha

Uma solução adotada por alguns sites, incluindo o Twitter, para limitar a busca de dados e evitar problemas legais é a oferta de Interfaces de Programação de Aplicativos (APIs). Somado às técnicas de ACP, ao fornecer APIs, os sites podem controlar a quantidade e a frequência com que os dados são acessados, permitindo que os *crawlers* obtenham informações de forma regulada e direta do banco de dados do site, sem interferir na navegação dos usuários. Esta abordagem não apenas reduz a carga nos servidores, mas também oferece uma alternativa mais transparente e eficiente para a coleta de dados. (LIU; FENG; SUN, 2023)

Do ponto de vista do site que fornece a API, disponibilizar uma interface traz várias vantagens, especialmente em termos de gestão de tráfego e segurança dos dados. Uma API dedicada permite ao site implementar um controle mais eficaz sobre a quantidade de dados acessados, estabelecendo limites que previnem o abuso do sistema e a degradação do desempenho. Isso não apenas ajuda a melhorar o

balanceamento de carga, distribuindo de forma mais equilibrada as requisições ao servidor, mas também protege a integridade dos dados ao limitar o acesso excessivo. Esses limites são essenciais para garantir que o site possa operar de maneira estável e segura, sem comprometer a experiência do usuário final ou a disponibilidade do serviço, mesmo sob demanda intensa de acessos por *crawlers*.

A disponibilidade de APIs pelos sites pode ser considerado um recurso vantajoso na implementação de web *crawlers*, apresentando uma solução para as complexidades legais e técnicas. Ao facilitar o acesso autorizado e estruturado aos dados, as APIs eliminam preocupações legais relacionadas ao acesso não autorizado e à violação de direitos autorais, alinhando-se com as legislações de proteção de dados como o GDPR. Além disso, as APIs garantem uma coleta de dados mais sustentável e eficiente, evitando o abuso de recursos dos sites, beneficiando tanto os provedores de conteúdo quanto os usuários finais.

Capítulo 3

Proposta

Este capítulo tem como objetivo discutir a motivação por trás deste trabalho, tratando sobre a necessidade de busca, filtragem e organização de dados do Twitter. Será abordado também a arquitetura do projeto, discutindo como foram estruturados a base de dados, controlador e o *crawler* em si.

3.1 Motivação

Nas últimas décadas, o aumento do uso da internet tem sido uma das transformações mais notórias da sociedade moderna. Com a crescente disponibilidade de dispositivos móveis e o avanço de tecnologias como da banda larga e das redes sem fio, a internet se tornou cada vez mais presente na realização de atividades cotidianas, como compras, entretenimento e trabalho. Segundo o Ministério das Comunicações (2022), o número de domicílios brasileiros com acesso a internet passou dos 90% em 2021.

Esta mudança permitiu o surgimento de novas formas de interagir socialmente, expressar opiniões, preferências e buscar conhecimento, mudando a maneira como as pessoas se relacionam e se informam sobre o mundo. O tempo médio de uso diário da

internet no Brasil já ultrapassa 8 horas[1]. A quantidade de informações que passam na internet é crescente a cada dia. Essa grande quantidade de dados torna a esta rede uma fonte valiosa para pesquisas e análises em diversos campos do conhecimento.

Neste vasto volume de conteúdo disponível em tempo real, é possível obter informações sobre praticamente qualquer assunto, desde notícias de última hora até pesquisas acadêmicas especializadas. As pessoas podem acessar sites de notícias, *blogs*, fóruns de discussão, enciclopédias online, vídeos educativos e outras fontes de informações. Este volume é tão grande que, muitas vezes, encontrar informações precisas, relevantes ou gerar conclusões sobre um determinado assunto pode ser um desafio.

As redes sociais têm um papel importante nessa disseminação de informações. Nelas são publicadas quantidades enormes de dados a cada minuto [2]. Com bilhões de usuários em todo o mundo compartilhando fotos, vídeos, textos e outros conteúdos, elas se tornaram uma das principais fontes de dados da era digital. Esses dados incluem informações sobre as preferências e interesses dos usuários, seus comportamentos de consumo, suas opiniões sobre diversos assuntos e muitas outras informações que podem ser utilizadas para fins diversos, como publicidade, pesquisa de mercado e análise de tendências.

O Twitter é uma plataforma proeminente em meio às redes sociais da atualidade. Com sua abrangência global e impacto significativo na disseminação de informações, essa plataforma tem conquistado uma imensa base de usuários ao redor do mundo. De acordo com Simon Kemp (2022), em julho de 2022, o Twitter tinha cerca de 486 milhões de usuários ativos mensais. Nele, o usuário pode compartilhar pensamentos, ideias e opiniões de forma concisa, por meio de postagens breves conhecidas como tweets. Além disso, a capacidade de se conectar e interagir com uma ampla gama de usuários, incluindo figuras públicas, empresas e organizações, torna o Twitter uma ferramenta poderosa para a expressão pessoal, promoção de produtos e serviços, bem

[1] *Brasileiro é um dos campeões em tempo conectado na internet* - G1 https://g1.globo.com/especial-publicitario/em-movimento/noticia/2018/10/22/brasileiro-e-um-dos-campeoes-em-tempo-conectado-na-internet.ghtml. Data de acesso: 17/05/2023

[2] Data Never Sleeps 11.0 - Quantidade de dados publicados na internet a cada minuto. https://www.domo.com/learn/infographic/data-never-sleeps-11. Data de acesso: 16/04/2024

como para o debate e discussão de temas de relevância social.

Essa rede social se consolidou como uma das redes sociais mais empregadas em estudos acadêmicos e de mercado, principalmente porque a maioria dos dados publicados nessa plataforma são de natureza pública e são disponibilizados por meio de uma API. Essa característica facilita significativamente o trabalho dos pesquisadores, permitindo a coleta de informações sem as complexidades legais e éticas associadas ao acesso a dados privados. Além disso, o Twitter conta com uma grande diversidade de assuntos, variando de discussões sobre eventos cotidianos a debates sobre grandes questões globais. Essa amplitude de tópicos disponíveis torna a plataforma extremamente valiosa para pesquisas que buscam entender tendências sociais, dinâmicas políticas, respostas a eventos atuais e muito mais.

A plataforma do Twitter, com sua vasta quantidade de dados publicados diariamente, se torna uma fonte valiosa e rica em potencial de informações. Segundo o infográfico anual Data Never Sleeps, são publicados, mundialmente, em média 360.000 tweets a cada minuto [2] No entanto, devido à natureza massiva desses dados, monitorar manualmente um tópico específico de interesse pode se tornar uma tarefa desafiadora e, em alguns casos, até mesmo impossível. O número de tweets relacionados a um assunto pode crescer rapidamente, o que dificulta acompanhar todas as postagens relevantes. Essa explosão de informações torna necessário o uso de soluções inteligentes e automatizadas para filtrar, analisar e extrair insights úteis desses fluxos contínuos de dados. Uma possível solução são a busca e filtragem desses dados são os *crawlers*.

Como discutido no Capítulo 2 deste trabalho, existem *crawlers* que se aproveitam das APIs para coletar dados de forma eficiente. Nesse contexto, o funcionamento desse tipo de crawler envolve o envio de solicitações HTTP para a API, com a definição dos parâmetros necessários para acessar os dados desejados. A API, por sua vez, processa essas solicitações e retorna os dados em um formato estruturado, como JSON ou XML. O *crawler*, então, realiza o tratamento e o armazenamento dessas informações para uso posterior ou análise. Essa abordagem automatizada permite a coleta regular de dados atualizados da API, tornando-a uma ferramenta poderosa

para manter informações em tempo real ou criar conjuntos de dados personalizados a partir de fontes externas.

Um *crawler* de dados de redes sociais é de extrema importância para diversas aplicações, principalmente de análise e tomada de decisões. Ele permite coletar informações valiosas a partir das plataformas de redes sociais, possibilitando a análise de tendências, opiniões e comportamentos dos usuários. Com a coleta de dados em larga escala, um *crawler* pode ser utilizado para monitorar a percepção pública sobre uma marca, produto ou evento, identificar influenciadores, detectar notícias falsas, analisar o sentimento do público, entender a demografia dos usuários e até mesmo ajudar em pesquisas acadêmicas.

Empresas podem utilizar *crawlers* para realizar monitoramento de marca, acompanhando como seus produtos e serviços são percebidos pelos usuários da plataforma. Além disso, pesquisadores e cientistas sociais podem empregar *crawlers* para coletar dados e analisar tendências em tempo real, auxiliando em estudos de opinião pública, comportamento eleitoral e pesquisa de mercado. Também é possível utilizar um *crawler* para rastrear eventos em tempo real, como desastres naturais, manifestações políticas ou surtos de notícias, contribuindo para a disseminação de informações precisas e auxiliando na resposta a crises. Em resumo, um *crawler* de dados é uma ferramenta poderosa para a coleta de informações valiosas que podem ser aplicadas em uma variedade de contextos, desde marketing até pesquisa acadêmica e gestão de crises.

É importante ressaltar que o Twitter disponibiliza uma robusta API que oferece uma ampla gama de recursos e funcionalidades, permitindo os desenvolvedores criarem serviços que interagem com a plataforma do Twitter. Um dos principais recursos fornecidos pela API é a possibilidade de buscar tweets juntamente com metadados relacionados a eles, como autor, data de criação, localização e interações. Essa funcionalidade abre a possibilidade de criar um *crawler* de monitoramento que pesquisa e coleta tweets relevantes sobre determinados tópicos, permitindo acompanhar as discussões em tempo real. Além disso, a busca de usuários permite identificar perfis específicos e suas atividades, permitindo a criação de serviços de

análise de influenciadores, temas, eventos, etc.

Este trabalho tem como objetivo desenvolver um crawler de dados do Twitter que seja capaz de buscar e armazenar grandes quantidades de informação de maneira automatizada, sem a necessidade de um grande esforço manual. A implementação desse *crawler* poderá servir como um recurso valioso para a comunidade de pesquisadores, empresas e instituições que buscam utilizar dados do Twitter para análises de mercado, pesquisa acadêmica, monitoramento de tendências e até mesmo na gestão de crises, destacando-se como uma ferramenta de alta utilidade.

Na próxima seção, iremos analisar trabalhos relacionados ao desenvolvimento e uso de *crawlers* específicos para a plataforma Twitter. Esses estudos fornecerão insights valiosos sobre as técnicas e desafios enfrentados na coleta de dados dessa rede social.

3.2 Trabalhos Relacionados

O grande número de usuários ativos fazendo publicações diariamente no Twitter tem chamado atenção de pesquisadores, tornando a rede um grande atrativo como fonte de dados para trabalhos e pesquisa. Alguns deles apresentam maior foco na análise dos dados, outros, na coleta deles, como este trabalho. A seguir, apresentaremos e articularemos sobre artigos e trabalhos que apresentam soluções para a coleta e organização automatizada de dados em larga escala no Twitter.

Bošnjak et al. (2012) desenvolveram o *TwitterEcho*, um *crawler* de código aberto que permite pesquisadores buscarem dados publicados por comunidades de interesse de maneira contínua. A arquitetura desta aplicação conta com vários clientes separados fazendo requisições à API do Twitter e enviando essas informações para um servidor central. Este gerencia os dados recebidos e organiza-os. Essa abordagem tem a vantagem de superar limitações de quantidade de tweets por tempo que uma aplicação sofreria ao ter apenas um acesso à Twitter API.

Já Pratikakis (2018), propôs um *crawler* de dados do Twitter chamado *twAwler* que possui dois principais objetivos. O primeiro foco do projeto é ser um programa

leve, que pode ser executado com apenas uma máquina, porém, mantendo um alto potêncial de busca. O segundo objetivo é buscar postagens e perfis de comunidades de linguagem específica, como o grego, que é utilizado como exemplo no trabalho. Assim, a ferramenta tem a capacidade de juntar e organizar muitos dados sobre falantes de uma língua sem necessitar de muitos recursos.

Como afirma Cardoso e Machado (2008) em seu trabalho, a Mineração de Dados é uma das alternativas mais eficazes para extrair conhecimento a partir de grandes volumes de dados, descobrindo relações ocultas, padrões e gerando regras para predizer e correlacionar dados, que podem ajudar as instituições nas tomadas de decisões mais rápidas ou, até mesmo, a atingir um maior grau de confiança. Tendo em mente este potencial da Mineração de Dados, Byun, Lee e Kim (2012) criaram um *crawler* de dados do Twitter como uma ferramenta para possibilitar a criação de bases de dados para usarem como escopo de algoritmos de mineração. Um diferencial deste é a possibilidade de utilizar múltiplas chaves da API do Twitter no mesmo software para lidar com suas limitações.

Sohail et al. (2021) discutem a coleta de dados do Twitter através de APIs, abordando tanto perspectivas técnicas quanto legais. Ele ressalta a importância da privacidade do usuário e dos desafios de segurança ao acessar informações pessoalmente identificáveis em redes sociais. O trabalho propõe um *framework* para equilibrar a necessidade de coletar dados para serviços de recomendação e a proteção da privacidade do usuário. Além disso, aborda exemplos práticos para ilustrar os riscos associados à manipulação de preferências e publicidade direcionada usando dados coletados. O artigo também sugere soluções técnicas e legais para proteger a privacidade dos usuários ao coletar e usar seus dados.

3.3 Proposta

O Twitter é uma plataforma que abriga uma imensa quantidade de dados sobre uma ampla variedade de assuntos. Com milhões de usuários compartilhando pensamentos, opiniões e notícias em tempo real, há uma riqueza de informações disponíveis

para análise. Diante disso, uma plataforma capaz de capturar e organizar esses dados por assunto é de grande interesse para empresas e pesquisadores. Essa abordagem permite uma compreensão mais aprofundada dos tópicos em destaque, possibilitando análises mais precisas e insights valiosos. Além disso, uma plataforma de organização por assunto pode facilitar a descoberta e o acompanhamento de discussões relevantes, contribuindo para um compreensão mais completa e eficiente de assuntos.

O objetivo deste trabalho é desenvolver uma aplicação que busque, colete e organize dados relevantes da rede social Twitter, como postagens e perfis. Ela visa aproveitar a vasta quantidade de informações disponíveis no Twitter e possibilitar a extração e o armazenamento eficiente desses dados. O trabalho busca implementar recursos de organização por assunto, permitindo uma análise mais precisa e um entendimento abrangente das discussões em curso.

O sistema proposto tem a capacidade de buscar e armazenar tweets relacionados a um determinado termo, juntamente com os metadados associados. Ele oferece aos usuários a funcionalidade de cadastrar termos de interesse, permitindo buscas manuais instantâneas ou configurações para buscas automáticas em intervalos periódicos. Essa flexibilidade permite o usuário ter um controle preciso sobre as informações que deseja coletar e analisar, adaptando-se às suas necessidades e objetivos específicos. Os termos são separados em categorias, promovendo uma organização eficiente e facilitando a gestão dos dados.

A próxima seção apresentará a arquitetura do sistema proposto. Serão abordados os componentes fundamentais que compõem o *crawler*, incluindo a integração com a *Twitter API V2*, os mecanismos de busca automatizada, o processo de armazenamento de dados e a implementação de estratégias para garantir eficiência e confiabilidade na coleta de informações.

3.3.1 Arquitetura

A Figura 3.1 representa uma visão geral do sistema desenvolvido neste trabalho. Nela, podemos ver uma representação de seus devidos componentes. De maneira resumida, o Controlador é o componente que orquestra o sistema, fazendo chamadas

ao *Crawler* e recuperando informações presentes na Base de Dados. O *Crawler* utiliza do Serviço de Comunicação com o Twitter para conseguir dados, podendo assim filtrá-los, tratá-los e armazená-los na Base de Dados. A comunicação com o Twitter é feita via API. Nas sessões seguintes discutiremos em mais detalhes cada um destes componentes a fim de entender o funcionamento do sistema.

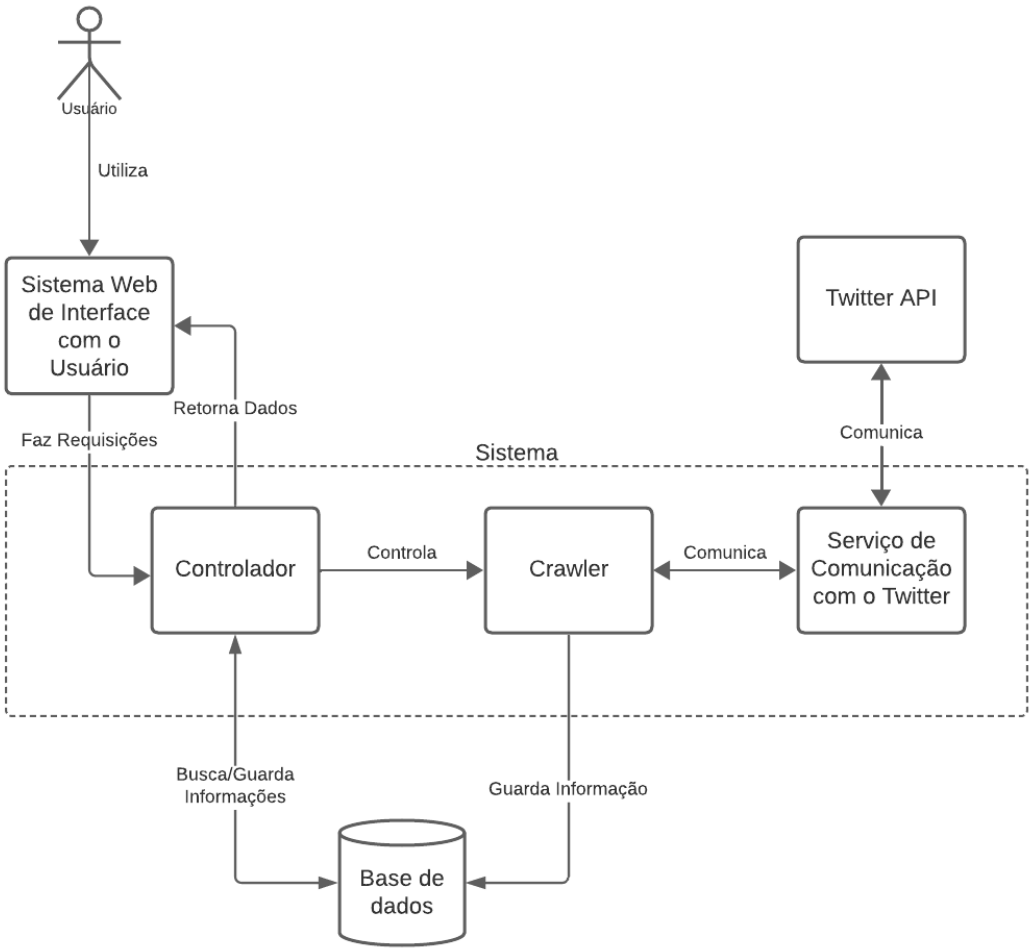

Figura 3.1: Diagrama simplificado de componentes do sistema

3.3.2 Base de Dados

Para uma compreensão abrangente das operações do sistema, é importante entender a estrutura dos dados no banco de dados relacional. Esse entendimento não apenas revela a organização interna dos dados, mas também proporciona conhecimento sobre os modelos e as relações entre eles. Ao analisar a arquitetura do

banco de dados, é possível identificar a lógica subjacente à armazenagem dos tweets, perfis de usuários, metadados e configurações de busca. Essa análise oferece uma base sólida para compreender como as diversas partes do sistema interagem e se complementam, contribuindo assim para uma visão completa do funcionamento do sistema como um todo.

A seguir, a Figura 3.2 mostra um diagrama que representa os modelos existentes no sistema com seus respectivos relacionamentos, descrevendo como os dados são organizados e como eles interagem entre si dentro. A fim de garantir a compreensão sobre o sistema, a seguir, as entidades presentes no diagrama serão apresentadas.

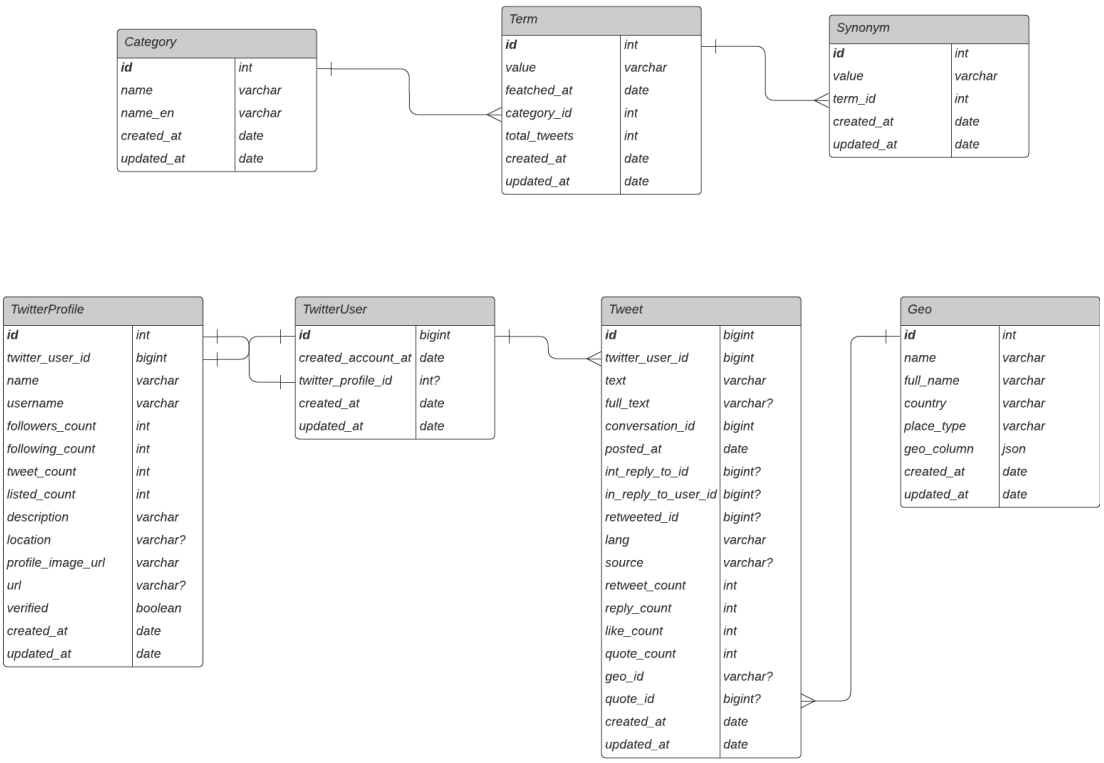

Figura 3.2: Diagrama de representação das entidades e relacionamentos do Banco de Dados do Sistema

A entidade *Term* atua como o núcleo para os termos de busca empregados pelo *crawler* que serão usados como parâmetro para a coleta de dados via Twitter API. O atributo *value* do *Term* é empregado como a chave principal na busca por novos

tweets, o que significa que é o critério de pesquisa principal. Cada *Term* está vinculado a uma entidade *Category*, o que organiza os termos de busca em grupos categorizados, facilitando assim a gestão e possível análise dos dados coletados. O relacionamento entre *Term* e *Synonym* permite que cada termo possa ter associado a ele vários sinônimos. Isso é essencial para garantir a inclusividade na busca por tweets, assegurando que variações de um termo não sejam tratadas como distintas, o que poderia fragmentar ou duplicar os dados durante a análise.

A entidade *Tweet* no diagrama é uma representação de um tweet coletado pelo sistema, armazenando não apenas o conteúdo textual no campo *text*, mas também vários metadados significativos. Estes incluem identificadores únicos para o fio de tweets (*conversation_id*), contagens de retweets, respostas, curtidas e citações, além de informações sobre a linguagem do tweet. Quando um tweet é recuperado, ele traz consigo uma referência ao *TwitterUser*, que é o autor da postagem, identificado pelo *twitter_user_id* associado.

O *TwitterUser*, por sua vez, está diretamente ligado à entidade *TwitterProfile*, que contém dados mais pessoais e estatísticas do usuário, como o *username*, *followers_count* (quantidade de seguidores), *following_count* (quantidade de perfis que segue), *tweet_count* (número total de tweets postados pelo usuário), e *profile_image_url* (URL da imagem de perfil). Esses dados permitem uma compreensão mais rica do contexto e do impacto do usuário dentro da plataforma do Twitter.

Além disso, se disponível, dados de geolocalização associados a um tweet são capturados na entidade *Geo*, onde informações como o nome do lugar, o país e o tipo de local (*place_type*) podem ser armazenados. Esses dados geográficos são essenciais para análises que consideram a localização das postagens, permitindo, por exemplo, estudos demográficos ou de disseminação de informações por região.

Combinando todas essas informações, o conteúdo do tweet, os dados do perfil do usuário que o postou, e possíveis detalhes geográficos, o sistema fornece uma base de dados robusta para análises complexas, como a filtragem de tweets para estudos de mercado, pesquisas de opinião, monitoramento de eventos em tempo real e muito mais.

3.3.3 Controlador

O controlador do sistema desempenha um papel central na orquestração e gestão das operações do *crawler* de dados do Twitter. Funcionando como a interface entre o usuário e o sistema, o controlador recebe requisições provenientes do usuário, especificando os parâmetros desejados para a busca de dados. Ao receber essas requisições, o controlador realiza a tarefa de invocar o *crawler* de dados, transmitindo os parâmetros fornecidos pelo usuário para que a coleta de informações seja conduzida de maneira personalizada e eficiente.

Ele é responsável por solicitar ao *Crawler* o inicio da busca de tweets com base nos termos cadastrados na base de dados. Juntamente a isso é realizada a busca dos perfis e usuários atrelados à publicação encontrada. O controlador pode solicitar atualizações dos perfis de usuários presentes no banco de dados para uma versão mais recente, trazendo as informações atuais disponíveis no Twitter.

Além disso, são oferecidos comandos intuitivos para a manipulação de termos, categorias e sinônimos. Através do controlador, os usuários têm a capacidade de solicitar a criação de novos termos de pesquisa, o que, por sua vez, amplia o escopo das busca do *crawler*. Também é possível modificar esses termos a fim de melhor acompanhar às tendências e assuntos emergentes ou refinar as estratégias de pesquisa. Quando um termo, categoria ou sinônimo não é mais necessário ou relevante, o controlador permite a sua deleção.

Além de coordenar as atividades do *crawler*, o controlador também desempenha um papel fundamental na interação com o banco de dados. Ao chamar o banco de dados, o controlador recupera as informações relevantes previamente armazenadas, permitindo a apresentação ao usuário de dados e históricos sobre os termos de pesquisa, categorias e sinônimos ou sobre detalhes dos perfis dos usuários e tweets. Dessa forma, ele se configura como um elemento essencial na operação integrada do sistema, garantindo uma resposta ágil e completa às necessidades do usuário ao facilitar a interação com o *crawler* e o banco de dados.

3.3.4 Crawler

O *crawler* é um componente crítico dentro do sistema. Ele é uma ferramenta projetada para extrair informações da plataforma Twitter, atuando sob a direção do controlador. Suas funções são especializadas e centrais para a operação do sistema, permitindo a aquisição de dados em grande escala. Sua responsabilidade é de interagir com o Serviço de Comunicação com o Twitter, realizando buscas baseadas em termos específicos, coletando tweets e seus metadados associados, perfis de usuários e outros dados relevantes. Este processo de coleta é essencial para alimentar a base de dados com grandes volumes de informações atualizadas.

Utilizando o Serviço de Comunicação com o Twitter, o *crawler* consegue dados recebidos via API do Twitter. Uma vez recebida a informação, o *crawler* executa um processo de tratamento dos dados, separando as informações relevantes, descartando as irrelevantes. Ele transforma os dados brutos em estruturas definidas, criando objetos que possam ser salvos no banco de dados seguindo os modelos já citados. Após a curadoria desses dados, as informações são inseridas na base de dados.

O *crawler* é capaz de efetuar requisições de tweets, perfis de usuários e dados de geolocalização. É possível definir parâmetros e preferências de buscas, que podem incluir o volume de tweets a serem recuperados por termo de busca ou a especificação de um intervalo temporal, capturando tweets de um determinado período. Essa funcionalidade permite uma coleta de dados personalizada, atendendo às necessidades específicas de cada pesquisa ou análise de dados. Além disso, o *crawler* pode realizar atualizações nos perfis dos usuários, garantindo que a base de dados possua as informações mais atuais dos usuários do Twitter.

3.3.5 Serviço de Comunicação com o Twitter

O Serviço de Comunicação com o Twitter atua como uma camada de abstração essencial entre o sistema de *crawler* e a API do Twitter, simplificando a complexidade das chamadas diretas à API. Este serviço elimina a necessidade de os desenvolvedores do sistema entenderem os detalhes dos endpoints da API do Twitter, bem como

a nomenclatura e o método de envio de parâmetros. Ao mascarar esse nível de complexidade, o serviço permite que os usuários se concentrem em definir o que desejam obter, sem se preocuparem com o "como" técnico da obtenção.

Esse serviço permite buscar tweets via termos específicos ou usuários via ID. As respostas às solicitações à API do Twitter são recebidas em formatado de JSON. Além disso, o serviço de comunicação isenta os usuários da gestão de autenticação, uma etapa técnica e propensa a erros, que exige o gerenciamento de tokens de acesso seguros. Ao automatizar a autenticação, o serviço garante que as interações com a API do Twitter sejam seguras e conformes com as políticas de uso, mantendo a integridade e a segurança dos dados.

Portanto, esse serviço de comunicação é fundamental para a eficiência operacional do *crawler*. Ele não só facilita a coleta de dados como também assegura que o sistema possa ser utilizado com facilidade por usuários que não possuem pleno conhecimento sobre a API do Twitter.

No entanto, é importante ressaltar que não é possível buscar todos os dados do Twitter de forma abrangente. A plataforma impõe limitações em termos de volume de pesquisa e acesso a tweets mais antigos, o que pode dificultar a obtenção de informações completas ou históricas em algumas situações. Sendo assim, é crucial ter em mente essas limitações ao realizar o desenvolvimento do projeto.

Gratuitamente, a API oferece dois níveis de acesso, sendo o Essencial o mais básico e o Elevado, com mais recursos. O nível Essencial permite a busca de 500k tweets por mês e acesso a partir de apenas um aplicativo. Já o nível Elevado permite a busca de 2M de tweets por mês e acesso a partir de três aplicativos.

A API v2 do Twitter estabelece restrições de frequência para as solicitações de busca de tweets, o que implica que um número definido de pedidos está disponível em intervalos de tempo específicos. Exceder esses limites pode resultar em erros ou bloqueios temporários. Adicionalmente, a API restringe a pesquisa de tweets exclusivamente aos mais recentes, e também impõe um limite na quantidade de resultados por consulta, o que pode afetar a eficácia da busca em situações que

demandam um grande volume de tweets correspondentes.

Algumas das limitações impostas pela API são, em um intervalo de 15 minutos, limita-se a busca de um máximo de 10.000 tweets, distribuídos em até 15 solicitações. Além disso, quando se trata de perfis de usuários, pode-se buscar até 3.000 perfis em um período de 24 horas, espalhados ao longo de 100 requisições [3]. Estas limitações são implementadas a fim de gerenciar o uso da API e garantir um acesso equitativo aos recursos do Twitter para todos os desenvolvedores.

O aplicação deve respeitar e lidar com essas limitações, a fim de manter um bom funcionamento, para isso, o próprio serviço de comunicação com o Twitter guarda a informação sobre a quantidade de dados que ainda podem ser resgatados dentro do período de tempo. Assim, caso não seja possível buscar mais tweets ou usuários no momento atual, a aplicação não tentará fazer uma chamada desnecessária à API.

[3]Informações fornecidas via: https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api. Data de acesso: 23/01/2022

Capítulo 4

Implementação

Este capítulo trata sobre a implementação do projeto, apresentando as ferramentas escolhidas e seus motivos, como ele foi desenvolvido, alguns exemplos de utilização e os resultados obtidos.

4.1 Tecnologias Utilizadas

Para o desenvolvimento deste projeto, utilizou-se o *AdonisJS*[1], um *framework* para desenvolvimento web em *Node.js*. Ele é reconhecido por facilitar a criação de aplicativos *web* robustos, oferecendo recursos prontos para uso, como gerenciamento de rotas, controladores. Para o armazenamento dos dados, escolheu-se o banco de dados *PostgreSQL*, uma solução de código aberto que fornece licença gratuita. A integração entre ambas as tecnologias é facilidada devido ao *ORM* (Mapeamento Objeto-Relacional) do *AdonisJs*, o *Lucid*. Além de agilizar o processo de desenvolvimento, a escolha dessas tecnologias permite um maior foco na lógica específica do projeto em si.

Nas próximas subseções serão apresentadas de maneira mais profunda as tecnologias utilizadas discutindo o motivo de sua escolha.

[1]Página do framework AdonisJS: https://adonisjs.com/

4.1.1 Node.Js

O Node.js é um ambiente de tempo de execução (Runtime Environment) de código aberto, construído sobre o motor *JavaScript V8* da *Google*, que permite aos desenvolvedores executar JavaScript no lado do servidor.

A escalabilidade é uma das características mais notáveis do *Node.js*, permitindo que as aplicações cresçam de forma eficiente para lidar com um grande número de solicitações simultâneas. Ele conta com entrada e saída não bloqueante, ou seja, quando uma operação de entrada ou saída é iniciada, em vez de esperar bloqueada até que seja concluída, o fluxo de execução do programa continua a processar outras tarefas. Isso é possível devido à abordagem assíncrona e baseada em eventos do Node.js. Quando a operação é finalizada, o programa é notificado por meio de *callbacks*, *promises* ou *async/await*, permitindo que o programa possa utilizar a resposta de sua solicitação.

Além disso, o *Node.Js* oferece uma vasta biblioteca de módulos via NPM (Node Package Manager), facilitando a integração com APIs de terceiros, como a API do Twitter, e a implementação de funcionalidades complexas com menos esforço. Esses pontos consolidam o *Node.Js* como uma escolha viável para a criação de um sistema crawler eficiente, escalável e de alto desempenho.

4.1.2 AdonisJS

O *AdonisJs*[2] é plataforma robusta para a criação de aplicações web no ambiente Node.js utilizando Typescript. Com sua arquitetura baseada no conceito MVC (*Model-View-Controller*), ele promove uma ótima organização e facilita a manutenção do código, contribuindo para a escalabilidade dos projetos. Este framework permite o desenvolvimento de aplicações do lado do servidor (*server-side*) utilizando o JavaScript. AdonisJs simplifica várias tarefas rotineiras, incluindo o gerenciamento de rotas, manipulação de requisições e respostas, controle de sessões e autenticação, permitindo um desenvolvimento mais fluído e produtivo.

[2]Página do framework AdonisJS: https://adonisjs.com/

AdonisJs integra de maneira nativa o *ORM Lucid*. O Lucid oferece uma abstração para interação com bancos de dados, permitindo que os desenvolvedores realizem consultas e operações de banco de dados de maneira eficiente, sem a necessidade de escrever consultas SQL complexas manualmente Esta integração simplifica o processo de modelagem de dados e interação com o banco de dados, contribuindo para um desenvolvimento mais rápido e menos propenso a erros. Além disso, o Lucid facilita a manutenção e a escalabilidade dos sistemas web, pois permite que alterações na estrutura do banco de dados sejam implementadas de maneira ágil.

O framework suporta o esquema de *Migrations*, simplificando o processo de evolução do esquema do banco de dados. Uma *migration* de banco de dados é um procedimento através do qual se faz a gestão de mudanças e atualizações na estrutura da base de dados de forma controlada. Esse processo permite que desenvolvedores implementem e compartilhem alterações na estrutura do banco de dados, como criação de novas tabelas, alteração de colunas existentes, índices, ou até mesmo mudanças mais complexas envolvendo lógica de dados, de maneira consistente e reversível.

Em resumo, o AdonisJs é uma boa escolha de *framework web* principalmente devido à sua vasta gama de funcionalidades pré-implementadas. Essa característica não só economiza um tempo de desenvolvimento, eliminando a necessidade de construir soluções do zero, mas também reduz significativamente a propensão a erros. O AdonisJs possibilita que os desenvolvedores se concentrem na lógica e nas características únicas de suas aplicações, promovendo uma eficiência e qualidade superior no desenvolvimento de projetos web robustos e escaláveis.

4.1.3 PostgreSQL

O PostgreSQL[3], é um Sistema Gerenciador de Banco de Dados Objeto-Relacional (SGBDOR) de código aberto que se destaca por sua robustez e versatilidade. A integridade dos dados é preservada mediante a implementação dos princípios ACID (Atomicidade, Consistência, Isolamento e Durabilidade) em transações, tornando o

[3]Página do SGBDOR PostgreSQL: https://www.postgresql.org/

PostgreSQL confiável. Ele permite a personalização do sistema por meio da criação de tipos de dados, operadores, funções e agregados personalizados, tornando possível fazer adaptações a requisitos específicos.

A capacidade do PostgreSQL para gerenciar operações concorrentes destaca-se como uma de suas características mais interessantes, possibilitando que múltiplos usuários acessem e manipulem dados simultaneamente sem perda de desempenho ou integridade. Além disso, a funcionalidade de replicação oferecida pelo PostgreSQL desempenha um papel crucial em assegurar a alta disponibilidade dos dados e a eficiência na recuperação de informações em caso de falhas ou interrupções do sistema. Essas capacidades reforçam a confiabilidade do sistema em cenários de uso críticos, onde a continuidade das operações e a segurança dos dados são pontos prioritários.

A escolha de um banco de dados relacional foi motivada pela natureza interconectada dos dados coletados da API do Twitter. Como o sistema trabalha com relações estruturadas entre modelos, a integridade das referências entre entidades, como tweets, usuários, termos, categorias, em um banco de dados relacional se sobressai quanto a capacidade de organização e consulta de maneira eficiente. A robustez, a segurança e as capacidades de escalabilidade e replicação do PostgreSQL o tornam particularmente adequado para suportar as demandas de um sistema de crawler que precisa gerenciar grandes volumes de dados com alta disponibilidade e integridade. Portanto, a escolha desse banco de dados relacional está alinhada com as necessidades do sistema.

O PostgreSQL possui uma ferramenta de *Full Text Search* (FTS) que permite a realização de buscas eficazes por texto dentro do banco de dados. Diferentemente das buscas tradicionais que comparam cadeias de caracteres exatas, o FTS analisa o texto armazenado para permitir buscas mais flexíveis. Essa extensão utiliza vetores de texto e índices que armazenam informações sobre as palavras contidas nos documentos, incluindo suas formas raízes (*stemming*) e sinônimos. Essa análise ainda conta com a aplicação de dicionários, podendo ser customizados, que ajudam a identificar e ignorar palavras comuns (*stop words*) que não contribuem para o significado da busca.

O FTS do PostgreSQL pode ser de grande utilidade para aplicações que pretendem consumir e analisar os dados capturados pelo *crawler* apresentado neste trabalho. A integração do FTS do PostgreSQL em um sistemas de análise de dados pode enriquecer significativamente a qualidade e a eficiência das consultas à base de dados.

4.2 Desafios Durante a Implementação

Nas próximas subseções, exploraremos as estratégias adotadas para superar alguns desafios enfrentados durante o desenvolvimento do sistema. Isso inclui a abordagem para a representação adequada dos identificadores de tweets, a implementação de um agendador para automatizar as operações do crawler, garantindo assim a coleta contínua de dados sem intervenção manual, e a gestão eficaz dos limites de recursos impostos pela API do Twitter Essas decisões não apenas permitiram a superação de obstáculos técnicos, mas também otimizaram o desempenho geral do sistema, assegurando sua eficiência.

4.2.1 Identificação por ID

As publicações, no sistema do Twitter, possuem identificadores únicos (UID). Esses identificadores são úteis por várias razões, principalmente por facilitarem a indexação e o acesso rápido aos itens dentro de uma base de dados extensa. Além disso, a unicidade dos UIDs permite o rastreamento preciso de cada tweet, independentemente de alterações em seu conteúdo ou status, possibilitando a criação de relações confiáveis entre dados, como respostas a um tweet original ou a associação de tweets a perfis de usuários específicos. Cada UID utiliza 64 bits para ser representado

No JavaScript, o maior valor numérico inteiro que pode ser representado de forma segura é limitado a 53 bits[4]. Essa limitação representa um desafio ao manipular de dados provenientes da API do Twitter, uma vez que os UIDs de tweet são expressos em 64 bits. Sem uma maneira adequada de lidar com esses valores, existe o risco de

[4]Informações fornecidas pela documentação sobre Javascript do Mozilla Firefox sobre maior inteiro seguro em Javascript: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number/MAX_SAFE_INTEGER

perda de precisão, o que pode comprometer a integridade dos dados e a confiabilidade das operações realizadas com esses UIDs.

A solução utilizada para esse problema foi a utilização do tipo *BigInt*, que permite a representação e manipulação de inteiros com mais de 53 bits em sua representação binária. Ao utilizar o *BigInt* para armazenar os UIDs de tweet, garantiu-se a precisão necessária para manusear esses identificadores sem comprometer a segurança ou a precisão dos dados, possibilitando que o sistema trate as informações de tweets de maneira confiável.

4.2.2 Agendador

O sistema permite que o usuário manualmente faça solicitação de dados mais atuais do Twitter. É possível solicitar a busca de novos tweets de um único termo cadastrado, ou que o sistema busque novas postagens de todos os termos cadastrados de uma só vez. Essa abordagem não é o método mais eficiente para buscar novos dados com constância. Embora essa funcionalidade proporcione controle direto sobre a coleta de dados, ela não representa a abordagem mais eficiente para a obtenção de novas informações continuamente. A dependência de intervenções manuais para iniciar buscas pode limitar a frequência e a abrangência da coleta de dados, sugerindo a necessidade de métodos mais automatizados para garantir a atualização contínua e abrangente das informações coletadas.

A necessidade de realizar solicitações manuais de novos dados implica que deve haver um usuário ativo para, periodicamente, instruir o sistema a buscar atualizações. Essa abordagem, além de demandar um esforço significativo diário, gera um gargalo potencial na aquisição de novas informações. Qualquer interrupção nas solicitações de busca, seja por algumas horas ou dias, pode resultar na perda de muitos dados, comprometendo a eficácia da coleta. Para eliminar a necessidade de intervenção manual frequente, o sistema possui um agendador implementado.

Um agendador em um sistema web é um componente que desempenha a função de gerenciar a execução de tarefas em intervalos de tempo específicos sem a necessidade de intervenção manual. Esse componente é útil para automatizar processos de

atualizações de conteúdo. O agendador trabalha em segundo plano, monitorando o relógio do sistema e aciona a execução de tarefas predeterminadas quando as condições configuradas são atendidas. A utilização de um agendador no sistema não apenas aumenta a eficiência operacional, mas também garante que operações sejam realizadas de forma consistente. Essa abordagem assegura a busca contínua e sistemática de dados mais recentes do Twitter, garantindo que o sistema permaneça atualizado e maximizando a captura de informações relevantes sem a necessidade de supervisão direta do usuário.

As tarefas agendadas podem ser de dois tipos: *System Cron Jobs* ou *In-Process Cron Jobs*. Os *System Cron Jobs* são agendados e gerenciados pelo sistema operacional, utilizando o serviço cron disponível em sistemas *Unix-like*, como Linux e macOS. Já os *In-process Cron Jobs* são tarefas agendadas que são executadas como parte de um aplicativo. Uma das vantagens é que bibliotecas de várias linguagens de programação implementam essa funcionalidade.

Este trabalho utilizou *In-Process Cron Jobs* no processo de agendar as atualizações de termos. Esse tipo de agendador se destaca pela sua capacidade de integrar-se de maneira eficiente à aplicação, garantindo que as tarefas agendadas sejam executadas de forma confiável e sem a necessidade de gerenciar processos externos na máquina onde o servidor está hospedado. A principal vantagem desse método é a simplicidade com que se integra ao ambiente de execução da aplicação, reduzindo a complexidade operacional. Para facilitar a implementação, foi adotado o *middleware adonis5-scheduler*. Este pacote fornece uma interface amigável para a configuração de tarefas agendadas, permitindo uma integração simplificada com o sistema.

A implementação de um agendador para automatizar a execução de tarefas em intervalos regulares oferece a vantagem de eliminar a necessidade de acionar manualmente os processos de coleta de dados várias vezes ao dia. No entanto, essa automatização introduz um desafio relacionado às limitações impostas pela API do Twitter, que restringe o volume de dados e a frequência das solicitações. Quando as chamadas são automatizadas, há risco de exceder esses limites, levando a potenciais bloqueios ou atrasos na coleta de dados. Para contornar esse obstáculo, foi essencial

adotar uma abordagem mais estratégica no gerenciamento dos limites de solicitação impostos pela API. A metodologia aplicada para lidar com essas restrições será detalhada na seção a seguir.

4.2.3 Rate Limiting

A API do Twitter implementa um rigoroso sistema de controle que monitora a frequência e o volume de requisições, estabelecendo limites claros sobre o número de chamadas permitidas. Este mecanismo de limitação é crucial para prevenir o uso abusivo da plataforma, incluindo potenciais ataques de negação de serviço (DDoS) e tentativas de força bruta, garantindo assim a segurança e a estabilidade do serviço. Além de proteger contra tráfego mal-intencionado, esses limites também têm o objetivo de preservar os recursos da plataforma, impedindo que desenvolvedores externos consumam uma quantidade excessiva de largura de banda ou dados, o que poderia levar à replicação indevida ou à criação de serviços similares que concorram diretamente com o Twitter.

A utilização de um agendador para realizar requisições periódicas à API do Twitter neste sistema apresenta o risco de exceder os limites estabelecidos pela plataforma, dependendo de como o agendador está configurado. Ultrapassar esses limites pode resultar em bloqueios temporários impostos pelo Twitter, comprometendo significativamente a eficácia da coleta de dados. Essa questão representa um desafio notável, pois interrupções na coleta podem impactar a análise e o monitoramento de tendências em tempo real. Para evitar esse problema e assegurar a continuidade da coleta de dados, implementou-se um sistema de monitoramento das chamadas à API.

Para o monitoramento eficaz das chamadas à API do Twitter e a gestão dos limites de requisição, o sistema utiliza a biblioteca de monitoramento *Twitter Api Rate Limit Plugin* [5], desenvolvida em *TypeScript*. Um serviço *Singleton* foi criado para centralizar todas as requisições à API, armazenando informações como a quantidade de dados recebidos, o número de requisições realizadas e a frequência dessas solicitações.

[5]Repositório da Biblioteca Twitter Api Rate Limit Plugin: https://github.com/alkihis/twitter-api-v2-plugin-rate-limit

Um serviço *Singleton* é um padrão de design utilizado no software para garantir que uma classe tenha uma única instância em toda a aplicação. Esse modelo é especialmente útil quando diversas partes de um sistema precisam acessar recursos comuns ou compartilhar informações de forma consistente. Ao aplicar esse padrão, o sistema assegura que todas as chamadas a esse serviço sejam direcionadas para a mesma instância, mantendo um estado unificado e evitando a duplicação desnecessária de objetos ou dados. Essa abordagem é particularmente vantajosa em situações que exigem controle centralizado, como o monitoramento e a gestão de requisições a APIs, onde a consistência e a integridade dos dados são cruciais.

Esse serviço analisa cada tentativa de chamada à API, verificando se a ação respeita os limites estabelecidos pela plataforma. Se uma chamada adicional implicar na violação desses limites, o serviço opta por não executar a requisição, evitando assim bloqueios temporários à API e garantindo a continuidade segura e eficiente da coleta de dados. Esse mecanismo de monitoramento e controle é fundamental para a operação ininterrupta do sistema, maximizando a eficácia da coleta de informações sem infringir as políticas de uso da API.

4.3 Monitoramento do Sistema

O propósito desta seção é realizar uma análise sobre os dados acumulados ao longo do período de monitoramento do sistema. Empregaremos o conjunto de informações coletadas para avaliar a eficácia do sistema em capturar dados relevantes durante o intervalo de tempo em que o sistema foi testado, entre a data 21 de agosto de 2022 e 30 de agosto de2023.

4.3.1 Categorias e Termos

Durante o período de testes, o sistema foi configurado para coletar dados relacionados ao tema de *doping*, um assunto relevância no contexto esportivo. Foram cadastrados termos especificamente associados a esportes e substâncias proibidas, visando buscar um conjunto de informações sobre que pudessem revelar padrões,

frequência e a evolução das discussões sobre o uso de substâncias ilícitas no esporte.

Nesse período, o sistema contou com 502 termos cadastrados distribuídos nas quatro categorias a seguir: Substâncias Proibidas, Esportes, Eventos e Outros. No entanto, nem todos os termos contaram com tweets encontrados, apenas 391 termos possuíram tweets encontrados. A Tabela 4.1 apresenta a distribuição de termos cadastrados e tweets encontrados para cada uma das categorias.

Categorias	**N° de Termos Cadastrados**	**N ° de Tweets Encontrados**
Substâncias Proibidas	428	4.053.204
Esportes	65	4.894.625
Eventos	1	57.250
Outros	8	142.224
Total:	**502**	**9.147.303**

Tabela 4.1: Distribução de Termos e Tweets entre as Categorias Cadastradas

É possível observar que, apesar de a categoria de esportes conter menos termos cadastrados em comparação à categoria de substâncias proibidas, ela conseguiu ultrapassar esta última em número de tweets coletados, sendo a categoria com o maior volume de tweets. Ela é seguida pelas categorias de substâncias proibidas, outros e eventos, nesta ordem, ao comparar-se a quantidade de tweets capturados. A categoria eventos, embora tenha apenas um termo cadastrado -*Jogos Sul-americanos 2022*-, conseguiu capturar uma quantidade significativa de tweets durante o período de testes do sistema.

A Figura 4.1 apresenta os dez termos cadastrados que resultaram no maior número de tweets coletados pelo sistema durante o período de testes. Liderando a lista, o termo “futebol” destaca-se com a maior quantidade de tweets, seguido por “cannabis” e “tênis”, cada um acumulando mais de 200 mil tweets. Essa expressiva quantidade de dados abre possibilidade para análises futuras, permitindo um estudo aprofundado sobre as percepções e opiniões dos usuários do Twitter a respeito desses tópicos. Esses insights podem ser extremamente valiosos para entender as tendências de discussão e o sentimento geral em torno desses assuntos na plataforma.

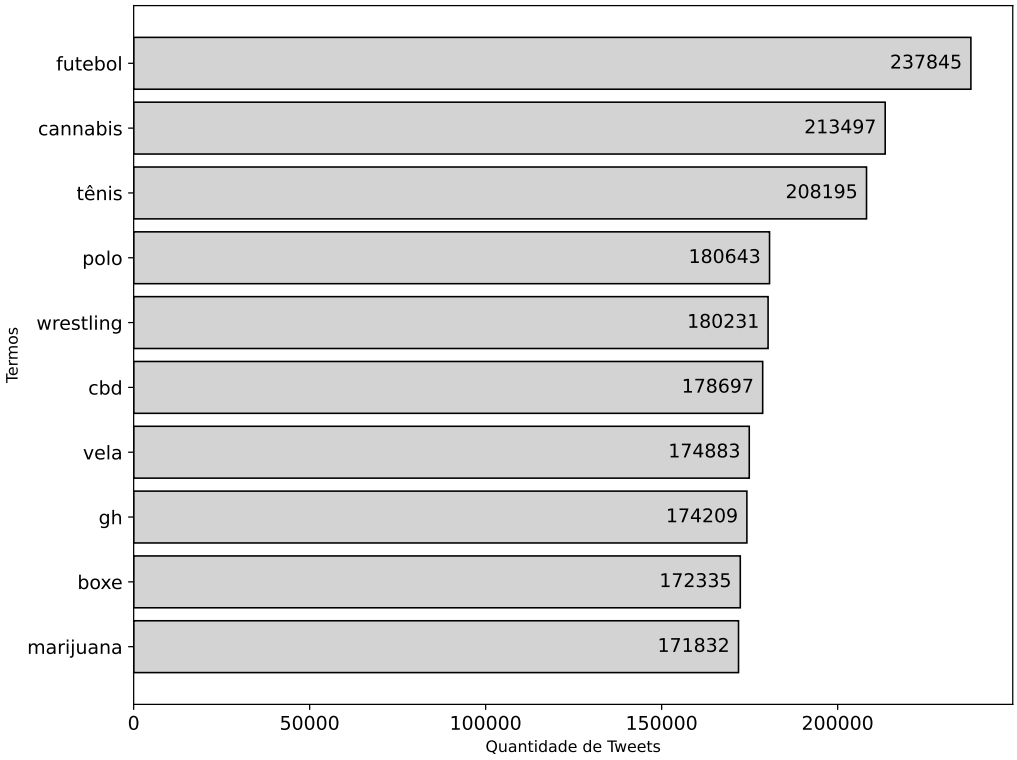

Figura 4.1: Termos com maior quantidade de tweets encontrados

4.3.2 Tweets

Ao longo do período monitorado, entre as datas 21/08/2022 e 30/08/2023, o sistema alcançou uma média aproximada de 830 mil tweets coletados por mês. A Figura 4.2 retrata a distribuição mensal. É importante destacar que esse resultado foi obtido apesar de um início lento, em que os primeiros meses apresentaram um desempenho significativamente menor devido às fases iniciais de teste e ajustes do sistema. Além disso, no mês de agosto de 2022, o projeto esteve em operação por apenas 10 dias.

Dentre os dados coletados pelo sistema, um total de 3.559.216 tweets são identificados como retweets, que são republicações de conteúdos de outros usuários. Esse volume corresponde a aproximadamente 38,9% do total dos tweets capturados, um ponto que merece atenção especial em casos da análise desses dados. A presença de

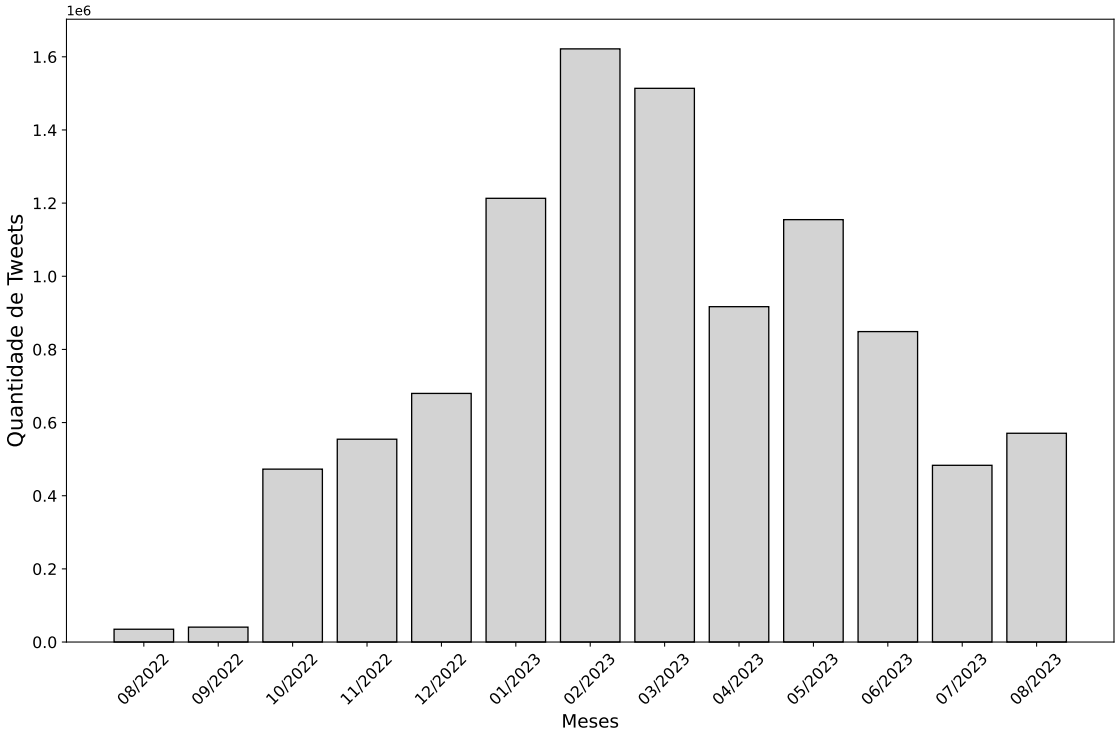

Figura 4.2: Distribuição de Tweets encontrados por mês

retweets indica que uma parcela dos dados consiste em informações repetidas, o que pode influenciar na interpretação dos dados.

Mesmo com a grande taxa de replicação de dados, essa base de dados adquirida durante o periodo de testes demonstrou como o sistema conseguiu buscar uma grande quantidada de tweets, ou seja, grande quantidade de informações de texto e metadados sobre as publicações. Vale ressaltar também que ao resgatar tweets, também são buscadas informações sobre usuários, que será aborada na próxima seção.

Apesar da elevada taxa de replicação de dados, a base de dados coletada durante o período de testes mostra a capacidade do sistema de capturar uma grande quantidade de informações, contando não somente o texto das publicações mas também o conjunto de metadados associados a cada uma delas. É importante destacar que, juntamente com os tweets, o sistema também efetua a coleta de dados detalhados sobre os usuários, aspecto que enriquece ainda mais a base de dados e que será explorado com mais detalhes na próxima seção.

4.3.3 Usuários e Perfis

O sistema incorpora uma funcionalidade de busca de usuários do Twitter, garantindo que, ao identificar um tweet, informações complementares sobre o autor da publicação também sejam coletadas. Cada usuário é registrado no banco de dados uma única vez, entretanto, distintas instâncias do perfil de um mesmo usuário podem ser armazenadas em momentos diferentes, capturando um histórico de dados do perfil. Essa abordagem permite a preservação de informações sobre as variações no perfil do usuário ao longo do tempo, enriquecendo a base de dados com detalhes que possibilitam análises aprofundadas sobre o estado do perfil em momentos específicos, especialmente no contexto da publicação de tweets.

Durante o período de testes, o sistema recuperou 5.305.153 usuários únicos e um total de 11.335.874 perfis desses usuários. A distribuição de usuários e perfis recuperados por mês pode ser observada na Figura 4.3. Nela, percebemos que o número de perfis, naturalmente, ultrapassa o número de usuários, já que cada usuário pode possuir várias atualizações de perfil no sistema.

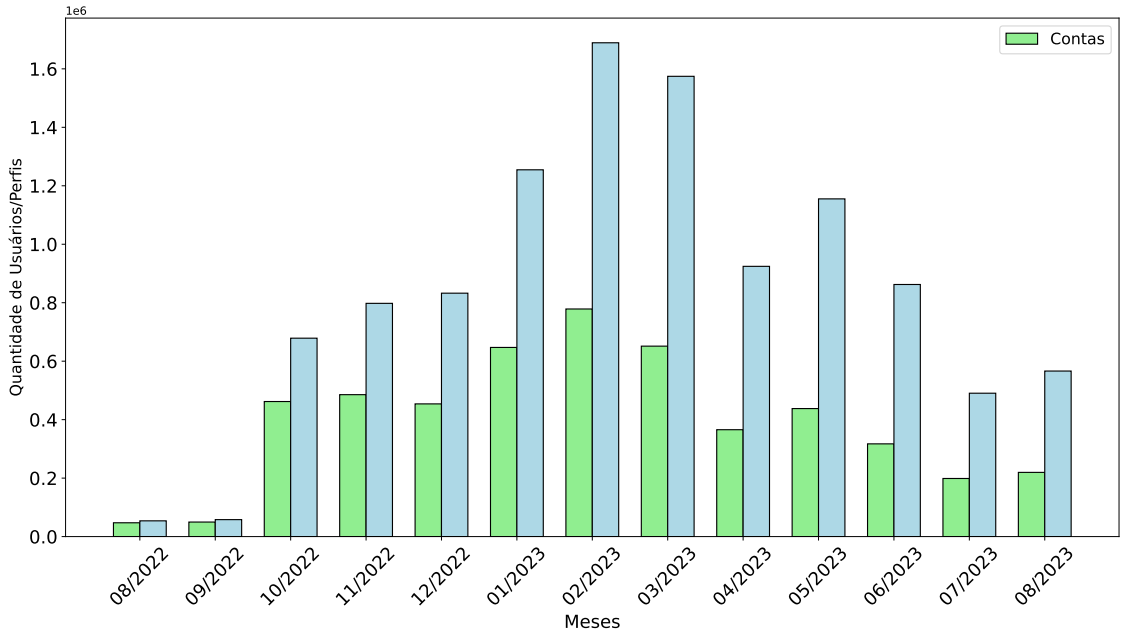

Figura 4.3: Distribuição de Usuários e Perfis por mês

4.3.4 Geolocalização

Durante o processo de busca por tweets, o sistema procura por informações de geolocalização associadas às publicações, sempre que estas estiverem disponíveis. Apesar da quantidade de dados coletados, apenas uma parte dos tweets contém dados de geolocalização, refletindo a natureza opcional dessa característica nas publicações do Twitter. Entre todos os dados capturados, um total de 112.881 tweets foram identificados com informações específicas de geolocalização, sendo apenas 1,2% dos tweets. Durante o período de testes, o sistema conseguiu capturar 13.407 geolocalizações únicas. Esses dados de geolocalização podem ser úteis para análises que buscam compreender padrões geográficos, distribuições regionais de tópicos ou tendências, e o impacto de eventos em diferentes locais.

O gráfico apresentado na Figura 4.4 destaca as geolocalizações com a maior quantidade de tweets capturados pelo sistema. Apesar de haver uma distribuição geográfica variada das publicações, o número de tweets por localização é relativamente baixo quando comparado ao volume total de tweets coletados. É notável que grande parte dessas geolocalizações identificadas encontra-se no Brasil, refletindo os termos cadastrados em grande maioria em português. Essa tendência destaca a importância de considerar fatores locais na interpretação de padrões e tendências nas publicações coletadas.

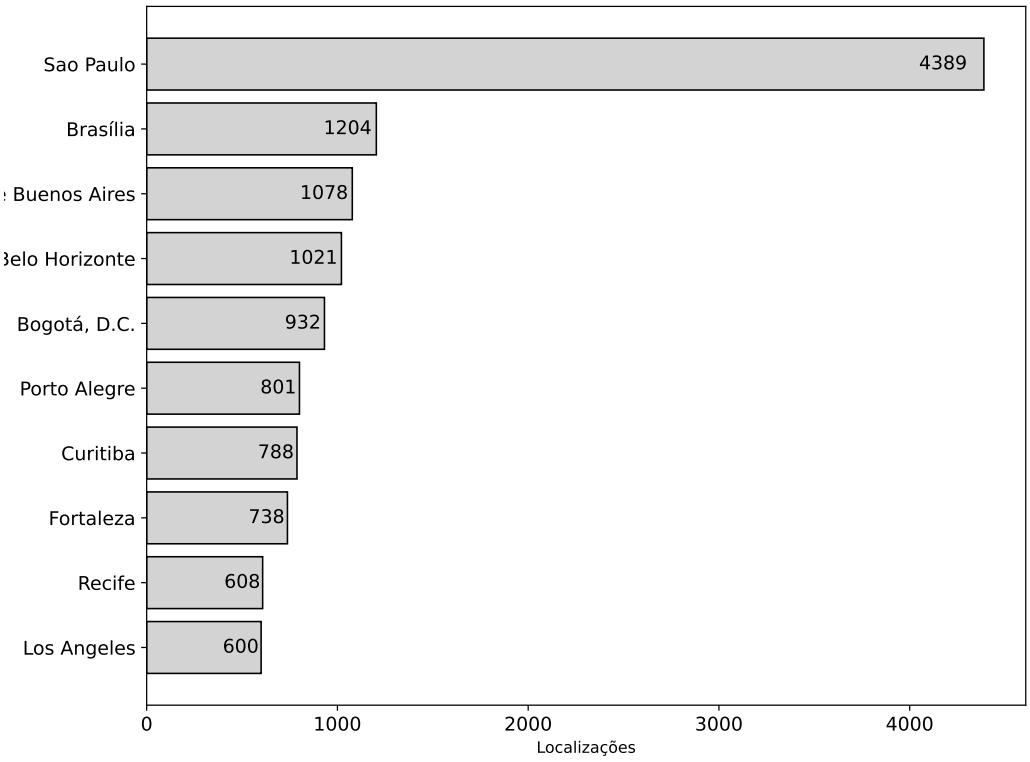

Figura 4.4: Geolocalizações com maiores quantidades de Tweets

Capítulo 5

Conclusão

Este capítulo tem como objetivo apresentar as conclusões finais deste estudo, discutindo sua eficácia, as limitações identificadas e as oportunidades para possíveis pesquisas futuras.

5.1 Considerações finais

Com base nas análises realizadas, observou-se que o sistema foi capaz de recuperar uma grande quantidade de dados referentes a tweets e perfis de usuários mensalmente. O volume de informações coletadas ao longo do período de testes mostra a eficácia do sistema na aquisição de dados. Esta capacidade de coleta ressalta o potencial do sistema em fornecer uma base de dados rica e diversificada, útil para pesquisas e análises profundas que visam explorar as dinâmicas e tendências manifestadas no Twitter.

Uma característica que se destaca no sistema é possibilidade de cadastrar termos específicos, facilitando a busca sobre assuntos particulares, o que se prova extremamente útil para conduzir pesquisas focadas em temas específicos. Por exemplo, durante os testes, o sistema conseguiu capturar 237.845 tweets relacionados a futebol, um conjunto de dados que oferece uma base sólida para estudos e análises sobre a popularidade, o engajamento e as perspectivas dos usuários a respeito desse esporte.

A capacidade de buscar e armazenar perfis de usuários se demonstrou muito eficaz, conseguindo encontrar uma variedade enorme de perfis, sendo possível assimilar perfis que se destacam em certos assuntos ao verificar metadados como quantidade de seguidores e likes e respostas de seus tweets.

A funcionalidade do sistema de buscar e armazenar perfis de usuários mostrou-se bastante eficaz, destacando-se pela capacidade de armazenar uma ampla gama de perfis. Essa característica permite uma análise dos usuários que ganham destaque em determinados temas, através da avaliação de metadados relevantes, como o número de seguidores, a quantidade de curtidas e as respostas recebidas por seus tweets. Esses dados facilitam a identificação de contas influentes em específicas áreas de interesse e também oferece visões sobre o impacto e o alcance de suas publicações.

5.2 Limitações e trabalhos futuros

A principal limitação enfrentada por este sistema é a restrição imposta pela API do Twitter, que limita a quantidade de tweets acessíveis para coleta. Adicionalmente, mudanças recentes implementadas pela plataforma resultaram na redução das possibilidades de coleta de dados para usuários não pagantes, impactando diretamente no custo de utilização do sistema. Essas mudanças tornaram a obtenção de informações mais restrita, dificultando significativamente a recuperação de grandes conjuntos de dados via Twitter API. Isso implica não apenas em desafios operacionais e financeiros para os desenvolvedores e pesquisadores, mas também na possibilidade de limitação na profundidade e na qualidade das análises realizadas com dados do Twitter.

Uma possível melhoria para superar a limitação imposta pela restrição de dados da API do Twitter envolve a implementação de um mecanismo que permita o cadastro de múltiplas keys de API. Isso pode ampliar a quantidade de dados acessíveis dentro de um determinado intervalo de tempo, já que diferentes keys podem ser utilizadas para contornar os limites de requisição impostos por cada chave. Ao distribuir as solicitações de coleta de dados entre várias keys de API, o sistema poderia aumentar seu alcance e capacidade de coleta, possibilitando a aquisição de um volume maior

de informações e superando o impacto das restrições da API.

Outra limitação que pode ser enfrentada pelo sistema é a necessidade de se adaptar às mudanças na API do Twitter para manter a compatibilidade com a interface da plataforma, caso ela seja alterada. Esse desafio ficou evidente com a transição da Twitter API v1.1 para a versão 2, uma atualização significativa que implicou na revisão e na adaptação do sistema para assegurar a continuidade da coleta de dados sem interrupções. A ausência de adaptações pode resultar em períodos de inatividade ou na perda de funcionalidades críticas, comprometendo a eficácia da coleta de dados.

Uma outra possível melhoria futura é a liberação da base de dados para ser acessada por outros pesquisadores e desenvolvedores para que possam realizar suas próprias análises. Essa liberação precisaria ser controlada por meio de credenciais de acesso

Uma possível melhoria para o futuro do projeto é a disponibilização da base de dados coletada pelo sistema para outros pesquisadores e desenvolvedores, possibilitando a realizaçãio de suas próprias análises e exploração dos dados. No entanto, para garantir a segurança e a privacidade dos dados, essa liberação deveria ser controlada através de um sistema de credenciais de acesso. Esse modelo de acesso controlado pode servir como um ponto de partida valioso para uma diversidade de pesquisas, fornecendo uma base sólida para a exploração de dados.

No futuro, a integração do sistema com um módulo de mineração de dados pode representar uma evolução significativa, permitindo a análise e a extração automatizada de informações valiosas a partir dos dados coletados. Essa adição transformaria o sistema em uma solução autossuficiente, capaz de não apenas coletar uma grande quantidade de dados de maneira eficiente, mas também de interpretá-los para identificar padrões e tendências sem a necessidade de grandes intervenções manuais. Essa funcionalidade ampliada potencializaria o valor do sistema, facilitando a realização de análises complexas e a geração de conhecimento de forma ágil. Com essa capacidade de autoanálise, o sistema poderia servir a uma gama ainda maior de propósitos e aplicações, desde pesquisas acadêmicas até soluções empresariais,

reforçando sua utilidade como uma ferramenta poderosa na compreensão dos dados gerados nas redes sociais.

Referências

ABKENARI, F. A.; SELAMAT, A. A clickstream-based focused trend parallel web crawler. *International Journal of Computer Applications*, v. 9, p. 1–8, 2010. Disponível em: <https://api.semanticscholar.org/CorpusID:2006164>.

BASTOS, V. M. *Ambiente de Descoberta de Conhecimento na WEB Para a Língua Portuguesa*. Tese (Doutorado) — Curso de Ciências em Engenharia Civil, Universidade Federal do Rio de Janeiro, UFRJ, 2006.

BOsNJAK, M. et al. Twitterecho: A distributed focused crawler to support open research with twitter data. In: *Proceedings of the 21st International Conference on World Wide Web*. New York, NY, USA: Association for Computing Machinery, 2012. p. 1233–1240.

BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, v. 30, p. 107–117, 1998. Disponível em: <http://www-db.stanford.edu/~backrub/google.html>.

BYUN, C.; LEE, H.; KIM, Y. Automated twitter data collecting tool for data mining in social network. In: *Proceedings of the 2012 ACM Research in Applied Computation Symposium*. New York, NY, USA: Association for Computing Machinery, 2012. p. 76–79.

CARDOSO, O. N. P.; MACHADO, R. T. M. Gestão do conhecimento usando data mining: estudo de caso na universidade federal de lavras. *Revista de Administração Pública*, Fundação Getulio Vargas, v. 42, n. 3, p. 495–528, May 2008. ISSN 0034-7612. Disponível em: <https://doi.org/10.1590/S0034-76122008000300004>.

CLAUSSEN, J.; PEUKERT, C. Obtaining data from the internet: A guide to data crawling in management research. *SSRN Electronic Journal*, 01 2019.

DR.P.PONMUTHURAMALING, S. A study on semantic web mining and web crawler. *International Journal of Engineering and Computer Science*, v. 2, n. 09, 2013. Disponível em: <https://www.ijecs.in/index.php/ijecs/article/view/1845>.

GOLD, Z.; LATONERO, M. Robots welcome: Ethical and legal considerations for web crawling and scraping. *Wash. JL Tech. & Arts*, HeinOnline, v. 13, p. 275, 2017.

HEYDON, A.; NAJORK, M. Mercator: A scalable, extensible web crawler. *World Wide Web*, v. 2, p. 219–229, 1999. Disponível em: <https://api.semanticscholar.org/CorpusID:207736356>.

LIU, Y.; FENG, G.; SUN, Y. To provide api or not? an analysis of the optimal anti-crawler strategy. *An Analysis of the Optimal Anti-crawler Strategy (March 15, 2023)*, 2023.

Ministério das Comunicações. *90% dos lares brasileiros já tem acesso à internet no Brasil, aponta pesquisa*. 2022. Disponível em <https://www.gov.br/casacivil/pt-br/assuntos/noticias/2022/setembro/90-dos-lares-brasileiros-ja-tem-acesso-a-internet-no-brasil-aponta-pesquisa>. Acesso em 12 de Maio 2023.

PRATIKAKIS, P. *twAwler: A lightweight twitter crawler*. 2018.

RIBEIRO, L. C. R. Web crawler em java: Coleta e filtragem de conteúdo da web. *MundojÁgil*, v. 1, n. 59, 2013.

Simon Kemp. *Twitter Statistics And Trends*. 2022. Disponível em <https://datareportal.com/essential-twitter-stats>. Acesso em 08 de Fevereiro 2023.

SOHAIL, S. S. et al. Crawling twitter data through api: A technical/legal perspective. *ArXiv*, abs/2105.10724, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:235166326>.

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.