UNIVERSIDADE FEDERAL FLUMINENSE

ALLAN CEZAR ALMEIDA CHAVES

ROBERTO RABELO GONZAGA JUNIOR

WEB SCRAPING: EXTRAÇÃO DE TWEETS COM OPINIÕES DE

MEDICAMENTOS PARA HIV

Niterói

2021

ALLAN CEZAR ALMEIDA CHAVES

ROBERTO RABELO GONZAGA JUNIOR

WEB SCRAPING: EXTRAÇÃO DE TWEETS COM OPINIÕES DE

MEDICAMENTOS PARA HIV

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Orientador:

ALTOBELLI DE BRITO MANTUAN

NITERÓI

2021

**ALLAN CEZAR ALMEIDA CHAVES**

ROBERTO RABELO GONZAGA JUNIOR

WEB SCRAPING: EXTRAÇÃO DE TWEETS COM OPINIÕES DE

MEDICAMENTOS PARA HIV

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Niterói, \_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_\_\_ de 2021.

Banca Examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Altobelli de Brito Mantuan, MSc. – Orientador

UFF – Universidade Federal Fluminense

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profa. Simone de Lima Martins, Dr. – Avaliadora

UFF - Universidade Federal Fluminense

ALLAN

Dedico este trabalho à toda comunidade acadêmica e científica, que realiza um trabalho excepcional para o bem-estar e evolução de nossa sociedade.

ROBERTO

Dedico este trabalho a toda comunidade de "loucos" que acreditam na área da computação e tecnologia.

Àqueles que dormem e acordam no meio da madrugada com soluções inesperadas.

Àqueles que confiam na arte como veículo transformador dentro da nossa sociedade.

Àqueles que dedicam a entender binário nas linhas coloridas de um monitor.

**AGRADECIMENTOS**

**ALLAN**

Ao meu orientador, Altobeli de Brito Mantuan, pelo empenho dedicado neste projeto e pelo apoio dado a mim durante este trabalho.

Aos tutores do consórcio CEDERJ, por todo empenho ao ensino e dedicação à aprendizagem dos alunos.

Aos colegas de curso pelo incentivo e troca de experiências.

A todos os meus familiares e amigos pelo apoio e colaboração.

AGRADECIMENTOS

ROBERTO

Agradeço meus pais(Janaina e Roberto) por sempre me impulsionarem em acreditar nos meus sonhos. Por me transferir valores familiares que edificaram o ser humano que sou hoje, por me explicarem que tudo será difícil, mas que "trabalho duro vence o dom natural".

Aos meus amigos de graduação (Allan, Hugo, Mateus, Maria), que em diversos momentos demostraram o quão forte e transformador pode ser o poder de um amigo quando se estar dentro de uma longa caminhada.

Ao meu irmão Pedro, que não se encontra mais nesse plano, porém sempre enxergou minhas qualidades e me fortaleceu para que eu pudesse desenvolvê-las. Você também é parte disso.

Ao Alan Mathison Turing, grande matemático considerado pai da computação que iniciou os primórdios dessa arte chamada computação e com ela nos mostrou um mundo inteiramente novo e cheio de possibilidades.

“Os que se encantam com a prática sem a ciência, são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino.”

Leonardo da Vinci

RESUMO

Trazendo uma nova forma de se comunicar e partilhar informações, a *internet* vem sendo uma fonte inesgotável e muito expressiva dentro da nossa sociedade, influenciando a nossa forma de gerar e consumir informações. Dentro desse contexto, com uma quantidade massiva de dados, também surgiram novas formas de manipular, tratar e armazenar esses dados. Entre elas, temos uma ferramenta denominada web scraping que via código é possível automatizar e tratar esse processo de captura dos dados. Neste trabalho procurou-se mostrar a implementação da ferramenta de busca, tendo como alvo opiniões sobre os medicamentos para o tratamento do vírus HIV, contando com auxílio de uma especialista farmacêutica para definições de palavras chaves dentro dessa busca, realizada na rede social Twitter, onde conseguimos capturar entre os anos 2015 e 2020, um total de 3.809.864 tweets.

Palavras-chaves: Web Scraping, rede social, Twitter, HIV.

LISTA DE ILUSTRAÇÕES

[Figura 1: Tipos de dados 24](#_Toc76694239)

[Figura 2: Tipos de dados qualitativos. 25](#_Toc76694240)

[Figura 3: Tipos de dados quantitativos 26](#_Toc76694241)

[Figura 4: Tipos organizações dos dados [13] 26](#_Toc76694242)

[Figura 5: Modelo da Tag HTML 29](#_Toc76694243)

[Figura 6: Estrutura básica do documento HTML 30](#_Toc76694244)

[Figura 7: Exemplo de um documento HTML 5 31](#_Toc76694245)

[Figura 8 : Exemplo de Website estruturado semânticamente 31](#_Toc76694246)

[Figura 9: Estrutura em árvore gerada pelo DOM 33](#_Toc76694247)

[Figura 10: Processo de requisição HTTP pelo navegador 34](#_Toc76694248)

[Figura 11: Aplicação *Web Scraping* 36](#_Toc76694249)

[Figura 12 Tipo de abordagem API 39](#_Toc76694250)

[Figura 13: Diagrama do Caso de Uso Requisitar Tweets 42](#_Toc76694251)

[Figura 14: Diagrama de Classes 45](#_Toc76694252)

[Figura 15: Mapa de Compreensão de domínios e subdomínos 47](#_Toc76694253)

[Figura 16: Resposta de uma requisição à API do Twitter 49](file:///C:\Users\Allan\Documents\TCC_CEDERJ_2021_ALLAN_ROBERTO\TCC_Roberto_Rabelo_Allan_Chaves.docx#_Toc76694254)

[Figura 17: Esquema de uso da API do Twitter 50](#_Toc76694255)

[Figura 18: Aplicativo Postman - *Client* para APIs 52](#_Toc76694256)

[Figura 19: Detalhes dos erros de requisições à API do Twitter 53](file:///C:\Users\Allan\Documents\TCC_CEDERJ_2021_ALLAN_ROBERTO\TCC_Roberto_Rabelo_Allan_Chaves.docx#_Toc76694257)

[Figura 20 – Arquivo de configuração dos parâmetros do aplicativo 55](#_Toc76694258)

[Figura 21 : Arquivo de palavras-chave da busca 56](#_Toc76694259)

[Figura 22 : Configuração da variável de ambiente *Bearer Token* 57](#_Toc76694260)

[Figura 23: Exemplo do arquivo de log gerado para as buscas 58](file:///C:\Users\Allan\Documents\TCC_CEDERJ_2021_ALLAN_ROBERTO\TCC_Roberto_Rabelo_Allan_Chaves.docx#_Toc76694261)

LISTA DE TABELAS

[Tabela 1: Tipos de dados não estruturados 28](#_Toc76114297)

[Tabela 2 Requisitos *RESTFull* 40](#_Toc76114298)

[Tabela 3: Lista de Endpoints da API - Versão 2 41](#_Toc76114299)

[Tabela 4: Descrição do Caso de uso “*Twitter API SEARCH*” 43](#_Toc76114300)

[Tabela 5: Tabela de descrição das classes 46](#_Toc76114301)

[Tabela 6: Limitações da conta *Academic Research* 48](#_Toc76114302)

[Tabela 7: Componentes de uma requisição ao endpoint Search da API do Twitter 50](#_Toc76114303)

[Tabela 8: Respostas HTTP da API do Twitter 52](#_Toc76114304)

[Tabela 9: Configurações de hardware utilizado 54](#_Toc76114305)

[Tabela 10: Pacotes Python instalados(as) 54](#_Toc76114306)

LISTA DE GRÁFICOS

[Gráfico 1: Total de buscas por ano 59](#_Toc76114307)

[Gráfico 2: Gráfico anual de *Tweets* por busca(1/3) 59](#_Toc76114308)

[Gráfico 3: Gráfico anual de *Tweets* por busca(2/3) 60](#_Toc76114309)

[Gráfico 4: Gráfico anual de *Tweets* por busca(3/3) 60](#_Toc76114310)

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interfade

BOT - Acrônimo para a palavra em inglês ROBOT

CSS - Cascade Style Sheet

DNS - Domain Name System

DOM - Document Object Model

HIV - Human Immunodeficiency Virus

HTML - HyperText Markup Language

HTTP - hypertext transfer protocol

IDE - Integrated Development Environment

IP - Internet Protocol

JSON - JavaScript Object Notation

LOG - Logging

ODT - Object Data Transfer

PNL - Precisiated Natural Language

POO - Programação Orientada a Objetos

RAM - Random Access Memory

RAM - Reações Adversas a Medicamentos

SGBD - Sistema Gerenciador de Banco de Dados

TAG - Acrônimo para a palavra em inglês tag

TCP - Transmission Control Protocol

TXT - Acrônimo para a palavra em inglês text

URI - Uniform Resource Identifier

URL - Uniform Resource Locator

WEB – World Wide WEB

XML - Extensible Markup Language

SUMÁRIO

[RESUMO 9](#_Toc76686708)

[LISTA DE ILUSTRAÇÕES 10](#_Toc76686709)

[LISTA DE TABELAS 11](#_Toc76686710)

[LISTA DE GRÁFICOS 12](#_Toc76686711)

[LISTA DE ABREVIATURAS E SIGLAS 13](#_Toc76686712)

[1 INTRODUÇÃO 17](#_Toc76686713)

[2 TRABALHOS RELACIONADOS 19](#_Toc76686714)

[3 FUNDAMENTAÇÃO TEÓRICA 23](#_Toc76686715)

[3.1 COLETA DE DADOS 23](#_Toc76686716)

[3.2 TIPOS DE DADOS 24](#_Toc76686717)

[3.2.1 QUALITATIVO 25](#_Toc76686718)

[3.2.2 QUANTITATIVO 25](#_Toc76686719)

[3.3 ORGANIZAÇÃO E ARMAZENAMENTO 26](#_Toc76686720)

[3.4 PÁGINAS WEB 28](#_Toc76686721)

[3.4.1 DOCUMETNOS HTML E TAGS 29](#_Toc76686722)

[3.4.2 DOCUMENT OBJECT MODEL (DOM) 32](#_Toc76686723)

[3.5 NAVEGADORES WEB 34](#_Toc76686724)

[3.6 WEB SCRAPING 35](#_Toc76686725)

[4 DESENVOLVIMENTO 37](#_Toc76686726)

[4.1 TWITTER API 37](#_Toc76686727)

[4.1.1 CONCEITOS DE API RESTFUL 39](#_Toc76686728)

[4.1.2 ENDPOINTS DA TWITTER API VERSÃO 2 41](#_Toc76686729)

[4.2 CASOS DE USO 42](#_Toc76686730)

[4.2.1 DIAGRAMA CASO DE USO "TWITTER API SEARCH" 42](#_Toc76686731)

[4.3 DIAGRAMAS DE CLASSES 45](#_Toc76686732)

[4.4 EXTRAÇÃO DE TWEETS COM OPINIÕES DE MEDICAMENTOS PARA HIV 47](#_Toc76686733)

[4.4.1 CONTA ACADEMIC RESEARCH 48](#_Toc76686734)

[4.4.2 DICIONÁRIO DE DADOS 49](#_Toc76686735)

[4.4.3 USO DA TWITTER API 50](#_Toc76686736)

[5 TESTES 54](#_Toc76686737)

[6 CONCLUSÕES E TRABALHOS FUTUROS 61](#_Toc76686738)

[REFERÊNCIAS BIBLIOGRÁFICAS 62](#_Toc76686739)

[ANEXOS 66](#_Toc76686740)

[ANEXO A – DOCUMENTO DE ACEITAÇÃO ASSINADO PELO ESPECIALISTA 66](#_Toc76686741)

[ANEXO B – DOCUMENTO DE DECLARAÇÃO DE TAGS DEFINIDAS PELO ESPECIALISTA 67](#_Toc76686742)

[ANEXO C – PREECHIMENTO DE FORMULÁRIO PARA APROVAÇÃO DA CONTA ACADEMIC RESEARCH. 68](#_Toc76686743)

# INTRODUÇÃO

É extremamente notável o extenso volume de informação que vem sendo produzido e consumido dentro da *internet*. Podemos notar uma infinidade de informação gerada para determinados assuntos.

Observando esse montante e agregando à função que as universidades têm assumido nos últimos tempos em contribuir com inovações e influenciar ativamente nas atividades humanas dentro da sociedade, se faz presente a necessidade de pesquisas científicas como grande fator de apoio na criação de novas tecnologias.

O *Twitter,* uma grande rede social, foi criado em 2006 e tem como principal objetivo ser um miniblog pessoal. Nele, podemos contar nossas experiencias diárias e instantâneas de forma rápida e informal, com versões em cerca de 35 idiomas e 316 milhões de usuários ativos . Ele vem sendo um dos contribuintes para esse enorme volume de informação gerada na *internet*, pois é um ambiente completamente propício para extração e análise das opiniões públicas sobre determinado assunto. Neste trabalho demos ênfase sobre informações de medicamentos sobretudo os que permeiam o vírus HIV. [1]

Para extração de conhecimento dentro dessa gama de informação, contamos com uma solução que nos auxilia otimizar a busca e a extração de informações de forma mais precisa, chamada de *web scraping.* [2]

Este trabalho implementa um software capaz de automatizar processos de busca, extração e organização de dados específicos, previamente definidos, que apoiam, de maneira relevante, decisões tomadas por farmacêuticos. Dentre as contribuições deste projeto podemos destacar*.*

* Desenvolvimento de *script* para automatização e tratamento das consultas estabelecidas. Foram feitos encontros virtuais com um pesquisador da área de farmácia para definições de parâmetros (*Tag*) a serem buscadas dentro do twitter*.*
* Definições do MongoDB (Banco de dados não relacional) para armazenamento, organização e consulta.

Todo material pertinente a este trabalho estará disponível em um repositório público no GitHub podendo ser acessado pelo endereço a seguir: https://github.com/altobellibm/CEDERJ\_2021\_ALLAN\_ROBERTO

# TRABALHOS RELACIONADOS

Muito se tem discutido, nos últimos anos, sobre o avanço da tecnologia e a quantidade massiva de dados criada na internet. Com o passar do tempo, começamos a perceber essa base como uma fonte infinita de informações. Ao manipular dados em massa, surgiram muitos desafios, tais como, a extração de informações relevantes, o tratamento apropriado e a análise correta. Sendo estes, processos importantes para obtenção de resultados precisos e opiniões sólidas para os interessados nestas informações [3].

Uma das formas de coletar dados na World Wide Web, é através do Web scraping*,* ou raspagem da web, traduzido para o português*.* Esta *é* uma técnica utilizada para extrair e combinar informações desejadas em sites de interesse. Esse processo é implementado com o uso de web BOTs, que extraem informações específicas, reestruturam e armazenam estes dados. Dessa forma, pode-se otimizar o método convencional de coleta de dados, onde alguém pesquisa em diversos documentos web, extrai e organiza manualmente as informações [4].

No trabalho trabalho relacionado [4], de Audeh et al(2017), os autores criam um framework de código aberto chamado Vigi4Med Scraper designado a extrair dados estruturados de fóruns da web e representá-los em uma estrutura semântica. Este framework faz parte de um projeto, da agência nacional de segurança de medicamentos Francesa, que busca detectar as reações adversas a medicamentos (RAM) em fóruns da web.

O Vigi4Med foi dividido em três blocos principais: extração dos dados, representação semântica e anonimização dos dados. Todo o framework é completamente configurável, permitindo seu uso em qualquer site de fórum. Além desta flexibilidade, os autores destacam outras vantagens em comparação com outros projetos, como eficiência, paginação dos resultados, detecção de objeto de dados retornado, representação conceitual, privacidade e disponibilidade. O framework foi capaz de extrair dados de mais de 200 gigabytes de dados, colhidos de 22 sites de fórum na web.

Em um segundo trabalho relacionado [5], os pesquisadores coletaram mais de 60 milhões de posts, extraídos de 21 fóruns web franceses. Para essa extração de informações, foi utilizado o framework Vigi4Med [3].

Foi feita uma correlação entre os medicamentos cadastrados em um banco de dados de fármacos francês, chamado *Theriaque*®, e as menções à fármacos, coletadas com o uso do framework Vigi4Med [3].

O objetivo principal neste trabalho [5] foi avaliar a usabilidade de menções à fármacos, através de fóruns web, em diversos estudos relacionados à farmacovigilãncia.

De forma que, para cumprir o objetivo, os pesquisadores analisaram quais eram os fármacos mais mencionados em fóruns na web. Também analisaram a hipótese das mídias tradicionais influenciarem a quantidade de menções nos fóruns. Ademais, outra análse importante foi correlacionar os fármacos mais prescritos na França com os mais mencionados nos fóruns web franceses. Segundo os autores, poucos trabalhos relacionados ao tema possuem esta abordagem.

O estudo [5] pôde elucidar, através de diversos gráficos estatísticos, o perfil de usuários dentre os 50 fármacos mais mencionados em fóruns franceses. Evidenciou-se que, mulheres entre 18 e 45 anos, correspondem à maior parcela de usuários dos fóruns. Também expôs o fato que entre os principais usuários dos fármacos mais mencionados estão os jovens.

Além de rastrear os perfís dos usuários dos fóruns, o estudo [5] constatou que os fármacos mais mencionados, correlacionam-se fracamente com os medicamentos mais prescritos na França, de acordo com a base de dados da *French Health Insurance System* (Sistema de plano de saúde Francês, traduzido para o Português).

Em [6] abordou-se que as Reações Adversas a Medicamentos (RAM), tornaram-se o foco dos sistemas de saúde públicos. Foi debatido como o uso de mídias sociais pode ser usado para coleta de dados na farmacovigilância, que se mostrou promissor devido ao fato de que os dados coletados serem em tempo real e por proverem informações sobre os hábitos dos consumidores.

Os autores explicam que ao se usar a API do Twitter como ferramenta de extração de dados para pesquisas, podem-se obter diversos benefícios, como: grande parte das informações serem disponibilizadas gratuitamente e o grande volume de dados obtidos. Portanto, eles consideraram o Twitter como uma ótima ferramenta para se alcançar o objetivo da pesquisa, que é identificar evidências sobre as reações adversas a fármacos nos usuários.

Dessa forma, os autores focaram em identificar automaticamente tweets que continham evidências pessoais (em primeira mão) sobre RAMs pelos usuários de dois tipos de fármacos. Contudo, os autores discorrem sobre uma das maiores dificuldades em se trabalhar com dados do Twitter, que é distinguir entre experiências pessoais das não pessoais e outros tipos de informações que mencionam os fármacos.

Assim sendo, para facilitar essa seleção, os pesquisadores utilizaram um conjunto de comentários que mencionavam apenas dois tipos de fármacos, escolhidos na pesquisa e eram escritos somente no idioma inglês.

O estudo [6] foi dividido em 3 fases distintas, a saber: pré-processamento, comentários e processamento.

No pré-processamento ocorreram 3 etapas, conforme listado abaixo:

1. Extração de um conjunto de tweets públicos com 420.983.64 mensagens.
2. Seleção de dados, selecionando apenas os tweets que mencionavam os fármacos de interesse.
3. Filtragem de dados, que buscou balancear as amostras entre os 11 fármacos selecionados na segunda etapa. Essa fase resultou em um conjunto com 1548 tweets.

Já na fase de comentários, foram criadas guias diretrizes para comentaristas leigos e experientes. Então, com o uso das diretrizes para comentaristas experientes, obteve-se um conjunto de 496 tweets de alto padrão de qualidade. Já com o uso de diretrizes para comentaristas leigos, obteve-se um conjunto de 899 tweets.

Por fim, na terceira fase de processamento, foram extraídos alguns atributos, que permitiram aos pesquisadores encontrar 356 tweets que relatavam experiência em primeira mão por parte dos usuários. Com esse último conjunto de dados, foram extraídos atributos específicos para o treinamento de modelos de aprendizado de máquina.

No quarto trabalho relacionado [7], o estudo objetivou apontar os principais efeitos colaterais ocorridos em determinados tratamentos de HIV. De modo que, para alcançar este objetivo, foram utilizados posts do Twitter, coletados entre os anos de 2010 e 2013.

A coleta de tweets foi feita com a aplicação de um conjunto de palavras-chave. Então, foi gerada uma base de dados inicial de 39.988.306 tweets. Após a aplicação de sucessivos filtros, previamente analisados e definidos, chegou-se a um grupo de 37.337 tweets.

A seleção de posts focou em um grupo específico da comunidade de usuários, que faziam menções às suas experiências pessoais com os fármacos. Para essa seleção, os pesquisadores utilizaram um serviço de colaboração coletiva, onde os tweets foram analisados e modelos foram criados para treinar os algoritmos de aprendizado de máquina.

Em seguida, esses algoritmos foram responsáveis pela seleção automática apenas dos tweets de usuários que eram portadores de HIV, que relatavam seus sintomas, opiniões e sentimentos.

Sendo assim, com a obtenção de tweets produzidos por um grupo específico, foi possível resumir as reações adversas causadas pelos fármacos escolhidos previamente. Constatou-se, que as principais reações estão relacionadas a problemas do sono. E 10% dos tweets relevantes não mencionavam reações adversas.

Ademais, o estudo em [7] pôde realizar uma análise de sentimentos em função dos dias, durante o intervalo da extração de tweets. Os pesquisadores, através de gráficos descritivos, expuseram que as análises de sentimentos tendem para valores negativos.

# FUNDAMENTAÇÃO TEÓRICA

Indubitavelmente vivemos em um mundo onde a hiper conectividade cresceu exponencialmente, graças à revolução digital. Com o surgimento de tecnologias como a *internet* das coisas, computação em nuvem, dispositivos móveis, e conexões com a *internet* de alta velocidade, passou-se a gerar uma quantidade massiva de dados [8]. “Por exemplo, o Twitter sozinho gera diariamente doze terabytes de tuítes”. [8, p. 14]

Em contrapartida, surgiram tecnologias e mecanismos para armazenamento e apuração desses dados, cada vez mais eficientes e precisos. Neste capítulo, serão abordados os processos de coleta de dados, armazenamento e organização.

## COLETA DE DADOS

Para análise de dados, temos como etapa inicial a coleta de dados. Com ela, conseguimos avaliar resultados, prever hipóteses, e levantar questionamentos sobre determinados temas. Esta etapa da pesquisa é comum a todas as áreas de estudo, podendo variar os métodos da coleta a partir da área solicitada, mas a filosofia sobre uma boa coleta de dados se mantém [9].

Uma coleta de dados sistemática define a qualidade de um estudo científico. As formas tradicionais de coleta de dados não conseguem gerar um resultado em tempo real, são caras e não acompanham as evoluções de uma população. Logo, pesquisas realizadas pela *internet* se mostram como a preferência entre os sujeitos dos estudos [10].

A coleta de dados pode se dar através de três fontes: primária, secundária e terciária [11].

A fonte primária pode ser definida como fonte original. É aquela onde não houve nenhuma análise anterior ou dado anterior a ela. Temos como exemplo as correspondências, pesquisa DataFolha, pergaminhos.

Já a fonte secundária, parte do princípio de que os dados já sofreram algum tipo de apuração ou foram modificados para facilitar seu consumo. Como por exemplo, sínteses, enciclopédias, dicionários.

Por último, a fonte terciária funciona como caminho ou guia para encontrar fontes primárias e secundárias ou uma soma das duas fontes de pesquisa. Alguns exemplos são guias de leituras, catálogos e manuais.

## TIPOS DE DADOS

Vimos que as fontes dos dados podem ser de 3 tipos (Primário, Secundário, Terciário) e para esses dados coletados, de acordo com suas respectivas origens, eles são separados em dois grupos: qualitativo e quantitativo. Estes, por sua vez, se subdividem em nominais e ordinais para qualitativo, e discreto e contínuo para quantitativo [11]. Essa divisão é ilustrada na Figura 1.



Figura 1: Tipos de dados

### QUALITATIVO

Dados qualitativos são aqueles que não possuem conteúdo numérico definido. Podem possuir diversas categorias, que são qualidades a serem classificadas pelo indivíduo, que recebe a informação [12]. Podemos ter diversas respostas para cada dado.

Podemos classificá-los como:

**Nominais** – Não existe uma ordenação entre as categorias.

**Ordinais** – Existe uma ordenação entre as categorias.

Na Figura 2, temos alguns exemplos das categorias. Observamos que nenhum deles são de teor numérico, ou seja, expressam quantidade [12]

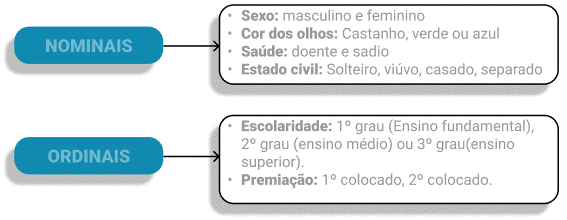


Figura 2: Tipos de dados qualitativos.

### QUANTITATIVO

Em contrapartida, conforme exibido na Figura 3,os dados que são representados como quantitativos tem como função trazer um conteúdo numérico em seu conteúdo, a fim de quantificar alguma questão [12].

Podemos classificá-los como:

**Discretos** – Conjunto com valores finitos e enumeráveis.

**Ordinais** – Valor dentro do intervalo de um conjunto infinito

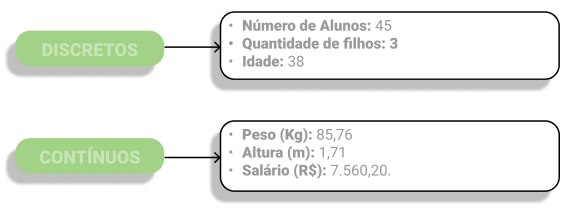


Figura 3: Tipos de dados quantitativos

Temos sempre um número como resposta central da informação a ser obtida.

## ORGANIZAÇÃO E ARMAZENAMENTO

Após a coleta dos dados, precisamos armazená-los e apresentá-los da melhor maneira possível. Os dados seguem o fluxo para serem organizados em três formas: Estruturados, semiestruturados e não-estruturados. Na Figura 4, segue uma ilustração da organização desses dados [12].



Figura 4: Tipos organizações dos dados [13]

Para os dados estruturados temos uma formatação bem definida com regras bem específicas e já estabelecidas antes do armazenamento dos registros.

Exemplos de dados estruturados são os bancos de dados relacionais, pois contém tabelas organizadas com linhas e colunas representadas por estruturas com atributos. Possuem um esquema de dados fixos e com pouca flexibilidade para mudanças.

* **Placa** – Atributo que representa um conjunto de caracteres alfanuméricos, com posições definidas e separados por um hífen.   
  Exemplo: KMO-0001, POX-1234.
* **Cor** – Atributo que representa uma cor através de um conjunto de nomes de cores definidos. Exemplo: azul-escuro, verde, vermelho, etc. .
* **Ano** – Atributo que representa um conjunto de números mutáveis.
* **Chassi** – Atributo que representa um conjunto de números imutáveis

O conjunto de valores, desses atributos específicos, representam um registro de um veículo banco de dados. Cada conjunto representa uma instância de um veículo [12].

Os dados semiestruturados não possuem, de certa forma, uma regra ou predefinição em relação ao conteúdo. Porém, existem algumas estruturas que definem um padrão na sua construção. Possuem grande flexibilidade para alteração de seu escopo. Alguns exemplos são os arquivos JSON e XML que possuem *tags* ou chaves para delimitar os dados e possuem uma forma mais amigável de visualização e manipulação dos dados. [14].

Diferentemente dos dados estruturados e dos semiestruturados, os dados não estruturados não possuem nenhum tipo de ordem ou regra para serem armazenados. Conforme exemplificado na Tabela 1, são dados que necessariamente não possuem um formato ou tamanho máximo.

Tabela 1: Tipos de dados não estruturados

|  |  |
| --- | --- |
| Arquivos | Descrição |
| Arquivo TXT | Não possuem limitação, podem ter vários idiomas, tamanhos, formatação e temas |
| Arquivo Imagem | Pode conter vários formatos, tamanhos em pixels, quantidade de cores, tamanhos em bytes |
| Arquivo Vídeo | Pode conter vários formatos, tamanhos e conteúdos |

## PÁGINAS WEB

O objetivo inicial da web foi o compartilhamento de informações entre pessoas e máquinas. Ela é definida como um universo de informações em uma rede global de computadores. As informações são disponibilizadas em páginas, que são interligadas por links, permitindo a interatividade por parte dos usuários [15].

A esse conceito de documentos interligados, deu-se o nome de hipertexto. Um dos primeiros trabalhos que utilizavam esse conceito, foi o de Vanevar Bush, em 1945, chamado “As we may think”. Nele, o autor propôs um equipamento chamado “*Memex*”, que permitia que documentos gravados em microfilmes fossem referenciados entre si, permitindo uma navegação automática entre documentos posteriormente. [15]

O termo HTML, significa *Hyper Text Markup Language*. A tradução para o português é Linguagem para Marcação de Hipertexto. Desde a invenção do documento HTML, foram criadas 8 versões [16].

Silva, afirma que: “Para o bom entendimento das definições, podemos resumir hipertexto como todo o conteúdo inserido em um documento para a web e que tem como principal característica a possibilidade de se interligar a outros documentos WEB” [16] . Para conseguir essa interligação, cada documento possui conectores, chamados de links, que conectam a outros documentos, provendo interatividade aos usuários. O uso de links possibilitou o surgimento dos documentos hipertextos e foi a principal característica na primeira versão do HTML.

Já a partir da segunda versão, chamada de “HTML+” e escrita por Dave Raggett, definia que os Documentos HTML+ possuiriam elementos como: títulos, parágrafos, listas, tabelas e figuras. Mas não haveria no documento definições de aparência de margem, tamanho, cores e demais formatações de textos. Essas definições seriam alteradas nos próprios softwares que renderizavam o documento, restando ao documento HTML apenas a função de estruturar o conteúdo a ser apresentado através de tags, que delimitam o início e o fim blocos de elementos [16].

Graças a essa padronização estrutural, foi possível desenvolver a WEB, de forma que os documentos HTML fossem independentes de plataforma. Não permitindo que ela crescesse com uma base proprietária ou sendo limitada por formatos incompatíveis de diversos fabricantes.

### DOCUMETNOS HTML E TAGS

As *tags* desempenham o papel principal na estruturação do HTML, delimitando os blocos com a mesma semântica. Todavia, existem regras que definem onde os elementos devem ser utilizados e a hierarquia entre eles.

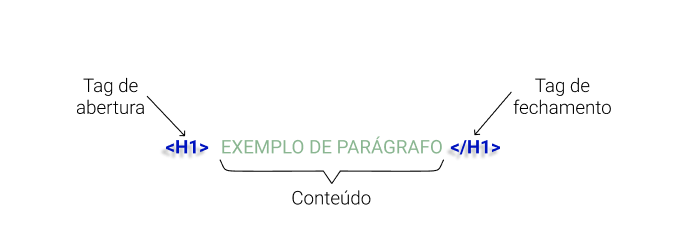


Figura 5: Modelo da Tag HTML

Conforme descrito por Ferreira, existem dois tipos de elementos no HTML: Os elementos de bloco e os elementos de linha. Os elementos de linha servem para delimitar marcações no texto. Já os elementos de blocos servem para delimitar blocos que representam seções no layout da página. Todos os elementos, tanto de linha quanto os de bloco, fazem parte de um grupo com as mesmas características. Tais elementos podem ser categorizados como elementos de: metadados, fluxo, seção, cabeçalhos, frase, incorporado e interativo [17].

Todo documento HTML5 possui uma estrutura básica, conforme mostrado na Figura 6.

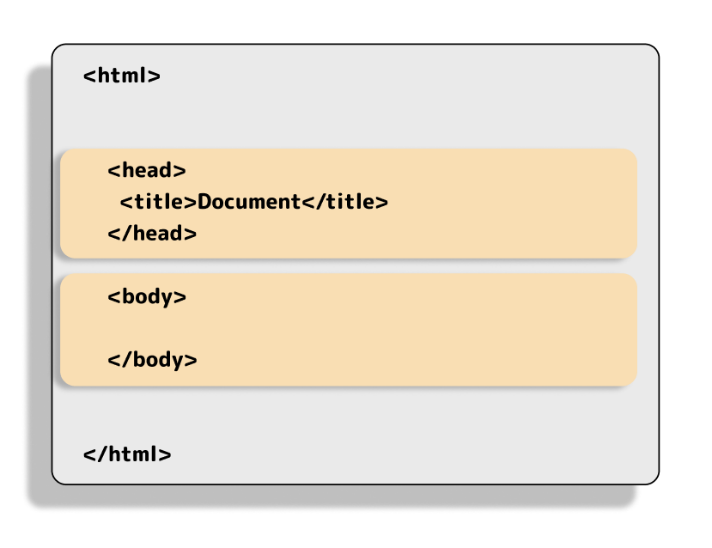


Figura 6: Estrutura básica do documento HTML

Conforme ilustrado na Figura 7, a *tag* <html> delimita a estrutura principal do documento. Ela estrutura o documento HTML em forma de árvore, sendo o elemento pai dos demais elementos dentro dele.

Já a *tag* <head> delimita os metadados do documento HTML, que podemos definir como informações que contêm informações importantes da categoria “Metadata”. Para organizar esses metadados, também são incluídas *tags* filhas, que servem para definições específicas na página, como: configurações de caracteres da página, links para importação de outros dados etc. Essa *tag*, deve estar dentro das *tags* <html></html> [18]

A *tag* <body> é a responsável por delimitar os elementos que são visualizados na página. Contém elementos das categorias “*Sectioning*”, “*Phrasing*”, “*Embedded*”, “*Interactive*”. Todos os elementos que são visíveis devem estar dentro do bloco delimitado por esta *tag*. Assim como a *tag* <head>, ela também deve estar dentro das tags <html ></html> [18]

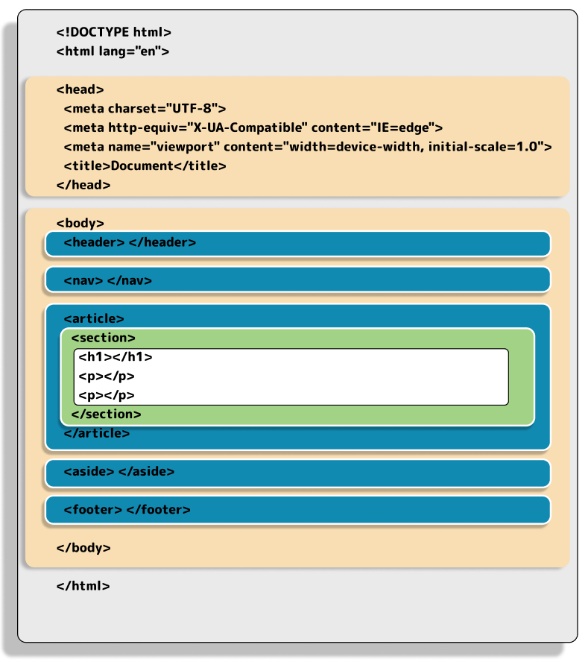


Figura 7: Exemplo de um documento HTML 5

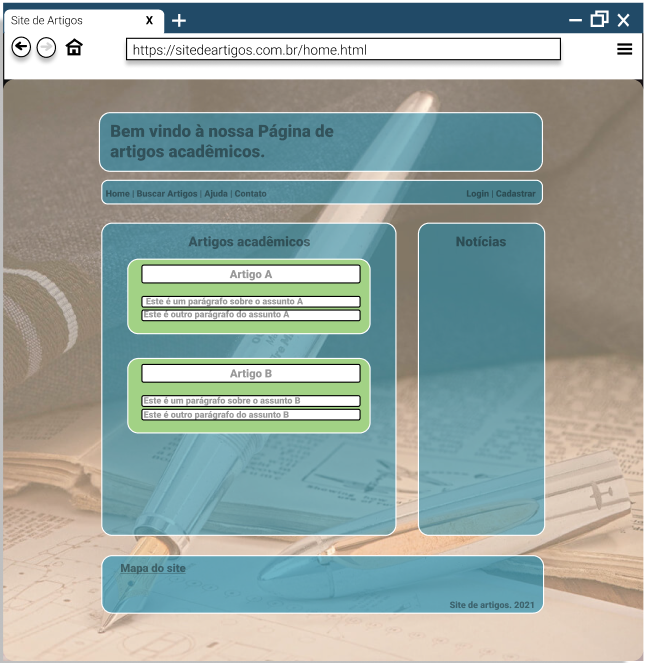


Figura 8 : Exemplo de Website estruturado semânticamente

“O HTML5 trouxe uma série de elementos que nos ajudam a definir setores principais no documen­to HTML.”. Na Figura 8, podemos ver um exemplo dessa estrutura semântica das páginas. Isso dá uma vantagem para sistemas como buscadores e sistemas de *web scraping*, por poderem guardar informações mais precisas, levando um tempo menor na busca dessas informações. [17]

Hoje em dia, na versão atual, existe uma forma diferente das versões anteriores para se escrever os documentos de hipertexto. Segundo Ferreira, a escrita do código é mais semântica e com menos código. Isso possibilita maior interatividade e cria um código interoperável, que pode ser usado em diversos dispositivos. [17]

### DOCUMENT OBJECT MODEL (DOM)

Ferreira afirma que o DOM – Document Object Model é a interface entre o documento HTML e o Javascript, O DOM disponibiliza uma estrutura dos objetos em um documento HTML para que o Javascript possa acessá-los [17].

Esse processo de estruturação dos objetos é realizado após o DOM construir uma árvore com os objetos do documento HTML. Os principais objetos são o “document”, “element”, “atribute” e “text” [19].

A Figura 9 mostra como podemos utilizar a linguagem Javascript para selecionar todos os objetos HTML com a classe “primeiro”, utilizando o código abaixo:

paragrafoPrincipal = document.getElementsByClassName(‘primeiro’)

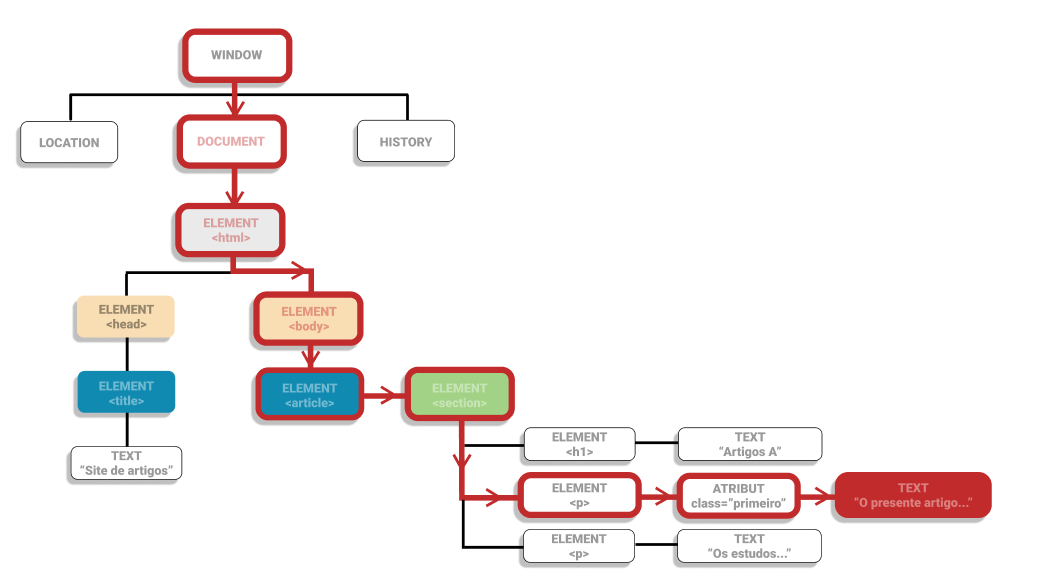


Figura 9: Estrutura em árvore gerada pelo DOM

Para permitir que o Javascript acesse os objetos do documento HTML, cada tipo de objeto, através da interface DOM, disponibiliza um método. Por exemplo, para o objeto do tipo “document” e todos os objetos abaixo na sua hierarquia, são providos métodos que possibilitam acesso aos objetos filhos através do ID, de classe ou da tag.

Alguns métodos mais conhecidos do objeto “document” são: getElementByID(), getElemenstByClassName(), getElementsByTagName(). Todavia, o DOM também disponibiliza métodos que executam seleções utilizando os seletores CSS, como: querySelector() e o querySelectorAll() [19].

Além de métodos de seleção, o DOM disponibiliza métodos para a criação de novos objetos na estrutura, adicionando novos nós na árvore de objetos do document. Também é possível adicionar eventos que interagem com os elementos HTML. Para isso, é disponibilizado o método addEventListener() [19].

## NAVEGADORES WEB

Os navegadores Web são softwares que provêm uma interface para os usuários navegarem nas páginas web desejadas. Atuam recuperando documentos WEB que estão localizados em servidores WEB e exibem na tela do equipamento do usuário (computador, celular, tablet etc.).

No domínio da World Wide Web, cada página ou recurso solicitado pelo usuário é referenciado por uma URI (Uniform Resource Identifier), ou “identificador uniforme de recurso”, traduzido para o Português. Eles podem assumir diferentes tipos, como imagens, documentos HTML, arquivos de som ou vídeos [20].

Para que o navegador no computador cliente possa recuperar um recurso em um servidor WEB, os dois devem estabelecer uma conexão de comunicação. Esse processo é realizado através do uso de protocolos de comunicação da suíte TCP/IP. Porém, antes do estabelecimento de uma conexão entre servidor web e o computador do usuário, este deve saber o endereço IP do servidor web. Para isso, existe o DNS(Domain Name System), que significa “sistema de nomes de domínios” ao se traduzir para o Português [21].

A WEB está baseada no sistema DNS, para poder traduzir os endereços das URLs para endereços de IP. Pois, seria inviável a expansão da WEB caso os usuários tivessem que utilizar números de endereços IPs ao invés de nomes de domínios [21].

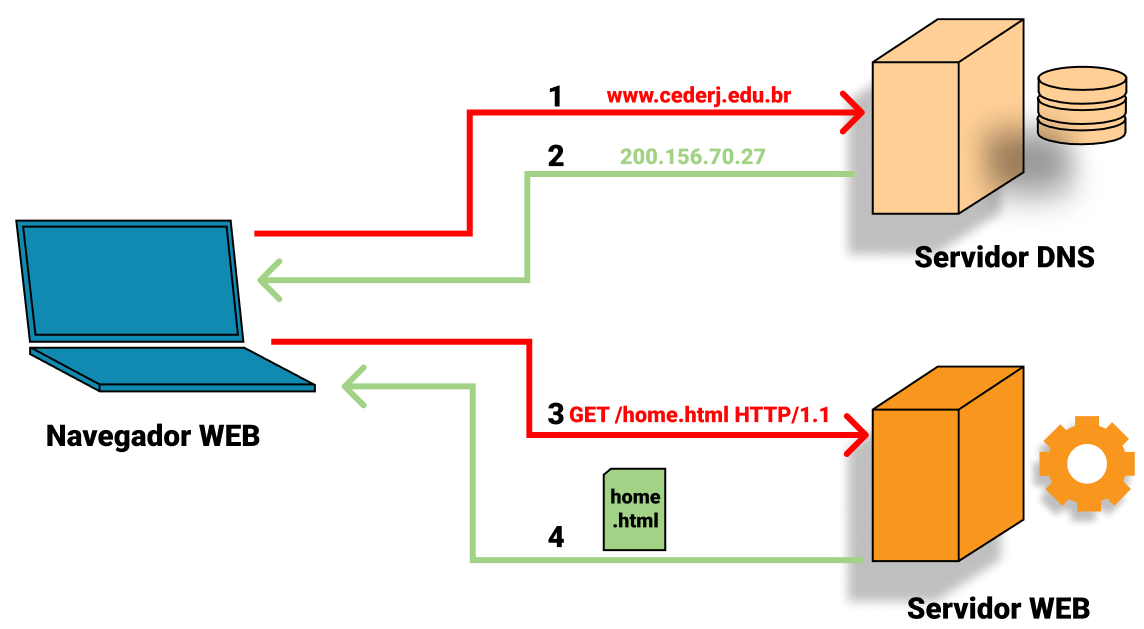


Figura 10: Processo de requisição HTTP pelo navegador

A figura 10 mostra um exemplo de processo de requisição HTTP:

1. O usuário digita o endereço www.cederj.edu.br e o navegador WEB envia a solicitação para um servidor DNS;
2. O servidor DNS devolve o endereço IP correspondente o domínio solicitado;
3. Com o endereço IP, o navegador estabelece uma conexão com o servidor web e requisita um recurso;
4. O Servidor envia o recurso solicitado ao navegador;

## WEB SCRAPING

Web scraping consiste em um conjunto de técnicas de coleta automatizada de dados da *internet* com o uso de aplicativos que requisitam informações de servidores Web e extraem dados dos documentos HTML. Em seguida, eles são analisados e têm seus dados estruturados, em um processo conhecido como “Parsing”

De acordo com Ryan(2019) “Na prática, web scraping engloba uma grande variedade de técnicas de programação e de tecnologias, por exemplo, análise de dados, parsing de idiomas naturais e segurança da informação.”. Esse conjunto de técnicas permite a coleta de um grande volume de dados, o que não seria viável manualmente. Para isso, são usados scripts, que coletam informações das páginas e armazenam em um banco de dados [22].

Diferentemente do acesso às páginas web através de navegadores, web scrapingnão utiliza uma camada de formatação de HTML, estilização com CSS ou scripts Javascript para a apresentação de dados. Todavia, esses recursos podem ser utilizados para a obtenção de dados.

Ryan descreve que o processo realizado pelo script ou programa de web scraping consiste em duas etapas sequenciais. Na primeira, através de uma URL, é feita uma requisição GET a um servidor WEB. Também pode ser feita com uma requisição POST. Após o servidor receber a requisição e processar as informações, os scripts recuperam os recursos solicitados, para serem analisados, reformatados, organizados e estruturados. Os formatos mais comuns para estruturar os dados são o XML e o JSON.

De forma análoga ao processo, Zhao (2017) aborda que um programa de webScraping é dividido em dois módulos.

O primeiro é usado para compor a requisição HTTP. Esse módulo lida não apenas com as requisições, mas com a autenticação, redirecionamentos e manipulação de cookies das páginas requisitadas.

Já o segundo, conforme a Figura 11, é o responsável pelo Parsing e extração de dados do documento HTML ou XML. Funções de bibliotecas específicas, implementadas por linguagens de programação, permitem a navegação, busca e a modificação de uma estrutura de árvore criada durante o parsing.

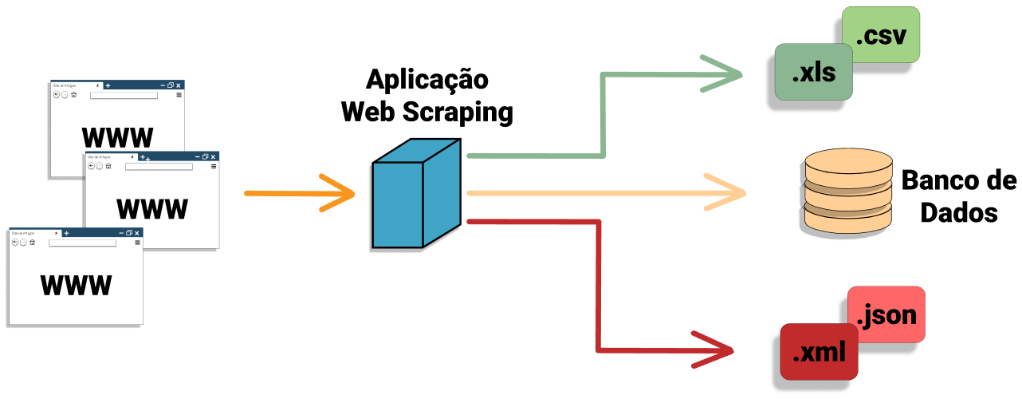


Figura 11: Aplicação Web Scraping

# DESENVOLVIMENTO

Este capítulo abordará a implementação do aplicativo BOT, desenvolvido neste trabalho, que coletará automaticamente tweets através da API do Twitter. Para uma melhor compreensão do aplicativo, apresentaremos os principais conceitos da API do Twitter.

Do mesmo modo, a compreensão do domínio e das necessidades dos pesquisadores envolvidos foi de suma importância para o desenvolvimento deste projeto. Para tal, foram realizadas diversas entrevistas com especialistas farmacêuticos. Nelas, se pôde compreender os requisitos funcionais, que nos permitiram desenvolver uma ferramenta útil e prática, que proporcionasse a coleta automática de dados aos pesquisadores envolvidos no projeto.

## TWITTER API

A API do Twitter surgiu com a versão 1.1, considerada legada pelo Twitter. Possuí três camadas de acesso, descritas abaixo [23]:

***Standard* –** É a camada que disponibiliza recursos gratuitos. Utilizada para pequenos projetos, testes de integração e validação de conceitos de aplicativos. Possui seis endpoints, que são: “Tweets”, “Users”, “Direct Messages”, “Media”, “Trends” e “Geo”[23]*.*

***Premium* –** É a camada da versão 1.1 que provê um acesso escalável aos dados do Twitter. Ela provê aos desenvolvedores recursos de busca mais aprimorados do que na versão Standard. Ela possui dois endpoints: “Search Tweets *API*”(com buscas de 30 dias e a busca completa na base de dados) e a “Account Activity API”(que permite uma coleta de tweets em tempo real de uma conta do Twitter) [24].

***Enterprise* –** É acamada onde o Twitter disponibiliza a maior quantidade de recursos na versão 1.1 da *API*, como ferramentas avançadas de filtros, mais operadores de buscas e um suporte dedicado para os contratos, como gerentes de contas e suporte técnico personalizado. Segundo a empresa Twitter, é a camada ideal para empresas que dependem dos dados e precisam de alta confiabilidade do serviço. Possui oito camadas, divididas em cinco grupos de usabilidade, conforme abaixo:[25]

Tempo Real – “Power Track API”, “Decahose API”, “Account Activity API”

Histórica – “Search API” e “Historical Power Track API”

Compreensão – “Engagement API”

Conformidade – “Compliance Firehose API”

Usabilidade – “Usage API”

Neste projeto abordaremos o uso da *API* versão 2, que foi a escolhida para a criação do aplicativo. Destacamos como alguns dos fatores para tal escolha fatores como: a disponibilização de recursos gratuitos para pesquisas, novos recursos de busca, melhorias na estrutura dos objetos de resposta JSON, ampla documentação e o fato do Twitter te anunciado a intenção de descontinuar a versão 1.1 em breve [26].

Dentre os novos recursos disponibilizados, podemos citar os seguintes:

* Habilidade de requisitar objetos e campos específicos, otimizando a resposta com apenas as informações desejadas e úteis na requisição.
* Mais detalhes e novos objetos de dados foram adicionados nas respostas.
* Métricas para uma melhor compreensão da performance na própria resposta da requisição.
* Dados de anotações que permitem filtrar tópicos através de contextos.
* Rastreio de conversações que permitem identificar quando um tweet faz parte de uma conversação.
* Conta para pesquisas acadêmica que disponibiliza aos pesquisadores acesso gratuito a toda base de dados do *Twitter*.

Em comparação com os endpoints da versão 1.1, houve algumas mudanças em alguns deles. A principal mudança foi o *“*Full-archive Search”, disponibilizada apenas para as contas aprovadas para o perfil “Academic Research”. Este endpoint será abordado em mais detalhes na próxima seção, além de serem apresentados com mais detalhes os principais conceitos de uma *API*.

### CONCEITOS DE API RESTFUL

Por muito tempo as empresas tinham necessidade de compartilhar ou vender seus serviços e dados de forma rápida, segura e sem grandes complicações. A partir dessa necessidade, temos criação dos termos API, API REST e API RESTFULL.

Entende-se como *API* (interface de programação de aplicações, traduzido do inglês)*,* o conjunto de definições e protocolos usados no desenvolvimento de integrações de software e aplicações. A *API* tem como objetivo permitir que desenvolvedores criem soluções ou serviços que se conectem com outros produtos e serviços sem necessariamente entender toda sua estrutura, padrões e forma que eles foram implementados. A mobilidade e flexibilidade trazidas pelas API’s para os desenvolvedores e empresas possibilitam poupança de tempo, dinheiro e melhor administração dos serviços contratados ou prestados. [27]

Em seu funcionamento as *APIs* vêm com um conjunto de regras e documentações para enviar e receber solicitações. Na Figura 12, vemos algumas abordagens das *APIs* [27]*.*

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 12 Tipo de abordagem API

A *API REST, onde* a sigla REST se junta ao termo API e tem como significado (Representacional State Transfer ou transferência representacional de estado), é uma arquitetura usada para construir APIs, tendo como alicerce principal em sua maioria o protocolo HTTP e definindo padrões de compartilhamento e comunicação que trabalham com os tipos de arquivo *JSON*, *XML* e GraphQL. Com essa base estrutura-se RESTFull a partir dos cinco princípios fundamentais de *Fielding,* conforme mostra aTabela 2*.* [28]

Tabela 2 Requisitos RESTFull

|  |  |
| --- | --- |
| Princípios  Fundamentais | Breve Descrição |
| Cliente/Servidor | A comunicação ocorre entre dois agentes, em rede, onde um agente é considerado cliente e outro é considerado servidor. O cliente, ativamente, faz requisições que devem ser respondidas pelo servidor. |
| Cache | Tanto o agente servidor quanto cliente podem implementar estratégias de caching com vistas a melhoria da performance de rede e a percebida pelo usuário. |
| Stateless | Uma única requisição deve conter todos os parâmetros necessários para ser atendida sem ter de contar com nenhum tipo de “memória” do agente servidor. |
| Layered | A comunicação entre cliente e servidor deve ocorrer através de camadas (arquitetura pipe-and-filter). Onde camadas podem ser adicionadas para melhorar a performance do sistema |
| Interface Uniforme | A troca de mensagens entre cliente e servidor deve ocorrer através de interfaces uniformes |

Uma API implementada seguindo as orientações do trabalho de *Roy Thomas Fielding* em essencial os cinco princípios fundamentais descritos na tabela acima, é denominadaRESTFull. Temos também um sexto princípio denominado “Code-on-Demand”, que consiste em baixar e executar código na forma de mini aplicativos ou *scripts*, com intuito de simplificar o cliente e reduzir recursos utilizados para grandes tarefas de execução [29]

### ENDPOINTS DA TWITTER API VERSÃO 2

Para uso dessa pesquisa, foi utilizado o endpoint“Search Tweets” mostrado na Tabela 3, localizado dentro do grupo de *endpoints* *Tweets* [30].

Com esse *endpoint* temos a possibilidade de acesso a todos *Tweet* públicos, criados desde março de 2006 até o dia de hoje, com base em nossa consulta programada [31].

Tabela 3: Lista de Endpoints da API - Versão 2

|  |  |  |
| --- | --- | --- |
| *Tweet* | Tweet lookup | Busca de Tweets por ID |
| Search Tweets | Busca dos últimos 7 dias |
| Busca completa na base de dados do *Twitter*. Disponível apenas para contas *Academic* *Research* |
| Tweet counts | Contagem dos *Tweets* dos últimos 7 dias |
| Contagem dos *Tweets*, de acordo com uma determinada busca, à base de dados do *Twitter*. Disponível apenas para contas *Academic* *Research* |
| Timelines | Retorna, cronologicamente, *Tweets* de uma conta do *Twitter* especificada ou das menções de uma conta do *Twitter* especificada. |
| Filtered stream | Filtra o fluxo de *tweets* públicos em tempo real |
| Sampled stream | Retorna uma amostra de 1% de todos os *Tweets* público de novos *Tweets* em tempo real. |
| Likes | Retorna uma lista de usuários que curtiram um *tweet* ou recupera uma lista de *tweets* que um usuário curtiu. |
| Hide replies | Oculta ou exibe uma resposta a um *Tweet* |
| *Users* | User lookup | Busca usuários pelo nome o pelo ID |
| Follows | Retorna os seguidores de um usuário ou retorna uma lista de quem um determinado usuário segue. Também permite seguir ou deixar de seguir um usuário |
| Blocks | Retorna uma lista de usuários que estão bloqueadas por uma conta. Também bloqueia ou desbloqueia um usuário |

## CASOS DE USO

Este capítulo abordará alguns detalhes do processo de arquitetura de software, o qual possibilitou o desenvolvimento da aplicação. A descrição do projeto é feita através de diagrama do caso de uso e sua respectiva descrição.

### DIAGRAMA CASO DE USO "TWITTER API SEARCH"

A fim de elucidarmos as principais funcionalidades do aplicativo, narramos através do diagrama de caso de uso, mostrado Figura 13 o papel do ator “usuário”, que opera o sistema.”

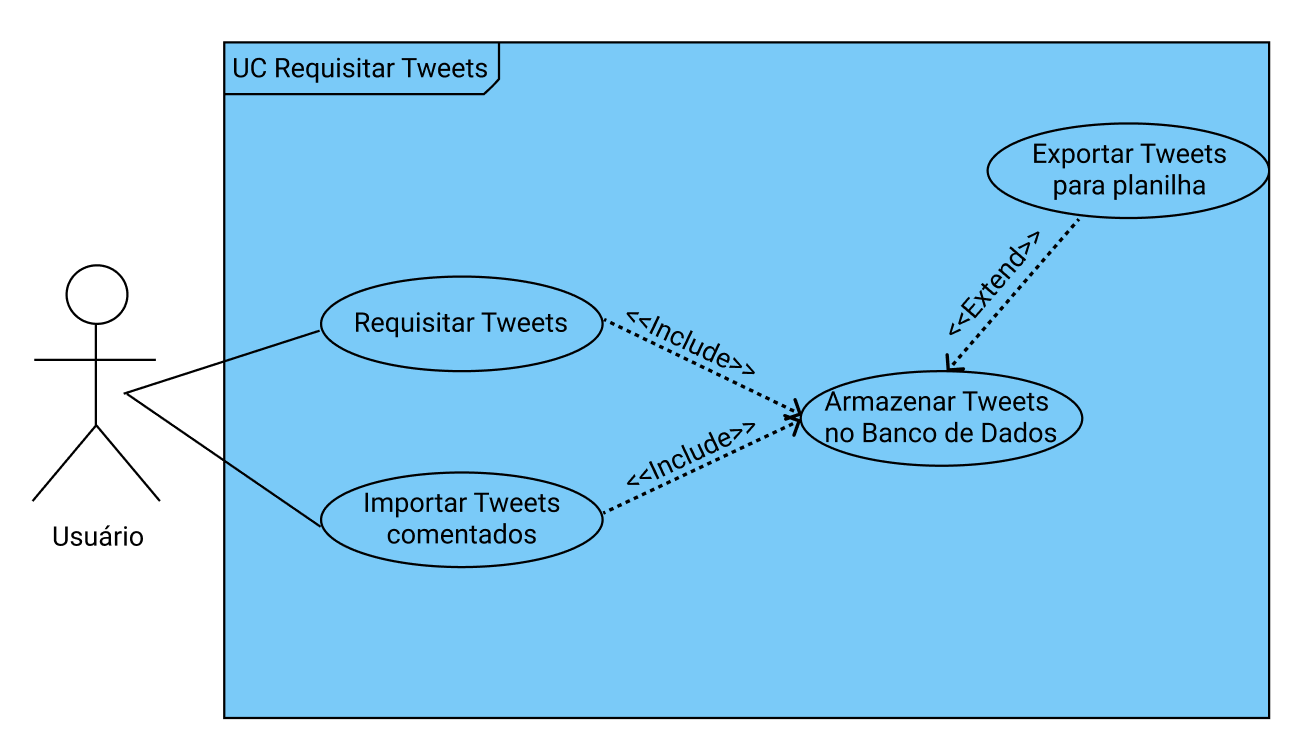


Figura 13: Diagrama do Caso de Uso Requisitar Tweets

Para uma melhor compreensão dos fluxos e regras de negócios do caso de uso, exibimos na Tabela 4, a descrição textual deste caso de uso.

Tabela 4: Descrição do Caso de uso “*Twitter API SEARCH*”

|  |  |
| --- | --- |
| ID: | UC01 |
| Objetivo: | Extrair tweets, de um determinado período, de acordo com o palavras-chave de interesse |
| Requisitos: | Conexão com a *internet*.  Conta do Twitter API *Acedemic Research* |
| Atores: | Usuário |
| Pré-Condições: | Os pacotes, linguagens, *frameworks* e SGBD usados pelo aplicativo, deverão ser instalados no ambiente. |
| Pós-Condições: | As informações serão armazenadas em um banco de dados não relacional  As informações serão exportadas para uma planilha, para posterior comentário do farmacêutico especialista. |
| Fluxo Principal: | 1. O Usuário configura o arquivo com as palavras-chave de interesse, de acordo com a definição do farmacêutico especialista. 2. O usuário define os parâmetros de configurações principais no arquivo “conf.json” 3. A ferramenta busca os dados na base de dados do Twitter, através do uso da API. 4. A ferramenta manipula as informações e salva em um banco de dados não relacional. 5. Cada busca é salva em uma coleção . |
| Fluxos Alternativos: | Não há. |
| Erros/Exceções: | Os erros são tratados pelo aplicativo. Em caso de indisponibilidade da API, o aplicativo faz novas tentativas de acesso a cada 60 segundos, até obter êxito. |
| Mensagens: | * É gerado um arquivo de log com a extensão “.log” * É gerada uma coleção das buscas no banco de dados, contendo as principais informações da busca |
| Regras de negócio: | **[RN01] –** As palavras-chave de interesse, deverão ser fornecidas no arquivo “entrada.py”, dentro do *array* “palavras-chave”.  **[RN02] –** O *array* deve possuir uma tupla para cada busca. As tuplas, por sua vez, possuem informações do domínio, subdomínio e as palavras-chave a serem buscadas. Para a correta leitura pelo aplicativo, as informações supracitadas devem ser encapsuladas entre aspas simples ou aspas duplas. As palavras-chave podem conter palavras com o caractere precedente “#”(definem assuntos de interesse) ou palavras a serem buscadas no texto. As palavras-chave podem ser combinadas com operadores lógico e expressões.  **[RN03] –** O código(*TOKEN*), que autentica o aplicativo na API do Twitter, deverá ser fornecido em uma variável de ambiente do sistema operacional, chamada “BEARER\_TOKEN”.  **[RN04] –** O resultado da requisição, deve ser alterado para que sejam adicionados ao dicionário de dados de cada tweet, as chaves “*target*”, “*sentiment*” e o “id” da busca atual.  **[RN05] –** Cada tweet deve ser único no banco de dados. Todavia, cada um deve conter informação do ID de todas as buscas aonde foi retornado.  **[RN05] –** AS buscas não podem ultrapassar o limite de taxa de 300 requisições a cada 15 minutos, na API do Twitter**.** |

## DIAGRAMAS DE CLASSES

A implementação da aplicação foi baseada em programação orientada a objetos, que tornou possível a solução de problemas com o uso de classes, conforme exposto na Figura 14.

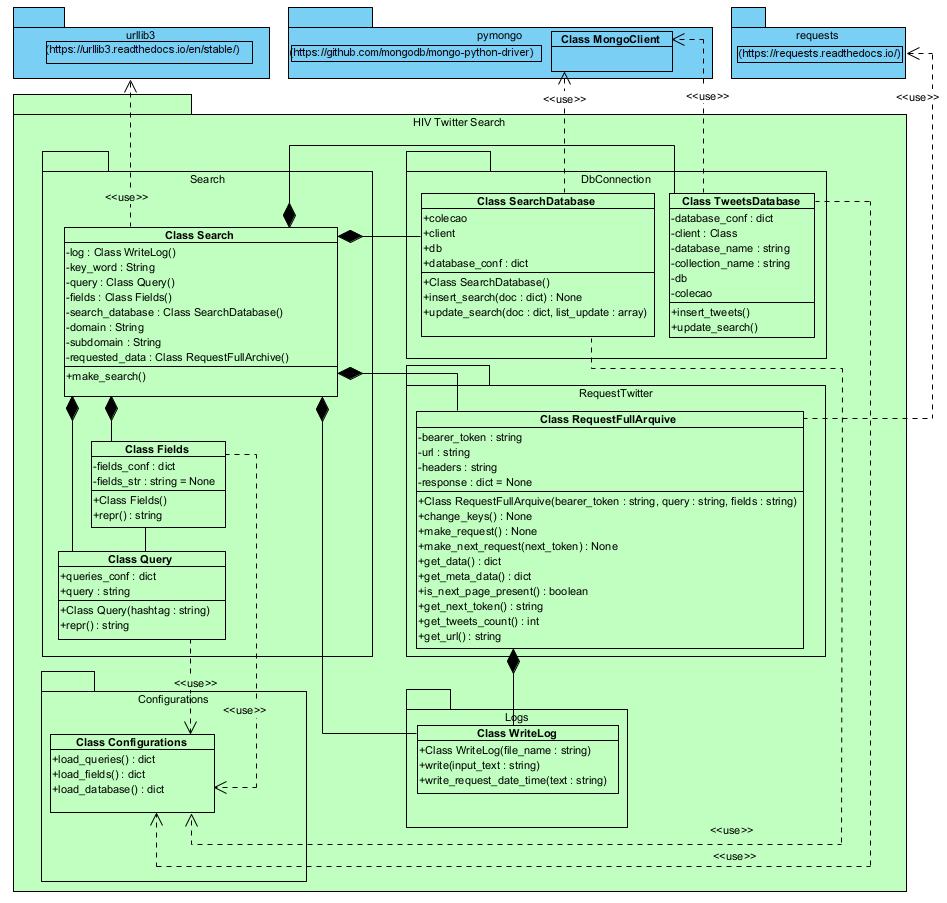


Figura 14: Diagrama de Classes

Além da diagramação das classes, exibimos na Tabela 5, a descrição textual das classes

Tabela 5: Tabela de descrição das classes

|  |  |
| --- | --- |
| Nome da Classe | Descrição |
| Configurations | O objetivo desta classe é fazer a leitura do arquivo de configurações “conf.json”. Ela possui três métodos estáticos, que retornam o dicionário de dados referente à chave “queries” , “fields” e “database” no arquivo de configuração. |
| *TweetsDatabase* | O objetivo desta classe é se conectar, escrever e atualizar dados no Banco de dados de tweets, referente ao ano da busca. Ela insere o Tweet, caso não exista no banco de dados e atualiza a chave “request\_id”, de cada tweet. |
| *SearchDatabase* | O objetivo desta classe é se conectar, escrever e atualizar dados no Banco de dados de buscas. Ela insere informações da busca no banco de dados de buscas. |
| *WriteLog* | Classe responsável por escrever informações das buscas e das requisições em um arquivo de log. |
| *RequestFullArchive* | Classe responsável por fazer requisições na API do Twitter. Ela recebe a resposta, um arquivo JSON, da API do Twitter. Ela também é a classe responsável por tratar os erros das requisições. |
| *Search* | Classe responsável por manipular cada busca de palavra-chave, de acordo com o ano escolhido. Nela, estão implementadas as principais regras de negócio do projeto. |
| *Query* | Classe responsável por obter a query, utilizada na busca de tweets. Esta classe obtém a string da query através da o arquivo de configurações (conf.json). |
| *Fields* | Classe responsável por obter os campos que serão retornados pela API do Twitter. Esta classe obtém a string dos campos retornados através da o arquivo de configurações (conf.json). |

## EXTRAÇÃO DE TWEETS COM OPINIÕES DE MEDICAMENTOS PARA HIV

As palavras-chave definidas pelo especialista(Anexo B) para as buscas foram inseridas em um arquivo Python. Conforme exibido na Figura 11, foi realizada uma análise sobre cada palavra-chave, que foram estruturadas hierarquicamente, estabelecendo assim, conceito de domínio e subdomínio. Essa divisão lógica permite a classificação por tags em cada busca.

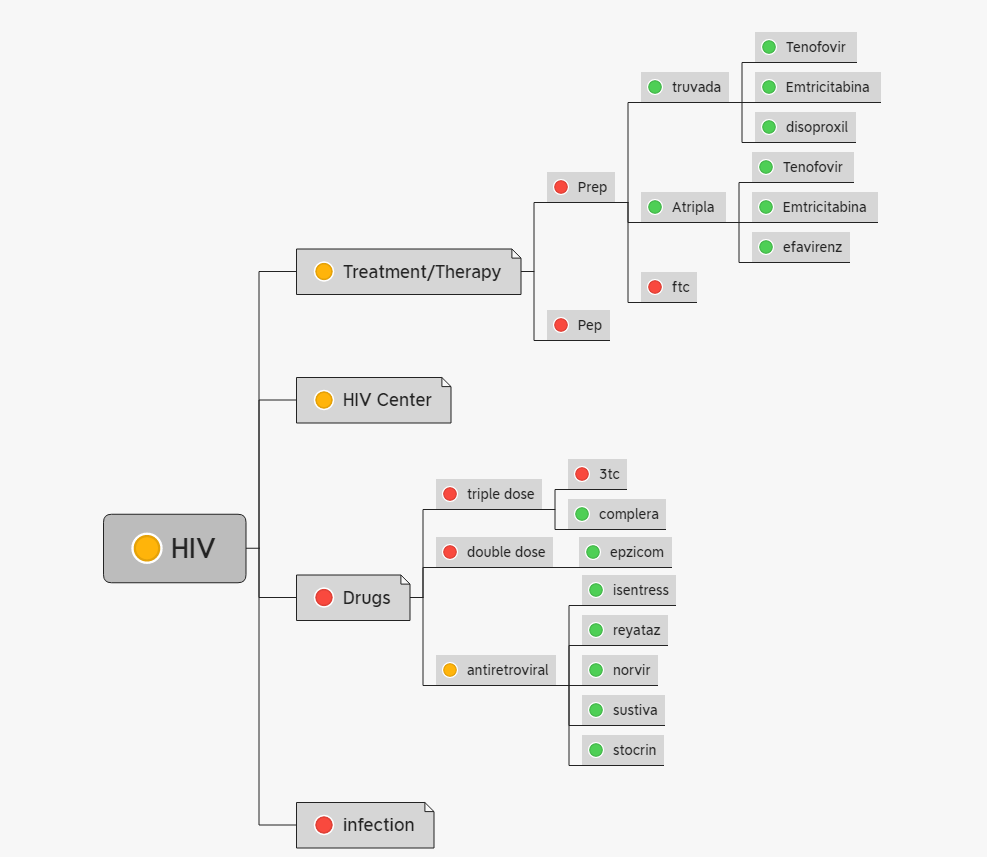


Figura 15: Mapa de Compreensão de domínios e subdomínos

### CONTA ACADEMIC RESEARCH

É do interesse do Twitter que pesquisadores acadêmicos desenvolvam trabalhos relacionados ao uso de dados de mídia social. Pois, para a equipe do Twitter, desenvolver estes trabalhos sozinha é um grande desafio. Com o apoio da comunidade acadêmica, o Twitter pode se beneficiar dos métodos de pesquisa ou modelagem dos dados, além dos métodos de aprendizado de máquina, tais como, métodos e técnicas de PNL para identificação de sentimentos em todos os idiomas e diversas outras técnicas aplicadas à ciência de dados [32].

A empresa acredita que quanto mais pesquisas houver, relacionadas a esses tópicos, mais próximo a empresa estará do seu objetivo, que é tornar o mundo um lugar melhor. O Twitter acredita que tem muito a aprender com a experiência da comunidade acadêmica.

Para viabilizar esta cooperação entre Twitter e a comunidade acadêmica, foi disponibilizada a conta “*Academic* *Research”*. Para utilizá-la deve-se aplicar para o acesso. Para se obter acesso, deve-se preencher um formulário e atender a alguns requisitos de perfil definidos pelo Twitter. No Anexo C, pode-se ver um exemplo do preenchimento deste formulário para a obtenção da conta *Academic* *Research,* utilizada neste trabalho [33].

A conta *Academic* *Research* possui limitações superiores das contas *Standard* descritas na Tabela 6[34].

Tabela 6: Limitações da conta *Academic Research*

|  |  |
| --- | --- |
| Limite Mensal por projeto | 10.000.000 de tweets por mês. Limite renovado todo dia 30. |
| Taxa de requisições | 300 requisições a cada 15 minutos. |
| Intervalo entre requisições | 1 requisição a cada 1 segundo |
| Comprimento da *query* | 1024 caracteres |

### DICIONÁRIO DE DADOS

As informações buscadas, são retornadas em um Objeto de Transferência de Dados, ODT, armazenado em um arquivo JSON.

A resposta da requisição, também conhecida como payload, varia de acordo com o endpoint utilizado. O endpoint usado nesse estudo, o Search, retorna um payload chamado de tweet object [35]. Conforme mostra a Figura 16, os dados são organizados em duas estruturas principais:

* “data” - Contém um array com todos os tweets retornados na busca
* “meta” – Chave com metadados sobre a resposta da requisição.

Figura : Resposta de uma requisição à API do Twitter

{

    "**data**": **[**

        {

            "id": "1410938831784890370",

            "text": "Delaying treatment allows for the #HIV virus to continue harming the immune system and puts you at risk of developing AIDS. Be sure to take treatment as prescribed and regularly visit your doctor. Learn more here: https://t.co/Mn2bydNxVv  #GetTestedNow"

        },

        {

            "id": "1410932590270369792",

            "text": "#RT @pozmagazine: What are the health consequences of overweight and obesity? @LizHIVHep https://t.co/wU5tfocpOT #BreastCancer #fattyliver #HIV #nash #stigma #nafld #COVID19"

        },

        {

            "id": "1410932202976776199",

            "text": "Congratulations Harold J. Phillips on your appointment to lead the Office of National AIDS Policy (ONAP), part of the Domestic Policy Council in the Biden White House! https://t.co/epNi5WYrxO #HIV #AIDS https://t.co/ksfvyHQM7A"

        }

**]**,

    "**meta**": {

        "newest\_id": "1410938831784890370",

        "oldest\_id": "1410932202976776199",

        "result\_count": 10,

        "next\_token": "b26v89c19zqg8o3fpdj6p8z05q1tw5tdml77thshwld31"

    }

}

### USO DA TWITTER API

Atualmente, existe o *framework* *Tweepy* que é amplamente utilizado para implementar soluções de requisições utilizando a API do *Twitter*. Todavia, este *framework* não contempla o uso da versão 2 da API, utilizada neste projeto. Portanto, o projeto consistiu em desenvolver um aplicativo que pudesse utilizar diretamente a API do Twitter, sem o uso de frameworks de terceiros, conforme mostra a Figura 17.

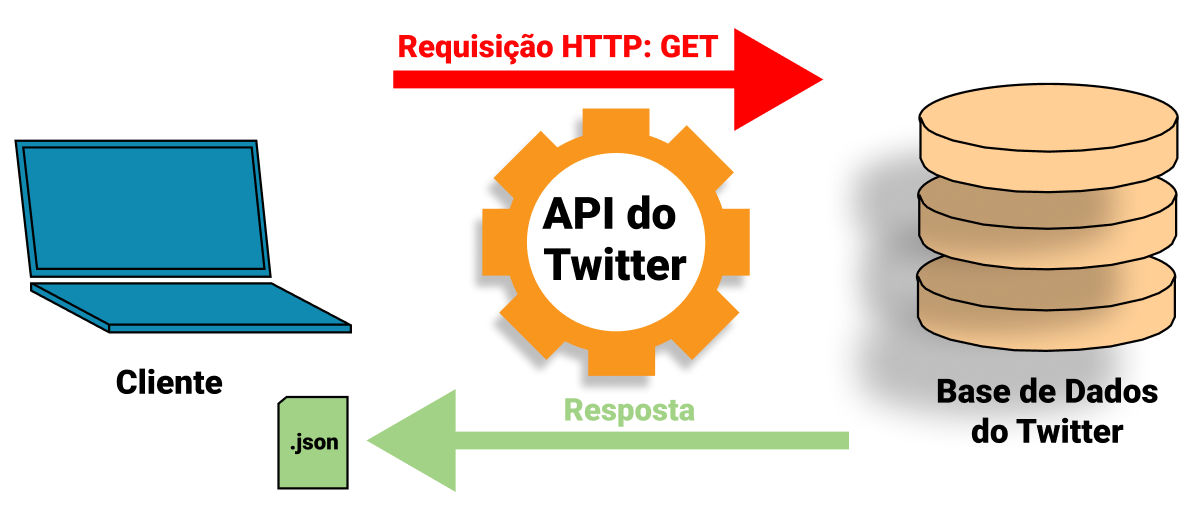


Figura 17: Esquema de uso da API do Twitter

Para requisitar os tweets, usa-se uma requisição HTTP do tipo GET. A requisição é construída através de uma URL, que por sua vez, possui o endereço do *endpoint* e parâmetros definidos pela API, conforme detalhado na Tabela 7.

Tabela : Componentes de uma requisição ao endpoint Search da API do Twitter

|  |  |
| --- | --- |
| **Endpoint Search** | https://api.twitter.com/2/tweets/search/all? |
| **Query** | **query=#HIV OR #truvada&lang:en&max\_results=500** |
| **Fields** | **&tweet.fields=** **author\_id,public\_metrics,text,created\_at** |
| **Expansions** | **&expansions=** **author\_id,geo.place\_id** |
| **Header** | **--header** **‘Authorization: Bearer $BEARER\_TOKEN’** |

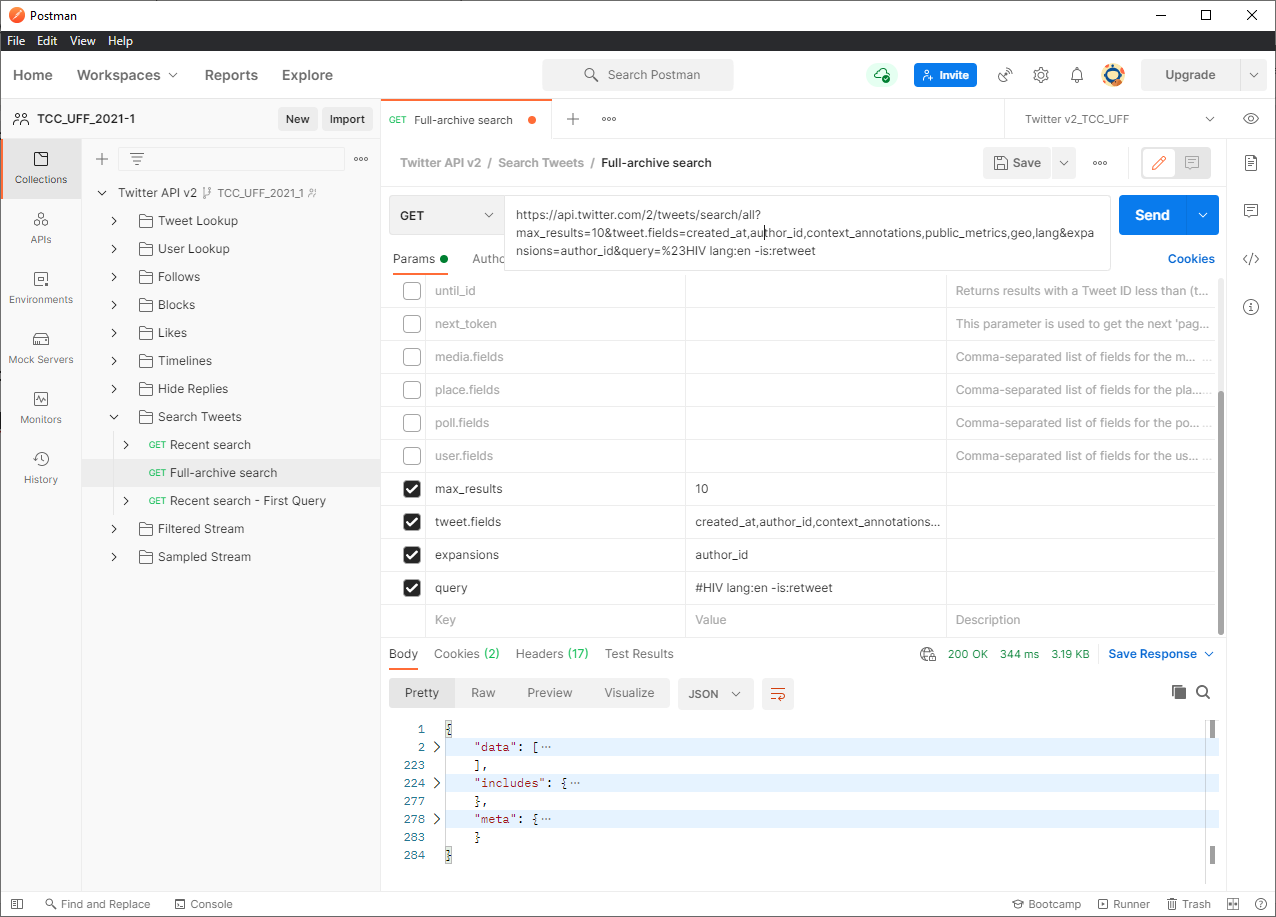
Para uma melhor compreensão da API, antes do desenvolvimento do aplicativo, decidimos fazer uma avaliação mais pragmática da API. Para isso, ao invés de focar em uma extensa leitura da documentação do *Twitter*, decidiu-se pelo uso do aplicativo Postman, que é um cliente de API. Ele possibilita aos desenvolvedores criar, compartilhar, testar e documentar APIs.

O Twitter disponibiliza uma coleção para o Postman, com as predefinições de todos *endpoints* da versão 2 da API. A coleção está disponível em https://documenter.getpostman.com/view/9956214/T1LMiT5U.

Na Figura 18, temos uma captura de tela do aplicativo Postman. Dentre as funcionalidades que ajudaram na compressão da API do Twitter, destacamos:

* + - * 1. Predefinições de requisições, de acordo com o *endpoint.*
        2. Parâmetros de busca.
        3. Visualização da URL de requisição.
        4. Visualização estruturada da resposta em JSON.

Tendo sido facilitada a compreensão estrutural de uma requisição, com o primeiro contato com a API através do Postman, observou-se uma melhor compreensão da documentação do próprio Twitter.



**(c)**

**(d)**

**(b)**

**(a)**

Figura 18: Aplicativo Postman - Client para APIs

O servidor retorna códigos de status HTTP no cabeçalho da resposta. Conforme mostrado na Tabela 8, esses códigos podem se referir a requisições bem-sucedidas ou malsucedidas. Os códigos de status seguem os padrões do protocolo HTTP.

Tabela 8: Respostas HTTP da API do Twitter

|  |  |  |
| --- | --- | --- |
| Código de Status | Resposta | Descrição |
| 200 | OK | A requisição foi bem-sucedida. |
| 400 | Bad Request | A requisição foi inválida ou não pôde ser provida. Requisições sem autenticação ou com a string de query inválidas, retornam essa resposta |
| 401 | Unauthorized | Problemas em autenticar a requisição. Requisições sem o token ou com token inválido retornam essa resposta. Alguns outros erros não definidos, também retornam este erro |
| 403 | Forbidden | A requisição possui uma sintaxe correta. Porém, ela foi recusada ou o acesso não é permitido |
| 404 | Not Found | A URI da requisição é inválida. Ou o recurso requisitado, como um usuário, não existe. |
| 429 | Too Many Requests | Não são permitidas mais requisições, pois o limite da taxa de requisições da API foi excedido. |
| 500 | Internal Server Error | Quando há sobrecarga de requisições ou o Endpoint está temporariamente enfrentando algum problema |
| 503 | Service Unavailable | Os servidores da API estão funcionais, porém sobrecarregados com excessivas requisições. |

Quando ocorre um erro na requisição, a API retorna no corpo da resposta um objeto ODT, implementado em JSON, com informações mais detalhadas, conforme exibido na Figura 19. Os campos de valor “title” e “detail” explicam com mais detalhes o erro. O campo “type” possui uma URI que direciona para uma página da documentação da API, onde o erro é explicado em maiores detalhes.

Figura : Detalhes dos erros de requisições à API do Twitter

1 {

2 "**client\_id**": "101010101",

3 "**required\_enrollment**": "Standard Basic",

4 **"registration\_url"**: "https://developer.twitter.com/en/account",

5 **"title"**: "Client Forbidden",

6 **"detail"**: "This request must be made using an approved developer account that is enrolled in the requested endpoint. Learn more by visiting our documentation.",

7 **"reason"**: "client-not-enrolled",

8 **"type"**: "https://api.twitter.com/2/problems/client-forbidden"

9 }

# TESTES

Este capítulo abordará os recursos computacionais utilizados e detalhes dos testes de execução do aplicativo *Twitter API Search,* que neste trabalho, foi utilizado para a coleta de Tweets relacionados ao HIV.

Os testes foram executados em um microcomputador com as seguintes configurações:

Tabela 9: Configurações de hardware utilizado

|  |  |
| --- | --- |
| Processador: | *Intel(R) Core(TM)* i7-4820K. *CPU* 3.70 GHz |
| RAM instalada: | 16,0 GB |
| Sistema Operacional: | Windows 10 Home |
| Tipo de sistema: | Sistema operacional de 64 bits, processador baseado em x64 |
| Versão: | 20H2 |
| Instalado em: | ‎18/‎03/‎2021 |
| Compilação do SO: | 19042.1052 |
| Experiência: | Windows Feature Experience Pack 120.2212.2020.0 |
| Versão do Interpretador Python: | Python 3.8.5, Jul 20, 2020 [MSC v.1924 64 bit (AMD64)] |
| Informações da IDE de desenvolvimento: | |  | | --- | | PyCharm 2020.3.4 (Professional Edition) | | Licenciado para Allan Chaves | | Subscrição ativa até 19 de Junho de 2022. | | For educational use only. | | Runtime version: 11.0.10+8-b1145.96 amd64 | | VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o. | | Windows 10 10.0 | | GC: ParNew, ConcurrentMarkSweep | | Memory: 1973M | | Cores: 8 | |

Tabela 10: Pacotes Python instalados(as)

|  |  |
| --- | --- |
| Pacotes instalados | Versão |
| Matplotlib: | 3.4.2 |
| Pymongo: | 3.11.4 |
| Requests: | 2.25.1 |
| urllib3: | 1.26.5 |

Para a execução do aplicativo, faz-se necessária a alteração no arquivo de parâmetros de configuração, *“*conf*.*json*”.* O aplicativo utiliza tais configurações para procedimentos de conexão ao banco de dados e as requisições dos tweets.

Na Figura 18, é exibida a estrutura e exemplos dos valores passados como parâmetros ao aplicativo.

1 {

2 "**database**": {

3 "host": "localhost",

4 "port": 27017,

5 "search\_database\_name": "search\_database",

6 "database\_name": "TCC",

7 "collection\_name": "2016"

8 },

9 "**queries**": {

10 "lang": "lang:en",

11 "is\_retweet": "-is:retweet",

12 "max\_results": "max\_results=500",

13 "start\_time": "start\_time=2016-01-01T00:00:00Z",

14 "end\_time": "end\_time=2016-12-31T23:59:59Z"

15 },

16 "**fields**": {

17 "tweet\_fields": true,

18 "tweet\_fields\_list": "attachments,author\_id,context\_annotations,conversation\_id,created\_at,entities,geo,id,in\_reply\_to\_user\_id,lang,public\_metrics,possibly\_sensitive,referenced\_tweets,reply\_settings,source,text,withheld",

19 "user\_fields": true,

20 "user\_fields\_list": "created\_at,description,entities,id,location,name,pinned\_tweet\_id,profile\_image\_url,protected,public\_metrics,url,username,verified,withheld"

21 }

22 }

23

Figura 20 – Arquivo de configuração dos parâmetros do aplicativo

Os parâmetros de conexão ao banco de dados estão organizados na estrutura com a chave “*database*”. Já a chave “*queries*” agrupa os parâmetros principais, que definem a busca. Por último, a chave “fields” define os campos de informações que serão retornados para cada tweet.

A implementação do conceito de domínios e subdomínios foi feita através de uma estrutura de dados de matriz, que na linguagem Python é implementada através da classe de estrutura de dados *array*. O *array*, por sua vez, contém uma série de tuplas, que é outra estrutura de dados da linguagem *Python*. Conforme a Figura 20, podemos ver a implementação das TAGs de domínio (em azul), subdomínio (em laranja) e da palavra-chave definida pelo especialista (em vermelho).

1 palavras\_chave = [

2 ("**treatment**", "**prep**", "**#prep** **(hiv OR treatment**)"),

3 ("treatment", "prep", "#truvada OR truvada"),

4 ("treatment", "prep", "#atripla OR atripla"),

5 ("infection", "-", "#hivinfection OR (hiv infection)"),

6 ("drug", "double dose", "#epzicom OR epzicom"),

7 ("drug", "triple dose", "#complera OR complera"),

8 ("treatment", "prep", "#ftc OR (ftc treatment) "),

9 ("drug", "-", "#drug OR (drug hiv)"),

10 ("infection", "-", "#hivtreatment OR hivtreatment OR (hiv treatment)"),

11 ("drug", "triple dose", "(#3tc OR 3tc) (hiv OR treatment)"),

12 ("drug", "triple dose", "#tripletherapy OR (triple therapy)"),

13 ("infection", "-", "#anti OR anti (hiv OR treatment)"),

14 ("drug", "antiretroviral", "#isentress OR isentress"),

15 ("drug", "antiretroviral", "#reyataz OR reyataz"),

16 ("drug", "antiretroviral", "#norvir OR norvir"),

17 ("infection", "-", "#livingwithaids OR (living aids)"),

18 ("infection", "-", "#NormalizingHIVChallenge OR (Normalizing HIV Challenge)"),

19 ("drug", "antiretroviral", "#sustiva OR sustiva"),

20 ("drug", "antiretroviral", "#stocrin OR stocrin"),

21 ("infection", "-", "#viread OR viread"),

22 ("treatment", "pep", "#pep (hiv OR treatment)"),

23 ("treatment", "pep", "#pepforhiv OR (pep hiv)"),

24 ("treatment", "pep", "#pepforearlyhiv OR (pep for early hiv) or (pep hiv)"),

25 ("treatment", "pep", "#pepindelhi OR (pep delhi)"),

26 ("treatment", "pep", "#peptreatment OR (pep treatment)"),

27 ("hiv center", "-", "#peptreatmentinmalviyanagar or (pep treatment malviyanagar) OR (pep malviyanagar)"),

28 ("hiv center", "-", "#pepcenterforhiv OR (pep center for hiv) "),

29 ("hiv center", "-", "#pephivcenter OR (pep center hiv)"),

30 ("infection", "-", "#pepforealryexposer OR (pep real exposer)"),

31 ("treatment", "-", "#pepandprep OR (pep prep) OR ((pep OR prep) treatment)")

32 ]

Figura 21 : Arquivo de palavras-chave da busca

Através de observações empíricas, obtidas através de testes preliminares, observou-se que algumas palavras-chave eram fracas, ou seja, promoviam buscas imprecisas, que retornavam tweets relacionados a domínios alheios ao interesse deste trabalho. Para resolver este resultado indesejado, foi feita a combinação da palavra-chave com palavras referentes aos domínios, como podemos observar em uma das tuplas (em verde) da Figura12.

Ademais, é necessário o fornecimento de uma chave eletrônica, denominada *Bearer Token,*  que permite a autenticação entre o aplicativo *Twitter API Search* e a API do Twitter através do protocolo HTTP.

O projeto do aplicativo busca assegurar a segurança da conta do *Twitter,* utilizada neste projeto, além de seguir as recomendações tanto dos padrões da RFC6750 quanto do *framework* *OAuth 2.0.* Para tal objetivo, o *token* deve ser armazenado em uma variável de ambiente do sistema operacional, chamada “*BEARER TOKEN*”, conforme a Figura 22. O aplicativo se encarregará de ler a variável de ambiente e autenticar na API do Twitter.

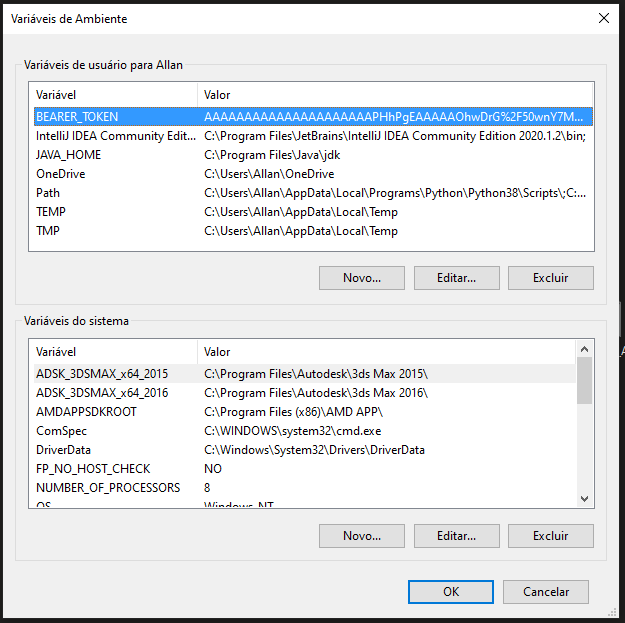


Figura 22 : Configuração da variável de ambiente *Bearer Token*

As principais informações sobre as buscas são gravadas em arquivos de LOG, conforme exemplo mostrado na Figura 23, permitindo extrações de informações importantes como: horário de início e do término (em cinza), quantidade de requisições (em verde e amarelo) e erros de cada busca (em vermelho).

Figura : Exemplo do arquivo de log gerado para as buscas

# Buscando Tweets com '#tripletherapy OR (triple therapy)'

# Início da requisicao efetuada em 23/6/2021, às 17:50:15

https://api.twitter.com/2/tweets/search/all?query=%23tripletherapy%20OR%20%28triple%20therapy%29%20lang%3Aen%20-is%3Aretweet&max\_results=500&start\_time=2017-01-01T00:00:00Z&end\_time=2017-12-31T23:59:59Z&&tweet.fields=attachments,author\_id,context\_annotations,conversation\_id,created\_at,entities,geo,id,in\_reply\_to\_user\_id,lang,public\_metrics,possibly\_sensitive,referenced\_tweets,reply\_settings,source,text,withheld&user.fields=created\_at,description,entities,id,location,name,pinned\_tweet\_id,profile\_image\_url,protected,public\_metrics,url,username,verified,withheld

Total de Tweets na requisicao: 500

next\_token: 1jzu9lk96gu5npw3jozruz9xxr9exxkjrn989d8nlv99

Total de Tweets na requisicao: 500

next\_token: 1jzu9lk96gu5npw3jg2jyxeq8hxnibpf14atsrjju1rx

Total de Tweets na requisicao: 500

next\_token: 1jzu9lk96gu5npw3j43gpgjyxi8g4yvtsqtqcb20sepp

Total de Tweets na requisicao: 500

next\_token: None

Total de Tweets na requisicao: 127

# Fim da requisicao efetuada em 23/6/2021, às 17:51:24

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Buscando Tweets com '#anti OR anti (hiv OR treatment)'

# Início da requisicao efetuada em 23/6/2021, às 17:51:24

https://api.twitter.com/2/tweets/search/all?query=%23anti%20OR%20anti%20%28hiv%20OR%20treatment%29%20lang%3Aen%20-is%3Aretweet&max\_results=500&start\_time=2017-01-01T00:00:00Z&end\_time=2017-12-31T23:59:59Z&&tweet.fields=attachments,author\_id,context\_annotations,conversation\_id,created\_at,entities,geo,id,in\_reply\_to\_user\_id,lang,public\_metrics,possibly\_sensitive,referenced\_tweets,reply\_settings,source,text,withheld&user.fields=created\_at,description,entities,id,location,name,pinned\_tweet\_id,profile\_image\_url,protected,public\_metrics,url,username,verified,withheld

Total de Tweets na requisicao: 499

next\_token: 1jzu9lk96gu5npw44kghujitc3zx70bcd9v79wq8imwt

Total de Tweets na requisicao: 498

Erro 503. Detalhes: {"title":"Service Unavailable","type":"about:blank","status":503,"detail":"Service Unavailable"}

next\_token: 1jzu9lk96gu5npw44kf831reiunnwwq7bx4p0twewewt

Total de Tweets na requisicao: 499

next\_token: 1jzu9lk96gu5npw44ket82uuudk1uzg1ncals5rgakcd

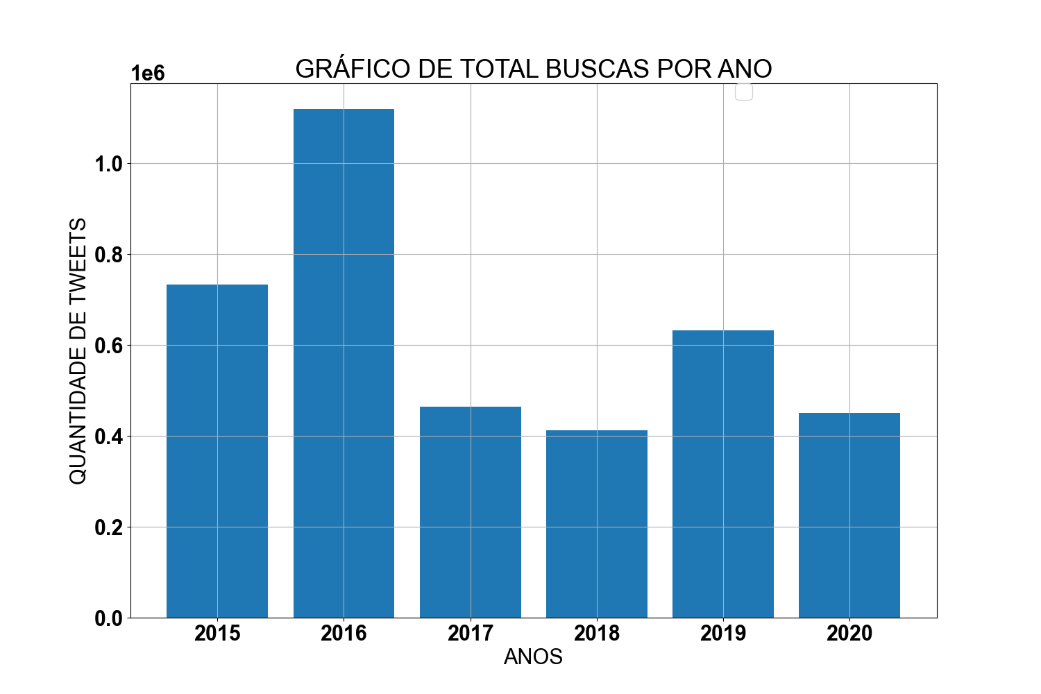
Total de Tweets na requisicao: 499

.

.

.

Nos testes realizados, foram extraídos tweets de 4 anos, de 2015 a 2020, totalizando 3.809.864 tweets. Conforme o Gráfico 1, podemos ver o total por ano e analisar a variação de comentários sobre medicamentos por HIV durante o intervalo supracitado.



**449.847**

**632.474**

**411.599**

**463.602**

**1.119.893**

**732.449**

Gráfico 1: Total de buscas por ano

Já nos Gráficos 2, 3 e 4, podemos analisar a variação de cada palavra-chave buscada, durante o intervalo de 2015 a 2020.

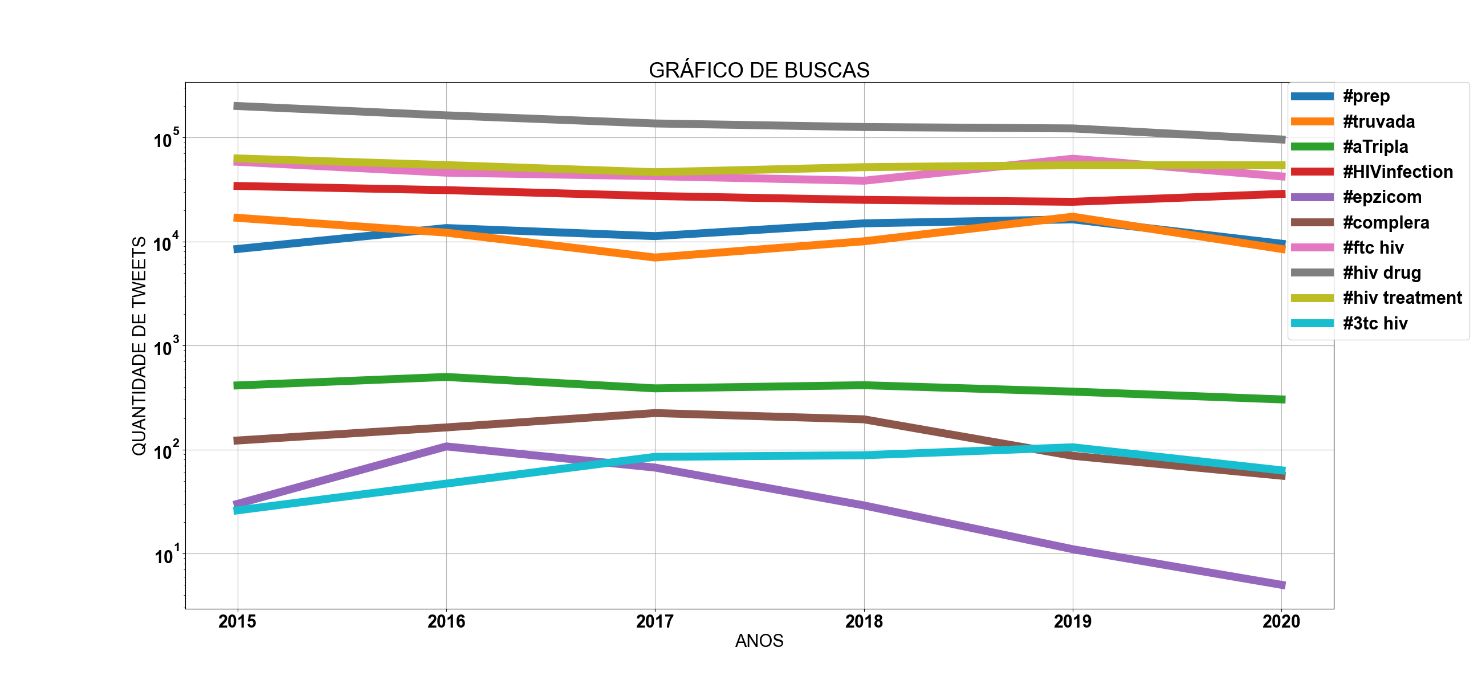


Gráfico 2: Gráfico anual de *Tweets* por busca(1/3)

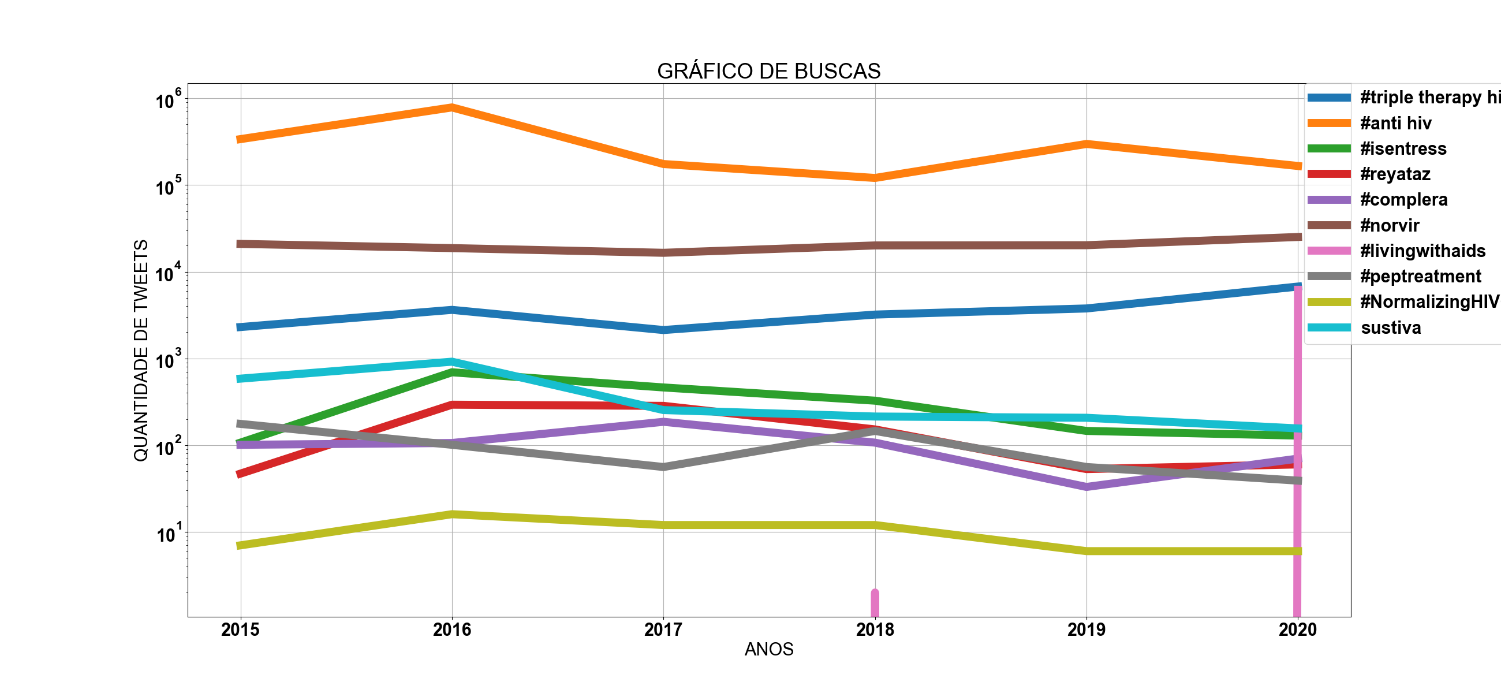


Gráfico 3: Gráfico anual de *Tweets* por busca(2/3)

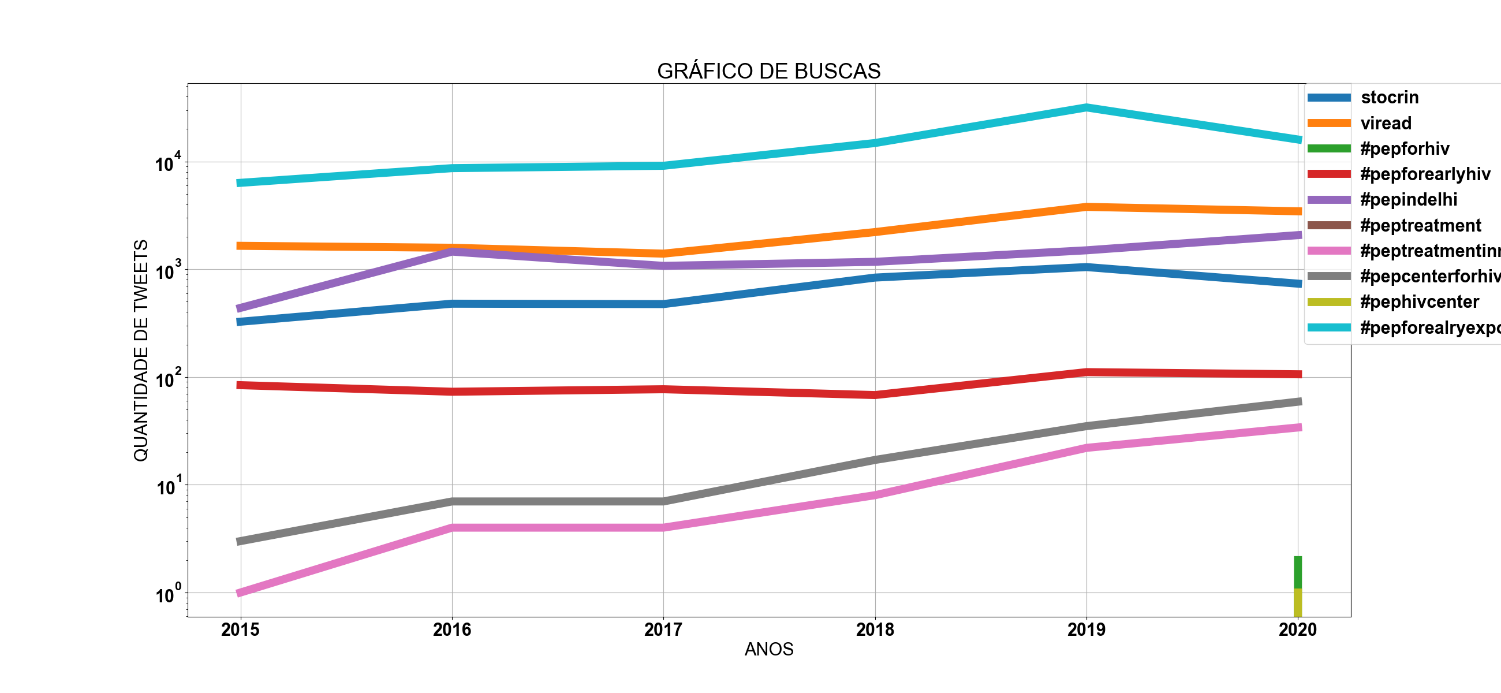


Gráfico 4: Gráfico anual de *Tweets* por busca(3/3)

De acordo com os gráficos 1, 2, e 3, fica evidente que palavras-chave que buscam textos sobre fármacos específicos, retornam menos tweets que as buscas por assuntos mais genéricos. Todavia, não faz parte deste trabalho a correlação desses dados a fatores externos. Sendo assim, os gráficos permitem-nos uma observação da quantidade de tweets achados de acordo com a linguagem utilizada pelos usuários do Twitter ao relatar o uso de fármacos nas redes sociais.

# CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho evidenciou-se a importância de dados de redes sociais em pesquisas. Com o uso da API do Twitter, foi possível coletar tweets relacionados aos medicamentos para HIV de forma sistemática. Organizando-os cronologicamente, por idioma, localização e de acordo com os assuntos de interesse.

O objetivo em se obter um grande volume de dados sobre medicamentos, para posterior análise, foi alcançado. Foi possível coletar uma quantidade considerável de tweets sobre assuntos de interesses específicos.

Através de observações empíricas dos testes preliminares, houve um aprimoramento no uso das palavras-chave, utilizadas para as buscas. Tal melhoria deu-se através da combinação de cada palavra-chave, provida pelo especialista, com o par de palavras referentes ao domínio e ao subdomínio. Limitando sempre as palavras-chave a um contexto.

Uma das maiores dificuldades neste projeto foi a compreensão do uso da API. Pois a documentação é extensa e a API se divide em diversas camadas de contexto, objetivos, público e de segurança.

Neste trabalho não foram utilizados tweets que fazem réplicas aos tweets encontrados. Todavia, consideramos que a análise de tweets com boas métricas de aceitação, por parte dos usuários, podem prover um conteúdo valioso para futuras pesquisas, já que essas réplicas estabelecem um padrão de conversação sobre o tweet original.

REFERÊNCIAS BIBLIOGRÁFICAS

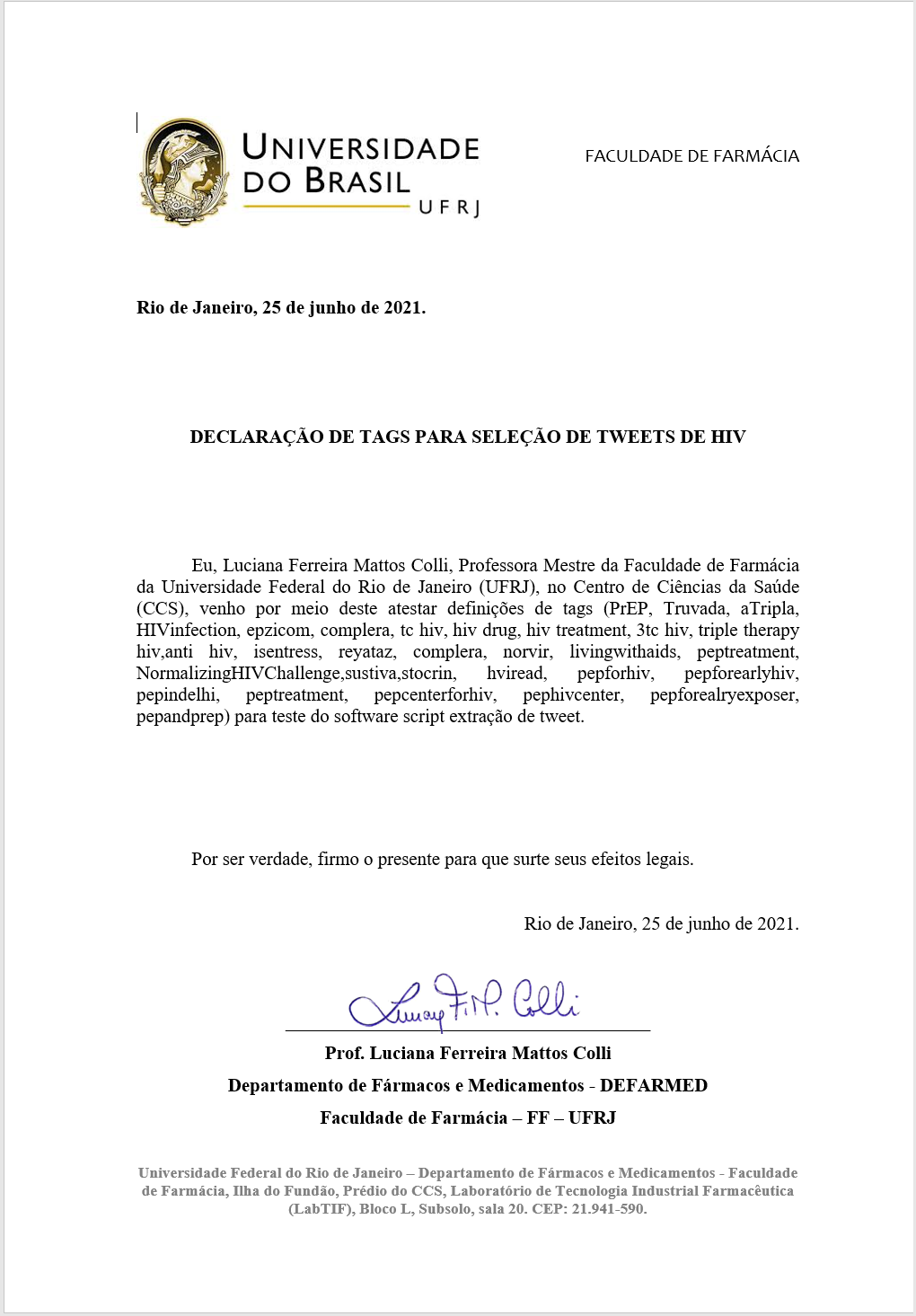
|  |  |
| --- | --- |
| [1] | C. Tech, “https://canaltech.com.br/,” Canal Tech, [Online]. Available: https://canaltech.com.br/empresa/twitter/. |
| [2] | B. FELKEY, H. LIANG e KRUEGER, “Data Mining for the Health,” 2003. |
| [3] | D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato e F. Fdez-Riverola, “Web scraping technologies in an API world,” *Briefings in Bioinformatics,* vol. 15, nº 5, pp. 788-797, 30 Abril 2013. |
| [4] | B. Audeh, M. Beigbeder, A. Zimmermann, P. Jaillon e C. Bousquet, “Vigi4Med Scraper: A Framework for Web Forum Structured Data Extraction and Semantic Representation,” *PLOS ONE,* vol. 12, nº 1, p. 12, 21 Janeiro 2017. |
| [5] | B. Audeh, F.-E. Calvier, F. Bellet, M.-N. Beyens, A. Pariente, A. Lillo-Le Louet e C. Bousquet, “Pharmacology and social media: Potentials and biases of web forums for drug mention analysis—case study of France,” *Health Informatics Journal,* pp. 1253-1272, Junho 2020. |
| [6] | N. Alvaro, M. Conway, S. Doan, C. Lofi, J. Overington e N. Collier, “Crowdsourcing Twitter annotations to identify first-hand experiences of prescription drug use,” *Journal of Biomedical Informatics,* vol. 58, pp. 280-287, 1 Dezembro 2015. |
| [7] | C. Adrover, T. Bodnar e M. Salathe, “Targeting HIV-related Medication Side Effects and Sentiment Using Twitter Data,” *arXiv:1404.3610,* 11 Abril 2014. |
| [8] | C. Taurion, Big Data, Brasport, 2013. |
| [9] | S. M. Kabir, BASIC GUIDELINES FOR RESEARCH: An Introductory Approach for All Disciplines., 2016. |
| [10] | F. Faleiros, C. Käppler, F. A. R. Pontes, S. S. d. C. Silva, F. d. S. N. d. Goes e C. D. Cucick, “USE OF VIRTUAL QUESTIONNAIRE AND DISSEMINATION AS A DATA COLLECTION STRATEGY IN SCIENTIFIC STUDIES,” *Scielo,* vol. 25, 24 Outubro 2016. |
| [11] | M. N. Magalhães e A. C. P. d. Lima, Noções de probabilidade e estatística, São Paulo: Editora da Universidade de São Paulo, 2002. |
| [12] | M. N. Zozus, The Data Book: Collection and Management of Research Data, Boca Raton: CRC Press, Taylor & Francis Group, 2017. |
| [13] | Salesforce, “Data Warehouse e Data Lake: O que são?,” Blog da Salesforce, [Online]. Available: https://www.salesforce.com/br/blog/2020/10/data-warehouse-e-data-lake.html. [Acesso em 30 Junho 2021]. |
| [14] | S. Abiteboul, P. Buneman e D. Suciu, Data on the web: from relations to semistructured data and XML, M. Kaufmann, Ed., San Francisco: Elsevier, 1999. |
| [15] | T. Berners-Lee, "The World Wide Web: Past, Present and Future," W3, Agosto 1996. [Online]. Available: https://www.w3.org/People/Berners-Lee/1996/ppf.html. [Accessed 19 Abril 2021]. |
| [16] | M. S. Silva, HTML5: a linguagem de marcação que revolucionou a web, 2019. |
| [17] | E. Ferreira e D. Eis, HTML5: Curso W3C Escritório Brasil, São Paulo: W3C, 2011. |
| [18] | C. J. Costa, Desenvolvimento para WEB, ITML press / Lusocredito, 2007. |
| [19] | L. Maldonado, “Entendendo o DOM (Document Object Model),” Tableless, 08 Fevereiro 2018. [Online]. Available: https://tableless.com.br/entendendo-o-dom-document-object-model/. [Acesso em 19 Abril 2021]. |
| [20] | A. Grosskurth e M. W. Godfrey, “Architecture and evolution of the modern web browser,” *lsevier Science,* pp. 235-246, 2006 Junho 20. |
| [21] | M. Walfish, H. Balakrishnan e S. Shenker, “Untangling the Web from DNS.,” *NSDI,* 2004. |
| [22] | R. Mitchell, Web Scraping com Python: Coletando mais dados da web moderna., Novatec Editora, 2019. |
| [23] | Twitter, “Standard v1.1 | Docs | Twitter Developer Platform,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/v1. [Acesso em 06 Junho 2021]. |
| [24] | Twitter, “Premium v1.1 | Docs | Twitter Developer Platform,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/premium. [Acesso em 07 Junho 2021]. |
| [25] | Twitter, “Enterprise | Docs | Twitter Developer Platform,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/enterprise. [Acesso em 07 Junho 2021]. |
| [26] | Twitter, “What's New with Twitter API v2,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/early-access. [Acesso em 08 Junho 2021]. |
| [27] | C. RedHat, “application-programming-interfaces,” RedHat, [Online]. Available: https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces. |
| [28] | E. Júnior, “Eximiaco,” Eximiaco Excelência Tecnológica , 13 09 2019. [Online]. Available: https://www.eximiaco.tech/origem-e-significado-de-rest-e-restful/. |
| [29] | R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” p. 96, 2000. |
| [30] | C. Twitter, “https://developer.twitter.com/,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/early-access. |
| [31] | C. Twitter, “https://developer.twitter.com,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction. |
| [32] | Twiiter, “Discover your next research project,” Twiiter, [Online]. Available: https://developer.twitter.com/en/use-cases/do-research/academic-research/research-areas. [Acesso em 01 Julho 2021]. |
| [33] | Twitter, “Getting historical Tweets using the full-archive search endpoint,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/tutorials/getting-historical-tweets-using-the-full-archive-search-endpoint. [Acesso em 01 Julho 2021]. |
| [34] | Twitter, “Twitter Developers,” Twitter, [Online]. Available: https://developer.twitter.com/en/portal/products/academic. [Acesso em 01 Julho 2021]. |
| [35] | Twitter, “Tweet object,” Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet. [Acesso em 01 Julho 2021]. |
| [36] | M. Meng, S. Steinhardt e A. Schubert, “Application Programming Interface Documentation: What Do Software Developers Want?,” *Journal of Technical Writing and Communication,* pp. 295-330, Julho 2018. |

ANEXOS

ANEXO A – DOCUMENTO DE ACEITAÇÃO ASSINADO PELO ESPECIALISTA



ANEXO B – DOCUMENTO DE DECLARAÇÃO DE TAGS DEFINIDAS PELO ESPECIALISTA



ANEXO C – PREECHIMENTO DE FORMULÁRIO PARA APROVAÇÃO DA CONTA ACADEMIC RESEARCH.

**What´s your research project´s name?**

Pharmaco epidemiological and therapeutic efficacy assessment of AIDStreatment with fixed-dose-combination (FDC) and its different drugcombinations

**What´s your research about?**

This research involving new treatments for AIDS has mobilized both private and public  
entities worldwide. Several pharmaceutical companies apply for patents of new drugs  
and launch innovative products with different dosage forms and treatment options.  
Currently, there are no standard treatment guidelines with an international scope for  
treatment using different combinations of antiretroviral. Therefore, the treatment  
efficacy of drug combinations in comparison with monotherapy is still not established.  
Thus, this research is an interesting approach to highlight the discrepancies  
among the treatments adopted in different parts of the world in patient’s opinion and comments in Twitter.

**In English, describe how Twitter data and/or Twitter APIs will be used in your research project.**

Twitter will be used in order to observe the comments of users of antiretroviral therapy for AIDS, the different drugs possibly used and the reports of adverse events, unwanted effects. This will all be done using the Twitter API.

**Will your research present Twitter data individually or in aggregate?**

The research will present data in aggregate way.

**In English, describe your methodology for analyzing Twitter data, Tweets, and/or Twitter users.**

The project aims to select a specific group of opinions on the Twitter platform. This group has the characteristics of being HIV-positive people and having commented on the drug's effect. We already have a list of keywords of interest to the project. We have already found some characteristics of spurious Twitter that deviate from the subject of the group of interest. In this case, we will define some features that will be used to train a classifier to separate these spurious twitters from the interest ones. We will make relevant statistics on the reported symptoms with drug combinations and monotherapy with the targeted community Twitter. Finally, we will apply the sentiment function to the two groups and identify positive and negative relations.

**In English, describe how you will share the outcomes of your research (include tools, data and/or resources).**

We will share the outcomes in scientific publication, in form of article.