

UNIVERSIDADE FEDERAL FLUMINENSE
LEANDRO BARIFOUSE DE SOUZA
RAFAEL MENDES MATOS

**QSCRAPER: WEB SCRAPING DE PERGUNTAS E RESPOSTAS DO
QUORA COM MENÇÕES A MEDICAMENTOS PARA HIV**

Niterói
2022

**LEANDRO BARIFOUSE DE SOUZA
RAFAEL MENDES MATOS**

**QSCRAPER: WEB SCRAPING DE PERGUNTAS E RESPOSTAS DO
QUORA COM MENÇÕES A 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(a):
ALTOBELLI DE BRITO MANTUAN**

**NITERÓI
2022**

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

S719q Souza, Leandro Barifouse de
Qscraper : Web scraping de perguntas e respostas do Quora com menções a medicamentos para HIV / Leandro Barifouse de Souza, Rafael Mendes Matos ; Altobelli de Brito Mantuan, orientador. Niterói, 2022.
68 f. : il.

Trabalho de Conclusão de Curso (Graduação em Tecnologia de Sistemas de Computação)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2022.

1. Mineração de dados (computação). 2. Python (Linguagem de programação de computador). 3. Framework (Programa de computador). 4. Produção intelectual. I. Matos, Rafael Mendes. II. Mantuan, Altobelli de Brito, orientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

LEANDRO BARIFOUSE DE SOUZA
RAFAEL MENDES MATOS

**QSCRAPER: WEB SCRAPING DE PERGUNTAS E RESPOSTAS DO
QUORA COM MENÇÕES A 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, 21 de junho de 2022.

Banca Examinadora:

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

Prof. RICARDO DE OLIVEIRA NOLASCO, Esp. – Avaliador
UFF - Universidade Federal Fluminense

Dedicamos este trabalho aos envolvidos direta e indiretamente na construção do conhecimento que culminou nesse trabalho.

“O caos é uma ordem por decifrar”.

José Saramago

RESUMO

Neste trabalho, a partir da demanda de um grupo de pesquisadores farmacêuticos, foi construída uma ferramenta computacional desenvolvida em *Python* capaz de obter dados sobre medicamentos usados em pessoas que contraíram HIV, que posteriormente serão analisados pelos demandantes. Para tanto, utilizou-se a técnica do *web scraping* - através da biblioteca *Scrapy* - para coleta de perguntas e respostas provenientes da rede social Quora e a aplicação desenvolvida os armazena em um banco de dados MongoDB. Após os testes realizados com 43 palavras-chave indicadas pelos pesquisadores farmacêuticos, foi possível coletar 2260 perguntas e 2997 respostas à essas perguntas, demonstrando o potencial da aplicação resultante deste trabalho.

Palavras-chaves: Quora, hiv, web scraping, python, scrapy, mineração de dados.

ABSTRACT

In this work, based on the demand of a group of pharmaceutical researchers, a computational tool developed in Python was constructed capable of obtaining data on drugs used in people who contracted HIV, which will later be analyzed by the plaintiffs. For this purpose, the web scraping technique was used - through the Scrapy library - to collect questions and answers from the quora social network and the application developed stores them in a MongoDB database. After the tests performed with 43 keywords indicated by the pharmaceutical researchers, it was possible to collect 2260 questions and 2997 answers to these questions, demonstrating the potential of the application resulting from this work.

Key words: Quora, hiv, web scraping, python, scrapy, data mining.

LISTA DE ILUSTRAÇÕES

Figura 1 - Tipos de dados.....	21
Figura 2 - Elemento HTML	24
Figura 3 - Página simples: (a) código HTML, (b) Renderização de página simples no navegador Microsoft Edge	25
Figura 4 - Representação do acesso ao conteúdo de um site.	27
Figura 5 - Monitor de rede do Firefox exibindo requisições da página de busca (grifo nosso)	35
Figura 6 - Filtros de busca pelo tipo de dado (a), pelo autor (b) e pelo período (c)...37	
Figura 7 - Página de busca com estruturas destacadas.	40
Figura 8 - Recorte de pergunta e resposta.....	41
Figura 9 - Monitor de rede do Firefox exibindo requisições da página de pergunta (grifo nosso)	42
Figura 10 - Página de pergunta com (a) anúncio (grifo nosso), (b) questões relacionadas (grifo nosso), (c) respostas relacionadas e (d) acesso limitado ao Quora+	45
Figura 11 - Página de pergunta com questões colapsadas (grifo nosso).....	46
Figura 12 - Fluxograma do aplicativo	48
Figura 13 - Diagrama Entidade-Relacionamento do banco de dados do aplicativo ..	48
Figura 14 - Diagrama de classes do aplicativo.....	52

LISTA DE TABELAS

Tabela 1 - Usuários ativos por redes sociais em 2021	14
Tabela 2 - Transcrição do robots.txt para user-agent genérico.	31
Tabela 3 - Filtros possíveis para formação do <i>payload</i> de uma busca.	38
Tabela 4 - Chaves do JSON de resposta à requisição da página de busca.	39
Tabela 5 - Chaves do JSON de resposta à requisição da página de pergunta.	44
Tabela 6 - Definição dos campos das coleções no banco de dados.	49
Tabela 7 - Descrição do Caso de Uso “Coletar Perguntas e Respostas”.	50
Tabela 8 - Hardware do ambiente de testes.	54
Tabela 9 – Software do ambiente de testes – ambiente de software geral.	54
Tabela 10 – Resultado quantitativo dos testes – categoria “ <i>fixed dose</i> ”.	56
Tabela 11 – Resultado quantitativo dos testes – categoria “ <i>not fixed dose</i> ”.	57
Tabela 12 - Resultado da coleta de dados por categorias	58

LISTA DE ABREVIATURAS E SIGLAS

AJAX	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
bot	<i>robot</i>
BTM	<i>Biterm Topic Model</i>
CSS	<i>Cascading Style Sheets</i>
DNS	<i>Domain Name System</i>
HIV	<i>Human Immunodeficiency Virus</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IP	<i>Internet Protocol</i>
JSON	<i>JavaScript Object Notation</i>
LDA	<i>Latent Dirichlet Allocation</i>
OECD	<i>Organization for Economic Co-operation and Development</i>
PLSA	<i>Probabilistic Latent Semantic Analysis</i>
SQL	<i>Structured Query Language</i>
TCP	<i>Transmission Control Protocol</i>
URL	<i>Uniform Resource Locator</i>
XHR	<i>XMLHttpRequest</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

RESUMO	7
ABSTRACT (opcional)	8
LISTA DE ILUSTRAÇÕES	9
LISTA DE TABELAS	10
LISTA DE ABREVIATURAS E SIGLAS	11
1 INTRODUÇÃO	14
2 TRABALHOS RELACIONADOS	16
2.1 DISCUSSÃO	19
3 FUNDAMENTAÇÃO TEÓRICA	20
3.1 COLETA DE DADOS	20
3.2 TIPOS DE DADOS	21
3.3 ORGANIZAÇÃO E ARMAZENAMENTO	22
3.4 PÁGINAS <i>WEB</i>	23
3.4.1 HTML	24
3.4.2 PÁGINAS <i>WEB</i> DINÂMICAS	26
3.5 NAVEGADORES <i>WEB</i>	26
3.6 <i>WEB SCRAPING</i>	27
4 DESENVOLVIMENTO	29
4.1 QUORA	29
4.1.1 POLÍTICA DO QUORA QUANTO À COLETA DE DADOS	30
4.1.2 ESTRUTURA DE INFORMAÇÕES NO QUORA	32
4.2 BUSCA POR UMA API	33
4.3 INVESTIGAÇÃO DE MÉTODOS PARA <i>SCRAPING</i> DO QUORA	33
4.3.1 ANÁLISE DA PÁGINA DE BUSCA DO QUORA	34
4.3.2 ANÁLISE DA PÁGINA DE PERGUNTA E RESPOSTAS ASSOCIADAS	40
4.3.3 <i>FRAMEWORK</i> SCRAPY	46
4.4 FUNCIONAMENTO DO APLICATIVO QSCRAPER	47
4.5 CASO DE USO	50
4.6 DIAGRAMA DE CLASSES	52

5	TESTES	54
	CONCLUSÕES E TRABALHOS FUTUROS	59
	REFERÊNCIAS BIBLIOGRÁFICAS	60
	ANEXOS	66

1 INTRODUÇÃO

O número de usuários de redes sociais vem aumentando consideravelmente desde 2015, chegando a 4,48 bilhões em 2021, o que corresponde a 56,8% da população global à época. O detalhamento do número de usuários ativos nas principais redes sociais em 2021 pode ser visualizado na Tabela 1 [1], informação esta que demonstra a extensão dessas redes e sua capacidade de conectar as pessoas.

Tabela 1 - Usuários ativos por redes sociais em 2021

Rede social	Usuários ativos (Milhões)
Facebook	2.853
YouTube	2.291
WhatsApp	2.000
Instagram	1.386
FB Messenger	1.300
WeChat	1.242
TikTok	732
QQ	606
Douyin	600
Telegram	550
Sina Weibo	530
Snapchat	514
Kuaishou	481
Pinterest	478
Reddit	430
Twitter	397
Quora	300

Fonte: adaptado de [1]

As redes sociais, enquanto meios de comunicação, tornaram-se mais abrangentes, pois seus usuários debatem múltiplos assuntos, merecendo destaque o uso para acesso a informações sobre cuidados com a saúde [2].

A participação de profissionais da saúde nas mídias digitais ainda carece de discussões mais profundas, pois não existem diretrizes provenientes dos órgãos competentes, nem treinamento específico para sua interação com o público geral. Porém, é notório que a população aumentou a busca de informações sobre questões de cuidado com a saúde nas redes sociais [2].

Conquanto o Quora apresente o menor número de usuários ativos na Tabela 1, este ainda é bastante expressivo, cabendo salientar que, quando comparado ao Twitter – considerado como o meio mais popular para a comunicação sobre cuidados com a saúde [2] –, o Quora está apenas uma colocação abaixo, de modo que o interesse de analisar os dados dessa plataforma é evidente.

Entretanto, em razão da vasta quantidade de dados que podem ser consultados no Quora, faz-se necessária uma ferramenta capaz de coletá-los de forma célere e confiável. Assim, o escopo do presente trabalho é a criação de *software* capaz de obter perguntas e respostas no Quora, organizando-as em banco de dados. A validação da aplicação criada foi realizada a partir do uso de palavras-chave relacionadas a HIV e fármacos utilizados no seu tratamento, especificadas por pesquisadores da área de Farmácia e fornecidas pelo orientador deste trabalho.

Após o alcance do objetivo do presente trabalho, a análise dos dados será de responsabilidade dos pesquisadores farmacêuticos, a qual permitirá a obtenção de informações acerca da aceitação do medicamento, de efeitos colaterais de seus componentes, bem como a definição de tendência do medicamento para viabilizar um estudo mais aprofundado, dentre outras.

O código gerado encontra-se disponível no GitHub, pelo *link* https://github.com/altobellibm/CEDERJ_2022_LEANDRO_RAFAEL.

2 TRABALHOS RELACIONADOS

O uso de *web scraping* para coleta e análise de dados da internet tem sido intenso nos últimos anos, em especial em razão da utilização massiva de plataformas online de interação social. Nesse sentido, faz-se necessário analisar trabalhos anteriores que tenham recorrido a *web scraping*, de forma a verificar suas vantagens e desvantagens e, por conseguinte, sua adequação ao propósito do presente estudo.

Inicialmente, temos que ALASMARI e ZHOU [3] conduziram pesquisas para investigar como os consumidores de informações de saúde que possuem multimorbidade - isto é, que são acometidos por múltiplas doenças simultaneamente - se comportam em plataformas de perguntas e respostas online. Para atingir esse objetivo, os autores decidiram coletar dados da plataforma Quora [4], definindo, através de consultas iterativas, os tópicos de interesse para a coleta de dados, focando naqueles relacionados a doenças renais, pois pacientes com essas enfermidades tendem a apresentar multimorbidade. Em seguida, considerando que a Quora não oferece conjunto de dados públicos nem uma Interface de Programação de Aplicações (*Application Programming Interface* – API) oficial, realizaram a coleta de dados usando um *web crawler* customizado, utilizando-se de diversas bibliotecas *Python*, como *Scrapy* [5], *Selenium* [6] e *Pandas* [7]. A estratégia foi coletar, inicialmente, os perfis de usuários que postaram perguntas ou respostas relacionadas aos tópicos de interesse e, posteriormente, dados detalhados sobre a interação de cada um desses perfis, como as perguntas e respostas postadas, os tópicos criados e as relações de seguimento (de *posts* ou outros perfis). Após análise dos dados coletados, concluiu-se que há significativas diferenças na interação de consumidores que estão interessados em diversas doenças em relação àqueles que estão focados em apenas uma doença. Em geral, os interessados em multimorbidade são mais ativos na postagem de perguntas e seguem mais usuários e tópicos, o que pode estar associado ao fato de as informações que buscam serem mais complexas, de forma a exigir uma diversidade maior de fontes de informação.

Concluiu-se, ainda, que perguntas desse grupo recebem mais respostas, mesmo possuindo menos visualizações, o que sugeriria que a elevada complexidade das perguntas ocasionaria maior atenção de usuários e a necessidade de um número maior de respostas para serem satisfeitas. Constatou-se também que as respostas dos interessados em multimorbidade não são tão bem avaliadas quanto aquelas ofertadas pelos interessados em uma única doença, o que poderia ser explicado pelo fato de que as respostas do primeiro grupo são mais complexas e personalizadas, enquanto as do segundo grupo são mais profundas e simples em relação às respectivas doenças, de modo a atender a um grupo maior de interessados. Apoiados no estudo, os autores apontam que as plataformas de perguntas e respostas e de sistemas de saúde em geral precisam ser adaptadas para melhorar a experiência de busca de informações de saúde para pacientes com multimorbidade, o que, considerando que estes são participantes ativos nesses ambientes, beneficiaria não apenas os próprios pacientes, mas também a população em geral.

Por sua vez, CHEN et al. [8] realizaram estudos com o objetivo de identificar as diferenças entre as necessidades de consumidores de informações relativas à hipertensão em plataformas de perguntas e respostas e em plataformas de comunidade online. Os autores afirmam que compreender essas diferenças possibilita melhorar a forma de fornecer informações concernentes à hipertensão e, por conseguinte, aumentar o nível de conscientização acerca dos diversos aspectos dessa doença, permitindo maior prevenção e controle na população em geral. No que tange à metodologia de pesquisa, foi utilizado, em 15 de janeiro de 2020, um *web crawler* desenvolvido em *Python* para obter as perguntas e suas respectivas respostas na plataforma Quora, bem como para obter as *postagens* da comunidade online denominada MedHelp [9], sendo que todos os itens possuíam data de criação entre janeiro de 2010 e janeiro de 2019. Após a coleta de dados, aqueles que não se referiram à hipertensão foram removidos manualmente e, em seguida, foi aplicado um processo de limpeza de três etapas para processamento de linguagem natural: formalização de caixa do texto, remoção de pontuação e lematização. No que tange à modelagem de tópicos, foi utilizado o *Biterm Topic Model* (BTM), que possui um desempenho melhor para textos curtos como os encontrados nas plataformas utilizadas, quando comparado com métodos tradicionais, como o *Probabilistic Latent Semantic Analysis* (PLSA) e o *Latent Dirichlet Allocation* (LDA). Os autores,

baseados em análise probabilística, determinaram que 10 seria o número de tópicos ótimo para o estudo, sendo que, em cada um, seriam listadas as 10 palavras mais comuns, de forma a permitir uma comparação entre os resultados obtidos em relação à plataforma MedHelp e à plataforma Quora. A comparação permitiu descobrir similaridades, isto é, verificou-se que, em ambas as plataformas, os usuários buscam informações relacionadas a controle de hipertensão, dietas recomendadas, medicação, aferição de pressão sanguínea, e doenças relacionadas. Por outro lado, constatou-se diferença também, consubstanciada no fato de que usuários do MedHelp discutiam mais sobre patologia, farmacologia e saúde mental relacionadas à hipertensão do que usuários do Quora. Os autores entendem que estes achados podem contribuir para que os desenvolvedores de plataformas desses tipos consigam organizar melhor a informação para seus usuários, por exemplo, utilizando *tags* associadas aos tópicos descritos no estudo. Além disso, os pesquisadores concluíram que o estudo também apresenta grande valia para qualquer consumidor de informações relativas à hipertensão, uma vez que os achados lhe permitem definir o tipo de plataforma que possui maior probabilidade de fornecer as informações que busca, como no caso de um usuário que esteja interessado em farmacologia associada à hipertensão e que saberá, a partir do estudo, que a melhor plataforma seria a de comunidade online, e não a de perguntas e respostas.

MUPPIDI et al. [10] desenvolveram um *framework* para identificar perguntas com intenções semelhantes em sites de perguntas e respostas, ou seja, se existem duplicatas. O processo utilizado é modelado como coleta de dados, pré-processamento, concatenação de sinônimos, extração de características, classificação, predição de nome de classe e validação. Para resolver os desafios de processamento de linguagem natural, foram utilizadas técnicas como *tokenization*, *stemming*, *lemmatization* e *fuzzy string matching*.

Por fim, XIE et al. [11] coletaram dados estruturados sobre criptomoedas, através de conhecidas bibliotecas em *Python* no intuito de treinar um chatbot capaz de responder questionamentos sobre este tema. A coleta de dados foi realizada através de requisições de protocolo HTTP com o auxílio da biblioteca *Requests*. A biblioteca *Selenium* foi utilizada para automatização de acesso através de navegadores *web*. Para extrair os dados do padrão HTML foram utilizados métodos da biblioteca *Beautiful Soup* [12]. O processo de coleta de dados foi dividido em

duas etapas, a primeira onde o código busca perguntas com palavras chaves e a segunda para coletar o título das perguntas e as respostas.

2.1 DISCUSSÃO

Os trabalhos supramencionados demonstram que *web scraping* é uma poderosa ferramenta para permitir a coleta de grande volume de dados a partir de plataformas online nas quais usuários interagem através de perguntas e respostas. Então, as técnicas descritas nos trabalhos relacionados se mostram potenciais ferramentas para a coleta de informações farmacêuticas através de uma plataforma de perguntas e respostas como o Quora, conforme o objetivo deste trabalho.

3 FUNDAMENTAÇÃO TEÓRICA

A Internet, em 2021, atingiu a marca de 4,66 bilhões de usuários, o que significa dizer que é utilizada por mais da metade da população mundial [13]. Estima-se que no Brasil, em 2020, 81% da população com dez anos ou mais eram usuários da Internet [14]. Nesse sentido, é patente que uma massa colossal de dados trafega pela Internet todos os dias.

O presente capítulo dedica-se à compreensão de como dados são coletados, classificados, organizados e armazenados. Além disso, aborda especificamente o funcionamento de páginas *web*, navegadores *web* e *web scraping*.

3.1 COLETA DE DADOS

A coleta de dados, em regra, é a tarefa de levantamento e reunião dos dados necessários à análise da matéria objeto do estudo que se pretende empreender. Essa tarefa pode ser realizada de diversas maneiras, isoladamente ou de forma combinada, como por pesquisa bibliográfica, pesquisa experimental (de laboratório ou de campo), estudo de casos, entre outras [15].

Além de diversas maneiras de executar o levantamento, é possível que existam também múltiplas fontes de informações, sendo certo que a literatura nos fornece uma classificação nesse particular [16]. As fontes primárias são aquelas que possuem informações novas, como descrições de ideias originais ou narrativa de fatos inéditos, como teses e dissertações, periódicos, projetos etc. Por sua vez, as fontes secundárias são aquelas que fazem referência a alguma fonte primária, isto é, aquelas que são arrumadas de tal forma que permitam encontrar informações expostas por fontes primárias, como dicionários e enciclopédias. Por fim, as fontes terciárias são aquelas que apenas indicam onde encontrar as fontes primárias e

secundárias, tal qual ocorre com bibliotecas e centros de informação, diretórios e mecanismos de busca.

Cabe observar que, em pesquisas que possuem como foco o *web scraping*, a tarefa de obtenção de dados é o próprio objeto do estudo, isto é, a análise recai sobre a própria forma de coletar informações - no caso, aquelas existentes em páginas da *web* -, razão pela qual essa tarefa ganha ainda maior relevo.

3.2 TIPOS DE DADOS

De acordo com BAKER [17], os tipos de dados podem ser classificados como qualitativos ou quantitativos. Os dados quantitativos são aqueles que medem uma grandeza e ainda podem ser subdivididos como discretos ou contínuos. Dados qualitativos são aqueles que classificam algo e podem ser subdivididos como nominais ou ordinais. A Figura 1 ilustra a classificação descrita.



Figura 1 - Tipos de dados

Dado discreto pertence a um conjunto de valores e não alcança maior precisão. Exemplos de dados discretos são os valores que podemos tirar em um lançamento de um dado de seis faces, um número que pertence ao conjunto {1, 2, 3, 4, 5, 6} ou o número de alunos em uma escola. É impossível tirar um número 3,5 em um lançamento do dado, bem como não é possível contar 100,5 pessoas em uma escola, o que demonstra a limitação da precisão deste tipo de dados.

Dado contínuo é aquele que pode possuir tanta precisão quanto a que o instrumento que realiza a medição é capaz de dar. Um exemplo é o tempo. Em um

relógio de sol, é possível descrever com alguma precisão a hora do dia, porém, com um relógio analógico, é possível medir na precisão de segundos, uma precisão 3.600 vezes maior. O que define a continuidade no caso do tempo, é que com medidores mais precisos será possível alcançar medidas mais exatas.

Os dados qualitativos representam uma classificação, como, por exemplo, a classe social à qual o indivíduo faz parte, cor da pele, gênero musical, dentre outras possíveis categorias. Esses dados podem ser nominais, hipótese em que não se pode ter uma ordem de grandeza do que é maior ou menor, ou ordinais, que é o caso oposto. Um exemplo de utilização deste tipo de dado seria a classificação dos alunos de uma escola em classes socioeconômicas para determinar quais alunos poderão ter acesso a programas sociais de apoio aos estudos, como distribuição gratuita de livros didáticos.

3.3 ORGANIZAÇÃO E ARMAZENAMENTO

A coleta de dados será realizada através de uma fonte de informações que deverá conter os dados de interesse. Esses dados podem estar armazenados de forma estruturada, semiestruturada ou não estruturada.

De acordo com trabalho desenvolvido pela OECD [18], os dados não estruturados são os mais comuns e exigem o maior potencial para as técnicas de análise de dados para coleta de informações. Este tipo de dado é o que se apresenta sem a existência de um modelo para organizá-los, ou seja, os dados contêm muita informação dentro de um contexto, como por exemplo e-mails, imagens e vídeos.

Dados estruturados são os que estão modelados de alguma forma, podem ser modelos explícitos, como por exemplo em um banco de dados SQL ou implícitos, como na estrutura de uma página da *web* onde é possível tornar o modelo explícito com relativa facilidade.

Por último, os dados semiestruturados não possuem modelos explícitos, mas estão ligados através de elementos semânticos, como *tags*, a um modelo estruturado. Por exemplo, arquivos JSON e XML.

3.4 PÁGINAS WEB

Na sociedade contemporânea, a utilização de páginas *web* é algo corriqueiro e através delas pode-se ler notícias, utilizar redes sociais para entrar em contato com outras pessoas, consultar o melhor trajeto entre uma localidade e outra, efetuar compras e fazer pesquisas das mais variadas.

Inicialmente, cabe ressaltar que a Internet, em seus primórdios, era uma rede capaz de interligar computadores distantes entre si (inclusive em países diferentes), sendo utilizada principalmente para comunicação por e-mail e newsgroups, em especial no meio universitário [19]. Contudo, foi a criação da chamada *World Wide Web* (*web*) que impulsionou a expansão da Internet para torná-la o que é hoje, isto é, uma rede verdadeiramente mundial e de amplo uso pela sociedade [20].

A *web* pode ser definida como um sistema através do qual clientes podem requisitar objetos, dentre os quais as páginas *web*, a servidores através da Internet. Nesse sistema, os componentes principais são (i) *Hypertext Transfer Protocol* (HTTP); (ii) *Uniform Resource Locator* (URL) e; (iii) *Hypertext Markup Language* (HTML) [21].

O HTTP é o protocolo que define a estrutura de mensagens que serão trocadas entre cliente e servidor *web* para transferência dos dados relativos ao objeto *web* que se pretende acessar [20].

Por sua vez, o URL é o endereço utilizado para localizar o objeto *web*. Então, o que ocorre é que “Cada URL tem dois componentes: o nome de hospedeiro (*hostname*) do servidor que abriga o objeto e o nome do caminho do objeto. Por exemplo, no URL <http://www.someSchool.edu/someDepartment/picture.gif>, <http://www.someSchool.edu> é o nome de hospedeiro e */someDepartment/picture.gif* é o nome do caminho.” [20].

Por fim, HTML é uma linguagem de marcação de hipertexto utilizada para criar o documento que define a estrutura de uma página *web*. Cabe salientar que a noção de hipertexto está atrelada à capacidade de um texto conter referências a outros textos, os quais podem ser acessados de imediato, através de *hiperlinks* [22]. No que tange ao conceito de linguagem de marcação, pode-se afirmar que se trata de uma linguagem que se utiliza de marcações para individualizar elementos em um

texto ou conjunto de dados [23], sendo que, no caso da HTML, essas marcações são denominadas *tags*.

3.4.1 HTML

As páginas *web*, como dito acima, são estruturadas em um documento em linguagem HTML, que utiliza as chamadas *tags* para individualizar/marcar os elementos que compõem tal estrutura, tais como títulos, parágrafos, imagens, hyperlinks (referência a outras partes desse documento ou a outro documento) etc. Nessa linguagem, cada elemento, em regra, é definido através de uma *tag* de abertura, seguida do conteúdo e, ao final, de uma *tag* de fechamento.

A *tag* de abertura é especificada através do uso de sinais de chevron [24], que englobam o nome da *tag*, ou seja, é composta de símbolo '<', nome da *tag* e símbolo '>'. A *tag* de fechamento possui composição similar, com a diferença de que, imediatamente após o símbolo '<', deve constar o símbolo '/'. A Figura 2 demonstra a composição básica de um elemento HTML:

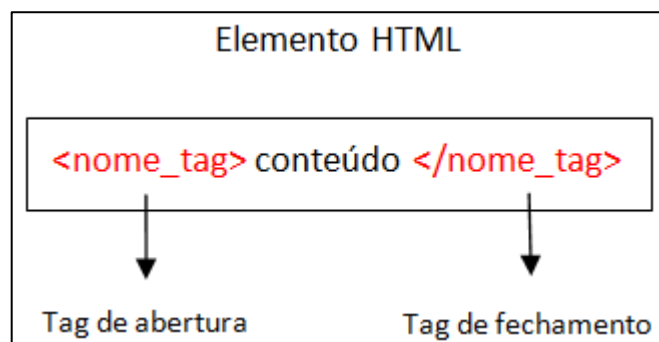


Figura 2 - Elemento HTML

Observe-se que alguns elementos não possuem conteúdo e, portanto, não necessitam de *tag* de fechamento, de modo que são definidos exclusivamente pela *tag* de abertura. Como exemplo, podemos citar o elemento que define uma quebra de linha, cuja composição é apenas: `
`.

Também é relevante notar que os elementos podem ser aninhados, de forma a descrever que um elemento pertence à estrutura definida por outro. Nesse caso, o elemento interior necessariamente possui toda sua definição (*tag* de abertura, conteúdo e *tag* de fechamento) englobada pela definição do elemento

exterior, não sendo possível haver sobreposições. Na Figura 3, vemos o código HTML de uma página simples, em que são aplicados os conceitos até agora expostos, bem como a sua respectiva renderização em um navegador *web*:

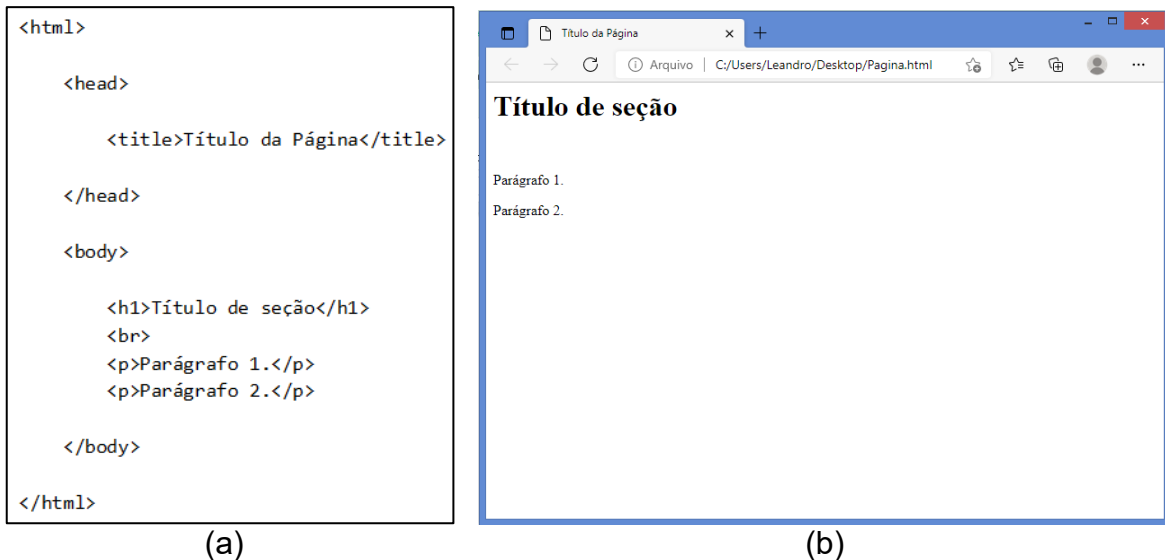


Figura 3 - Página simples: (a) código HTML, (b) Renderização de página simples no navegador Microsoft Edge

Embora a linguagem HTML seja suficiente para estruturar o conteúdo de uma página *web*, o atual estágio de evolução da Internet exige muito mais que isso, sendo necessário prover estilização e comportamento.

A estilização de páginas *web* é feita através de *Cascading Style Sheets* (CSS), que é uma linguagem utilizada para especificar como documentos (HTML, dentre outros) são apresentados aos usuários, permitindo a definição de *layouts*, cores, fontes e até mesmo animações, além de viabilizar a adaptação do documento a diferentes dispositivos (computadores, celulares etc.) [25].

Por sua vez, é possível adicionar comportamento às páginas *web* através de Javascript, uma linguagem de programação que fornece meios para a criação de funcionalidades complexas que tornam uma página *web* dinâmica e interativa. Dentre os comportamentos que são possíveis de serem obtidos através do uso de Javascript, um merece especial atenção quando o assunto é *web scraping*, a saber, a capacidade de manipular o próprio HTML, criando, removendo ou alterando elementos da página [26].

3.4.2 PÁGINAS WEB DINÂMICAS

Na *web*, existem páginas cujo conteúdo está integralmente definido no documento HTML, de modo que, para alterá-lo, faz-se necessário modificar o próprio código HTML da página, caso em que são denominadas páginas estáticas. Por outro lado, existem páginas cujo conteúdo é carregado ou renderizado de acordo com determinadas condições ou circunstâncias, como um evento disparado pelo usuário ou por um temporizador, hipótese em que são chamadas de páginas *web* dinâmicas [26].

Essas páginas dinâmicas contam com a utilização de linguagem de programação para lhes conferir essa característica, sendo extremamente comum atualmente o uso da linguagem Javascript no desenvolvimento *web*, através da qual é possível a manipulação dos elementos de uma página.

São várias as formas utilizadas para obtenção dessa característica com o uso do Javascript. Porém, há uma técnica específica que se fez necessário conhecer para cumprir o objetivo do presente trabalho, chamada *Asynchronous JavaScript and XML* - AJAX, que viabiliza a realização de requisições assíncronas para obtenção de dados do servidor em diversos formatos (e não apenas XML, como o nome faz parecer), utilizando-se o objeto existente nos navegadores chamado XMLHttpRequest [27]. Essa particularidade da assincronia significa que não há necessidade de recarregar a página por completo, sendo possível atualizar apenas partes dela [28].

3.5 NAVEGADORES WEB

Mozilla [29] define os navegadores *web* (*browsers*) como um tradutor, que recupera a informação contida em outras partes da *web* através do *Hypertext Transfer Protocol* (HTTP) que define como textos, figuras e vídeos serão apresentados ao usuário.

No que tange à forma como um navegador funciona, o primeiro passo é descobrir onde os ativos dessa página estão localizados [30]. Se o site nunca foi

visualizado, uma requisição DNS (*Domain Name System*) deverá ser realizada através de um DNS *lookup*, que busca em um servidor de nomes e retorna endereço IP dos ativos do site.

Após se conhecer o endereço IP (*internet protocol*), o segundo passo é realizar uma conexão TCP (*transmission control protocol*) via um TCP *three-way handshake*, mecanismo pelo qual duas entidades (navegador e servidor neste caso) negociam parâmetros para realizar uma conexão de rede através de um *socket* antes de realizar a transmissão de dados.

Após o estabelecimento da conexão HTML, o navegador envia uma requisição HTTP GET inicial. Após o servidor receber a requisição, ele enviará os cabeçalhos e conteúdo HTML. Esse fluxo é representado na Figura 4.



Figura 4 - Representação do acesso ao conteúdo de um site.

Fonte: adaptado de [30].

3.6 WEB SCRAPING

Mitchell (2019) [31] define *web scraping* como a coleta de dados por qualquer meio que não seja um programa interagindo com uma API ou um ser humano usando um navegador web. O *web scraping* engloba várias técnicas como análise de dados e processamento de linguagem natural.

Normalmente o *web scraping* é realizado através de um programa:

1. automatiza a consulta um servidor web;
2. requisita dados (através de HTML e outros arquivos que compõem as páginas web);
3. realiza o *parse* dos dados.

Isto posto, o presente trabalho pretende utilizar técnicas de *web scraping* para acessar dados disponíveis na plataforma Quora. Com essas técnicas é possível acessar milhares de páginas, coletando grande volume de dados em pouco tempo.

Mitchell (2019) define que o *web scraping* não é a utilização de APIs, pois a API é “projetada para fornecer um *stream* conveniente de dados bem formatados de um computador para o outro”. Quando se coleta dados da *web*, nem sempre existe uma API ou a API existente pode não mapear as informações que se busca. Dessa forma, o *web scraping* é uma poderosa técnica onde, se os dados podem ser visualizados em um navegador, normalmente eles podem ser acessados através de um script Python.

4 DESENVOLVIMENTO

Este capítulo abordará questões relativas ao desenvolvimento do aplicativo *QScraper* que se propõe a coletar automaticamente conteúdo do Quora.

Com esse propósito, descreve-se todas as etapas de desenvolvimento que se desdobra desde o entendimento da relevância das informações presentes no ambiente Quora, passando pela busca sem sucesso por uma API que auxiliasse o trabalho e finalizando com a explicação da técnica utilizada para realizar a coleta de dados. Dentro da parte técnica se destaca o entendimento das estruturas existentes no Quora e a forma é realizada a comunicação com base de dados.

4.1 QUORA

O Quora é uma página *web* que cumpre o papel de rede social destinada ao compartilhamento de conhecimento entre todas as pessoas, nos mais variados assuntos e campos do saber. Pretende conectar pessoas para viabilizar a troca de ideias, percepções e informações em escala global, o que, em última análise, acarreta o desenvolvimento do conhecimento humano. A definição de sua missão expõe isso claramente:

A missão do Quora é ampliar e compartilhar o conhecimento do mundo. Grande parte do conhecimento que seria valioso para muitas pessoas ainda está disponível somente para poucos - presa dentro das mentes das pessoas ou acessível para poucos. Nós queremos conectar aqueles que detêm o conhecimento com os que precisam dele, reunindo pessoas com diferentes perspectivas para que elas possam se entender melhor, permitindo que compartilhem seu conhecimento para o benefício de todos [32].

Essa rede social de compartilhamento de conhecimento, de acordo com dados de 2018, contava com 300 milhões de usuários ativos por mês [33], o que torna o Quora uma relevante fonte para a coleta de dados e informações relativas a medicamentos utilizados no tratamento de HIV, sendo este o objetivo do presente trabalho.

4.1.1 POLÍTICA DO QUORA QUANTO À COLETA DE DADOS

Os termos de serviço do Quora, no item **4.d.** referente aos “Usos Permitidos”, tratam de regras específicas para uso da plataforma quando são utilizadas ferramentas automatizadas para coleta de dados:

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: i) você deve usar um cabeçalho de agente de usuário descritivo; ii) você deve seguir *robots.txt* o tempo todo; iii) seu acesso não deve afetar negativamente qualquer aspecto do funcionamento da Plataforma do Quora; e iv) 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 não irá utilizar ferramentas automáticas como inteligência artificial ou aprendizado de máquina i) para 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. [34]

O agente de usuário define-se como “uma cadeia de caracteres característica que permite servidores e pares de rede identificar a aplicação, sistema operacional, fornecedor, e/ou versão do agente de usuário requisitante” [35]. Nesse sentido, o que se impõe é que o usuário da ferramenta de *scraping*, em cada requisição HTTP, identifique-se e estabeleça canais para eventuais contatos, a fim de que o Quora saiba quem a realizou e possa, caso entenda necessário, comunicar-se com ele.

Por sua vez, no que se refere ao *robots.txt*, trata-se de arquivo que fica na raiz de uma página *web* e prevê quais caminhos de arquivo podem ou não ser acessados [36], de acordo com o protocolo de exclusão de robôs [37]. Nesse sentido, a ferramenta QScraper deverá observar os caminhos permitidos no *robots.txt* do Quora [38], sendo certo que, para agentes de usuário em geral, esse arquivo estabelece permissões conforme a Tabela 2.

Tabela 2 - Transcrição do robots.txt para user-agent genérico.

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

Como será detalhado nos capítulos 4.3.1 e 4.3.2, para realizar o *scraping* de perguntas e respostas, as requisições são realizadas com as seguintes URL's, respectivamente:

- https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery;
- [https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery.](https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery)

Observando a Tabela 2 e realizando a interpretação das correspondências dos endereços URL [39], não há proibição expressa a caminhos dos tipos “/graphql/” ou “/*_POST?q=”. Saliente-se que as proibições em relação a “/*_POST\$” e “/*_POST/” não são aplicáveis às URL's acima, uma vez que aquela diz respeito a URL's que terminam no termo “POST” e esta se refere a URL's de quaisquer itens em diretórios que terminam no termo “POST”.

De toda sorte, é importante observar que a ferramenta desenvolvida no presente trabalho não possui fins lucrativos e/ou comerciais, tratando-se de pesquisa com finalidade acadêmica. Além disso, também não se utiliza de inteligência artificial ou aprendizado de máquina, estando em conformidade com os termos de serviço neste particular. O uso da ferramenta por terceiros deverá obedecer aos termos de serviço do Quora.

4.1.2 ESTRUTURA DE INFORMAÇÕES NO QUORA

No presente tópico será explicada a forma como as informações são apresentadas e estruturadas no ambiente Quora, pois este entendimento permite estabelecer a maneira mais eficaz de coletar os dados desejados.

O Quora determina a disposição do conteúdo compartilhado utilizando-se das seguintes estruturas: perfis (*profiles*), tópicos (*topics*), perguntas (*questions*), respostas (*answers*), espaços (*spaces*) e postagens (*posts*).

Perfis (*profiles*) estão relacionados aos usuários do Quora, sendo que cada usuário possui um perfil, através do qual pode compartilhar informações sobre si, como nome, localização, formação acadêmica, emprego, foto, entre outras.

Por sua vez, os tópicos (*topics*) são utilizados para categorizar perguntas e postagens, com o intuito de descrever o assunto a que se referem. Usuários podem seguir tópicos ou colocar tópicos em mudo, respectivamente recebendo ou deixando de receber conteúdo correlato, bem como podem indicar perícia ou interesse em determinado tópico, de modo a mostrar a outros usuários que suas respostas possuem maior qualificação [40].

No que tange às perguntas (*questions*) e respostas (*answers*), estas formam o núcleo do funcionamento do Quora, incentivando-se os usuários a formular perguntas sempre que as possuírem e a enviar respostas sempre que puderem. Ademais, os usuários podem votar nas melhores respostas, de forma a aumentar sua relevância e ajudar outros usuários a encontrá-las [41].

Os espaços (*spaces*) podem ser criados por usuários com assuntos de seus interesses e esses espaços podem ser mantidos de forma individual ou com um grupo de colaboradores [42]. Além disso, colaboradores de um espaço podem realizar postagens (*posts*) para adicionar conteúdo aos espaços [43].

Este trabalho terá como foco as estruturas de perguntas e respostas para a coleta de dados, ficando as demais como proposta para pesquisas futuras.

4.2 BUSCA POR UMA API

De acordo com IBM Cloud Education (2020) [44], uma *Application Programming Interface* (API) é uma aplicação que permite que desenvolvedores acessem dados e funcionalidades de aplicativos de empresas por meio de uma interface documentada, sem precisar saber como a API em si foi implementada.

Inicialmente, foi realizada uma busca por uma API oficial do Quora que permitisse a otimização da coleta de informações, mas não foi obtido sucesso nesse particular, sendo que a última notícia oficial de que seria desenvolvida uma API foi postada há aproximadamente uma década por Edmond Lau, à época engenheiro chefe da empresa [45]. São inúmeros os usuários que através do próprio Quora buscam uma API oficial sem êxito [46].

No que se refere a API's não oficiais, isto é, desenvolvidas por terceiros, também não foi possível encontrar uma API funcional, ou seja, capaz de coletar os dados conforme objetivo proposto. Como não foi encontrada API oficial, as não oficiais na linguagem *Python* que foram testadas encontram-se desatualizadas. São exemplos destas API's:

- pyquora [47] – API desatualizada, com a última versão em 05/08/2019;
- quora-api [48] – API desatualizada, com a última versão em 19/06/2015.

Portanto, concluiu-se que, para realizar a coleta de dados do Quora, seria necessária uma extração sem uso de uma API.

4.3 INVESTIGAÇÃO DE MÉTODOS PARA SCRAPING DO QUORA

Ao realizar uma busca no Quora, como, por exemplo, para o termo “hiv”, o usuário é redirecionado para a página <https://www.quora.com/search?q=hiv>. Numa primeira tentativa de coleta de dados do referido endereço com o módulo de *Python* chamado *Requests* [49], foram retornados diversos *scripts* em *JavaScript*, porém, nenhum conteúdo que é apresentado no navegador para o usuário é visualizado

dentre os dados capturados, indicando que o conteúdo é carregado de forma dinâmica.

Além disso, conforme exposto no item 4.1.1, os caminhos do tipo “/search?q=” fazem parte da lista de proibição do robots.txt, de forma que não seria possível fazer o *scraping* por este caminho sem desobedecer aos termos de uso do Quora. Portanto, ficou clara a necessidade de investigar outras formas de realizar a coleta de dados, como explicitado a seguir.

4.3.1 ANÁLISE DA PÁGINA DE BUSCA DO QUORA

Em primeiro lugar, é digno de nota que, a fim de entender como os dados são requisitados pelo navegador ao servidor do Quora, foram utilizadas as ferramentas de desenvolvimento do navegador Firefox [50], observando-se a troca de pacotes que ocorre ao se utilizar o mecanismo de busca do Quora.

Então, valendo-se do monitor de rede desse navegador [51], constatou-se que são enviadas requisições do tipo *POST* para o endereço “https://www.quora.com/AJAX/receive_POST”, levando-se à conclusão de que o site foi implementado utilizando AJAX.

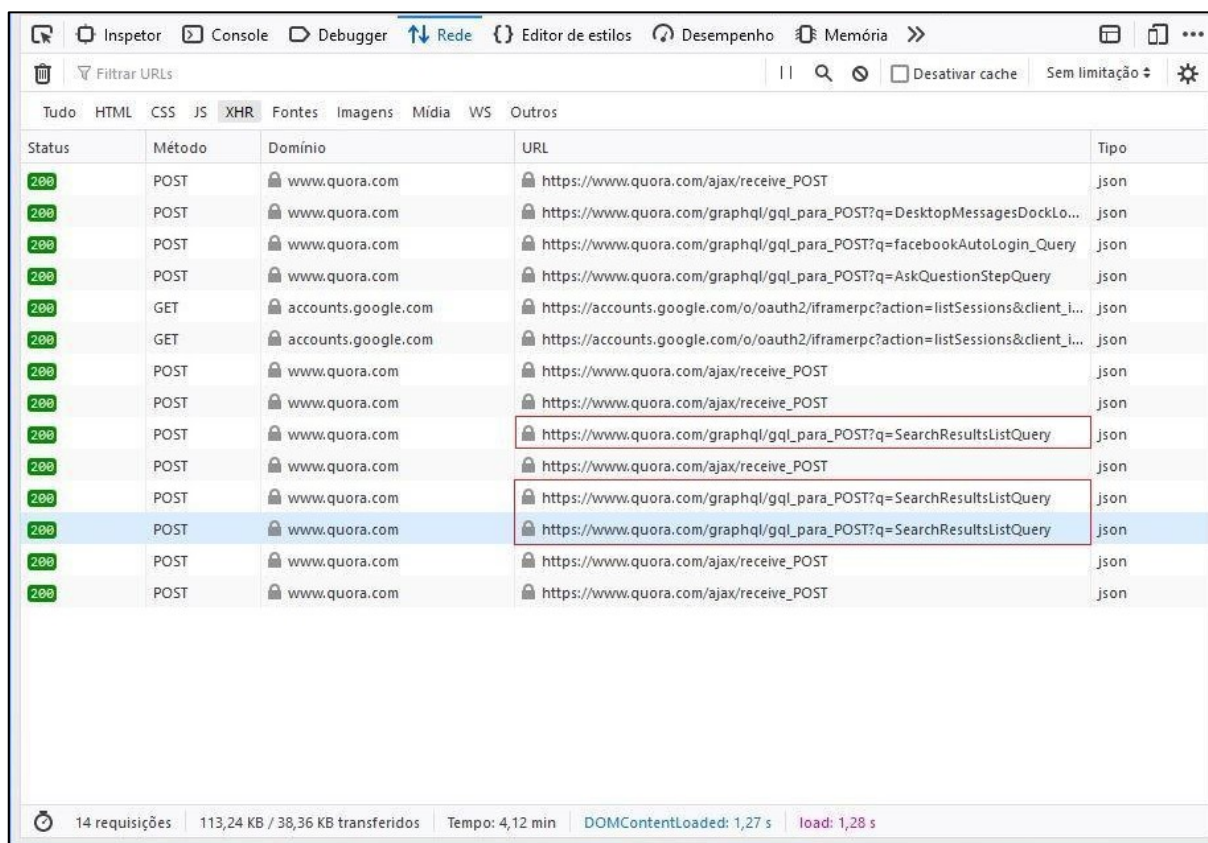
Como supramencionado, numa simples requisição do tipo *GET* feita pelo *Requests*, houve a coleta dos dados estáticos do HTML, mas as requisições AJAX não ocorreram. Ou seja, os dados dinâmicos não foram enviados pelo servidor em resposta à requisição *GET*, razão pela qual não foram observados no conteúdo capturado.

Para resolver essa dificuldade, foram levantadas as seguintes possibilidades: a utilização de navegadores automatizados, como o *Selenium* e o *Splash* [52]; ou a interpretação e utilização das requisições AJAX para realizar requisições diretas ao servidor e assim obter os dados dinâmicos gerados por esta interação.

Como a utilização de navegadores automatizados incrementaria a complexidade da configuração e implementação, além de comprometer o desempenho final da ferramenta, decidiu-se pela realização de requisições utilizando AJAX.

Sendo assim, foi necessário compreender como as requisições AJAX são feitas ao servidor do Quora para posteriormente implementar na ferramenta uma funcionalidade que pudesse reproduzi-las. Para essa finalidade, recorreu-se novamente ao monitor de rede do Firefox, filtrando-se as requisições que utilizam o *XMLHttpRequest* (XHR), objeto responsável por requisitar dados ao servidor e carregar o conteúdo dinâmico na página sem a necessidade de atualizá-la, conforme já referido no item 3.4.2.

Logo, ao realizar a busca por “hiv” e rolar a página para baixo, de modo a disparar novas requisições AJAX e, por conseguinte, o carregamento de mais conteúdo dinamicamente, verificou-se, conforme Figura 5, que eram efetuadas requisições do tipo *POST* utilizando-se o endereço URL https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery.



Status	Método	Domínio	URL	Tipo
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=DesktopMessagesDockLo...	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=facebookAutoLogin_Query	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=AskQuestionStepQuery	json
200	GET	accounts.google.com	https://accounts.google.com/o/oauth2/iframe?pc?action=listSessions&client_i...	json
200	GET	accounts.google.com	https://accounts.google.com/o/oauth2/iframe?pc?action=listSessions&client_i...	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json

14 requisições | 113,24 KB / 38,36 KB transferidos | Tempo: 4,12 min | DOMContentLoaded: 1,27 s | load: 1,28 s

Figura 5 - Monitor de rede do Firefox exibindo requisições da página de busca (grifo nosso)

Analisando uma das requisições *POST* que utilizou a mencionada URL no próprio monitor de rede, foi possível extrair dados de cabeçalho (*headers*) e o JSON referente à sua carga útil (*payload*). Isso permitiu a delimitação das informações

necessárias para envio ao servidor do Quora quando se pretende o carregamento de novos dados na página de busca. Seguem cabeçalhos e *payload*

- cabeçalho geral:

```
{
  "Request URL":
    "https://www.quora.com/graphql/gql_para_POST?q=SearchResultsListQuery",
  "Request Method": "POST",
  "Status Code": "200 ",
  "Remote Address": "151.101.1.2:443",
  "Referrer Policy": "strict-origin-when-cross-origin"
}
```

- cabeçalho da requisição:

```
{
  "accept": "*/*",
  "accept-encoding": "gzip, deflate, br",
  "accept-language": "pt-BR,pt;q=0.9",
  "content-length": "276",
  "content-type": "application/json",
  "cookie": "m-b=qzYSFxFJ22W0lZ6wvc35JAg==; m-b_lax=qzYSFxFJ22W0lZ6wvc35JAg==; m-b_strict=qzYSFxFJ22W0lZ6wvc35JAg==; m-s=pEui-5EC0SLC342vhDmJ4A==; m-uid=None; m-ans_frontend_early_version=15e8ef229a88a1e1; m-dynamicFontSize=regular; G_ENABLED_IDPS=google",
  "dnt": "1",
  "origin": "https://www.quora.com",
  "quora-broadcast-id": "main-w-chan52-8888-react_kwosdyehhfzdgvcprRlZ",
  "quora-canary-revision": "false",
  "quora-formkey": "75dd19863076e742ebc56253a7c597cf",
  "quora-revision": "90219283569000dd0ble6dcld4fb507d4ffcb42",
  "quora-window-id": "react_kwosdyehhfzdgvcpr",
  "sec-ch-ua": "\"Not A;Brand\";v=\"99\", \"Chromium\";v=\"99\", \"Microsoft Edge\";v=\"99\"",
  "sec-ch-ua-mobile": "?0",
  "sec-ch-ua-platform": "\"Windows\"",
  "sec-fetch-dest": "empty",
  "sec-fetch-mode": "cors",
  "sec-fetch-site": "same-origin",
  "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36 Edg/99.0.1150.39"
}
```

- *payload*:

```
{
  "queryName": "SearchResultsListQuery",
  "extensions": {
    "hash":
      "9424e16ff7d82a79e4dcd279cbf4af135766640ba2a5afe8bd28553aa0680a90e"
  },
  "variables": {
    "query": "hiv",
    "disableSpellCheck": null,
    "resultType": "all_types",
    "author": null,
  }
}
```

```

    "time": "all_times",
    "first": 10,
    "after": "9",
    "tribeId": null
  }
}

```

Dentre os campos do cabeçalho da requisição pode-se observar a presença do campo *user-agent*, que, de acordo com o exposto em 4.1.1, deve conter formas de contato com o responsável pelas requisições ao Quora. Outros campos do cabeçalho que se mostraram imprescindíveis para que a requisição fosse devidamente respondida pelo servidor foram: *accept*, *content-type*, *quora-formkey* e *cookie*. É importante ressaltar que as buscas foram realizadas de forma anônima, e, portanto, nenhum dos campos se refere à conta do usuário que realiza a busca.

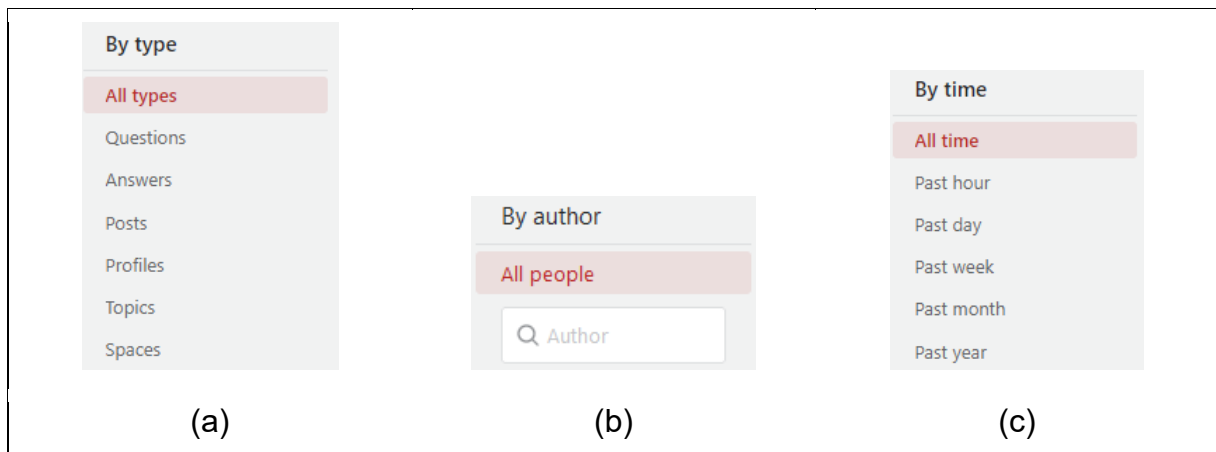


Figura 6 - Filtros de busca pelo tipo de dado (a), pelo autor (b) e pelo período (c).

No que tange ao *payload*, inicialmente podemos constatar que este contém os campos “*query*”, com o termo buscado; “*result_type*”, que se refere ao filtro do tipo de conteúdo buscado; “*author*” (autor da estrutura - pergunta, resposta, postagem etc.); e o campo “*time*”, que se refere ao filtro temporal. Valores podem ser associados aos filtros para determinar quais as características dos dados que serão retornados. A Figura 6, mostra como os filtros estão disponíveis para o usuário da página de buscas. Na Tabela 3, explica-se como cada um dos filtros se comporta na formação do *payload*.

Tabela 3 - Filtros possíveis para formação do *payload* de uma busca.

Nome do campo	Valores possíveis	Retornado do servidor
<i>result_type</i>	<i>all_types</i>	todos os resultados
	<i>question</i>	perguntas
	<i>answer</i>	respostas
	<i>post</i>	postagens
	<i>profile</i>	perfis
	<i>topic</i>	tópicos
	<i>tribe</i>	espaços
<i>author</i>	<i>null</i>	todos os autores
	número de uid	resultados do autor com o número especificado de usuário do Quora (uid)
<i>time</i>	<i>all_times</i>	conteúdos criados em qualquer tempo
	<i>hour</i>	conteúdos criados na última hora
	<i>day</i>	conteúdos criados no último dia
	<i>week</i>	conteúdos criados na última semana
	<i>month</i>	conteúdos criados no último mês
	<i>year</i>	conteúdos criados no último ano

Além desses campos presentes no *payload*, observam-se mais dois que são importantes para a construção das requisições, a saber: "*first*" e "*after*". Esses campos significam, respectivamente, o número de itens retornados a cada requisição ao servidor e o contador que indica a posição após a qual devem ser enviados os próximos itens do servidor.

Exemplificando, o trecho do *payload* mostrado anteriormente indica que a requisição deverá ser respondida com 10 resultados ("*first*": 10). Além disso, o primeiro desses 10 resultados deverá ser o seguinte ao 9º resultado disponível para exibição na página ("*after*": "9"). Logo, a resposta deverá conter os resultados que são exibidos na página entre a 10ª e a 19ª posições. Entende-se que essas posições

relativas são fruto de algoritmos internos ao Quora, o que não é escopo deste trabalho.

Após realizar novas rolagens da página, verificou-se que as novas requisições somente alteram o campo *"after"* do *payload*, sendo certo que o novo valor é igual à soma do valor do *"after"* usado na requisição anterior acrescido do valor de *"first"*. Desta forma, para obter os primeiros resultados exibidos na página, verificou-se a necessidade de atribuir o valor *"-1"* ao campo *"after"*.

Compreendida a forma como é realizada a requisição, faz-se necessário examinar a resposta dada pelo servidor com os resultados da busca no formato JSON. Dessa forma, a Tabela 4 mostra as chaves relevantes para este trabalho, onde elas estão localizadas na estrutura do JSON e seu significado.

Tabela 4 - Chaves do JSON de resposta à requisição da página de busca.

Chave	Estrutura	Significado
edges	data/ searchConnection/ edges	Contém todos dados referentes aos x resultados de busca contidos no JSON, sendo x igual (ou possivelmente menor, no caso dos últimos resultados) ao valor atribuído ao campo <i>first</i> do <i>payload</i> .
node	edges/[i] ¹ /node	Nó contendo diversos dados do resultado.
searchResultType	node/searchResultType	Define o tipo do resultado.
question	node/question	Contém os dados do i-ésimo resultado, desde que seja do tipo pergunta (searchResultType = "question"). Se não for desse tipo, a estrutura referida será diferente, pois cada tipo de resultado possui a sua.
qid	question/qid	Identificação numérica interna do Quora para uma pergunta.
hasNextPage	data/searchConnection/ pageInfo/hasNextPage	Valor booleano que indica se ainda existem novos resultados que possam ser carregados na página.

¹ [i] é o número do resultado dentre o total de resultados compreendidos na resposta, variando de 0 a, no máximo, total de resultados - 1.

4.3.2 ANÁLISE DA PÁGINA DE PERGUNTA E RESPOSTAS ASSOCIADAS

A página de busca do Quora é o meio através do qual um usuário pode encontrar perguntas referentes aos termos de seu interesse. Após receber os resultados de sua busca, o usuário clica em uma pergunta específica, sendo redirecionado para a respectiva página, onde encontrará as respostas fornecidas pela comunidade.

Observe-se que, na página de busca, caso não seja utilizado o filtro de tipo, os resultados podem ser dos mais variados, conforme estruturas explicadas no item 4.1.2. Assim, por exemplo, a busca pelo termo “hiv” retorna os primeiros resultados conforme Figura 7, em que são destacadas as estruturas *Topic* (1); *Space* (2); e *Question* (3 e 4). Além disso, foram identificadas também as estruturas *Answer* (6) e *Profile* (5), embora estivessem incorporadas a um resultado do tipo *Question*, não figurando, nesse caso, como resultados autônomos da busca efetuada.



Figura 7 - Página de busca com estruturas destacadas.

Considerando que a página de busca já foi objeto de análise no item 4.3.1, resta examinar o que ocorre quando um usuário seleciona uma pergunta dentre os resultados da busca, isto é, cabe investigar o redirecionamento para a página da pergunta e a sua estrutura.

Em relação ao redirecionamento, é utilizada uma URL composta por “<https://www.quora.com/>” concatenada com o título da pergunta, substituindo-se os espaços por “-” e removendo-se os caracteres de pontuação. Então, à guisa de exemplo, quando selecionada a pergunta “*Is there a cure for HIV? If not, when will there be one?*”, o redirecionamento ocorre para a página da pergunta com a URL “<https://www.quora.com/Is-there-a-cure-for-HIV-If-not-when-will-there-be-one>”, exibida parcialmente na Figura 8.

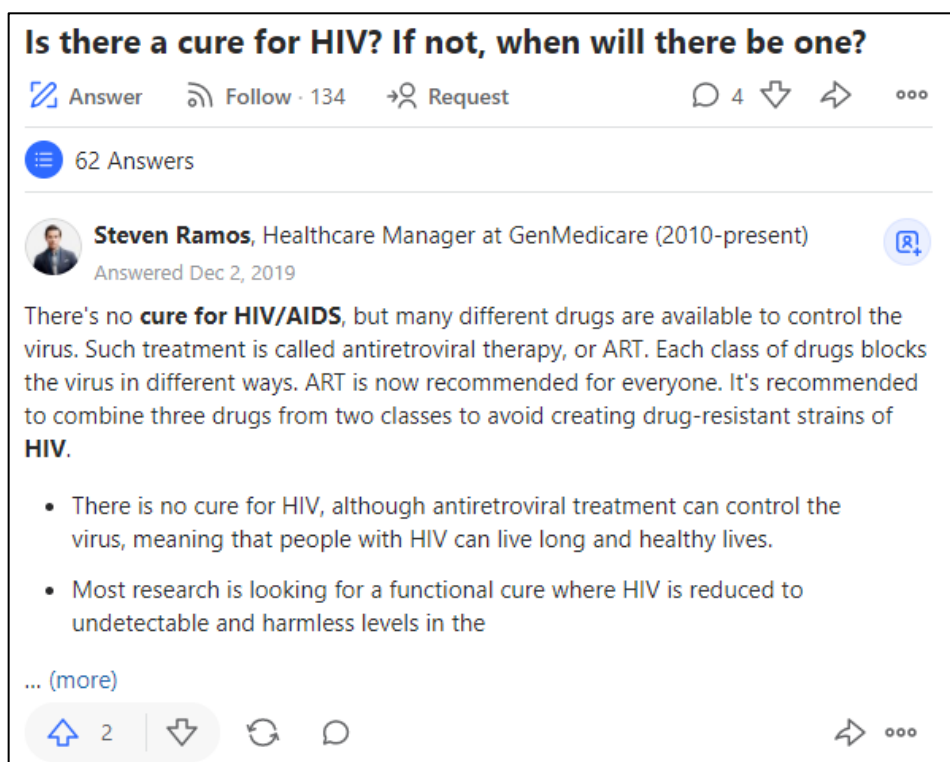


Figura 8 - Recorte de pergunta e resposta

De maneira idêntica ao que foi realizado na página de busca, utilizou-se o monitor de rede do Firefox para visualizar o tráfego de requisições na página de pergunta, verificando-se que também são trocadas mensagens de rede do tipo POST para o endereço https://www.quora.com/AJAX/receive_POST. Rolando-se a página para baixo, de modo a ocasionar novas requisições AJAX, observa-se, conforme Figura 9, que são efetuadas requisições utilizando-se o endereço URL https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery.

Status	Método	Domínio	URL	Tipo
	GET	pagead2.googlesyndication.c...	https://pagead2.googlesyndication.com/pcs/activeview?xai=AKAOjssincZNFYtUvMk_NiG...	
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=DesktopMessagesDockLoaderQuery	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=facebookAutoLogin_Query	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=AskQuestionStepQuery	json
200	GET	securepubads.g.doubleclick...	https://securepubads.g.doubleclick.net/pagead/ppub_config?ippd=www.quora.com	json
200	GET	c.amazon-adsystem.com	https://c.amazon-adsystem.com/e/dtb/bid?src=600&u=https%3A%2F%2Fwww.quora.com	js
200	GET	c.amazon-adsystem.com	https://c.amazon-adsystem.com/cdn/prod/config?src=600&u=https%3A%2F%2Fwww...	xml
200	GET	c.amazon-adsystem.com	https://c.amazon-adsystem.com/bao-csm/aps-comm/aps_csm.js	js
200	GET	accounts.google.com	https://accounts.google.com/o/oauth2/iframe?pc?action=listSessions&client_id=917071...	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery	json
200	GET	accounts.google.com	https://accounts.google.com/o/oauth2/iframe?pc?action=listSessions&client_id=917071...	json
200	GET	pagead2.googlesyndication...	https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gpt&tv=20220...	json
200	GET	securepubads.g.doubleclick...	https://securepubads.g.doubleclick.net/gampad/ads?pvsrcid=3507424005434351&corr...	plain
200	GET	www.facebook.com	https://www.facebook.com/x/oauth/status?client_id=13660945963&input_token&or...	plain
200	POST	d9.flashtalking.com	https://d9.flashtalking.com/lgc	json
200	GET	pagead2.googlesyndication...	https://pagead2.googlesyndication.com/getconfig/sodar?sv=200&tid=gda&tv=r2022...	json
200	GET	securepubads.g.doubleclick...	https://securepubads.g.doubleclick.net/pcs/view?xai=AKAOjstcEoi_kmsh4hhKRrQpTRY...	gif
200	POST	www.quora.com	https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedListQuery	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json
200	POST	www.quora.com	https://www.quora.com/ajax/receive_POST	json

25 requisições 395,81 KB / 101,29 KB transferidos Tempo: 1,11 min DOMContentLoaded: 1,76 s load: 1,80 s

Figura 9 - Monitor de rede do Firefox exibindo requisições da página de pergunta (grifo nosso)

Examinando-se uma das requisições para a mencionada URL, foi possível extrair os cabeçalhos (*headers*) e a carga útil (*payload*) em formato JSON, cujas informações constam a seguir:

- cabeçalho geral:

```
{
  "URL da Solicitação":
    "https://www.quora.com/graphql/gql_para_POST?q=QuestionAnswerPagedLis
tQuery",
  "Método de solicitação": "POST",
  "Código do status": "200 ",
  "Endereço Remoto": "104.18.0.74:443",
  "Política de referenciador": "strict-origin-when-cross-origin"
}
```

- cabeçalho da requisição:

```
{
  "accept": "*/*",
  "accept-encoding": "gzip, deflate, br",
  "accept-language": "pt-BR,pt;q=0.9",
  "content-length": "488",
  "content-type": "application/json",
  "cookie": "m-b=qzYSFxFJ22W0lZ6wvc35JAg==; m-
b_lax=qzYSFxFJ22W0lZ6wvc35JAg==; m-b strict=qzYSFxFJ22W0lZ6wvc35JAg==;
```

```

m-s=pEui-5ECoSLC342vhDmJ4A==; m-uid=None; m-dynamicFontSize=regular;
G_ENABLED_IDPS=google; __stripe_mid=aelf229f-465a-4aae-ab9c-
1dc7be3c3cec394e46; m-ans_frontend_early_version=ae5f6cbbbb75641;
__cf_bm=b3NIEtbC_huYkkYR_lPc_8tbihCftjTLmoLdflup4UU-1648925961-0-
ATPCoHy/x0Vhn4z79qqVQsIU7nM9AAYSfQnQ21l5kjl5n7Fp3ATSXBQWPtwKi+Tr0Q99
G3EXimNiB2xmdXt1Jw=; __aaxsc=2; m-sa=1; __stripe_sid=1314df65-28d3-
4b36-9bcc-8df34e8ab2d28d1c61; aasd=2%7C1648926091473",
"dnt": "1",
"origin": "https://www.quora.com",
"quora-broadcast-id": "main-w-chan49-8888-react_nhednaxpeaathzoj-8NBH",
"quora-canary-revision": "false",
"quora-formkey": "75ddl9863076e742ebc56253a7c597cf",
"quora-revision": "0d20e0757d455b8d9468f977ff07cfce0424d9b4",
"quora-window-id": "react_nhednaxpeaathzoj",
"sec-ch-ua": "\" Not A;Brand\";v=\"99\", \"Chromium\";v=\"99\", \"Microsoft Edge\";v=\"99\"",
"sec-ch-ua-mobile": "?0",
"sec-ch-ua-platform": "\"Windows\"",
"sec-fetch-dest": "empty",
"sec-fetch-mode": "cors",
"sec-fetch-site": "same-origin",
"user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51
Safari/537.36 Edg/99.0.1150.39"
}

```

- **payload:**

```

{
  "extensions": {
    "hash": "58520df560d929eaabbc4c7e96cf7fbc6f53ba52ea026b9f12ef7ebbd1336663"
  },
  "queryName": "QuestionAnswerPagedListQuery",
  "variables": {
    "adsData": {
      "answer_intervals": [3, 4],
      "ads": {
        "answer_intervals": [3, 4],
        "ads": [
          {
            "ad_id": 211106236445379,
            "query_id": 1116306240300897581
          },
          {
            "ad_id": 211106235403260,
            "query_id": 1116306240300897581
          },
          {
            "ad_id": 211106236472041,
            "query_id": 1116306240300897581
          }
        ]
      },
      "after": "16",
      "first": 12,
      "forceScoreVersion": null,
      "qid": 929440,
      "refetch": false,
      "topAid": null
    }
  }
}

```

No que tange aos cabeçalhos, a lógica é bem similar àquela delineada no item 4.3.1, referente à análise da página de busca. Por sua vez, no que se refere ao *payload*, a estrutura é um tanto diversa, sendo relevante notar que um dos campos - o “qid” - é o número identificador de uma pergunta do Quora. Além disso, também é

de se observar que o campo “*first*” possui valor padrão 12, e não 10 como no caso de requisições da página de busca.

Em relação à resposta da requisição em formato JSON, foi realizada a análise nos mesmos moldes daquela efetuada no item 4.3.1, resultando na Tabela 5, que mostra as chaves relevantes para este trabalho, onde elas estão localizadas na estrutura do JSON e seu significado.

Tabela 5 - Chaves do JSON de resposta à requisição da página de pergunta.

Chave	Estrutura	Significado
<code>pagedListDataConnection</code>	<code>data/question/ pagedListDataConnection</code>	Estrutura com o conteúdo das respostas.
<code>edges</code>	<code>pagedListDataConnection / edges</code>	Contém todos dados referentes aos x itens da página de pergunta contidos no JSON, sendo x igual (ou possivelmente menor, no caso dos últimos itens) ao valor atribuído ao campo <i>first</i> do <i>payload</i> .
<code>__typename</code>	<code>edges/[i]/node/__typena me</code>	Define o tipo do i-ésimo item da página constante do JSON.
<code>answer</code>	<code>edges/[i]/node/answer</code>	Contém os dados do i-ésimo item, desde que seja do tipo resposta (<code>__typename = "QuestionAnswerItem 2"</code>). Se não for desse tipo, a estrutura referida será diferente, pois cada tipo de item possui a sua.
<code>aid</code>	<code>edges/[i]/node/answer/a id</code>	Identificação numérica interna do Quora para uma pergunta.
<code>hasNextPage</code>	<code>pagedListDataConnection / pageInfo/hasNextPage</code>	Valor booleano que indica se ainda existem novos resultados que possam ser carregados na página.

A menção a “itens” na Tabela 5 é motivada pelo fato de que a página de uma pergunta contém não apenas suas respostas, mas também anúncios, perguntas relacionadas e respostas relacionadas, conforme Figura 10. Sendo assim, esses variados “itens” podem estar presentes no JSON de resposta à requisição da página de uma pergunta. Saliente-se que, no presente trabalho, o escopo de coleta abrange a pergunta relativa a um termo de interesse e as respectivas respostas, relegando a trabalhos futuros os demais itens.

The figure displays a Quora page for the question "How many years can a person live with HIV/AIDS?". The page is divided into four sections labeled (a) through (d). (a) shows an advertisement for Duolingo English Test. (b) shows related questions, such as "Is there a cure for HIV?". (c) shows related answers, including one from an anonymous user. (d) shows a promotional banner for Quora+ membership, which offers access to more answers and ad-free browsing.

Figura 10 - Página de pergunta com (a) anúncio (grifo nosso), (b) questões relacionadas (grifo nosso), (c) respostas relacionadas e (d) acesso limitado ao Quora+

Ressalte-se que é possível determinadas respostas terem sua visualização bloqueada para usuários comuns, exigindo-se, para acesso a seu conteúdo, a filiação ao “Quora Plus” [53], que é uma versão paga do Quora, conforme exibido em (d) da Figura 10. Nesses casos, como a ferramenta ora

A ideia original de se utilizar o módulo *Request* mostrou-se ineficiente, pois as requisições realizadas por intermédio deste são síncronas, isto é, a execução do código ficava pausada até que a resposta de uma requisição fosse recebida do servidor do Quora. Nesse sentido, foi necessário encontrar uma solução assíncrona, que tornasse a coleta de dados mais célere e conferisse maior eficiência à execução do código.

A solução adotada no presente trabalho foi o *framework Scrapy*, que é definido como um *framework* rápido e de alto nível para *web crawling* e *web scraping*, usado para uma variedade de propósitos, dentre os quais a coleta de dados estruturados de páginas da *web* [5].

Esse *framework* possui como uma das principais vantagens o agendamento e processamento assíncrono das requisições, o que significa dizer que as requisições são feitas de forma independente, isto é, uma requisição pode ser enviada mesmo que a anterior não tenha sido respondida [55], satisfazendo, assim, as necessidades para a coleta em larga escala de dados do Quora.

4.4 FUNCIONAMENTO DO APLICATIVO QSCRAPER

O funcionamento do aplicativo QScraper foi concebido conforme fluxograma da Figura 12, iniciando-se pelas leituras dos arquivos de palavras-chave e parâmetros para, em seguida, executar a coleta de perguntas e respostas do site do Quora, com o respectivo carregamento para o banco de dados do MongoDB [56], realizado com o auxílio do módulo PyMongo [57].

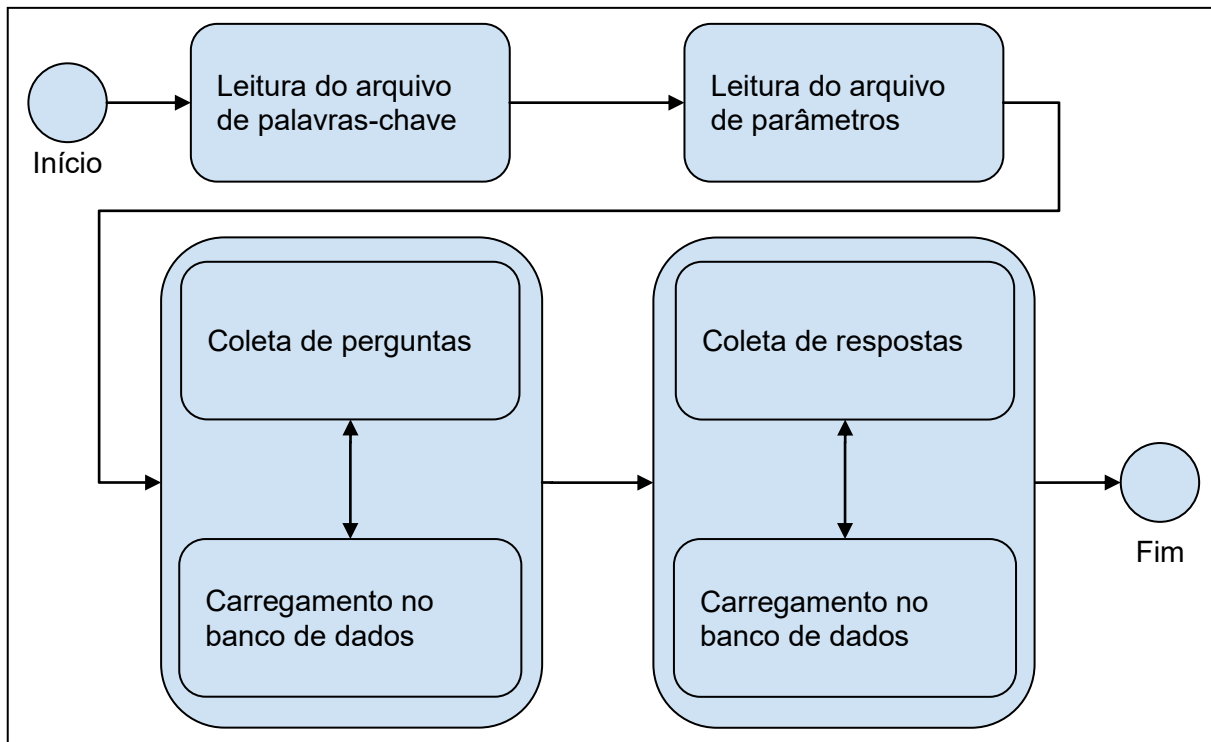


Figura 12 - Fluxograma do aplicativo

Os dados coletados foram armazenados no MongoDB online conforme o diagrama entidade-relacionamento (DER) constante da Figura 13.

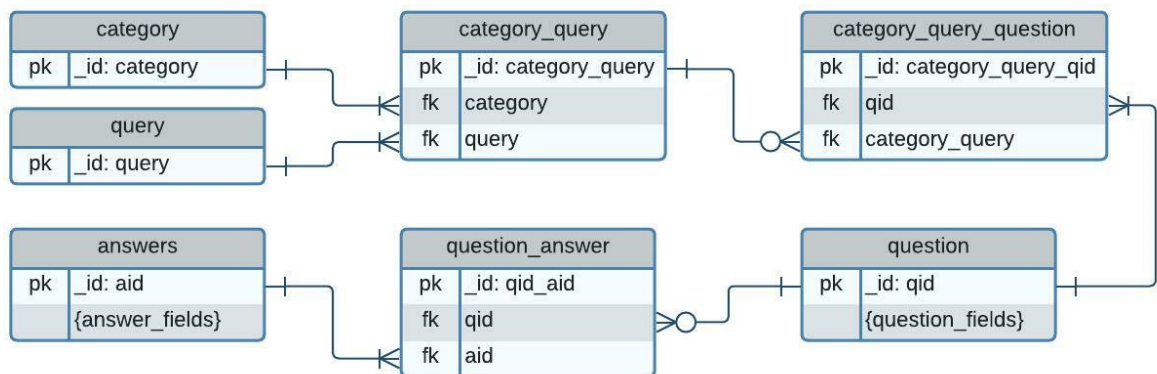


Figura 13 - Diagrama Entidade-Relacionamento do banco de dados do aplicativo

A Tabela 6 traz as coleções criadas no MongoDB, os campos criados, com o tipo da chave, onde “*pk*” (*primary key*) indica uma chave primária e “*fk*” (*foreign key*) indica uma chave estrangeira; além do significado de cada chave da coleção.

Tabela 6 - Definição dos campos das coleções no banco de dados.

Coleção	Campo	[Tipo]	Definição
query	_id	[pk]	Texto da palavra-chave.
category	_id	[pk]	Texto da categoria.
category_query	_id	[pk]	Textos da categoria e da palavra-chave, separadas por “_”.
	category	[fk]	Aponta para a coleção <i>category</i> .
	query	[fk]	Aponta para a coleção <i>query</i> .
category_query_question	_id	[pk]	Textos da categoria, da palavra-chave e da pergunta, separadas por “_”.
	category_query	[fk]	Aponta para a coleção <i>category_query</i> .
	qid	[fk]	Aponta para a coleção <i>question</i> .
question	_id	[pk]	Identificador da pergunta no Quora (<i>qid</i>).
	{question_fields}		Demais dados da pergunta que constam no JSON coletado do Quora.
answers	_id	[pk]	Identificador da resposta no Quora (<i>aid</i>).
	{answer_fields}		Demais dados da resposta que constam do JSON coletado do Quora.
question_answers	_id	[pk]	Textos da qid e da aid, separadas por “_”.
	qid	[fk]	Aponta para a coleção <i>question</i> .
	aid	[fk]	Aponta para a coleção <i>answers</i> .

4.5 CASO DE USO

O projeto desenvolvido compreende apenas o caso de uso relativo à funcionalidade através da qual um usuário realiza a coleta de perguntas e respostas relacionadas a um ou mais termos na plataforma Quora, com o respectivo armazenamento no banco de dados MongoDB. A descrição deste caso de uso pode ser visualizada na Tabela 7.

Tabela 7 - Descrição do Caso de Uso “Coletar Perguntas e Respostas”.

Nome:	UC01 - Coletar Perguntas e Respostas
Objetivo:	Obter as perguntas e repostas da plataforma Quora que se relacionam aos termos fornecidos pelo usuário
Atores:	Usuário
Pré-condições:	Python e pacotes utilizados instalados; Banco de dados MongoDB local ou na nuvem configurado; Arquivo de parâmetros de requisições ao Quora configurado; Arquivo de termos de busca configurado.
Trigger:	Usuário executa aplicativo QScraper
Fluxo Principal:	<ol style="list-style-type: none"> 1. Usuário indica o caminho (<i>path</i>) do arquivo de parâmetros das requisições HTTP, bem como seu nome e e-mail para adição ao cabeçalho <i>user-agent</i>. 2. Usuário indica o caminho (<i>path</i>) do arquivo de termos de busca e uma instância de um cliente MongoDB. 3. Sistema realiza requisições para coleta de perguntas relacionadas aos termos de busca. 4. Sistema salva dados das respostas às requisições em banco de dados. 5. Sistema realiza requisições para coleta das respostas relacionadas a cada pergunta coletada. 6. Sistema salva dados das respostas às requisições

	<p>em banco de dados.</p> <p>7. Sistema encerra sua execução</p>
Fluxo Alternativo:	<p>1.1. O caminho (<i>path</i>) do arquivo de parâmetros indicado não contém um arquivo.</p> <ol style="list-style-type: none"> 1. O sistema lança exceção, informando que é necessário especificar corretamente o caminho do arquivo de parâmetros. 2. O sistema encerra sua execução. <p>2.1 O arquivo de termos de busca indicado contém alguma categoria sem termos de busca.</p> <ol style="list-style-type: none"> 1. O sistema lança exceção, informando que a categoria em questão não possui termos de busca. 2. O sistema encerra sua execução. <p>3.1 O sistema não consegue formar uma requisição válida para coleta de perguntas por configuração inadequada do arquivo de parâmetros.</p> <ol style="list-style-type: none"> 1. O sistema lança exceção, informando que o arquivo de parâmetros deve ser revisto. 2. O sistema encerra sua execução. <p>5.1 O sistema não consegue formar uma requisição válida para coleta de respostas por configuração inadequada do arquivo de parâmetros.</p> <ol style="list-style-type: none"> 1. O sistema lança exceção, informando que o arquivo de parâmetros deve ser revisto. 2. O sistema encerra sua execução.
Pós-condições:	Perguntas e Respostas salvas no banco de dados
Regras de negócio:	<p>RN1: Todo termo de busca deve estar relacionado ao menos a uma categoria.</p> <p>RN2: Toda categoria deverá conter ao menos um termo de busca.</p> <p>RN3: O usuário deverá fornecer nome e e-mail para coleta de dados no Quora.</p>

As demais classes utilizadas são provenientes de pacotes de terceiros, a saber: pymongo, scrapy e twister [58].

5 TESTES

O presente capítulo se dedica à exposição do ambiente de execução e dos resultados dos testes realizados em 02/05/2022 com o aplicativo QScraper para coleta de perguntas e respostas na plataforma Quora envolvendo termos e expressões relacionados ao tratamento de HIV.

Inicialmente, são listadas na Tabela 8 informações relativas ao *hardware* do computador onde foi executado o programa *QScraper*.

Tabela 8 - Hardware do ambiente de testes.

Computador	Dell Inspiron One 2330 (All in one)
Processador	Intel Core i5-3330S 2.70GHz
Memória RAM	8,00 GB
Placa de Vídeo	AMD Radeon 7600A
Disco Rígido	Seagate 2TB SATA III
Adaptador de Rede	Qualcomm Atheros AR8161 PCI-E Gigabit Ethernet
Monitor	Dell 23 polegadas

No que tange às informações de *software* utilizado para a realização dos testes, na Tabela 9 são detalhadas tanto aquelas relativas ao sistema instalado quanto as que se referem ao ambiente virtual utilizado.

Tabela 9 – Software do ambiente de testes – ambiente de software geral.

Software	Versão
Sistema operacional	Windows 8.1 Single Language 64 bits
Ambiente de desenvolvimento integrado (IDE)	Pycharm Community Edition 2021.2.4
Interpretador Python	Python 3.9.7

Foram utilizados os pacotes do *Python: Scrapy* (versão 2.5.1), *PyMongo* (versão 4.0.1), além de suas dependências, para a construção do *QScraper* e realização dos testes.

Os termos e expressões relacionados ao tratamento de HIV foram fornecidos pelo orientador do presente trabalho, baseando-se nos termos definidos pelo especialista no ANEXO A, no formato de um arquivo JSON, conforme segue:

```
{
  "general": ["#prep (hiv OR treatment)", "#tripletherapy OR (triple
therapy)", "#anti OR anti (hiv OR treatment)", "#hivinfection OR (hiv
infection)", "#drug OR (drug hiv)", "#NormalizingHIVChallenge OR
(Normalizing HIV Challenge)", "#livingwithaids OR (living aids)",
"#hivtreatment OR hivtreatment OR (hiv treatment)", "#pep (hiv OR
treatment)", "#pepforhiv OR (pep hiv)", "#pepforearlyhiv OR (pep for
early hiv) or (pep hiv)", "#pepindelhi OR (pep delhi)",
"#peptreatment OR (pep treatment)", "#peptreatmentinmalviyanagar or
(pep treatment malviyanagar) OR (pep malviyanagar)",
"#pepcenterforhiv OR (pep center for hiv) ", "#pephivcenter OR (pep
center hiv)", "#pepforealryexposer OR (pep real exposor)",
"#pepandprep OR (pep prep) OR ((pep OR prep) treatment)"],
  "fixed dose": ["#truvada OR truvada", "#atrimpla OR atrimpla", "#epzicom
OR epzicom", "#complanera OR complera", "#cimduduo OR cimduduo", "#combivir
OR combivir", "#descovy OR descovy", "#Temixys or Temixys",
"#Trizivir or Trizivir", "#Delstrigo or Delstrigo", "#Odefsey or
Odefsey", "#SymfiLo or SymfiLo", "#Biktarvy or Biktarvy", "#Dovato or
Dovato", "#Triumeq or Triumeq", "#Juluca or Juluca", "#Genvoya or
Genvoya", "#Stribild or Stribild", "#Kaletra or Kaletra", "#Kivexa or
Kivexa", "#Triomune or Triomune", "#Duovir or Duovir", "#Evotaz or
Evotaz", "#Prezcobix or Prezcobix", "#Rezolsta or Rezolsta",
"#Dutrebiz or Dutrebiz", "#Symfi or Symfi", "#Eviplera or Eviplera",
"#Symtuza or Symtuza", "#Cabenuva or Cabenuva"],
  "not fixed dose": ["#viread OR viread", "#ftc OR (ftc treatment) ",
"#3tc OR 3tc", "hiv OR treatment", "#isentress OR isentress",
"#reyataz OR reyataz", "#norvir OR norvir", "#sustiva OR sustiva",
"#stocrin OR stocrin", "#tivicay OR tivica", "#lamivudine OR
lamivudine", "#epivir OR epivir"]
}
```

Como se pode observar, os termos foram divididos em três categorias: “*general*”, “*fixed dose*” e “*not fixed dose*”. Esses termos foram preparados para se realizar a raspagem de dados de outra rede social e não possuiriam o mesmo efeito no *Quora*.

Em acordo com o orientador, os termos de “*general*” foram excluídos, por trazerem muitas informações generalistas. Além disso, os demais termos das demais categorias foram tratados para uso no *Quora*, pois inicialmente refletiam expressões que foram usadas em uma API do Twitter, em um projeto anterior com objetivo semelhante ao deste, e cujo formato não se aplica à busca no *Quora*. O resultado é mostrado a seguir.

```
{
  "fixed dose": ["truvada", "atrimpla", "epzicom", "completa", "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"]
}
```

Utilizando-se o aplicativo criado, obteve-se os resultados a categoria “*fixed dose*”, conforme apresentado na Tabela 10.

Tabela 10 – Resultado quantitativo dos testes – categoria “*fixed dose*”.

Palavra-chave	Perguntas	Respostas
atrimpla	15	13
biktarvy	55	51
cabenuva	16	8
cimduo	0	0
combivir	4	3
completa	1	0
delstrigo	3	0
descovy	34	19
dovato	11	1
duovir	3	1
dutrebiz	0	0
epzicom	1	3
eviplera	0	0
evotaz	0	0
genvoya	7	1
juluca	5	0
kaletra	6	13
kivexa	11	18
odefsey	6	14

Palavra-chave	Perguntas	Respostas
prezcobix	1	0
rezolsta	0	0
stribild	6	6
symfi	0	0
symfilo	0	0
symtuza	3	2
temixys	0	0
triomune	0	0
triumeq	8	5
trizivir	0	0
truvada	184	229

A Tabela 11 apresenta os resultados para todas as palavras-chave relativas à categoria “*not fixed dose*”.

Tabela 11 – Resultado quantitativo dos testes – categoria “*not fixed dose*”.

Palavra-chave	Perguntas	Respostas
3tc	22	26
epivir	0	0
ftc	1.093	1.083
ftc treatment	0	0
hiv treatment	703	1.469
isentress	8	6
lamivudine	25	11
norvir	4	1
reyataz	2	2
stocrin	4	3
sustiva	1	0
tivicay	4	1
viread	6	5

Por fim a Tabela 12 sumariza o quantitativo de perguntas e respostas de cada categoria.

Tabela 12 - Resultado da coleta de dados por categorias

Palavra-chave	Perguntas	Respostas
<i>Fixed dose</i>	380	388
<i>Not fixed dose</i>	1.872	2.607

CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho alcança seus objetivos ao efetuar com sucesso a coleta de perguntas e respostas do Quora, salvando o respectivo conteúdo em banco de dados. Dessa forma, os dados estarão estruturados e disponíveis para o aprofundamento de conhecimentos após cada coleta efetuada com o software desenvolvido.

Foi possível observar que as palavras-chave ligadas ao nome de remédios não retornam muitos resultados. Isto foi exemplificado com a palavra-chave “*truvada*”, em relação à qual houve o maior número de perguntas e respostas coletadas (184 perguntas e 229 respostas). Por outro lado, a expressão “*hiv treatment*”, mais generalista e não necessariamente relacionada a medicações para o HIV, retornou 703 perguntas e 1.469 respostas, a indicar que uma busca mais generalista no Quora pode retornar uma maior massa de dados, como esperado. Além disso, verificou-se que a palavra-chave “*ftc*” gerou 1.093 perguntas e 1.083 respostas associadas, porém, este termo também corresponde a uma sigla de *Federal Trade Commision*², que não está relacionada com o objetivo deste trabalho, de modo que a sua escolha como palavra-chave gera uma série de dados espúrios.

A partir disso, conclui-se que o Quora contém uma massa de dados relevante para a pesquisa de farmacêuticos, mas as palavras-chave deverão ser bem selecionadas, além de tratados os dados coletados para a remoção de dados espúrios, como foi observado no caso do termo “*ftc*”.

A partir do presente trabalho, é possível sugerir para trabalhos futuros a coleta de dados provenientes de outras estruturas dentro do Quora não utilizadas pelo software ora desenvolvido, como *posts*, *spaces* e *topics*. A otimização da coleta de dados também seria campo fértil de pesquisa, por exemplo, com a possibilidade de utilização de processamento paralelo.

² Comissão norte americana que trabalha para prevenir práticas comerciais fraudulentas, enganosas e injustas. Eles também fornecem informações para ajudar os consumidores a localizar, parar e evitar golpes e fraudes. [61]

REFERÊNCIAS BIBLIOGRÁFICAS

1. DEAN, B. How Many People Use Social Media in 2022? (65+ Statistics). **Backlinko**, 2021. Disponível em: <<https://backlinko.com/social-media-users>>. Acesso em: 07 Maio 2022.
2. PERSHAD, Y. et al. Social Medicine: Twitter in Healthcare. **Journal of Clinical Medicine**, v. 7, Maio 2018. Disponível em: <<https://doi.org/10.3390/jcm7060121>>. Acesso em: 07 Maio 2022.
3. ALASMARI, A.; ZHOU, L. How multimorbid health information consumers interact in an online community Q&A platform. **International Journal of Medical Informatics**, v. 131, Setembro 2019. ISSN 1386-5056.
4. QUORA. Quora - Um lugar para compartilhar conhecimento e entender melhor o mundo. **Quora**. Disponível em: <<https://www.quora.com/>>. Acesso em: 10 Abril 2022.
5. SCRAPY 2.6 documentation. **Scrapy 2.6.1 documentation**. Disponível em: <<https://docs.scrapy.org/en/latest/>>. Acesso em: 02 Fevereiro 2022.
6. MUTHUKADAN, B. Selenium Python Bindings 2 documentation. **Selenium with Python**. Disponível em: <<https://selenium-python.readthedocs.io/>>. Acesso em: 10 Abril 2022.
7. PANDAS - Python Data Analyses Library. **Pandas**, 2022. Disponível em: <<https://pandas.pydata.org/>>. Acesso em: 09 Maio 2022.
8. CHEN, Y. et al. What Concerns Consumers about Hypertension? A Comparison between the Online Health Community and the Q&A Forum. **International Journal of Computational Intelligence Systems**, v. 14, p. 734-743, Janeiro 2021. ISSN 1875-6891.
9. MEDHELP. Health community, health information, medical questions, and medical apps. **MedHelp**. Disponível em: <<https://www.medhelp.org/>>. Acesso em: 10 Abril 2022.
10. MUPPIDI, S.; RAO, P. S.; KIRUBAKARAN, S. S. An efficient framework for blog recommender system. **Materials Today: Proceedings**, Fevereiro 2021. ISSN

2214-7853.

11. XIE, Q. et al. Chatbot Application on Cryptocurrency. **IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFER)**, Shenzhen, Maio 2019.
12. BEAUTIFUL SOUP. Beautiful Soup 4.4.0 documentation. **Beautiful Soup Documentation**. Disponível em: <<https://beautiful-soup-4.readthedocs.io/en/latest/>>. Acesso em: 10 Abril 2022.
13. NÚMERO de usuários de Internet no mundo chega aos 4,66 bilhões. **ISTOÉ DINHEIRO**, 2021. Disponível em: <<https://www.istoedinheiro.com.br/numero-de-usuarios-de-internet-no-mundo-chega-aos-466-bilhoes/>>. Acesso em: 04 Abril 2022.
14. CETIQ. **Resumo Executivo: Pesquisa TIC Domicílios 2020**. CETIQ. [S.l.], p. 8. 2020.
15. PÁDUA, E. M. M. D. **Metodologia da pesquisa**: Abordagem teórico-prática. Campinas: Papirus, 2018.
16. CUNHA, M. B. D. **Para saber mais**: fontes de informação em ciência e tecnologia. Brasília: Briquet de Lemos, 2001.
17. BAKER, L. **Data Types**. [S.l.]: CSI Publishing, 2020.
18. OECD. **Data-Driven Innovation**: Big Data for Growth and Well-being. Paris: OECD Publishing, 2015.
19. COSTA, D. G. **Uma breve história da computação**. Feira de Santana: UEFS Editora, 2016.
20. KUROSE, J. F.; ROSS, K. W. **Redes De Computadores E A Internet**: uma abordagem top-down. 3ª. ed. São Paulo: Pearson Addison Wesley, 2006.
21. WORLD Wide Web - MDN Web Docs Glossary: Definitions of Web-related terms. **MDN**. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/World_Wide_Web>. Acesso em: 21 Novembro 2021.
22. HTML: Linguagem de Marcação de Hipertexto. **MDN**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 27 Dezembro 2021.
23. LINGUAGEM de marcação. **Wikipédia, a enciclopédia livre**. Disponível em:

- <https://pt.wikipedia.org/wiki/Linguagem_de_marcação>. Acesso em: 27 Dezembro 2021.
24. WIKIPEDIA. Chevron (tipografia). **Wikipedia, a enciclopédia livre**. Disponível em: <[https://pt.wikipedia.org/wiki/Chevron_\(tipografia\)](https://pt.wikipedia.org/wiki/Chevron_(tipografia))>. Acesso em: 03 Janeiro 2022.
25. W3C. HTML & CSS. **W3C**. Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss>>. Acesso em: 10 Janeiro 2022.
26. WHAT is JavaScript? - Learn web development. **MDN**. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Acesso em: 10 Janeiro 2022.
27. XMLHTTPREQUEST - APIs da Web. **MDN**, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/API/XMLHttpRequest>>. Acesso em: 10 Março 2022.
28. PRIMEIROS passos - Guia do desenvolvedor web. **MDN Web Docs**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/Guide/AJAX/Getting_Started>. Acesso em: 10 Março 2022.
29. MOZILLA. What is a web browser? **Mozilla**. Disponível em: <<https://www.mozilla.org/en-US/firefox/browsers/what-is-a-browser/>>. Acesso em: 03 Novembro 2021.
30. POPULATING the page: how browsers work - Web Performance. **MDN Web Docs**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work>. Acesso em: 03 Novembro 2021.
31. MITCHELL, R. **Web Scraping com Python**: Coletando mais dados da web moderna. 2ª. ed. São Paulo: Novatec Editora, 2019.
32. QUORA. Sobre. **Quora**. Disponível em: <<https://pt.quora.com/about>>. Acesso em: 02 Março 2022.
33. 21 Quora Statistics Marketers Need to Know For 2021. **Foundation**. Disponível em: <<https://foundationinc.co/lab/quora-statistics/>>. Acesso em: 09 Abril 2022.
34. QUORA. Termos de Serviço. **Quora**, 2020. Disponível em:

- <<https://pt.quora.com/about/tos>>. Acesso em: 09 Abril 2022.
35. USER-AGENT - HTTP. **MDN**, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Headers/User-Agent>>. Acesso em: 09 Abril 2022.
 36. CRIAR e enviar um arquivo robots.txt. **Google Developers**, 2022. Disponível em: <<https://developers.google.com/search/docs/advanced/robots/create-robots-txt?hl=pt-br>, acesso em 09/04/2022>. Acesso em: 09 Abril 2022.
 37. ROBOTS exclusion standard. **Wikipedia**, 2022. Disponível em: <https://en.wikipedia.org/wiki/Robots_exclusion_standard>. Acesso em: 09 Abril 2022.
 38. QUORA. robots.txt. **Quora**. Disponível em: <<https://www.quora.com/robots.txt>>. Acesso em: 09 Abril 2022.
 39. COMO o Google interpreta a especificação de robots.txt. **Google Developers**, 2022. Disponível em: <https://developers.google.com/search/docs/advanced/robots/robots_txt?hl=pt-br>. Acesso em: 09 Abril 2022.
 40. OFFICIAL QUORA ACCOUNT. What are Topics on Quora? **Quora Help Center**, 2021. Disponível em: <<https://help.quora.com/hc/en-us/articles/115004755623-What-are-Topics-on-Quora-?share=1>>. Acesso em: 10 Abril 2022.
 41. OFFICIAL QUORA ACCOUNT. How do I get started using Quora? **Quora Help Center**, 2021. Disponível em: <<https://help.quora.com/hc/en-us/articles/115004145086-How-do-I-get-started-using-Quora->>. Acesso em: 02 Março 2022.
 42. OFFICIAL QUORA ACCOUNT. About Quora Spaces. **Quora Help Center**, 2021. Disponível em: <<https://help.quora.com/hc/en-us/articles/360061486651-About-Quora-Spaces>>. Acesso em: 02 Março 2022.
 43. OFFICIAL QUORA ACCOUNT. How do I add content to my Space? **Quora Help Center**, 2021. Disponível em: <<https://help.quora.com/hc/en-us/articles/360061074272/>>. Acesso em: 02 Março 2022.
 44. IBM CLOUD EDUCATION. What is an Application Programming Interface (API). **IBM**, 2020. Disponível em: <<https://www.ibm.com/cloud/learn/api>>. Acesso em: 02 Março 2022.

45. LAU, E. Quora Extension API - Edmond Lau's Posts. **Quora**. Disponível em: <<https://edmondlausposts.quora.com/Quora-Extension-API>>. Acesso em: 21 Novembro 2021.
46. WHAT is the status of the full Quora API? **Quora**. Disponível em: <<https://www.quora.com/What-is-the-status-of-the-full-Quora-API>>. Acesso em: 21 Novembro 2021.
47. SU, C. csu/pyquora: A Python module for fetching and parsing data from Quora. **GitHub**, 2019. Disponível em: <<https://github.com/csu/pyquora>>. Acesso em: 02 Março 2022.
48. SU, C. quora-api/quora-api: An unofficial API for Quora. **GitHub**, 2016. Disponível em: <<https://github.com/csu/quora-api>>. Acesso em: 02 Março 2022.
49. REQUESTS 2.26.0 documentation. **Requests: HTTP for Humans™**. Disponível em: <<https://docs.python-requests.org/en/master/index.html>>. Acesso em: 10 Abril 2022.
50. FERRAMENTAS do Firefox para desenvolvedores. **MDN**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Tools>>. Acesso em: 21 Novembro 2021.
51. MONITOR de Rede - Ferramentas do Firefox para desenvolvedores. **MDN**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Tools/Network_Monitor>. Acesso em: 22 Novembro 2021.
52. SPLASH - A javascript rendering service. **Splash 3.5 documentation**. Disponível em: <<https://splash.readthedocs.io/en/stable/>>. Acesso em: 22 Novembro 2021.
53. QUORA. Quora Plus. **Quora**. Disponível em: <<https://www.quora.com/quoraplus>>. Acesso em: 02 Março 2022.
54. OFFICIAL QUORA ACCOUNT. Qual é a política do Quora para ocultar respostas? Quais tipos de respostas e revisões não são permitidos no Quora? **Quora Help Center**, 2018. Disponível em: <<https://help.quora.com/hc/pt/articles/360000470346>>. Acesso em: 02 Março 2022.
55. SCRAPY at a glance. **Scrapy 2.6.1 documentation**. Disponível em:

<<https://docs.scrapy.org/en/latest/intro/overview.html>>. Acesso em: 02 Fevereiro 2022.

56. MONGODB. MongoDB: A Plataforma De Aplicação De Dados. **MongoDB**. Disponível em: <<https://www.mongodb.com/pt-br>>. Acesso em: 14 Março 2022.

57. PYMONGO 4.1.1 Documentation. **PyMongo 4.1.1 documentation**. Disponível em: <<https://pymongo.readthedocs.io/en/stable/>>. Acesso em: 10 Abril 2022.

58. TWISTED. **Twisted**, 2022. Disponível em: <<https://twistedmatrix.com/>>. Acesso em: 09 Maio 2022.

59. FEDERAL Trade Commission. **USAGov**. Disponível em: <<https://www.usa.gov/federal-agencies/federal-trade-commission>>. Acesso em: 07 Maio 2022.

ANEXOS

ANEXO A – TAGS SELECIONADAS PELO ESPECIALISTA SOBRE HIV QUE SERVIRAM COMO BASE PARA A DEFINIÇÃO DA COLETA DE DADOS



UNIVERSIDADE
DO BRASIL
UFRJ

FACULDADE DE FARMÁCIA

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, Prezcoibx, 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.

Universidade Federal do Rio de Janeiro – Departamento de Fármacos e Medicamentos - Faculdade de Farmácia, Ilha do Fundão, Prédio do CCS, Laboratório de Tecnologia Industrial Farmacêutica (LabTIF), Bloco L, Subsolo, sala 20. CEP: 21.941-590.

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