UNIVERSIDADE FEDERAL FLUMINENSE

MAYK CALDAS RAMOS
PEDRO HENRIQUE RIBEIRO BICHARA

QScrapper2.0: Web scraping de dados do Quora referentes à medicamentos para HIV

NITERÓI

2022

MAYK CALDAS RAMOS
PEDRO HENRIQUE RIBEIRO BICHARA

QScrapper2.0: Web scraping de dados do Quora referentes à 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

2022

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

R175q Ramos, Mayk Caldas
QScrapper2.0: Web scraping de dados do Quora referentes à medicamentos para HIV / Mayk Caldas Ramos, Pedro Henrique Ribeiro Bichara. - 2022.
56 f.: il.

Orientador: Altobelli de Brito Mantuan.
Trabalho de Conclusão de Curso (graduação)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2022.

1. Web Scraping. 2. Quora. 3. Scrapy. 4. HIV. 5. Produção intelectual. I. Bichara, Pedro Henrique Ribeiro. II. Mantuan, Altobelli de Brito, orientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

MAYK CALDAS RAMOS
PEDRO HENRIQUE RIBEIRO BICHARA

QScrapper2.0: Web scraping de dados do Quora referentes à 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.

Aprovada em 10 de Dezembro de 2022.

BANCA EXAMINADORA:

Prof. ALTOBELLI DE BRITO MANTUAN, Dr. - Orientador
Universidade Federal Fluminense - UFF

Prof. Dr. LEANDRO SOARES DE SOUSA - Avaliador
Universidade Federal Fluminense - UFF

Niterói
2022

"Science is like sex: sometimes something useful comes out,
but that is not the reason we are doing it."
Richard Feynmann

"I'm woven in a fantasy.
I can't believe the things I see.
The path that I have chosen now has led me to a wall"
The wall by Kansas.

Resumo

A coleta dos dados é a primeira etapa de um projeto baseado em dados. No campo farmacêutico, muitas vezes é necessario o entendimento sobre o impacto que um medicamento, já existente ou em desenvolvimento, tem em um determinado grupo. Informações de cunho geral sobre esses medicamentos podem ser obtidas em grupos de discussão expalhados por toda a internet. Redes sociais são uma fonta de informação em constante e exponencial crescimento. O *web scraping* é uma técnica desenvolvida a fim de obter dados relacionados a um determinado tema de uma página da web. Esses dados devem ser, posteriormente, tratados por especialistas uma vez que a internet é uma fonte insegura de informação. O presente trabalho se propõe a desenvolver uma ferramenta para extração de dados da plataforma Quora à partir de palavras chaves pré-selecionadas. O QScraper é desenvolvido utilizando o *scrapy*, um *framework* python para *web scraping* e os dados são armazenados em um bando de dados MongoDB. Foram obtidos 2617 perguntas, 1213 respostas, 67 tópicos, 3698 postagens e 2361 espaços com a utilização de 43 palavras-chave relacionadas à medicamentos contra HIV.

Palavras-chave: *Web Scrapping*, Quora, HIV, *scrapy*, python.

Abstract

Data gathering is the first step in a data-driven project. In the pharmaceutical field, it is often needed to study the impact that a medicine, already commercially produced or in development, may cause in a specific group of people. General information about these drugs can be obtained in online discussion groups. Social networks are a ever-growing source of information. Web scraping is a tool developed aiming to extract data concerning a specific topic from a web page. Usually, these data need to be processed since the internet is not a safe source of information. The current study has as objective the development of a tool for data extraction from Quora, a question-answer portal, based on pre-defined queries. QScraper was developed using scrapy, a python framework for web scraping, and the obtained data was stored using a MongoDB data bank. We obtained 2617 questions, 1213 answers, 67 topics, 3698 posts e 2361 spaces upon search using 43 query words related to HIV drugs.

Keywords: Web Scrapping, Quora, HIV, scrapy, python.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACK acknowledgment packets

API Application Programming Interface

ARPA Advanced Research Projects Agency

CSS Cascade Style Sheet

DNS Domain Name System

HIV Human Immunodeficiency Virus

HTML Hypertext Markup Language

HTTP HyperText Transfer Protocol

IFES Instituto Federal do Espírito Santo

IP Internet Protocol

ITC International Telecommunication Union

JSON JavaScript Object Notation

NCP Network Control Protocol

SQL Structured Query Language

SYN synchronization packages

TCP Transmission Control Protocol

URL Uniform Resource Locator

W3C World Wide Web Consortium

WSL Windows Subsystem for Linux

WWW World Wide Web

XHR XMLHttpRequest

XHTML Extensible Hypertext Markup Language

XML Extensible Markup Language

Sumário

1 Introdução

Dados abertos são imprescindíveis para o avanço do conhecimento científico, visto que constituem dados úteis de livre acesso que têm sido publicados por diferentes organizações, muitas delas ligadas à comunidade científica. Quando esses dados são interligados em um contexto semântico, potencializam o avanço do conhecimento. Esse contexto semântico que se deseja é onde se instaura a Web Semântica, um ambiente onde a informação disponibilizada venha com significado bastante preciso e coeso, viabilizando que máquinas e pessoas trabalhem em conjunto. Para isto, a Web semântica requer que as pessoas façam um esforço extra na codificação de informações em representações passíveis de processamento automático, para que computadores possam processar, interpretar e concatenar dados facilmente (1).

Juntamente com o crescimento do número de páginas codificadas de forma semântica, o número total de usuários utilizando a web cresce num ritmo exponencial.(2) Em 2021, o número de usuários acessando a internet consistentemente chegou a 4.977 bilhões, em âmbito mundial (ver Tabela 1).(3)

Usuários	2005	2010	2017	2019	2021
População mundial (em bilhões)	6.5	6.9	7.4	7.75	7.9
Global	16%	30%	48%	53.6%	63%
Países desenvolvidos	51%	67%	81%	86.6%	90%
Países em desenvolvimento	8%	21%	41.3%	47%	57%

Tabela 1: Avanço da população mundial e da porcentagem desses que tem acesso regular à internet em função do tempo. Adaptado de (3).

A quantidade de informação compartilhada em redes sociais, naturalmente, segue esse padrão de crescimento. A Figura 1 mostra o número de usuários ativos em diversas redes sociais em 2022. Essa enorme quantidade mostra a capacidade que redes sociais tem de difundir informação.(4)

O entendimento de como quanto à veracidade das informações disseminadas é de crucial importância para o combate à desinformação e para o entendimento do impacto

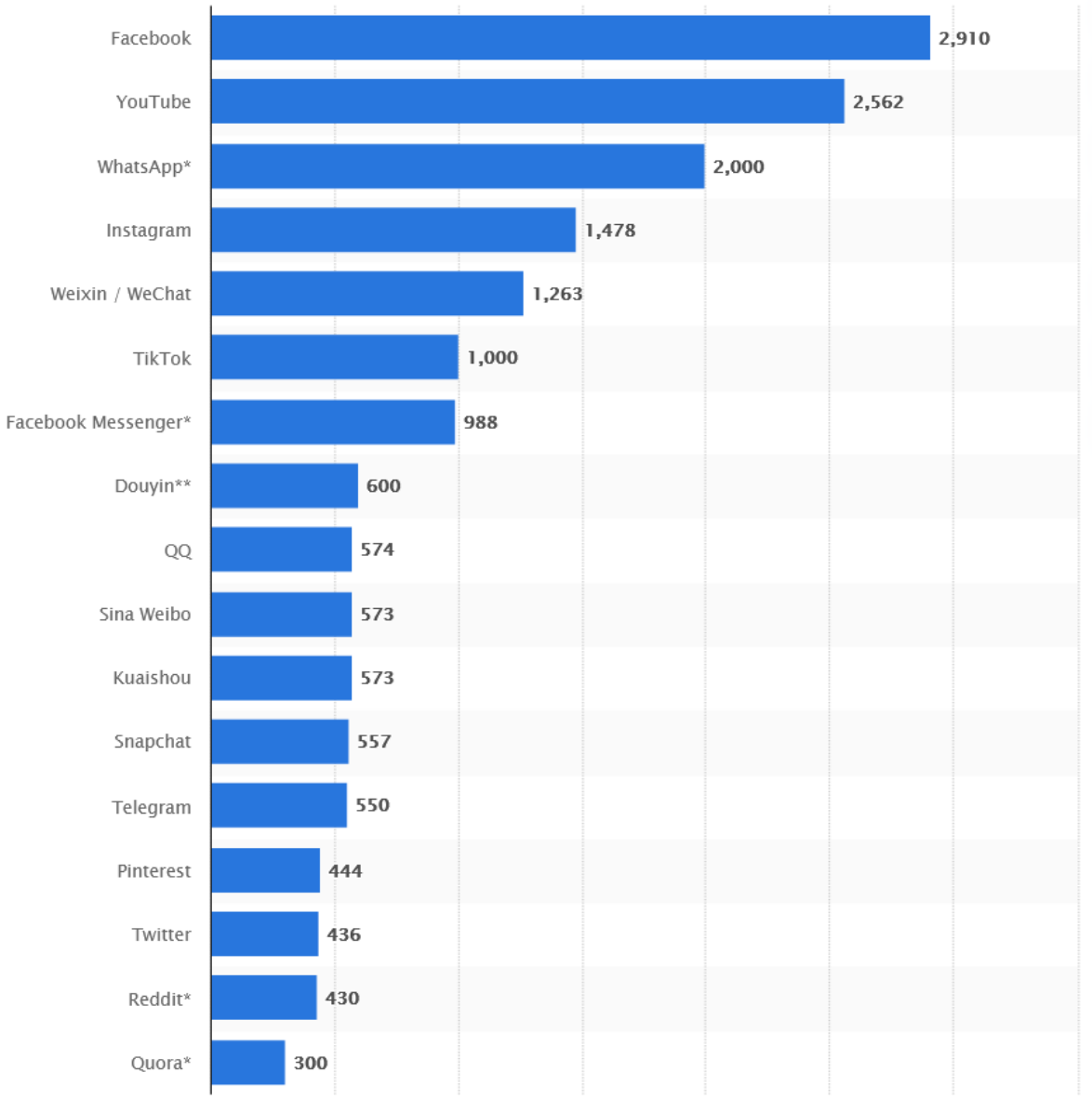

Figura 1: Número de usuários de redes sociais em 2022 (em milhões). Retirado de (4)

que as pesquisas acadêmicas tem fora da comunidade científica. Um assunto de interesse que se coloca nesse cenário, é a pesquisa sobre o quão bem informada a sociedade está acerca de medicamentos para combate ao vírus da HIV. A taxa de mortalidade dessa doença sofreu uma diminuição drástica nos últimos anos, alcançando a taxa de 9,84% em 2018, uma diminuição de 29,16% em 32 anos, a nível mundial. Os medicamentos disponíveis para o tratamento da doença são os responsáveis por essa diminuição(5). Contudo, o tratamento depende da adesão do portador do vírus e os efeitos colaterais são um dos principais motivos para que essa adesão não ocorra.

Neste contexto, o *web scraping* pode ser uma ferramenta útil para obtenção de dados de uma determinada rede para análise do sentimento do usuário de um determinado

medicamento contra a HIV. Dentre as redes disponíveis, o Quora é uma opção devido à semântica contida em seu conteúdo. A plataforma Quora é um grande fórum de perguntas e respostas onde especialistas disponibilizam seu tempo para responder perguntas sobre os mais diversos temas. Contudo, mesmo que o tráfico diário de usuários seja menor (Ver Figura 1), o conteúdo é mais específico e em nichos.

O presente trabalho tem como objetivo o desenvolvimento de uma ferramenta, o QScraper, para extração de dados relevantes sobre medicamentos contra HIV do Quora. O Quora não possui uma API (do inglês, *Application Programming Interface*) que permita a extração dos dados de forma direta. Por isso, o desenvolvimento do QScraper se faz altamente necessário não só para a extração dos dados de interesse nesse trabalho, mas também para estudos futuros. O presente trabalho é uma continuação da pesquisa feita por Leandro Barifouse De Souza e Rafael Mendes Matos(6), onde o desenvolvimento do QScraper foi iniciado. No trabalho anterior, o QScraper foi capaz de obter dados acerca de perguntas e respostas do Quora. No presente trabalho, o objetivo é a extensão desse escopo e a implementação necessária para realizar a extração de outros componentes do Quora (a saber, tópicos, postagens e espaços de discussão).

O QScraper foi utilizado para extração dos elementos referidos anteriormente relevantes a medicamentos contra HIV. Para isso, 43 palavras-chaves para pesquisa foram definidas em parceria com o LabTIF (Laboratório de Tecnologia Industrial Farmacêutica) da Faculdade de Farmácia da Universidade Federal do Rio de Janeiro (UFRJ) para obtenção dos dados (Ver Anexo A). A análise e pós-processamento dos dados foge do escopo deste trabalho e será realizada por pesquisadores especialistas da área.

Este trabalho é composto por esta introdução, pelo Capítulo 2 que trata de trabalhos relacionados para reforçar e validar ainda mais a tese aqui exposta e a importância desta pesquisa, pelo Capitulo 3 onde se encontra toda a fundamentação teórica para a realização desta pesquisa, pelo Capitulo 4 que demonstra todo o desenvolvimento da ferramenta Qscraper para atingir o objetivo proposto e pelo Capítulo 5 que discorre sobre os resultados alcançados por este trabalho.

O código desenvolvido no presente trabalho está hospedado no GitHub e pode ser acessado através do *link*: https://github.com/altobellibm/CEDERJ_2022_MAYK_PEDRO.

2 Trabalhos relacionados

Neste projeto, o objetivo foi de obter opiniões de usuários que utilizam medicamentos para o tratamento de HIV (Do inglês, *Human immunodeficiency virus*). É uma pesquisa que será de grande valia para outras áreas da ciência, principalmente a medicina e a farmácia. Segundo pesquisa realizada pela PUCRS em 2019 a grande maioria dos portadores de HIV preferem ficar em silêncio sobre sua doença.

> [..]Para 81% das pessoas entrevistadas ainda é muito difícil revelar que vivem com HIV. Em geral, as pessoas responderam que não têm boas experiências ao revelar sua condição positiva para o HIV a quem não é próximo.

Essa pesquisa relatou ainda que "Mais de 64% das pessoas vivendo com HIV ou AIDS já sofreram alguma forma de estigma ou discriminação". Visamos obter um feedback mais completo analisando os dados disponíveis na internet por conta da possibilidade do usuário poder se expressar de forma anônima, tendo em vista que uma parte considerável das pessoas soropositivas decidem reduzir seu contato social, conforme ilustrado na Figura 2.

Atitudes que reduzem contato social	
Eu decidi não participar de eventos sociais ($n = 1.665$)	371 (22,3%)
Eu decidi não procurar atendimento de saúde ($n = 1.736$)	180 (10,4%)
Eu decidi não me candidatar para um emprego ($n = 1.427$)	272 (19,1%)
Eu decidi não buscar apoio social ($n = 1.633$)	253 (15,5%)
Eu me isolei de minha família ou amiga/o(s) ($n = 1.731$)	502 (29,0%)
Eu decidi não fazer sexo ($n = 1.702$)	519 (30,5%)

Figura 2: Número de pessoas diagnosticadas como soropositivo para HIV quando perguntadas sobre atitudes que reduzem o contato social nos últimos 12 meses.

2.1 Web scraping na pandemia

O Web Scraping pode ser definida como "raspagem"de dados diretamente da web, onde extraímos informações relevantes de sites através de *bots*, submetendo os dados à análise posterior. Essa técnica é importante pois a análise auxilia a encontrar padrões e a tomar decisões com maior probabilidade de acerto. Utilizando essa ferramenta na plataforma Quora, é possível encontrar uma comunidade de pessoas discutindo sobre o uso de medicamentos para o tratamento do HIV.

Em 2020, o Facebook juntamente com a Carnegie Mellon University conseguiu criar um mapa de contágio da Covid-19 nos Estados Unidos através de coleta de dados pela própria plataforma, conforme ilustrado pela Figura 3.

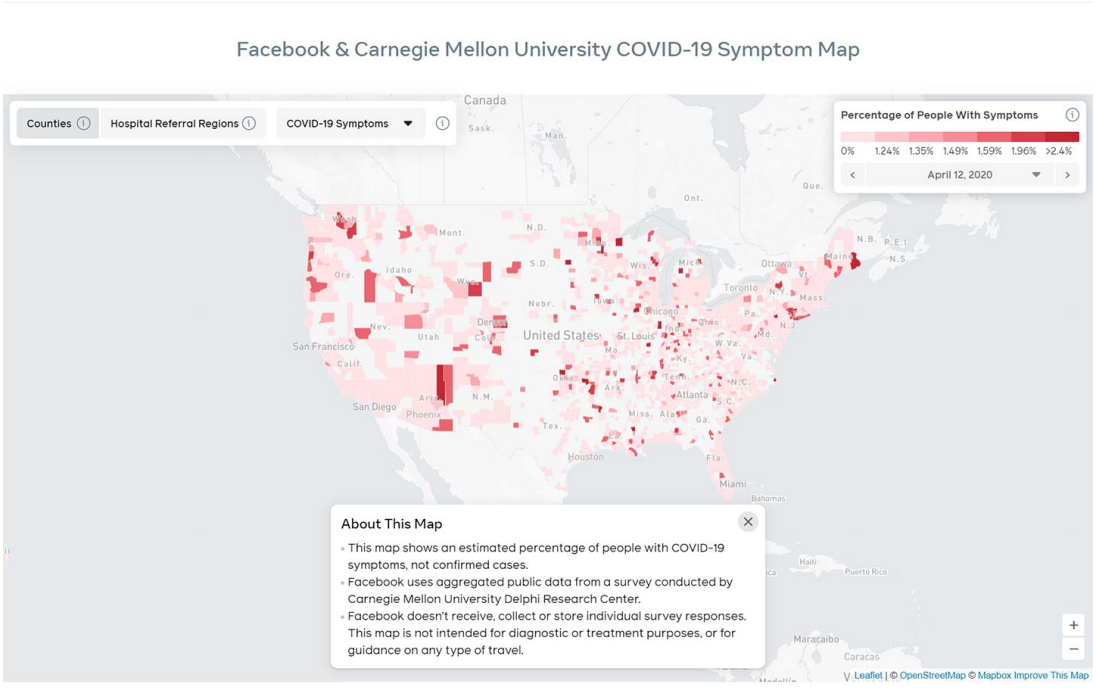

Figura 3: Mapa de contágio de Covid-19 publicado pelo Facebook juntamente com a Carnegie Mellon University.

Esta pesquisa demonstrou como a extração de dados em redes sociais podem ser úteis para outras áreas da ciência. Com este mapa, foi possível que as autoridades responsáveis inferissem se uma determinada área estaria com risco de um contágio descontrolado do vírus.

2.2 Normalização no Quora

A pesquisa a seguir se propôs a resolver um problema muito comum na plataforma Quora: A enorme quantidade de perguntas repetidas na rede social, que atrapalha não somente os usuários, mas também cria uma dificuldade enorme de se armazenar tantos dados.

> [..] Devido ao enorme número de exceções, nuances e complexidades de língua, a exploração automática de significados em texto não é uma tarefa trivial. Em ambientes modernos como redes sociais, essa problemática é amplificada, já que a informação, em geral, se encontra de maneira não estruturada. (**p.9**, 7)

No processo foram utilizados dois métodos diferentes para comparação das perguntas, e os resultados estão ilustrados nas Figuras 3 e 4.

Figura 4: Comparação entre os métodos por entropia cruzada, Quora.Fonte: (7)

Foi possível perceber que existe um aumento significativo em acurácia com o auxílio das técnicas de Deep Learning e NLP dentro do modelo. Percebe-se que o modelo mais robusto, a partir de *word embeddings* e redes neurais recorrentes, apresenta um resultado muito superior, tanto em termos da entropia cruzada (0.59 vs 2.2) quanto de acurácia (69% vs 58%). Isso comprova a efetividade do mapeamento semântico e das redes neurais com capacidade de memória.

2.3 Web scraping em redes sociais

O Web Scraping pode ser utilizado na coleta de diferentes tipos de dados. Quando se trata de redes sociais, cada plataforma tem sua peculiaridade e nos trazem diferentes tipos de informações para serem extraídas. A pesquisa a seguir teve como objetivo auxiliar uma autoavaliação institucional do Campus Serra do Instituto Federal do Espírito Santo (Instituto Federal do Espírito Santo (IFES)), visto que houve uma baixa adesão por parte de ex-alunos à pesquisa, que foi feita de forma digital. Foram utilizadas as plataformas Linkedin e Escavador, que possuem informações a respeito dos vínculos profissionais e acadêmicos dos usuários. A Figura 5 demonstra os resultados obtidos na pesquisa.

Curso	Egressos	Ambos os sites	Só Escavador	Só LinkedIn	Coletados	%
BSI	72	29	14	8	51	70,83%
TADS	38	6	2	14	22	57,89%
TRC	39	5	9	8	22	56,41%
	149	40	25	30	95	63,75%

Figura 5: Egressos identificados em cada site e em ambos os sites, por curso.

"De uma forma bem simplificada, neste trabalho obtivemos um total de 63,75% (95 dos 149 egressos) de perfis identificados com os resultados integrados dos dois sites" (- p. 56). Esta pesquisa demonstra a utilidade do Web Scraping em redes sociais e como essa ferramenta pode auxiliar a resolver, de forma automatizada variados, tipos de problemas.

3 Fundamentação teórica

Dados do International Telecommunication Union (ITC) (do inglês, International Telecommunication Union), estimam que o número de usuários na internet cresceu 27 vezes no período de 1998 até 2018. (2) Em 1998, apenas 3 a cada 100 habitantes tinha acesso regular à internet, já em 2018 esse número aumentou para 48 a cada 100 habitantes, em âmbito global (Ver Figura 6). Com o aumento exponencial de acesso à internet, a quantidade de informação disponível também experienciou um aumento na mesma proporção. Dado a abundância de dados, projetos que objetivam a análise de dados da internet normalmente devem realizar tarefas como: coleta de dados, classificação e categorização, recuperação da informação, agrupamento dos documentos, extração da informação e, finalmente, análise.(2)

O presente capítulo introduz a ideia de como dados podem ser coletados, classificados e armazenados. Para isso, a estrutura de páginas webs e seu funcionamento assim como o funcionamento de navegadores web é discutido. Finalmente, conceitos de web scraping são apresentados.

3.1 Coleta de dados

A coleta de dados é uma etapa preliminar de pesquisa que visa a obtenção de dados de uma determinada fonte para uso posterior. Dependendo do tipo de pesquisa que esteja sendo feita, a fonte pode ser experimentos, questionários com pacientes, revisão bibliográfica, entre outros.(9) A coleta de dados é uma das etapas cruciais de um projeto. Nessa etapa, as informações pertinentes ao projeto são reunidas e servem como base para toda a argumentação feita para suportar as hipóteses do estudo. Visto isso, essa etapa deve ser feita com cuidado e atenção para que todo o estudo subsequente não seja prejudicado.(10)

Focando especificamente na coleta de dados da internet, a principal técnica utilizada é o web-scraping.(11) Essa técnica consiste no uso de um *software* que simule a navegação

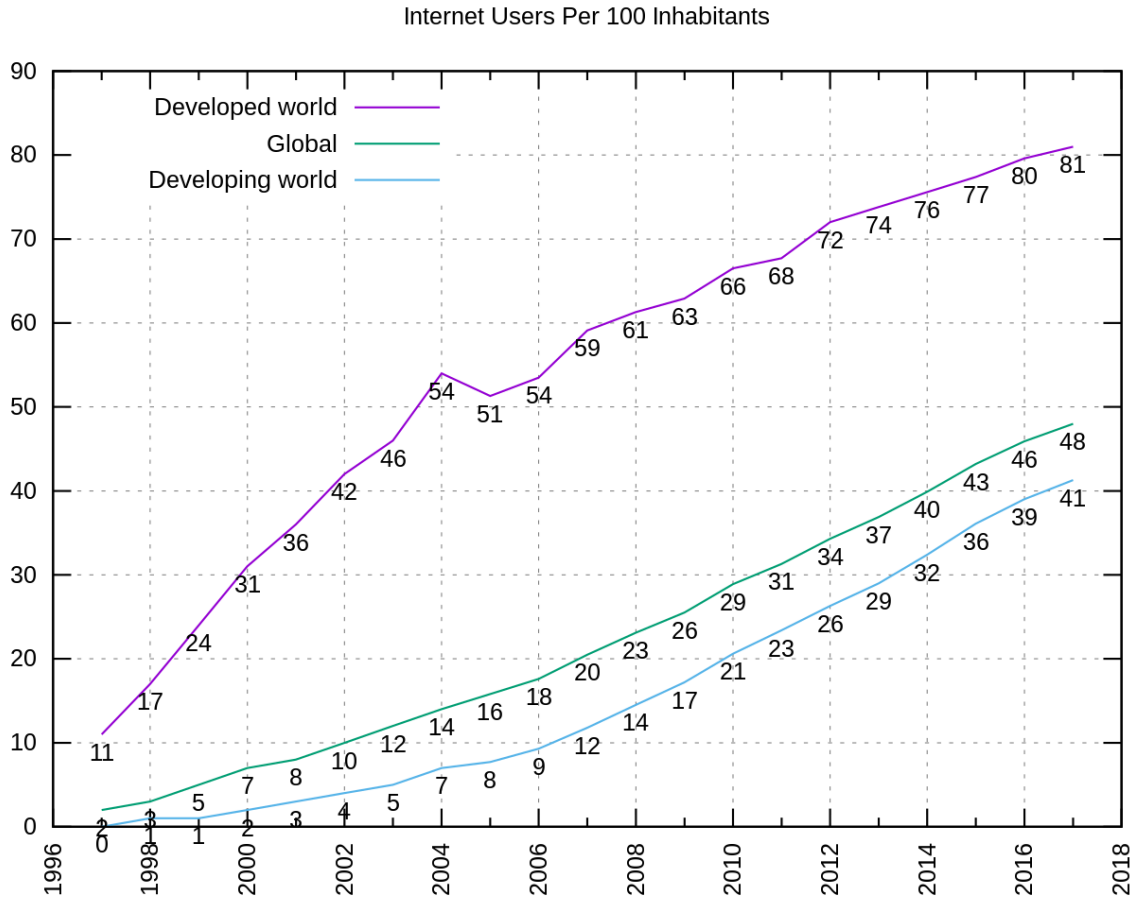

Figura 6: Números de usuários da internet a cada 100 habitantes. A curva verde mostra as estatísticas considerando todo o globo. Enquanto as linhas roxa e azul representam essa mesma estatística considerando somente países desenvolvidos e em desenvolvimento, respectivamente. Dados disponíveis em (8)

pela internet enquanto coleta dados das páginas que são visitadas.(12) Existem diversos algoritmos para promover uma extração utilizando o web-scraping. Eles são baseados no reconhecimento da estrutura do HTML (do inglês, HyperText Markup Language) para promover a coleta da informação semi-estruturada contida na página. (2) Dentre os *softwares* disponíveis, alguns funcionam como programas independentes, outros são desenvolvidos como bibliotecas para implementação do algoritmo em alguma linguagem de programação e alguns utilizam Application Programming Interface (API)s (do inglês, *Application Programming Interface*) disponibilizadas ou pelo Hypertext Markup Language (HTML), ou pelo próprio *website* que se deseja extrair informação.(12)

Por se tratar de uma técnica relativamente nova, muitas questões ainda são levantadas quanto às implicações éticas e legais de se extrair dados de toda a web.(13) Nessa linha, tem sido discutida a possibilidade de websites proibirem explicitamente seu conteúdo

de ser extraído por esses algoritmos atualizando sua política de termos de uso que esteja publicamente disponível em seu website. Camada de proteção que é de grande importância para manter segura informação protegida por copyright.

O presente trabalho, objetiva o estudo do web-scraping como ferramenta de coleta de dados e de obtenção de informação da web. Ou seja, a tarefa de extração de dados é o objetivo *per se*. Dessa forma, não cabe ao escopo deste estudo a análise dos dados obtidos. O principal objetivo é analisar a eficiência da aplicação de algoritmos de web-scraping para realização da etapa de coleta de dados de um projeto focado em informações disponíveis na internet.

3.2 Tipos de dados

Dados podem ser classificados de diversas formas.(2) Uma das classificações de interesse no presente estudo se propõe a dividir os dados entre qualitativos ou quantitativos (Ver Tabela 2).

Tipos de dados			
Qualitativos		Quantitativos	
Discretos	Contínuos	Ordinal	Nominal

Tabela 2: Tipos de dados classificados de acordo com sua qualidade.

Dados quantitativos medem uma grandeza. Eles podem ser representados de forma numérica. Esse tipo de dado ainda pode ser subdividido em discretos ou contínuos. Dados discretos assumem somente valores em um conjunto discreto (como o conjunto dos números naturais). Idade e número de produtos em um inventário são exemplos de dados discretos, pois possuem valores apenas no intervalo [0, 1, 2, 3, 4, 5, ...]. Já os dados contínuos, quando podem ser representados por um conjunto infinito (como o conjunto de números reais). Altura, largura, peso e velocidade são exemplos de dados contínuos.(10)

Já os dados qualitativos expressam uma qualidade. Ainda que possam ser expressos por valores numéricos, também podem ser representados por outros tipos de valores, como textuais. Dados quantitativos podem ser subdivididos em ordinal ou nominal. Dados ordinais podem ser ordenados de alguma forma. Um exemplo de dado qualitativo ordinal pode ser o número e período de um aluno da universidade. O dado é uma característica de cada aluno e pode ser ordenado de forma crescente ou decrescente. Por outro lado, dados nominais não apresentam ordenação intrínseca dos dados. Dados como cor dos olhos, categorias e modelos de produtos são exemplos de dados nominais.(14)

3.3 Organização e Armazenamento

No processo de coleta de dados, é necessário observar como os dados estão organizados. Conforme a sua estrutura, podemos identificar três possibilidades para a informação estar organizada.(2) São elas as formas: Estruturada, semi-estruturada ou não-estruturada.

Informação organizada de forma estruturada é aquela em que os dados são armazenados seguindo um modelo explícito e bem definido. Dados estruturados tem como principal vantagem a facilidade para consulta, uma vez que toda informação é salva de forma pré-definida. Bancos de dados são um bom exemplo de dados estruturados, onde os dados são organizados em tabelas criadas anteriormente. Por isso, consultas Structured Query Language (SQL) funcionam bem nesses tipos de dados. Além de bancos de dados, arquivos binários e imagens são outros exemplos de arquivos que armazenam os dados de forma estruturada.

O armazenamento de informação de forma semi-estruturada é similar à forma estruturada. Porém, dados estruturados de forma semi-estruturada não seguem um modelo estritamente bem definido. Isto é, esse tipo de armazenamento segue uma sintaxe e semântica rígida, mas como essa sintaxe é utilizada pode variar. Exemplos de armazenamento de informação de forma semi-estruturada são os arquivos JavaScript Object Notation (JSON) e linguagens de marcação (*markup languages*) como HTML, Extensible Markup Language (XML) ou Markdown. Nesses arquivos, o conteúdo é inserido utilizando estruturas conhecidas, como as *tags* no caso do HTML, mas essas estruturas podem ser inseridas em ordens diferentes.

Por outro lado, conteúdos não-estruturados são informações que não são organizadas de forma alguma, como dados textuais disponíveis em emails, blogs, websites, documentação, entre outros. Devido a sua simplicidade e por se assemelhar à forma de escrita humana, é a organização mais amplamente utilizada na internet. Informação não-estruturada normalmente é rica semanticamente e de difícil processamento por computadores. Contudo, Moldwin e colaboradores(15) treinaram modelos de *deep neural network* utilizando dados não-estruturados (anotações clínicas) e estruturados (relatórios de equipamentos). Esse estudo mostrou que os modelos treinados utilizando dados não-estruturados foram mais acurados na classificação dos 172 tipos de fenótipos considerados no estudo.

3.4 Páginas Web

A internet é uma megarrede, implementada em nível mundial que interliga diversas redes de computadores e sistemas informáticos cuja finalidade é a transferência de dados e informações de forma direta. (16)

A origem da internet começa em 1969 em um cenário de guerra fria entre os Estados Unidos e a União Soviética. Visando criar uma rede de dados descentralizada de comunicações, um projeto do Departamento de Defesas do Estados Unidos criou a Advanced Research Projects Agency (ARPA) para permitir partilhamento de informações e recursos militares pela rede.

O sucesso dessa rede foi tão grande que deixou de ser exclusivamente de uso militar. centenas de universidades faziam seu uso para implementar pesquisas e disseminar informações no mundo acadêmico. Com isso, em 1983 a ARPAnet foi subdividida em duas redes distintas - a ARPAnet para investigação e a MILnet (*Military network*) para operações militares e começou a usar o Transmission Control Protocol (TCP)/Internet Protocol (IP), abolindo os seus protocolos originais. O grande diferencial do TCP/IP para o antigo protocolo utilizado, o Network Control Protocol (NCP), era a facilitação da comunicação entre as redes sem a necessidade de que estas fizessem alterações em sua interface. Além de garantir o envio e a ordem dos pacotes enviados.(17)

A Primeira página web foi criada em 06 de agosto de 1991 pelo cientista britânico Tim Berners-Lee que revolucionou a ARPAnet apresentando uma forma diferente de compreender a internet. Nesse *web site*, ele descreveu breve detalhes sobre a Rede de Alcance Mundial - vulgarmente chamada de "World Wide Web (WWW)" que é até hoje o modelo utilizado globalmente. A página Web é um documento escrito em linguagem Hyper Text Markup Language (HTML), que se constitui por texto e representações gráficas, armazenado num computador ligado à Internet (18).

A primeira fase da web é conhecida como a "internet das empresas", onde os consumidores (usuários) tinham o poder de apenas consumir o conteúdo colocado pelas empresas. A internet 1.0 é definida pela baixa interatividade entre o usuário e o conteúdo contido nas páginas.(19)

Já a internet 2.0, diante de um crescente número de usuários, teve o surgimento natural de um ambiente marcado pela interatividade. O internauta não somente se comportaria como um espectador passivo, mas sim um gerador de conteúdo através de *blogs*, *sites* para publicações de vídeo, redes sociais e dentre outros. Além disso, também se destaca

a evolução dos mecanismos de busca, liderada pelo Google, que permitiu o usuário a encontrar com mais facilidade o que desejava. Foi nessa era que foram criadas as grandes redes sociais como Orkut, Facebook, Youtube e dentre outras que contribuíram para o que viria a ser a internet 3.0.(20)

Diante de uma internet totalmente interativa e dinâmica e com um aumento significativo de usuários de redes sociais, a quantidade de informações que se obtinha do usuário/consumidor era imensa. Com isso, a chamada internet 3.0 veio para organizar essas informações e usar isto a favor tanto do entretenimento do internauta como também uma forma mais direta de se fazer *marketing*. Se o usuário pesquisar por exemplo, "celulares" no Google, começará a ser exibido para ele diversos anúncios de celulares nos sites que ele frequenta. Essa possibilidade de se fazer o *marketing* digitalmente e de forma direta ao público-alvo é o fator marcante nessa era da internet.

Atualmente, estamos passando pela fase da internet 4.0 onde o foco é o uso de inteligência artificial para otimizar a interpretação da grande quantidade de informações extraídas do usuário. A web 4.0 visa a integração não só de computadores, mas também em objetos que utilizamos no dia a dia. Essa integração é conhecida como "internet das coisas" é uma nova visão para a internet, em que a internet passa a abarcar não só computadores, como, também, objetos do cotidiano.(21)

3.4.1 HTML

O HTML (do inglês, *HyperText Markeup Language*) é uma linguagem de marcação utilizada para construção de páginas web que nasceu juntamente com a página web em 1990, criada pelo cientista Tim Bernes-Lee. A primeira versão do HTML era bem simples porém muito útil, pois implementava o sistema de hipertexto que era bem dinâmico. O hipertexto é um modo de escrita e leitura personalizado sem uma hierarquia pré-estabelecida que podem ser acessados aleatoriamente, pois são idependentes entre si (22)

Em 1995 foi lançada a segunda versão do HTML, que agora ficava a cargo de um grupo chamando HTML Working Group. O HTML 2 foi muito bem aceita no mercado e começou a ter auxílio de empresas de construção de navegadores para o desenvolvimento da linguagem. Entretanto, esse auxílio começou a ciar uma grande problemática. As empresas buscavam desenvolver especificidades para uso próprio de seus navegadores. Com isso, ainda em 1995 foi proposto a criação do HTML 3, que seria controlada por um consórcio chamado World Wide Web Consortium (W3C), a World Wide Web Consortion, que determinaria o padrão de criação de novas tecnologias.(23)

Já em 1997 foi proposto a criação do HTML 4, que ficou conhecida por muitos anos e é utilizada inclusive até hoje. Entretanto, em 2004 o W3C propôs o lançamento de uma nova versão do HTML chamada Extensible HTML ou simplesmente Extensible Hypertext Markup Language (XHTML). Essa decisão extinguiria o HTML 4, fazendo uma quebra no processo de desenvolvimento do HTML.(23)

Neste cenário de criação do XHTML para substituir o processo de desenvolvimento do HTML, as empresas desenvolvedoras de navegadores se juntaram para formar uma grupo chamado WHATSWG. A fundação Mozzila, a Ópera e Apple se viam preocupadas com a possível extinção do HTML e decidiram continuar desenvolvendo o mesmo independentemente da decisão do W3C.(23)

Em 2007, o projeto XHTML estava próximo a chegar na versão 2.0, entretanto o grupo WHATSWG apresentou ao W3C um projeto com retrocompatibilidade com a versão 4 do HTML que chamou a atenção do W3C. Com isso, foi decidido parar o projeto do XHTML e voltar ao projeto HTML, criando assim o HTML 5.

O principal diferencial do HTML 5, que é utilizado até hoje, é a separação entre semântica, estilo e interatividade. Ou seja, tudo que é feito em HTML é semântico, tem significado. Os outros aspectos ficam por conta de outras tecnologias. A estilização é feita por Cascade Style Sheet (CSS) e a interatividade é feita por JavaScript.

3.4.2 Páginas Web Dinâmicas

Com a constante evolução da *web* como um todo e um crescente número de usuários, se viu a necessidade de dinamizar o ambiente virtual da página web, possibilitando a interação do usuário para com a página web. Em sua origem, o conteúdo das páginas web era gerado por uma HTML estática ou arquivos de imagem. A arquitetura utilizada nestas aplicações era cliente-servidor, onde a camada do cliente era composta por um navegador *web*, capaz de interpretar e renderizar as páginas na tela (Conforme indicado na figura 7).(24)

O conteúdo dinâmico, por sua vez, é gerado por uma combinação de um servidor *front-end*, um aplicativo servidor e um banco de dados back-end *(Conforme ilustrado na figura 8)*. O conteúdo dinâmico do *site* é armazenado no banco de dados e o servidor de aplicativos fornece métodos que implementam a lógica das tarefas do aplicativo.Os três servidores (*web*, aplicativo e servidor de banco de dados) podem todos executar em uma única máquina, ou cada uma delas pode executar em uma máquina separada ou em um

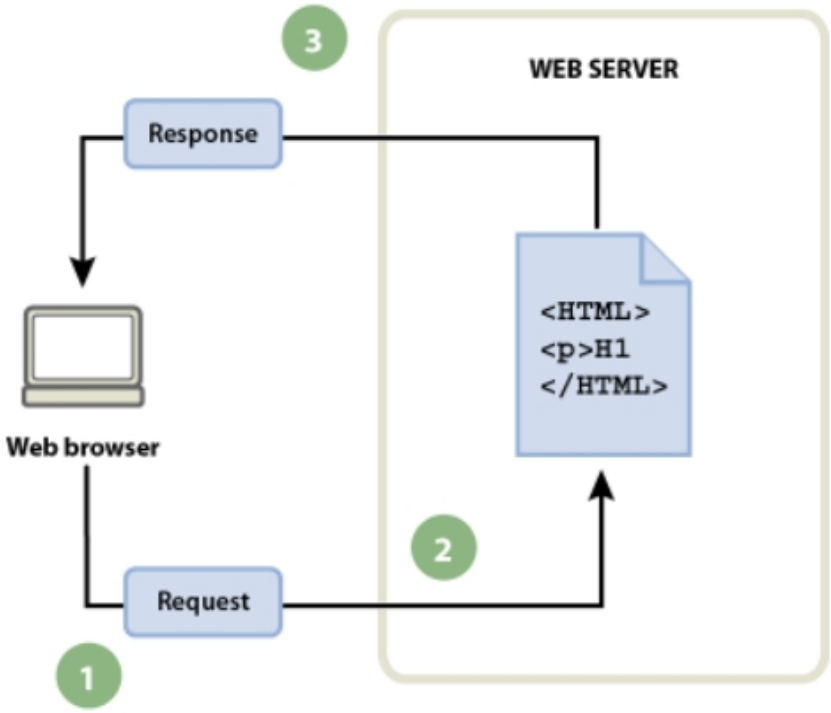

Figura 7: 1. Navegador da Web solicita a página estática. 2. Servidor Web Localiza a página. 3. Servidor Web envia a página para o navegador solicitante.

cluster de máquinas.(25)

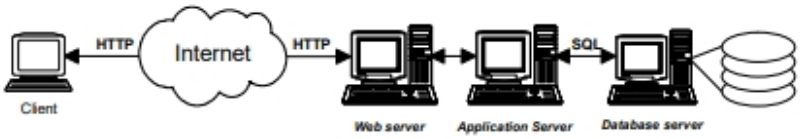

Figura 8: Configuração típica de um conteúdo dinâmico local na rede internet

A *web* dinâmica começou em 1993, quando foram introduzidos os formularios HTML. Entretanto, a HTML limita-se a criar rótulos e campos a serem preenchidos por um usuário e nada mais, não sendo possível processar os dados nem mesmo enviá-los a outro servidor. Para realizar essas tarefas foram então utilizadas linguagens de programação para adicionar e processar dados nas páginas web. Destacam-se as linguagens PHP, ASP, Java, Ruby, Python e ColdFusion, entre outras. As linguagens podem rodar no lado do servidor como por exemplo Python, Java, etc, ou, rodar do lado do cliente, como por exemplo a linguagem Javascript. A difereça é que as linguagems que rodam do lado do servidor necessitam de uma máquina remota para armazenar as funcionalidades capazes de interpretar os programas, enquanto as linguagens que rodam do lado do cliente têm um interpretador hospedado do navegador.(26)

As linguagens podem rodar no lado do servidor como por exemplo Python, Java, etc,

ou, rodar do lado do cliente, como por exemplo a linguagem Javascript. A difereça é que as linguagems que rodam do lado do servidor necessitam de uma maquina remota para armazenar as funcionalidades capazes de interpretar os programas, enquanto as linguagens que rodam do lado do cliente têm um interpretador hospedado do navegador.

3.5 Navegadores Web

No início da internet, o acesso era restrito a pesquisadores, cientistas e militares. Ela era utilizada principalmente para troca de informação entre partes usando protocolos de comunicação desenvolvidos na época, como o HyperText Transfer Protocol (HTTP) (do inglês, *HyperText Transfer Protocol*; o qual é utilizado até hoje. O HTTP define regras sobre como a comunicação entre o cliente e o servidor deve ser feita. A implementação do protocolo HTTP se baseia em definir a estrutura como as mensagens entre essas duas entidades serão enviadas.(27)

Atualmente, os navegadores web usam esse protocolo para requisitar dados de páginas web aos servidores e exibí-las em tela. Os navegadores implementam o lado do cliente do protocolo HTTP enquanto os servidores implementam o restante do protocolo.

O protocolo funciona da seguinte forma. Primeiramente, o usuário insere a página web que deseja acessar. Normalmente, essa página é inserida na forma de um endereço Uniform Resource Locator (URL) por ser uma forma mais conveniente para o usuário do que memorizar endereços IP. Uma requisição é feita ao servidor Domain Name System (DNS) mais próximo e o endereço de IP é retornado. Com o conhecimento do endereço da página pedida, o navegador pode começar a execução do protocolo HTTP. Para isso, é necessária uma conexão entre o cliente e o servidor, onde os pacotes synchronization packages (SYN) e acknowledgment packets (ACK) são utilizados. O protocolo HTTP usa o protocolo TCP (do inglês, *transmission control protocol* para isso. Isto é, ele envia uma mensagem TCP (SYN) requisitando uma conexão e, em Seguida, o servidor deve responder (pacote SYN+ACK) informando se a conexão foi estabelecida ou não. Com a conexão estabelecida do lado do servidor, o navegador - o lado do cliente - deve enviar uma mensagem de concordância (ACK) para estabelecer a conexão dos dois lados. Esse processo do protocolo IP é conhecido como *three-way handshake* e está ilustrado na Figura 9.

Com a conexão TCP estabelecida, o navegador faz uma requisição HTTP ao servidor, que responde enviando os dados da página web.(29) Quando os dados requisitados são

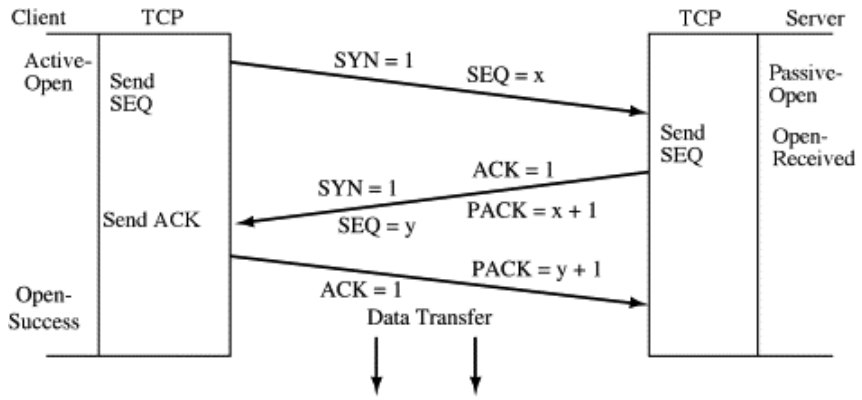

Figura 9: Representação do Three-way handshake. O cliente inicial a comunicação enviando um pacote SYN para o servidor, que deve responder com um pacote SYN+ACK aceitando a conexão. Em seguida o cliente deve enviar um pacote ACK estabelecendo a conexão. Após esse processo, a requisição de dados pode ser feita.(28)

enviados, a conexão pode ser encerrada.

3.6 Web Scraping

Web scraping é uma técnica recente que permite a extração de dados de websites e de mídias sociais.(30) Devido à sua eficiência, essa técnica tem sido amplamente utilizada para reduzir o esforço humano na busca por informação através da internet.(31) Existem diversos programas e algoritmos que podem ser utilizados. De forma geral, o funcionamento de um algoritmo de web scraping segue o seguinte procedimento: (*i*) Primeiramente um site é selecionado. O programa então (*ii*) faz uma requisição ao endereço fornecido obtendo o HTML e os dados contidos na página. Em seguida, (*iii*) esses dados podem ser convertidos e armazenados de forma conveniente conforme a proposta do projeto.(32) Em seguida, esses dados podem ser utilizados para análise.

Navegadores web são uma forma conveniente para coleta de dados se feito por um agente humano. Com a execução de scripts codificados em JavaScript, os dados podem ser exibidos de uma forma visual de fácil entendimento para o leitor. Contudo, coletar dados uma página por vez é um processo custoso e ineficiente. Por isso, para obtermos dados de forma mais eficiente, podemos utilizar web scrappers para coletar uma abundância de dados de uma só vez. Dessa forma, bases de dados composta de milhões de páginas podem ser acessadas de uma só vez.(12)

Web scraping é ainda mais conveniente por não depender de uma API desenvolvida. Como o web scraping acessa o html da página, qualquer dado que pode ser acessado pelo

browser pode ser acessado pelo *script* do *scrapper*. Assim, podemos coletar dados de sites que não possuem uma API disponível. Até mesmo páginas que disponibilizem APIs, muitas vezes limitam a quantidade de tráfego por acesso; o que não é um problema para os algoritmos de *web scraping*.(32)

4 Desenvolvimento

O presente trabalho foca no desenvolvimento do QScraper, um aplicativo para *web scraping* da plataforma Quora. A fim de realizar essa raspagem, diferentes métodos foram buscados. Como nenhuma biblioteca eficiente foi encontrada, a estrutura do *website* foi investigada para que o QScraper pudesse ser desenvolvido. O presente capítulo descreve o processo de análise da página do Quora e implementação do aplicativo QScraper.

4.1 Política de coleta de dados do quora

As condições para o uso da plataforma quando são utilizadas ferramentas automatizadas para coleta de dados podem ser encontradas nos termos de serviço do Quora, no item **4d** referente aos "Usos Permitidos" podemos encontrar as exigências do Quora para prática de tal atividade:

> Se você opera uma ferramenta de buscas, web crawler, bot, scraping tool, ferramenta de data-mining, ferramenta de download em massa, wget, ou qualquer outra ferramenta de coleta e extração de dados, você pode acessar a Plataforma do Quora de acordo com as seguintes regras adicionais:
>
> 1. você deve usar um cabeçalho de agente de usuário descritivo;
> 2. você deve seguir robots.txt o tempo todo;
> 3. seu acesso não deve afetar negativamente qualquer aspecto do funcionamento da Plataforma do Quora; e
> 4. você deve deixar claro como podemos entrar em contato com você, na própria informação de agente de usuário ou em seu website, se você possui um.
>
> Você representa e garante que que não irá utilizar ferramentas automáticas como inteligência artificial ou aprendizado de máquina para (i) criar trabalhos

> derivados a partir de Nosso Conteúdo e Materiais; (*ii*) para criar qualquer serviço que seja um competidor da Plataforma do Quora; ou (*iii*) para qualquer outro fim comercial exceto quando expressamente permitido por estes Termos de Serviço ou com o consentimento por escrito do Quora.

É necessário então, criarmos um agente de usuário na ferramenta de *scraping* em cada requisição HTTP, contendo alguma forma de contato para que se necessário for a plataforma Quora consiga entrar em contato usuário. No que se diz respeito ao robot.txt, é uma página web que estabelece quais caminhos podem ou não ser acessados. Na Tabela 2 podemos verificar quais os usos permitidos para um user-agent genérico.

No presente trabalho foram respeitadas todas as condições impostas pela plataforma Quora para o uso de aplicativo de *scraping*. As requisições foram feitas com as seguintes URL's:

- https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery
- https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery
- ttps://www.quora.com/graphql/gql_para_POST?q=MultifeedQuery

Tabela 3: Permissões presentes no robot.txt para user-agents genéricos

PERMITIDO	**NÃO PERMITIDO**		
/$	/	/*/posts$	/digest/
/about$	/AJAX/	/*/posts/	/email_optout/
/about/	/@async	/*/questions	/qemail/
/challenges$	/*/@async	/*/related	/invite/
/press$	/log/	/*/reviews$	/widgets/content_iframe/
/login/	/*/log	/*/reviews/	/widgets/content_js/
/signup$	/*/about	/*/share	/_/
	/*/action	/*/top_questions	/*_POST$
	/*/activity	/*/topic-bio	/*_POST/
	/*/all_questions	/*/topic_bio	/webnode2/
	/*/all_posts$	/*/topics	/anonymous/
	/*/all_posts/	/*/comment	/q/*/admin_log
	/*/blogs$	/comment/	/q/*/stats
	/*/blogs/	/*/comments	/q/*/settings
	/*/followers	/*/all_comments	/q/*/queue
	/*/following	/*/answer_comments	/q/*/suggestions
	/*/link/	/*/mobile_collapsed	/q/*/submissions
	/*/manage	/*/mobile_expanded_voter_list	/q/*/quality
	/*/mentions	/home/global_feed	/q/*/earnings
	/*/merged	/search?q=	/profile/*/Rss
	/*/open_questions	/search/?q=	/topic/*/Rss

Vale ressaltar que o presente trtabalho não possui fins lucrativos e/ou comerciais, pois se trata de pesquisa com finalidade acadêmica. Além disso não se utiliza de inteligência artificial ou aprendizado de máquinam cumprindo rigorosamente as exigências feitas pela plataforma Quora.

4.2 Investigação de métodos para scrapping do quora

Uma das formas mais simples de realizar a coleta de informações de um *website* é através do uso de uma API (do inglês, *Application Programming Interface*) disponibilizada pelos desenvolvedores do próprio *site*. Um API é definida como um conjunto de protocolos para integrar uma aplicação.(33) Ela é a interface que permite que serviços se comuniquem entre si sem a necessidade de acesso ao código fonte. Dessa forma, desenvolvedores podem acessar funcionalidades e dados de forma controlada e encapsulada, já que a API pode disponibilizar recursos específicos e proteger outros. Diversas aplicações web disponibilizam APIs para uso da comunidade de desenvolvedores. Entre elas, podemos citar: Reddit(34), Twitter(35), Telegram(36), Facebook(37), Discord(38), Google maps(39) e muitas outras.

Especificamente para a plataforma Quora, não existe uma API oficial disponibilizada. Usuários tem postado no próprio fórum do Quora questionamentos sobre o desenvolvimento de uma API oficial sem sucesso.(40) Os questionamentos datam de 10 anos atrás e permanecem sendo feito até a presente data. Por outro lado, a comunidade de usuários do Quora desenvolveu algumas APIs não-oficiais ao longo desses anos. Após busca na literatura, as seguintes APIs foram encontradas: pyquora(41), quora-api(42), quora-scrapper(43), quoras(44), pyquora(45). Notavelmente, existem duas aplicações criadas independentemente com nome "pyquora". A primeira aplicação nomeada pyquora não foi atualizada nos últimos 3 anos. Já o novo pyquora foi criado há 2 anos e teve sua última atualização no ano passado. Enquanto isso, as outras APIs encontradas também não são atualizadas há 2 anos ou mais. A API mais antiga encontrada foi a Qnowledge(46), descontinuada após pedido do escritório de finanças do Quora.

Devido à inexistência de uma ferramenta confiável para extração de dados do Quora que ainda seja suportada, o presente projeto se propõe a desenvolver uma ferramente para *web scrapping* de dados do Quora sem o uso de uma API. O presente trabalho é a continuação da monografia dos alunos Leandro Barifouse de Souza e Rafael Mendes Matos(6) e se propõe a estender a referida ferramenta através da implementação de novas

funcionalidades.

4.2.1 Análise da página do quora

De acordo com o trabalho passado(6), requisições utilizando o método GET da biblioteca *Requests* retorna somente o HTML estático da página. Contudo, a página é populada por requisições AJAX feitas dinamicamente conforme o usuário navega pela página. Como os dados de interesse não estão disponíveis no HTML original do *site*, não é possível a extração desses dados através da extração do conteúdo estrutural do mesmo. Serão feitas requisições diretas ao servidor do Quora utilizando os mesmos cabeçalhos e payload que o sistema do próprio site utiliza para obtenção dos dados de interesse.

Para que a extração de dados através da requisição direta ao servidor do Quora possa ser feita, precisamos entender como as requisições são feitas internamente ao sistema. Para isso, as ferramentas de desenvolvimento do navegador foram utilizadas. As ferramentas de desenvolvimento estão disponíveis na maioria dos navegadores disponíveis atualmente (normalmente, são ativadas pela tecla F12). O navegador Firefox foi utilizado nesse trabalho.

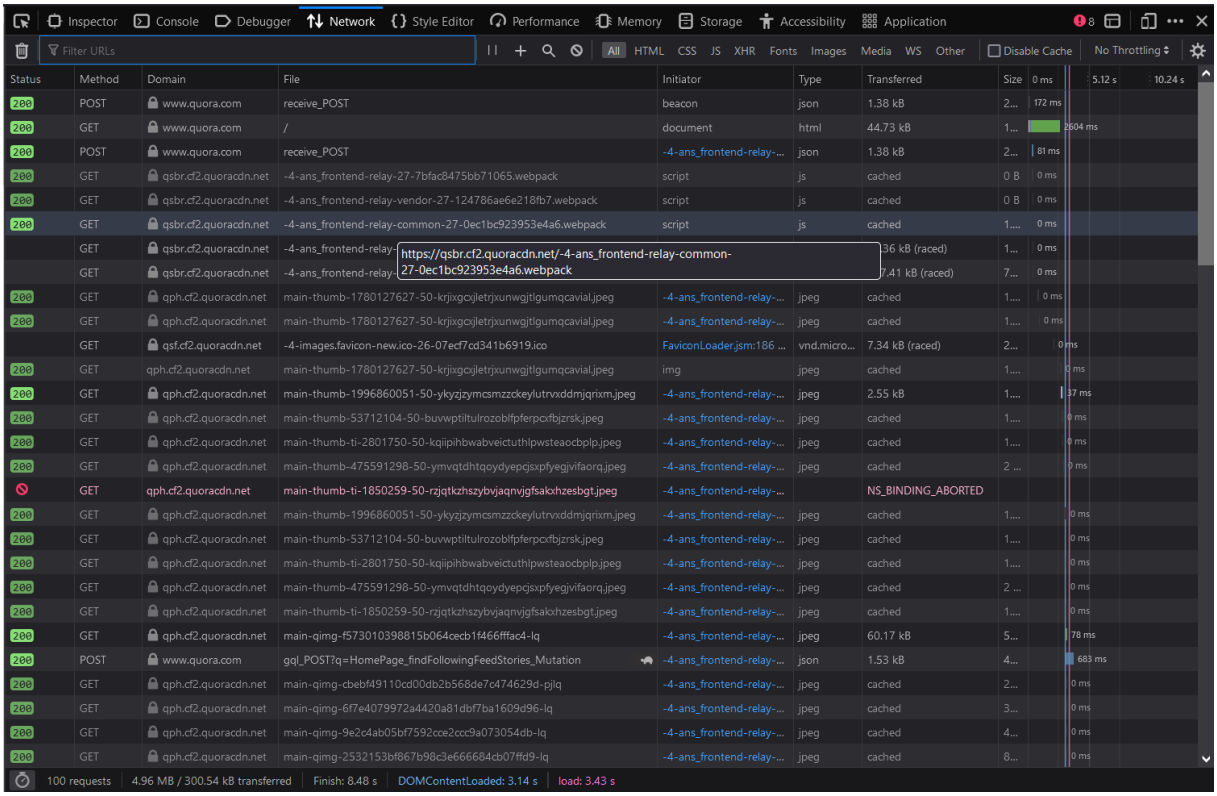

Figura 10: Lista de todo fluxo de rede realizado ao acessar a página do Quora.

Existem diversas ferramentas de desenvolvedor disponíveis no Firefox. O interesse

nesse trabalho foi entender como é o fluxo de rede entre o cliente utilizando a página do Quora e o servidor enviando os dados para popular a página. Portanto, utilizaremos a aba "Rede" da janela de desenvolvedor ("*Network*" no caso do navegador estar configurado em inglês). Na aba de redes, podemos ver todo o fluxo de informação que ocorre entre o cliente e o servidor, como ilustrado na Figura 10. Como não há o interesse em obter todo arquivo html, css, script, imagens e toda mídia do *site*, vamos filtrar as requisições para mostrar somente fluxo referentes às requisições XMLHttpRequest (XHR). A resposta dessas requisições são utilizadas para atualizar a página dinamicamente, sem atualizá-la.

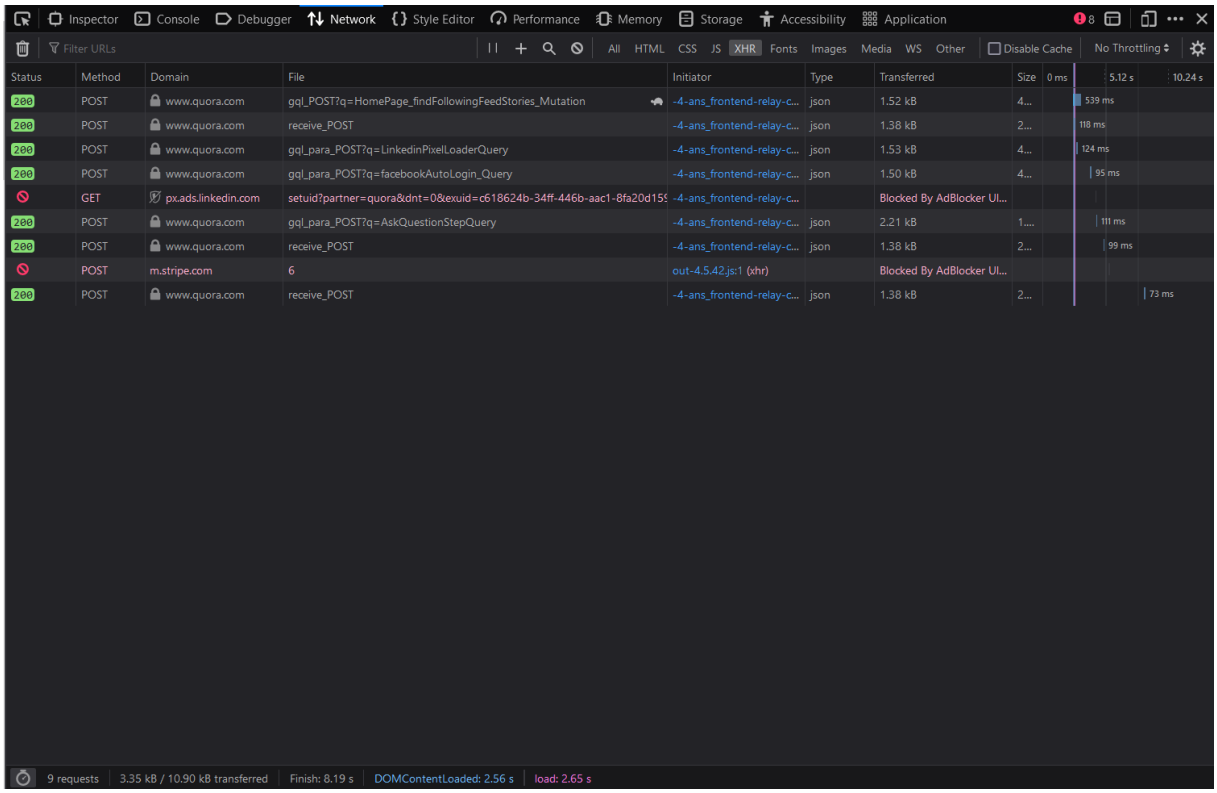

Figura 11: Lista do fluxo de rede filtrado para exibir somente objetos XHR. Essas são as requisições feitas no momento em que a página é carregada.

Ao carregar a página `https://www.quora.com/` no navegador e abrir a aba de ferramentas de desenvolvedor listando somente as requisições XHR da aba de redes, veremos uma lista parecida com a mostrada na Figura 11. Essa lista contem as requisições feitas para popular o carregamento da página exibida ao usuário inicialmente. Conforme descemos a barra de rolagem da página e nova informação é necessária para continuar a alimentação dinâmica da página, novas requisições são feitas. A Figura 12 mostra a lista atualizada após novos dados serem carregados na página.

De forma geral, dois tipos de requisição podem ser vistos. O URL relativo para onde a requisição é feita pode ser visto na coluna "file" da lista mostrada na Figura 12. A

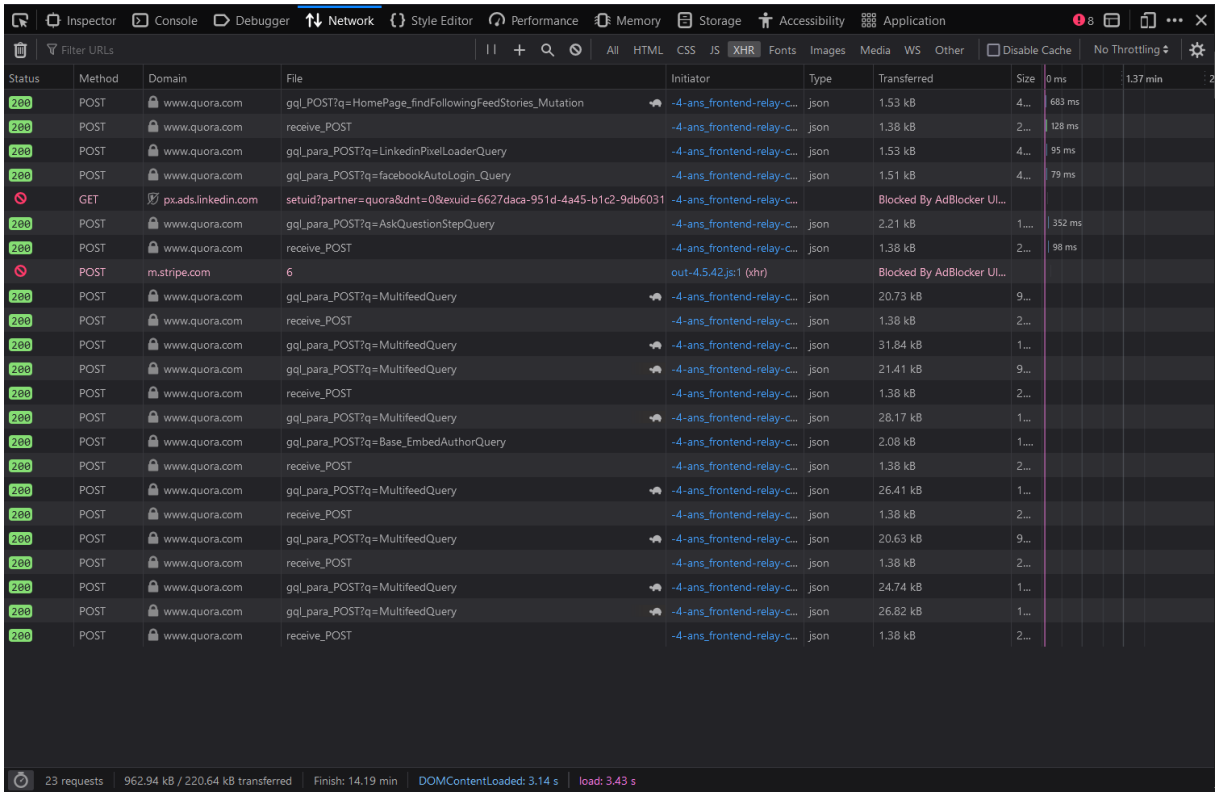

Figura 12: Lista do fluxo de rede filtrado para exibir somente objetos XHR. Novas requisições são feitas conforme descemos a barra de rolagem da página. Isto é, quando novos dados são necessários para popular a página e, consequentemente, novas requisições são feitas.

URL https://www.quora.com/ajax/receive_POST é o caminho para onde é enviada a requisição ajax enquanto a URL https://www.quora.com/graphql/gql_para_POST?q=<Query> é o caminho para onde a requisição ao banco é feita. Nesse endereço, a chave "<Query>" corresponde ao tipo de requisição que está sendo feita. Veremos que existem alguns tipos de querys utilizadas ao longo do *site* e cada uma dela retorna uma informação específica. Na Figura 12, por exemplo, encontra-se o endereço com *query* ?q=facebookAutoLogin_Query, responsável pelo retorno da informação de *login* do usuário.

Para obter os dados retornados pela busca, é preciso entender quais requisições são feitas quando uma busca é executada. Buscando por "hiv" e descendo a barra de rolagem para que novas requisições sejam feitas (ver Figura 13), a página de busca envia suas requisições para o endereço https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery.

Inspecionando os dados mostrados para essa requisição, é possível obter os cabeçalhos e *payloads* necessários para replicar as requisições. O cabeçalho é responsável por informar o servidor dados como o conteúdo, a origem, o destino, o usuário, idioma e tipo de *encoding*

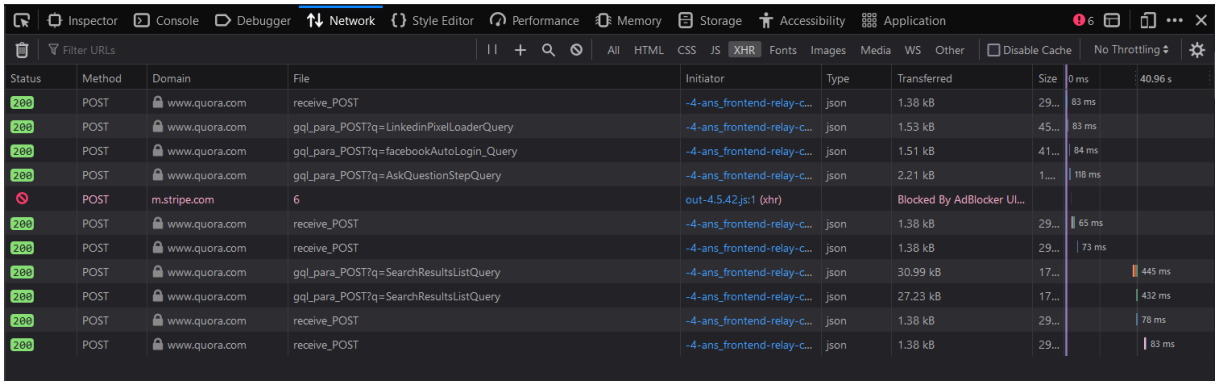

Figura 13: Lista de requisições XHR realizadas durante uma busca na página do quora

aceitáveis e diversas outras informações técnicas sobre a requisição. Já o *payload* contém a informação sobre os dados que estão sendo requisitados. O comportamento de cada parâmetro foi investigado manualmente alterando as opções de busca na interface *front-end* da página e conferindo as consequências tanto nos cabeçalhos quanto no *payload*.

Uma lista extensiva mostrando o significado e opções para os parâmetros mais importantes está disponível em um trabalho anterior(6).

4.2.2 Aplicativo QScrapper

A ferramenta cujo desenvolvimento foi continuado no presente trabalho se chama QScrapper. Ele é um programa desenvolvido utilizando o *framework* scrapy(47) para *web scrapping* do *website* do Quora.

4.2.2.1 Framework scrapy

O Scrapy é um *framework* desenvolvido para *web crawling* e *web scraping* para extração de dados estruturados de páginas.(47) A grande vantagem do Scrapy quando comparada a utilizar uma biblioteca comum de requisição HTTP é que o scrapy processa as requisições de forma assíncrona. Isto é, o fluxo de requisições pode ser paralelizado uma vez que novas requisições podem ser feitas mesmo que alguma requisição passada ainda não tenha retornado a resposta do servidor.

O funcionamento básico do Scrapy se baseia na instanciação de um *crawler*, chamado de *spider*, que será o responsável pelo gerenciamento das requisições.(48) Esse *crawler* deve herdar da classe scrapy.Spider e os métodos start_requests() e parse() devem ser implementados. O método start_requests() deve retornar um iterável contendo as requisições como objetos da classe scrapy.Requests. Esse iterável pode ser uma lista, mas o

uso de geradores python é mais eficiente e mais indicado). Quando o *crawler* é invocado, esse método é chamado e a requisição é feita. Quando a requisição é respondida, o método parse() é chamada como uma função *Callback*. O parse(), portanto, é o método responsável por tratar a resposta obtida, um objeto da classe scrapy.TextResponse que encapsula o conteúdo da resposta e outros métodos úteis para dar continuidade ao *web crawling*.

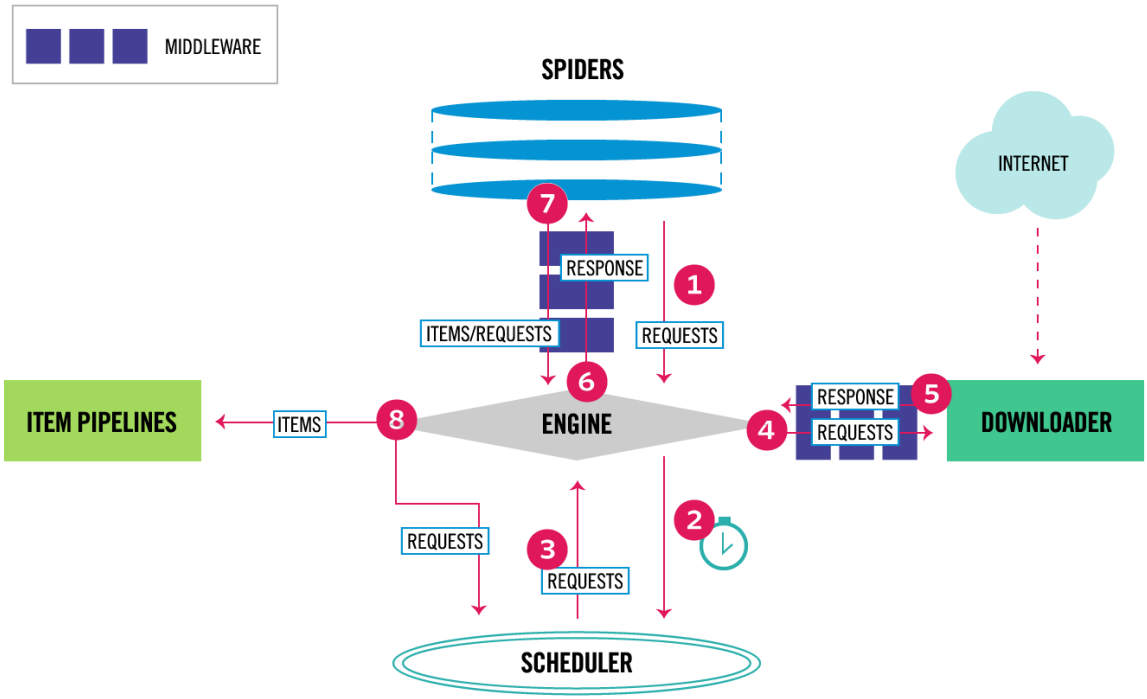

Figura 14: Ilustração da arquitetura do framework scrapy. Nesse esquema, as *Spiders* são os crawlers que o usuário deve implementar para utilizar o mecanismo do framework. O *Engine* é o programa central responsável por gerenciar e processar as requisições feitas pelas *Spiders*. Esquema retirado da documentação do scrapy(49).

O *crawler* envia cada uma dessas requisições para o mecanismo central do Scrapy que os gerencia através do *scheduler*, a parte do framework que agenda a efetivação das requisições, e do *downloader*, onde a requisição é efetivamente enviada ao servidor pedido (Ver Figura 14). Uma explicação mais aprofundada da arquitetura desse *framework* pode ser encontrada na sua documentação.(49)

4.2.2.2 Fluxo de informação

O funcionamento do QScraper é ilustrado pelo fluxograma mostrado na Figura 15. O processo se inicia pela leitura dos parâmetros e das palavras-chave referentes à busca.

Esses dados são utilizados para submeter a requisição ao servidor do quora. Dependendo do tipo de requisição que é pedida (*answers*, *topics* ou *posts*), um objeto da classe scrapy.Spider referente é criada. Essa aranha é responsável por gerenciar os headers e o payload para fazer a requisição. Ela também trata o retorno da requisição e salva os dados extraídos no banco de dados.

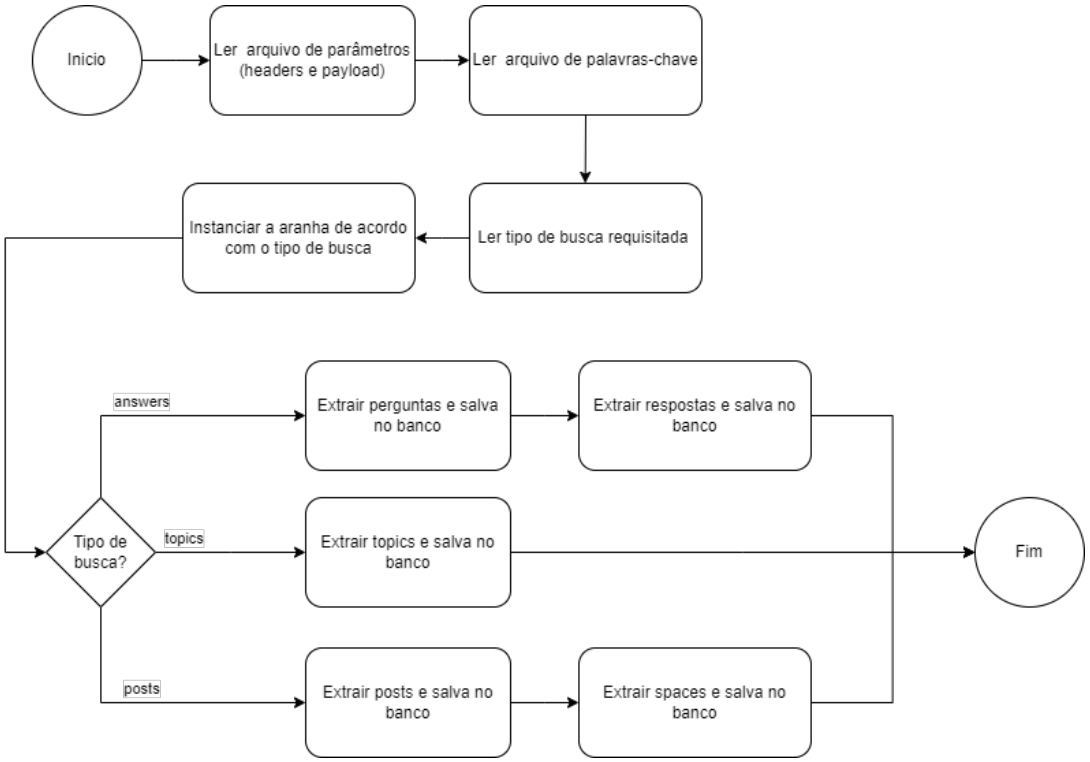

Figura 15: Fluxograma de execução do QScrapper.

Foi utilizado o banco MongoDB para salvar os dados extraídos. Uma vez que os dados são retornados no formato json, o uso de coleções (estrutura básica do Mongo) é facilitada.

4.2.2.3 Estrutura do banco MongoDB

O Mongo é um banco de dados noSQL orientado a documentos. Nesse paradigma, os dados são armazenados em uma estrutura de grafo. O MongoDB apresenta melhor performance e escalabilidade por ser uma estrutura mais simples. Entretanto, a consistência do banco é responsabilidade do desenvolvedor que o está implementando, já que não existem mecanismos para tratamento dos dados.

A arquitetura do banco implementado nesse trabalho é mostrado na Figura 16. Por não ser um banco SQL, não tem sentido pensar nos conceitos de chaves primarias e estrangeiras. Nesse caso, nos baseamos no id (uma variável chamada "_id") para identificação de

uma entrada específica em uma coleção e em algumas coleções suporte para descrevermos as relações que permitem a referência à atributos de outras coleções.

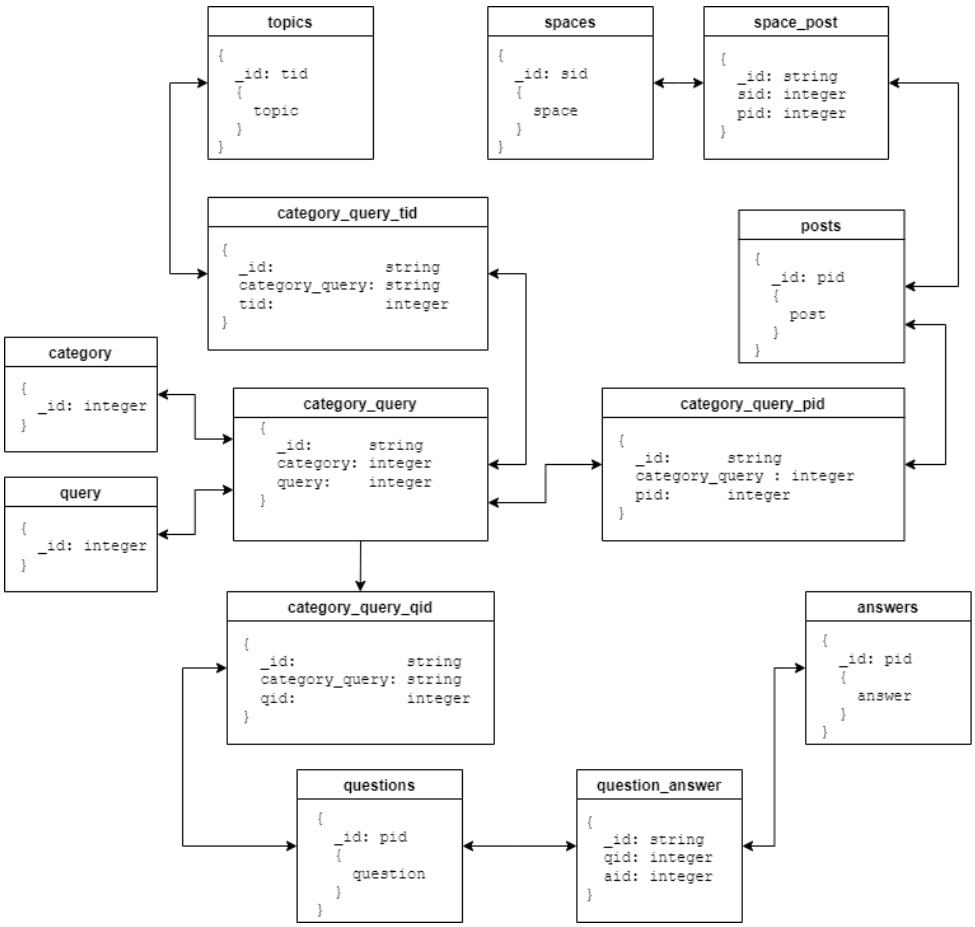

Figura 16: Diagrama do banco de dados orientado à documentos implementado no MongoDB.

Nessa modelagem, cada coleção tem um atributo "_id" identificador de cada elemento. As coleções questions, answers, topics, posts e spaces tem, respectivamente, atributos ilustrados como {questions}, {answers}, {topics}, {posts} e {spaces} que são referências para todos os dados resposta retornados no JSON pela requisição. Três tipos de coleções suportes foram criadas. Uma para agrupar *queries* em suas categorias, outra para relacionar cada resultado de busca com a *query* e categoria que foi utilizada para buscá-lo e uma última para agrupar os *posts* em relação aos *spaces* onde foram postados. A "category_query" é do primeiro tipo. Enquanto as coleções "category_query_qid", "category_query_tid" e "category_query_pid" relacionam, respectivamente, perguntas, topicos e posts com as queries. Finalmente, "space_post" agrupa os *posts* em relação aos *spaces*.

4.3 Caso de uso

Como caso de uso único do projeto, é descrito o processo em que o usuário realiza a extração de dados da plataforma Quora e salvamento desses dados em uma instância do MongoDB fornecida também pelo usuário. A utilização do aplicativo Qscrapper segue o fluxo do caso de uso mostrado na Tabela 4. O caso de uso tem como objetivo modelar o comportamento do sistema e ajudar a capturar os requistos necessários. É uma descrição das funçoes de alto nível e do escopo do sistema.

Tabela 4: Caso de uso do aplicativo QScrapper.

Nome:	Coleta de dados do Quora
Descrição:	Este caso de uso permite ao usuário coletar perguntas e respostas bem como informações das categorias Topics e Posts do servidor do Quora
Objetivo:	Extração de dados da plataforma Quora relacionado com os parâmetros fornecidos e salvamento no banco de dados
Atores:	Usuário
Pré-Condições:	Python instalado(De preferência versão 3.9.7) e pacotes instalados; Banco de dados MongoDB configurado; Arquivo JSON de parâmetros das requisições ao quora configurado; Arquivo das palavras-chave que serão utilizadas como termo de busca configurado;
Gatilho:	Execução do applicativo QScrapper
Fluxo Principal:	1 - Usuário indica o caminho do arquivo de parâmetros das requisições HTTP a serem feitas de acordo com qual sessão deseja realizar a raspagem *(topic, post, question, answer)*. 2 - Usuário indica o caminho do arquivo das palavras-chave para serem usadas como termo de busca. 3 - Usuário indica uma instância de um cliente do MongoDB que será usada para armazenamento dos dados coletados. 4 - Usuário indica o tipo de informação que deve ser raspada (*question*, *topic*, *post*) 5 - Sistema realiza as requisições para o servidor do Quora de acordo com os arquivos previamente configurados para a coleta das perguntas. 6 - Sistema salva os dados referente às requisições no banco de dados. 7 - Sistema realiza as requisições para o servidor do Quora de acordo com os arquivos previamente configurados para a coleta das respostas. 8 - Sistema salva os dados referente às requisições no banco de dados. 9 - Sistema encerra sua execução.

Continua na próxima página

– Continuação da Tabela 4

Fluxo Alternativo:	1.1 O caminho (path) do arquivo de parâmetros indicado não contém um arquivo. 1. O sistema lança exceção, informando que é necessário especificar corretamente o caminho do arquivo de parâmetros. 2. O sistema encerra sua execução. 2.1 O arquivo de termos de busca indicado contém alguma categoria sem termos de busca. 1. O sistema lança exceção, informando que a categoria em questão não possui termos de busca. 2. O sistema encerra sua execução. 3.1 O sistema não consegue conectar a instância da MongDB passada. 1. O sistema lança exceção, informando que há um erro com o banco. 2. O sistema encerra sua execução. 4.1 O tipo de dado a ser raspado pedido pelo usuário não existe ou não está implementado. 1. O sistema lança exceção, informando que o tipo de dado solicitado não existe e fornece uma lista com as opções disponíveis. 2. O sistema encerra sua execução. 5.1 O sistema não consegue formar uma requisição válida para coleta de respostas por configuração inadequada do arquivo de parâmetros. 1. O sistema lança exceção, informando que o arquivo de parâmetros deve ser revisto. 2. O sistema encerra sua execução.
Pós-Condições:	Dados referentes ao tipo de busca pedido pelo usuário salvos no banco de dados

– Fim da Tabela 4

4.4 Diagrama de classes

O aplicativo foi construido adotando o paradigma de orientação a objetos. Este projeto manteve o mesmo paradigma, tendo adicionado somente novas funcionalidades e novas classes para cumprir o objetivo proposto.

A classe QScrapeRunner é a responsável pela leitura dos arquivos dos parâmetros para envio das requisições e dos termos de busca. Além disso, a classe inicializa o crawler que utiliza as classes SearchSpider, AnswerSpider, TopicSpider e PostSpider para efetuar a coleta de dados no site do Quora. Basicamente as classes funcionam de forma semelhante, se diferenciando apenas pelos dados contidos nos arquivos json de parâmetros que contém

as espicificações das requisições ajax feitas para o servidor do Quora. (*post* e *spaces*, *topic* ou *questions* e *answers*) As demais classes utilizadas são provenientes de pacotes de terceiros, a saber: pymongo, scrapy e twister.

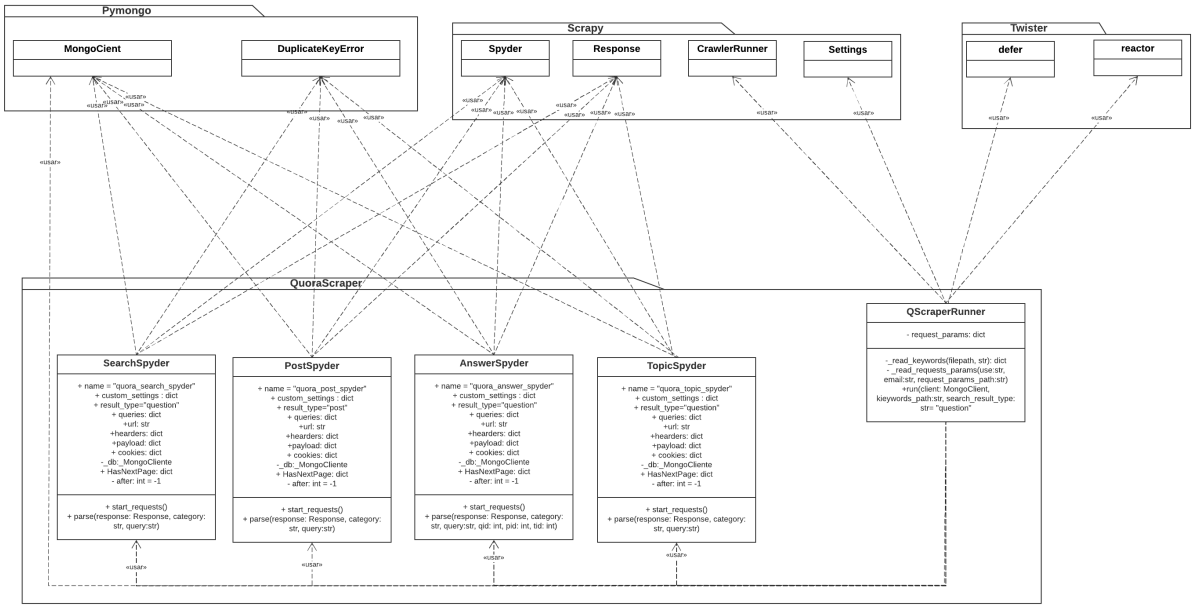

Figura 17: Diagrama de classes do aplicativo.

5 Resultados

O aplicativo desenvolvido foi utilizado para extração de dados. Após testes e validação, a execução do aplicativo se deu ao longo dos dias 9 e 16 de Novembro de 2022. Foi observado que, de forma imprevisível, o Quora encerrava a conexão antes de todas as requisições terem sido finalizadas. Nesses momentos, a execução do programa entrava em modo ocioso. Possivelmente, o *framework* Scrapy é preparado para lidar com esse problema e as requisições voltariam a ser feitas após um *time-out*. Essa possibilidade não foi investigada. Nesse estudo, a execução do programa foi manualmente interrompida assim que a ociosidade foi observada. Em seguida, o programa era reiniciado realizando a extração de outros elementos. Esse procedimento foi repetido até que fosse observado que novos dados não estavam mais sendo obtidos antes que o Quora encerrasse a conexão com o QScraper.

Sistema	Biblioteca	Versão
Windows (WSL)	scrapy	2.7.0
	pymongo	4.3.2
	Twisted	18.9.2
	tqdm	4.64.1
MacOS	scrapy	2.6.3
	pymongo	4.2.0
	Twisted	22.8.0
	tqdm	4.64.0

Tabela 5: Versões utilizadas para executar o QScraper e obter os dados apresentados no presente estudo divididas em categorias.

O QScraper foi testado em diferentes ambientes de desenvolvimento. Ele foi executado em MacOS, utilizando um interpretador Python 3.10.8, e em Windows 10 Pro 10.0.19044 utilizando Ubuntu 22.04.1 LTS através do (Windows Subsystem for Linux (WSL)), um interpretador Python 3.8.10 foi utilizado nesse ambiente. As versões das dependências utilizadas estão disponíveis na Tabela 5.

As palavras-chave para busca foram definidas de acordo com trabalho prévio(6). Termos foram definidos por especialista e salvos em formato JSON. A declaração de seleção

das palavras-chaves feita pelo especialista está disponível no Anexo A. Alguns dos termos inicialmente definidos retornaram informações generalistas e, portanto, foram removidos. A relação de palavras-chave utilizada no presente trabalho é apresentada no Algoritmo 5.1:

Algoritmo 5.1: Palavras-chave utilizadas nesse estudo.

```
{
    "fixed dose": [
        "truvada", "atripla", "epzicom", "complera",
        "cimduo", "combivir", "descovy", "Temixys",
        "Trizivir", "Delstrigo", "Odefsey", "SymfiLo",
        "Biktarvy", "Dovato", "Triumeq", "Juluca",
        "Genvoya", "Stribild", "Kaletra", "Kivexa",
        "Triomune", "Duovir", "Evotaz", "Prezcobix",
        "Rezolsta", "Dutrebiz", "Symfi", "Eviplera",
        "Symtuza", "Cabenuva"
        ]
    "not fixed dose": [
        "viread", "ftc", "\"ftc treatment\"", "3tc",
        "\"hiv treatment\"", "isentress", "reyataz", "norvir",
        "sustiva", "stocrin", "tivicay", "lamivudine",
        "epivir"
        ]
}
```

As palavras-chave são separadas em duas categorias: (*i*) "fixed dose" e (*ii*) "not fixed dose". Cada uma dessas categorias tem *queries* específicas.

O aplicativo QScrapper foi utilizado para extração de dados de perguntas e respostas, topics e posts e spaces do Quora utilizando as palavras-chave apresentadas. O número de elementos obtidos é apresentado nas Tabelas 6 (para a categoria "fixed dose") e 7 (para a categoria "not fixed dose").

Foi observado no trabalho anterior(6) que mesmo entre as palavras-chaves selecionadas, existem termos muito gerais. No presente trabalho, termos gerais também foram encontrados. O termo com maior número de dados extraídos, referente à palavra-chave "ftc", também retorna dado sobre diferentes assuntos como "Federal Trade Commission", "Fundamental Theorem of Calculus" e "First Tech Challenge" (uma competição de robó-

	fixed dose				
Query	**questions**	**answers**	**topics**	**posts**	**spaces**
atripla	18	10	0	11	10
biktarvy	60	28	1	6	5
cabenuva	18	4	1	2	1
cimduo	0	0	0	0	0
combivir	4	2	0	14	12
complera	1	0	1	5	4
delstrigo	2	0	0	1	1
descovy	34	15	1	9	7
dovato	11	1	1	2	2
duovir	4	1	0	4	4
dutrebiz	0	0	0	0	0
epzicom	1	1	1	5	4
eviplera	0	0	0	0	0
evotaz	0	0	0	3	3
genvoya	7	1	0	6	4
juluca	5	0	1	10	7
kaletra	7	5	1	27	20
kivexa	80	8	2	2	1
odefsey	6	5	2	6	4
prezcobix	1	0	1	7	5
rezolsta	0	0	0	4	2
stribild	6	4	1	7	5
symfi	0	0	0	0	0
symfilo	0	0	0	0	0
symtuza	3	1	0	9	5
temixys	0	0	0	1	1
triomune	0	0	2	1	1
triumeq	8	3	0	5	3
trizivir	0	0	0	6	6
truvada	193	103	1	28	25

Tabela 6: Relação quantitativa de elementos obtidos do Quora relacionados à categoria "fixed dose".

tica). Essa palavra-chave precisa ser melhor processada e, análise posterior, é necessária para remoção dos dados espúrios. Contudo, como o presente trabalho é focado na implementação do QScraper e extração dos dados, o tratamento dos dados foge do escopo deste trabalho.

O resumo da quantidade de elementos extraídos por categoria foi apresentado na Tabela 8.

not fixed dose					
Query	**questions**	**answers**	**topics**	**posts**	**spaces**
"ftc treatment"	0	0	0	0	0
3tc	22	15	1	4	4
epivir	0	0	0	0	0
ftc	1268	477	32	1098	796
hiv treatment	793	504	11	2413	1417
isentress	9	5	1	0	0
lamivudine	29	10	1	0	0
norvir	4	1	1	0	0
reyataz	2	1	0	0	0
stocrin	5	3	1	0	0
sustiva	2	0	0	0	0
tivicay	7	1	1	0	0
viread	7	4	1	2	2

Tabela 7: Relação quantitativa de elementos obtidos do Quora relacionados à categoria “not fixed dose”.

Query	**questions**	**answers**	**topics**	**posts**	**spaces**
Total					
fixed dose	469	192	17	181	142
not fixed dose	2148	1021	50	3517	2219

Tabela 8: Relação quantitativa de elementos obtidos do Quora separados por categoria.

6 Conclusão e trabalhos futuros

Os dados do Quora foram satisfatóriamente extraídos e salvos de forma estruturada no banco da dados. Assim, o tratamento dos dados e posterior análise são possibilitadas e facilitadas. Além disso, o programa é estruturado de forma encapsulada. O que facilita a manutenção e extensão do código.

Foi possível perceber que o perfil de busca não foi muito alterado desde o último semestre. Na categoria "fixed dose", a palavra-chave com maior quantidade de dados extraídos foi a "truvada". Nós conseguimos obter 193 perguntas e 103 respostas para essa palavra-chave. Impressionantemente, utilizando a palavra-chave "truvada" o trabalho anterior extraiu 184 perguntas e 229 respostas. Isso mostra que mais perguntas foram feitas nos últimos meses. Entretanto, menos respostas foram obtidas no presente estudo. Isso se dá ao fato do servidor do Quora encerrar a conexão impedindo que todas as respostas fossem extraídas. Esse mesmo padrão é observado na categoria "not fixed dose". O termo com maior retorno é o "hiv treatment", mais generalista. O presente estudo obteve 793 perguntas e 504 respostas, enquanto o trabalho anterior raspou 703 perguntas e 1469 respostas.

Foi visto que em determinados momentos, a conexão com o servidor do Quora era interrompida. A versão atual do QScraper não está preparada para lidar com isso muito bem. A implementação de algum tipo de ordenação da extração para que a extração possa ser retomada à partir da última página já extraída é uma das perspectivas futuras desse trabalho. Outra possibilidade é esperar o tempo de *time-out* para que as requisições voltem a ser realizadas. Essa preocupação com a obtenção dos dados é crucial uma vez que foi observado que existem muitas respostas que não foram obtidas. Esse cenário pode estar ocorrendo também nos campos de *topic*, *post* e *space*.

De forma geral, pôde-se mostrar que a plataforma do Quora possui uma grande quantidade de dados referentes à pesquisa farmacêuticas com HIV. O QScraper pode ser utilizado principalmente de três formas: (i) Para obter relações de perguntas e respostas para

entender as dúvidas existentes na comunidade e para obter informação sobre as perguntas mais comuns, (*ii*) para extrair número de seguidores em tópicos e espaços, o que pode dar informação sobre a popularidade e o interessa da comunidade em um determinado tema e (*ii*) para obter postagens (*posts*) que podem dar entendimento sobre as opiniões presentes na comunidade estudada.

REFERÊNCIAS

1 CAMPOS, Linair Maria et al. Dados abertos interligados e o espaço do profissional de informação: Uma aplicação no domınio da enfermagem., 2012.

2 LAMBA, Manika; MADHUSUDHAN, Margam. **Text Mining for Information Professionals**. [S. l.]: Springer International Publishing. DOI: 10.1007/978-3-030-85085-2.

3 INTERNET access. [S. l.: s. n.]. Disponível em https://en.wikipedia.org/wiki/Internet_access, acessado em 20/11/2022.

4 MOST popular social networks worldwide as of January 2022, ranked by number of monthly active users (in millions). [S. l.: s. n.]. Disponível em https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/, acessado em 20/11/2022.

5 DE SOUZA, Gilson Engelkes Salgado; OLIVEIRA CHRISTOFF, Adriana de. AVALIAÇÃO DA ADESÃO AO TRATAMENTO COM ANTIRRETROVIRAIS EM PACIENTES HIV-POSITIVO ATENDIDOS NO CENTRO DE CONTROLE DE AGRAVOS DE PINHAIS. **Cadernos da Escola de Saúde**, v. 22, n. 1, 2022.

6 SOUZA, L. B e Matos R. M. de. **QSCRAPER: WEB SCRAPING DE PERGUNTAS E RESPOSTAS DO QUORA COM MENÇÕES A MEDICAMENTOS PARA HIV**. Niterói: [s. n.], 2022. P. 68.

7 KALEJAIYE, Gabriel Bayomi Tinoco. Deep learning para processamento de linguagem natural: extração de significado em redes sociais, 2017.

8 GLOBAL Internet usage. [S. l.: s. n.]. Disponível em https://en.wikipedia.org/wiki/Global_Internet_usage, acessado em 20/11/2022.

9 BAPTISTA, Sofia Galvão; CUNHA, Murilo Bastos da. Estudo de usuários: visão global dos métodos de coleta de dados. pt. **Perspect. ciênc. inf.**, Escola de Ciência da Informação da UFMG, v. 12, n. 2, p. 168–184, ago. 2007. ISSN 1413-9936, 1981-5344. DOI: 10.1590/S1413-99362007000200011.

10 OLIVEIRA; MARTINS. Coleta de dados para agregação de repositórios digitais: Entidades vinculadas à Secretaria Especial de Cultura do Brasil. **Advanced Notes in Information**, pub.colnes.org, 2022.

11 ZHAO, Bo. Web Scraping. In: SCHINTLER, Laurie A; MCNEELY, Connie L (Ed.). **Encyclopedia of Big Data**. Cham: Springer International Publishing, 2017. P. 1–3. ISBN 9783319320014. DOI: 10.1007/978-3-319-32001-4_483-1.

12 DIOUF, Rabiyatou et al. Web Scraping: State-of-the-Art and Areas of Application. In: 2019 IEEE International Conference on Big Data (Big Data). [S. l.: s. n.], dez. 2019. P. 6040–6042. DOI: 10.1109/BigData47090.2019.9005594.

13 KROTOV, Vlad; SILVA, Leiser. Legality and Ethics of Web Scraping. In.

14 PEDROSO; FERREIRA; SILVA et al. Coleta de dados para pesquisa quantitativa online na pandemia da COVID-19: relato de experiência. **Rev. Esc. Enferm. USP**, periodicos.ufsm.br, 2022. ISSN 0080-6234.

15 MOLDWIN, Asher; DEMNER-FUSHMAN, Dina; GOODWIN, Travis R. Empirical Findings on the Role of Structured Data, Unstructured Data, and their Combination for Automatic Clinical Phenotyping. en. **AMIA Jt Summits Transl Sci Proc**, v. 2021, p. 445–454, mai. 2021. ISSN 2153-4063.

16 EÇA, Teresa Almeida de; FIGUEIREDO, António Dias de. **NetAprendizagem: a Internet na educação**. [S. l.: s. n.], 1998.

17 FOROUZAN, Behrouz A; FEGAN, Sophia Chung. **Protocolo TCP/IP-3**. [S. l.]: AMGH Editora, 2009.

18 CARVALHO, Marta Pinto de. **Integração da Internet nas aulas de Educação Visual e Tecnológica**. 2008. Tese (Doutorado).

19 VAZ, Welton Rodrigues. A Evolução da Internet 1.0 a 3.0. **Montes Belos**, 2015.

20 PRIMO, Alex. O aspecto relacional das interações na Web 2.0. In: E-COMPÓS. [S. l.: s. n.], 2007. v. 9.

21 FACCIONI FILHO, Mauro. Internet das coisas. **Unisul Virtual**, 2016.

22 BALADELI, Ana Paula Domingos. Hipertexto e multiletramento: revisitando conceitos. **Revista e-scrita: Revista do Curso de Letras da UNIABEU**, v. 2, n. 4, p. 1–11, 2011.

23 FLATSCHART, Fábio. **HTML 5-Embarque Imediato**. [S. l.]: Brasport, 2011.

24 SOTTO, Eder Carlos Salazar; LUCINIO, Gleydson. ADAPTANDO TÉCNICAS DE TESTE DE SOFTWARE TRADICIONAIS PARA APLICAÇÕES WEB. **Revista Interface Tecnológica**, v. 13, n. 1, p. 7–22, 2016.

25 AMZA, Cristiana et al. Specification and implementation of dynamic web site benchmarks. In: CONF. 5TH Workshop on Workload Characterization. [S. l.: s. n.], 2002.

26 SILVA, Maurıcio Samy. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. [S. l.]: Novatec Editora, 2010.

27 MAH, B A. An empirical model of HTTP network traffic. In: PROCEEDINGS of INFOCOM '97. [S. l.: s. n.], abr. 1997. v. 2, 592–600 vol.2. DOI: 10.1109/INFCOM.1997.644510.

28 HUNT, Ray. Transmission Control Protocol/Internet Protocol (TCP/IP). In: BIDGOLI, Hossein (Ed.). **Encyclopedia of Information Systems**. New York: Elsevier, 2003. P. 489–510. ISBN 978-0-12-227240-0. DOI: https://doi.org/10.1016/B0-12-227240-4/00187-8. Disponível em: <https://www.sciencedirect.com/science/article/pii/B0122272404001878>.

29 KUROSE, James F; ROSS, Keith W. **Computer Networking: A Top-down Approach**. [S. l.]: Addison-Wesley, 2010. ISBN 9780136079675.

30 IDRIS, Aizal Yusrina; BAMOALLEM, Razan; MOHAMAD HATTA, Mohamad Harith Azfar. Web scraping and regression analysis based on machine learning for COVID-19 with rapid software platform. **Mathematical Sciences and Informatics Journal**, UiTM Press, Universiti Teknologi MARA, v. 3, n. 1, p. 75–85, mai. 2022. ISSN 2735-0703. DOI: 10.24191/mij.v3i1.18278.

31 NIU et al. Web Scraping Tool For Newspapers And Images Data Using Jsonify. **J. Appl. Sci. South. Afr.**, jase.tku.edu.tw, 2022. ISSN 1019-7788.

32 MITCHELL, Ryan. **Web Scraping with Python: Collecting More Data from the Modern Web**. [S. l.]: "O'Reilly Media, Inc.", mar. 2018. ISBN 9781491985526.

33 WHAT is an API? [S. l.: s. n.], 2022. Disponível em https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces, acessado em 07/11/2022.

34 API Documentation. [S. l.: s. n.]. Disponível em https://www.reddit.com/dev/api/, acessado em 07/11/2022.

35 TWITTER API. [S. l.: s. n.]. Disponível em https://developer.twitter.com/en/docs/twitter-api, acessado em 07/11/2022.

36 TELEGRAM APIs. [S. l.: s. n.]. Disponível em https://core.telegram.org/, acessado em 07/11/2022.

37 FACEBOOK Developer Docs. [S. l.: s. n.]. Disponível em https://developers.facebook.com/docs/, acessado em 07/11/2022.

38 DISCORD Developer Portal. [S. l.: s. n.]. Disponível em https://discord.com/developers/docs/intro/, acessado em 07/11/2022.

39 GOOGLE Maps Plataform. [S. l.: s. n.]. Disponível em https://developers.google.com/maps, acessado em 07/11/2022.

40 WHAT is the status of the full Quora API. [S. l.: s. n.]. Disponível em https://www.quora.com/What-is-the-status-of-the-full-Quora-API?q=%22Quora%20API%22, acessado em 07/11/2022.

41 PYQUORA. [S. l.: s. n.]. Disponível em https://github.com/csu/pyquora, acessado em 07/11/2022.

42 QUORA-API. [S. l.: s. n.]. Disponível em https://github.com/csu/quora-api, acessado em 07/11/2022.

43 QUORA-SCRAPPER. [S. l.: s. n.]. Disponível em https://github.com/banyous/quora-scraper, acessado em 07/11/2022.

44 DAS, Dipto; SEMAAN, Bryan. quoras: A Python API for Quora Data Collection to Increase Multi-Language Social Science Research. In: CONFERENCE Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing. Virtual Event, USA: Association for Computing Machinery, out. 2020. (CSCW '20 Companion), p. 251–256. ISBN 9781450380591. DOI: 10.1145/3406865.3418333.

45 PYQUORA. [S. l.: s. n.]. Disponível em https://github.com/TheShubhendra/pyquora, acessado em 07/11/2022.

46 QNOWLEDGE. [S. l.: s. n.]. Disponível em https://github.com/karan/Qnowledge, acessado em 07/11/2022.

47 SCRAPY. [S. l.: s. n.]. Disponível em https://scrapy.org/, acessado em 07/11/2022.

48 SCRAPY 2.7 documentation. [S. l.: s. n.]. Disponível em https://docs.scrapy.org/en/latest/, acessado em 07/11/2022.

49 SCRAPY: Architecture overview. [S. l.: s. n.]. Disponível em https://docs.scrapy.org/en/latest/topics/architecture.html, acessado em 07/11/2022.

ANEXO A

Declaração de seleção das palavras-chave para extração de dados de medicamentos contra HIV. As palavras-chave foram propostas pela pesquisadora Luciana Ferreira Mattos Colli, professora da faculdade de farmácia da Universidade Federal do Rio de Janeiro. Por serem escolhidas por uma especialista na área, essas palavras-chave mostram uma aplicação de real interesse para pesquisa no tema, motivando o trabalho e evidenciando uma necessidade real experienciada pelos cientistas farmacêuticos.

Rio de Janeiro, 28 de novembro de 2021.

DECLARAÇÃO DE TAGS PARA SELEÇÃO DE SUBMISSÕES E COMENTÁRIOS DE HIV

Eu, Luciana Ferreira Mattos Colli, Professora Mestre da Faculdade de Farmácia da Universidade Federal do Rio de Janeiro (UFRJ), no Centro de Ciências da Saúde (CCS), venho por meio deste atestar definições de tags (PReP HIV, Prep treatment, tripletherapy, triple therapy, anti, anti HIV, anti treatment, HIVinfection, drug, drug HIV, NormalizingHIVChallenge, Normalizing HIV Challenge, livingwithaids, living aids, HIVtreatment, HIV treatment, pep HIV, pep treatment, pepforhiv, pepforearlyhiv, pep for early hiv, pepindelhi, pep delhi, peptreatment, peptreatmentinmalviyanagar, peptreatment malviyanagar, pep malviyanagar, pepcenterforhiv, pep center for hiv, pephivcenter, pep center hiv, pepforealryexposer, pep real exposer, pepandprep, pep prep, Truvada, Atripla, Epzicom, Complera, Cimduo, Combivir, Descovy, Temixys, Trizivir, Delstrigo, Odefsey, SymfiLo, Biktarvy, Dovato, Triumeq, Juluca, Genvoya, Stribild, Kaletra, Kivexa, Triomune, Duovir, Evotaz, Prezcobix, Rezolsta, Dutrebiz, Symfi, Eviplera, Symtuza, Cabenuva, Viread, FTC, FTC treatment, 3TC hiv, 3TC treatment, Isentress, Reyataz, Norvir, Sustiva, Stocrin, Tivicay, Lamivudine, Epivir) para teste do software script extração de submissões e comentários.

Por ser verdade, firmo o presente para que surte seus efeitos legais.

Rio de Janeiro, 28 de novembro de 2021.

_____

Prof. Luciana Ferreira Mattos Colli

Departamento de Fármacos e Medicamentos - DEFARMED

Faculdade de Farmácia – FF – UFRJ