UNIVERSIDADE FEDERAL FLUMINENSE

MARCELLO TELLES DOS SANTOS

YASMIM DE LIMA MARTINS

WEB SCRAPING E TEXT MINING: UMA SOLUÇÃO PARA CAPTURA DOS COMPONENTES UTILIZADOS PELOS PRINCIPAIS MEDICAMENTOS CATALOGADOS

Niterói

2020MARCELLO TELLES DOS SANTOS

YASMIM DE LIMA MARTINS

WEB SCRAPING E TEXT MINING: UMA SOLUÇÃO PARA CAPTURA DOS COMPONENTES UTILIZADOS PELOS PRINCIPAIS MEDICAMENTOS CATALOGADOS

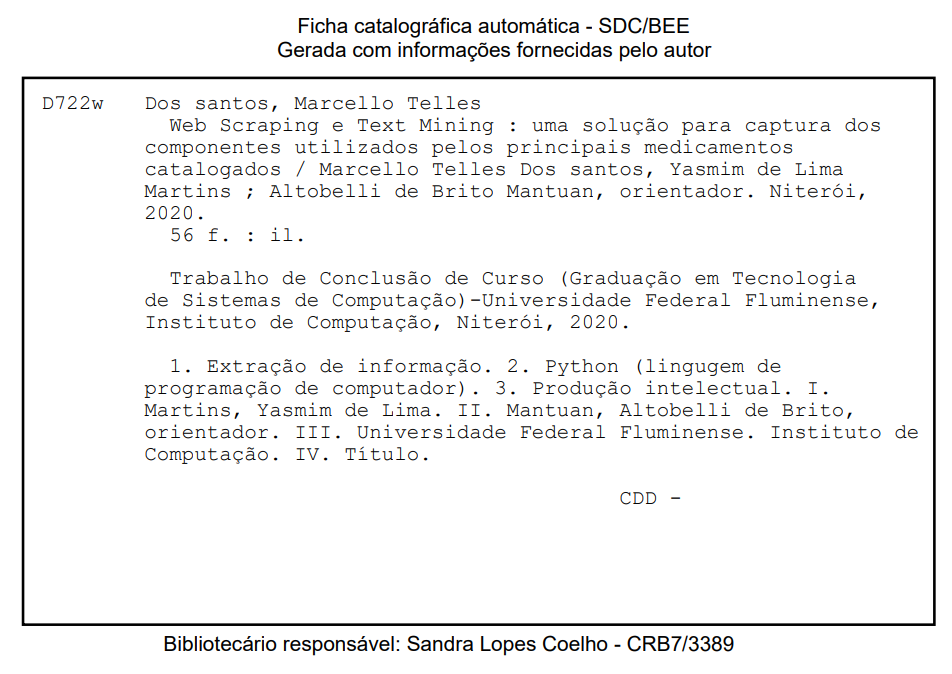
Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Orientador(a):

ALTOBELLI DE BRITO MANTUAN

NITERÓI

2020



MARCELLO TELLES DOS SANTOS

YASMIM DE LIMA MARTINS

WEB SCRAPING E TEXT MINING: UMA SOLUÇÃO PARA CAPTURA DOS COMPONENTES UTILIZADOS PELOS PRINCIPAIS MEDICAMENTOS CATALOGADOS

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

Niterói, \_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ de 2020.

Banca Examinadora:

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

Prof. Altobelli de Brito Mantuan, MSc. – Orientador

UFF – Universidade Federal Fluminense

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

Prof. Kauê Francisco Corrêa de Souza e Souza, Dr. Ciências e Biotecnologia

FIOCRUZ – Fundação Oswaldo Cruz

AGRADECIMENTOS

MARCELLO

A minha família, principalmente minha mãe, Simone, pelo constante apoio, preocupação e por sempre ter sido uma fonte de inspiração ao longo da minha vida.

A minha namorada, Renata, pelo estímulo para que eu fizesse um curso de graduação, e pela inspiração e apoio ao longo da jornada.

Ao meu orientador Altobelli de Brito Mantuan pelos direcionamentos e apoio concedido na confecção deste trabalho.

Aos colegas de curso pela troca ao longo dos últimos 3 anos, que enriqueceram muito a experiência de aprendizado, em especial a minha companheira de trabalho Yasmim, que foi exemplo em dedicação a este trabalho.

YASMIM

A minha família por todo apoio e incentivo durante toda minha vida e em especial a minha mãe Janaina (*in memoriam*), que como educadora sempre batalhou pelos meus estudos e continua sendo minha força e alicerce. Sua lembrança me inspira e me faz persistir.

Ao meu noivo, Willer, que sempre me apoiou e acreditou em mim. Agradeço por todo amor, cuidado e suporte.

Ao meu colega Marcello e ao meu orientador Altobelli de Brito Mantuan pela dedicação, ensinamentos e compreensão durante a trajetória desse trabalho.

RESUMO

A obtenção de dados referentes a excipientes utilizados em medicamentos para uma pesquisa farmacêutica é um tipo de atividade que demanda tempo e esforço elevados, comparando ao método de extração manual. A partir do desenvolvimento dos *bots*, é possível coletar esses dados de forma automatizada através da técnica de *web scraping* e realizar a interpretação dos dados através da técnica de *text mining*. Neste estudo, expomos a implementação de um robô que simula as ações humanas para obter os dados relevantes à pesquisa do medicamento de interesse. Os requisitos do projeto foram levantados junto a um pesquisador farmacêutico especialista que indicou o procedimento e as fontes de dados utilizadas por eles para a extração das informações. O sistema desenvolvido foi testado e os resultados obtidos foram validados pelo pesquisador considerando o objetivo de definir os pontos de interesse a serem capturados.

Palavras-chaves: web scraping, text mining, excipiente, extração automática de dados e pesquisa farmacêutica.

ABSTRACT

Obtaining data about excipients utilized in medicines for a pharmaceutical research is a kind of activity that spends elevated time and requires high efforts, considering the manual extraction method. From the development of bots, it is possible to gather these data in an automated way, through web scraping techniques and the data interpretation with text mining. In this study, we expose the implementation of a robot that simulates the human actions to obtain relevant data on the research of the medicine of interest. The project requirements were raised together with a specialist pharmaceutical researcher who pointed the procedure and the data sources used by them to extract the information. The system developed was tested and the obtained results were validated by the researcher, considering the objective in define the points of interest to be captured.

Key words: web scraping, text mining, excipient, automatic data extraction e pharmaceutical research.

LISTA DE ILUSTRAÇÕES

[Figura 1: Dados não estruturados, semiestruturados e estruturados 20](#_Toc44782573)

[Figura 2: Estrutura básica de um documento HTML 23](#_Toc44782574)

[Figura 3: Visualização de um documento HTML em um navegador 23](#_Toc44782575)

[Figura 4: Estrutura básica de um documento HTML representado como texto 25](#_Toc44782576)

[Figura 5: Representação da estrutura básica de um documento HTML 25](#_Toc44782577)

[Figura 6: Diagrama do caso de uso Scrap Anvisa 27](#_Toc44782578)

[Figura 7: Diagrama do caso de uso Text Mining Anvisa 29](#_Toc44782579)

[Figura 8: Diagrama do caso de uso Text Mining Handbook 31](#_Toc44782580)

[Figura 9: Diagrama de classes do projeto 35](#_Toc44782581)

[Figura 10: Fluxograma dos módulos da extração do dado 36](#_Toc44782582)

[Figura 11: Fluxograma de atividades para obter as bulas do profissional. 37](#_Toc44782583)

[Figura 12: Exemplo de lista de sugestões de nomes completos do fármaco. 37](#_Toc44782584)

[Figura 13: Fluxograma de atividades para extração dos excipientes. 39](#_Toc44782585)

[Figura 14: Exemplo de bula contemplada no CASO 1. 40](#_Toc44782586)

[Figura 15: Exemplo de bula contemplada no CASO 2. 40](#_Toc44782587)

[Figura 16: Exemplo de bula contemplada no CASO 3. 40](#_Toc44782588)

[Figura 17: Exemplo de bula contemplada no CASO 4. 41](#_Toc44782589)

[Figura 18: Exemplo de bula contemplada no CASO 5. 41](#_Toc44782590)

[Figura 19: Exemplo de bula contemplada no CASO 6. 41](#_Toc44782591)

[Figura 20: Exemplo de bula contemplada no CASO 7. 42](#_Toc44782592)

[Figura 21: Exemplo de bula contemplada no CASO 8. 42](#_Toc44782593)

[Figura 22: Exemplo de bula contemplada no CASO 9. 42](#_Toc44782594)

[Figura 23: Exemplo de bula não contemplada. 42](#_Toc44782595)

[Figura 24: Fluxograma de atividades para extração de informações do excipiente. 43](#_Toc44782596)

[Figura 25: Comando para execução do primeiro módulo 46](#_Toc44782597)

[Figura 26: *Log* de execução do primeiro módulo 46](#_Toc44782598)

[Figura 27: Comando para execução do segundo módulo 46](#_Toc44782599)

[Figura 28: *Log* de execução do segundo módulo 1 46](#_Toc44782600)

[Figura 29: *Log* de execução do segundo módulo 2 47](#_Toc44782601)

[Figura 30: *Log* de execução do segundo módulo 3 47](#_Toc44782602)

[Figura 31: Comando para execução do módulo de metadados 47](#_Toc44782603)

[Figura 32: Visão de bulas dos metadados de execução 48](#_Toc44782604)

[Figura 33: Visão de excipientes dos metadados de execução 48](#_Toc44782605)

LISTA DE TABELAS

[Tabela 1: Descrição do caso de uso Scrap Anvisa 28](#_Toc44633515)

[Tabela 2: Descrição do caso de uso Text Mining Anvisa 31](#_Toc44633516)

[Tabela 3: Descrição do caso de uso Handbook 32](#_Toc44633517)

[Tabela 4: Especificações utilizadas nos testes 46](#_Toc44633518)

LISTA DE ABREVIATURAS E SIGLAS

CSS - *Cascading Style Sheets*

CSV - *Comma-separated Values*

FDA – *U.S. Food and Drug Administration*

FIPE – Fundação Instituto de Pesquisas Econômicas

HTML - *Hypertext Markup Language*

HTTP - *Hypertext Transfer Protocol*

JS – JavaScript

JSON - *JavaScript Object Notation*

PDF - *[Portable Document Format](https://pt.wikipedia.org/wiki/Portable_Document_Format)* [SQL –](https://pt.wikipedia.org/wiki/Portable_Document_Format) *[Structured Query Language](https://pt.wikipedia.org/wiki/Portable_Document_Format)*

[URL –](https://pt.wikipedia.org/wiki/Portable_Document_Format) *[Uniform Resource Locator](https://pt.wikipedia.org/wiki/Portable_Document_Format)*

WSL – *Windows Subsystems for Linux*

XML *- Extensible Markup Language*

SUMÁRIO

[RESUMO 8](#_Toc44620830)

[ABSTRACT 9](#_Toc44620831)

[LISTA DE ILUSTRAÇÕES 10](#_Toc44620832)

[LISTA DE TABELAS 11](#_Toc44620833)

[LISTA DE ABREVIATURAS E SIGLAS 12](#_Toc44620834)

[1 INTRODUÇÃO 15](#_Toc44620835)

[2 TRABALHOS RELACIONADOS 16](#_Toc44620836)

[2.1 DISCUSSÃO 17](#_Toc44620837)

[3 FUNDAMENTAÇÃO TEÓRICA 18](#_Toc44620838)

[3.1 COLETA DE DADOS 18](#_Toc44620839)

[3.2 TIPOS DE DADOS 18](#_Toc44620840)

[3.2.1 QUALITATIVOS 19](#_Toc44620841)

[3.2.2 QUANTITATIVOS 19](#_Toc44620842)

[3.2.3 PRIMÁRIOS 19](#_Toc44620843)

[3.2.4 SECUNDÁRIOS 20](#_Toc44620844)

[3.3 ORGANIZAÇÃO E ARMAZENAMENTO 20](#_Toc44620845)

[3.3.1 ESTRUTURADOS 20](#_Toc44620846)

[3.3.2 NÃO ESTRUTURADOS 21](#_Toc44620847)

[3.3.3 SEMIESTRUTURADOS 21](#_Toc44620848)

[3.4 FUNCIONAMENTO DAS PÁGINAS WEB 21](#_Toc44620849)

[3.4.1 HISTÓRIA DA INTERNET 21](#_Toc44620850)

[3.4.2 NAVEGADORES E PÁGINAS WEB 22](#_Toc44620851)

[3.5 WEB SCRAPING 24](#_Toc44620852)

[3.5.1 BUSCANDO INFORMAÇÕES EM UM DOCUMENTO HTML 24](#_Toc44620853)

[4 IMPLEMENTAÇÃO 27](#_Toc44620854)

[4.1 CASOS DE USO 27](#_Toc44620855)

[4.1.1 DIAGRAMA DO CASO DE USO SCRAP ANVISA 27](#_Toc44620856)

[4.1.2 DESCRIÇÃO DO CASO DE USO SCRAP ANVISA 28](#_Toc44620857)

[4.1.3 DIAGRAMA DO CASO DE USO TEXT MINING ANVISA 30](#_Toc44620858)

[4.1.4 DESCRIÇÃO DO CASO DE USO TEXT MINING ANVISA 30](#_Toc44620859)

[4.1.5 DIAGRAMA DE CASO DE USO HANDBOOK 32](#_Toc44620860)

[4.1.6 DESCRIÇÃO DO CASO DE USO HANDBOOK 32](#_Toc44620861)

[4.2 TECNOLOGIAS 34](#_Toc44620862)

[4.3 DIAGRAMA DE CLASSES 35](#_Toc44620863)

[4.3.1 DESCRIÇÃO DAS CLASSES 35](#_Toc44620864)

[4.4 FONTES 37](#_Toc44620865)

[4.4.1 ANVISA 37](#_Toc44620866)

[4.4.1.1 DOWNLOAD DAS BULAS DO PROFISSIONAL 37](#_Toc44620867)

[4.4.1.2 ANÁLISE DAS BULAS E EXTRAÇÃO DOS EXCIPIENTES 40](#_Toc44620868)

[4.4.2 HANDBOOK OF PHARMACEUTICAL EXCIPIENTS 43](#_Toc44620869)

[5 TESTES 46](#_Toc44620870)

[CONCLUSÕES E TRABALHOS FUTUROS 47](#_Toc44620871)

[REFERÊNCIAS BIBLIOGRÁFICAS 48](#_Toc44620872)

[ANEXOS 51](#_Toc44620873)

# INTRODUÇÃO

Os novos medicamentos passam por um processo extenso e trabalhoso para sua pesquisa e desenvolvimento. Dentre várias etapas para a produção, é necessário decidir o excipiente a ser utilizado para se obter a forma desejada, tais como cápsula, líquido e comprimido. A substância escolhida completa a massa ou o volume e deve-se considerar se serve de veículo para o princípio ativo, se é inerte e se não reage com o fármaco. Uma boa prática para a escolha é a análise de excipientes já utilizados em medicamentos com o mesmo princípio ativo. Assim é possível garantir a segurança, potencialização do efeito terapêutico e a minimização dos erros farmacotécnicos, favorecendo a eficácia e qualidade do medicamento.

Contudo, a averiguação manual de formulações catalogadas é custosa e demanda muito tempo do pesquisador. É necessário coletar os excipientes aplicados nas numerosas bulas disponíveis, e em seguida verificar o uso mais comum de cada substância de forma exata e precisa. Analisando as dificuldades dessa pesquisa, foi debatido o uso das técnicas de *web scraping* e *text mining* para coletar de forma automatizada as informações de interesse nas fontes da web e em documentos previamente conhecidos.

Este estudo apresenta um projeto de desenvolvimento de um sistema para obtenção de dados de excipientes presentes em medicamentos catalogados, afim de apoiar as decisões e análises de pesquisadores farmacêuticos. O programa realiza a extração de diversos dados em poucos minutos através das técnicas supracitadas. A linguagem de programação escolhida foi Python, já que esta possui diversas bibliotecas que auxiliam na construção do robô. O objetivo principal foi direcionar a extração dos excipientes, bem como sua formulação e proporção, nas fontes confiáveis e previamente definidas, identificando as zonas de interesse.

Está disponível em um repositório público no GitHub [35] o conteúdo e código-fonte referente ao trabalho em questão, no seguinte endereço: https://github.com/altobellibm/CEDERJ\_2020\_YASMIM\_MARCELLO

# TRABALHOS RELACIONADOS

Visando o entendimento e análise da importância da implementação de técnicas de coletas de dados, foram ponderados artigos cujo o conceito principal é análogo ao do presente trabalho.

O estudo [1] de autoria de Carolina Franco Zanon, apresenta a criação de um sistema, utilizando a técnica *Web Scraping*, para auxiliar na definição do perfil epidemiológico dos pacientes da Clínica de Estomatologia da FO-UFRJ. Os dados foram coletados de um sistema já existente denominado Estomato Web, um *software* incapaz de fornecer estatísticas ou exportar grande quantidade de dados de uma única vez. A técnica foi implementada fazendo a coleta dos conjuntos de dados e transferindo-os para planilhas do Excel. Foi possível agrupar os dados por categorias, analisando detalhadamente o perfil dos pacientes e concluindo que a utilização do novo sistema, chamado Estomato Web Scraper, facilitou a extração e qualidade dos dados, e, portanto, está credenciado a ser utilizado para diversos estudos na área de estomatologia.

Um segundo artigo [2], que tem Dhaniel Nunes Mazini e Renato Cesar Sato como autores, e abordou uma análise da rentabilidade dos dividendos das empresas que compõem o IBrX 50. O artigo demonstra a implementação de um *Web Scraper* a partir da linguagem Python [29], sendo todas as informações extraídas do site Yahoo!Finanças [4], onde os dados não estão disponíveis para uma coleta em massa. As análises feitas para a área do mercado financeiro têm como característica uma grande massa de dados, fazendo-se necessária extração em diferentes fontes. No entanto, a utilização da tecnologia proposta automatizou a coleta de dados em uma escala de tempo reduzida, sendo os elementos obtidos armazenados em um arquivo que favoreça a manipulação e possibilite a realização de uma análise de regressão múltipla para atingir o objetivo proposto pelo estudo.

Por fim, o trabalho [3] de autoria de Thiago da Cunha Borges e Zeus Olenchuk Ganimi discorre sobre a implementação de um *Web Scraper* para capturar dados de valores de automóveis, extraídos do site da FIPE (Fundação Instituto de Pesquisas Econômicas). A ferramenta permite a coleta automatizada dos dados semiestruturados do módulo de consulta da página da FIPE e persistência das informações estruturadas em banco de dados PostgreSQL [30]. A análise do conteúdo estruturado pode ser utilizada, por exemplo, para otimização de recursos de empresas adquirentes de veículos, e sua visualização foi possibilitada através do software de Tableau Desktop [31].

A partir dos trabalhos supracitados é possível analisar que existem diversas aplicações da tecnologia de *Web Scraping* para a obtenção de dados. Constatou-se que a implementação da técnica é capaz de auxiliar a automação, assiduidade e estudo dos dados obtidos, além de dar suporte para distintas áreas de estudo.

Uma das áreas na qual foi identificada a oportunidade de automatização é a farmacêutica. Parte do procedimento de proposta de um novo fármaco envolve a consulta às substâncias que compõe o fármaco em suas diversas composições, tendo como parâmetro de busca o princípio ativo do medicamento. Não há um veículo de comunicação oficial que permita aos pesquisadores acompanharem essas tendências de utilização de ingredientes eficientemente, de forma que as informações sejam extraídas de sites especializados da área.

Nesse contexto, a aplicação de *Web Scraping* para a extração das informações de sites farmacêuticos especializados através da busca por elementos-chave agiliza a etapa de criação de novos medicamentos, favorecendo a elaboração de soluções rápidas e eficientes contra possíveis patógenos infecciosos, assim como os agentes recém descobertos e que afligem a nossa sociedade.

# FUNDAMENTAÇÃO TEÓRICA

Com objetivo de disponibilizar embasamento teórico para uma acertada análise e interpretação do presente estudo, neste capítulo serão abordados os principais conceitos e definições encontrados na literatura: tipos de dados, técnicas de coleta e de análise e armazenamento.

## COLETA DE DADOS

Ao iniciar uma pesquisa é necessário um estudo intensivo sobre o conjunto de dados que possam gerar informações relevantes para uma análise. A obtenção desses dados possibilita um posterior exame pormenorizado de diferentes fontes para o estudo da área de interesse. [18]

De acordo com a finalidade da pesquisa e tipo de dado, a coleta poderá ser realizada através de questionários, entrevistas, observações, experimentos ou estudo e revisões de documentos já disponíveis, além de poder ser necessária a utilização de mais de um método para explorar com profundidade as informações desejadas. No mesmo sentido, cabe realçar que independentemente do tipo de coleta utilizado, a prioridade será a garantia da precisão, coerência e confiabilidade nas informações obtidas.

## TIPOS DE DADOS

Para compreendermos melhor o desenvolvimento do trabalho, que será descrito no capítulo a seguir, é importante definirmos com clareza os tipos de dados que foram definidos até hoje. Estes dados são divididos em dois grupos, quanto a sua característica: qualitativos e quantitativos, e em dois outros grupos, quanto a sua origem: primários e secundários.

### QUALITATIVOS

Os dados qualitativos possuem a função de rotular e atribuir uma identificação ao objeto analisado. Como exemplos de dados qualitativos, temos: nível de escolaridade, estado civil ou nome [7].

### QUANTITATIVOS

Os dados quantitativos são numéricos e, como o nome sugere, possuem o objetivo de quantificar as amostras, além de serem amplamente utilizados em modelos estatísticos. São considerados discretos, quando assumem valores dentro de um intervalo enumerável (por exemplo, minutos inteiros em uma hora), e contínuos, quando assumem valores reais que não podem ser enumerados [7].

### PRIMÁRIOS

Os dados denominados primários são aqueles coletados diretamente de sua fonte. A obtenção de dados primários é planejada especialmente para o estudo de interesse e, portanto, são mais adequados aos seus objetivos. Em contrapartida, o elevado esforço para a coleta de dados o torna desvantajoso. São exemplos de dados primários as respostas de entrevistas e questionários concedidas diretamente pelo entrevistado [7].

### SECUNDÁRIOS

Dados secundários são obtidos a partir de coletas previamente realizadas. Por já terem sido publicados, os dados secundários comumente não possuem o formato ou a integridade adequados ao estudo. Desta forma, embora o esforço seja menor na obtenção dos dados, em comparação aos dados primários, os dados secundários requerem maiores esforços nas etapas posteriores definição de fontes e na metodologia empregada em sua manipulação [7].

## ORGANIZAÇÃO E ARMAZENAMENTO

Posterior a coleta dos dados, os mesmos precisam ser armazenados de acordo com a sua organização que pode ser estruturada, não estruturada ou semiestruturada. É priorizada formas acessíveis e a escolha leva em consideração a análise e nível de controle necessário dos dados.

### ESTRUTURADOS

São dados que possuem um formato específico e bem definido. Nesse tipo de organização, cada registro possui características obrigatórias que devem ser respeitadas (figura 1) [7].



Figura 1: Dados não estruturados, semiestruturados e estruturados

### NÃO ESTRUTURADOS

São dados que não possuem estrutura fixa, como arquivos de texto comuns, cujo conteúdo pode conter inúmeras variações no formato, idioma, formatação, entre outros (figura 1) [7].

### SEMIESTRUTURADOS

Os dados semiestruturados apresentam um avanço com relação aos dados não estruturados, pois são dotados de regras que orientam a organização do seu conteúdo, mesmo ainda possuindo considerável liberdade com relação à composição como um todo (figura 1) [7].

## FUNCIONAMENTO DAS PÁGINAS WEB

A técnica de coleta de dados proposta no presente trabalho terá como base fundamental a navegação através da estrutura de sites para a captura dos dados desejados. Portanto, é de suma importância a compreensão dos conceitos envolvendo as páginas *web* e os navegadores que conhecemos.

### HISTÓRIA DA INTERNET

As motivações que culminaram na estruturação da rede que hoje conhecemos como internet datam-se nas primeiras décadas da Guerra Fria. No ano de 1962, os militares americanos, com o apoio do governo, concluíram o desenvolvimento da rede que ficou conhecida como ARPANET. Concebida como uma resposta ao Projeto Sputnik da URSS, lançado alguns anos antes, a rede era limitada e foi utilizada principalmente para a comunicação entre universidades de ponta e institutos de pesquisa da época [5].

Até o final da década de 80, a Net foi majoritariamente um meio de comunicação entre militares e acadêmicos, porém não mais para fins de defesa dos EUA [5]. Em 1979, concretizou-se a visão comercial da rede através do primeiro vendedor de serviços on-line americano. Dez anos depois, o pesquisador Tim Berners-Lee, que posteriormente ficou conhecido como o pai da internet, imaginou o que chamou de *World Wide Web* (ou apenas web). Os alicerces da proposta de Berners-Lee foram os mesmos da rede de ampla utilização que conhecemos hoje. Segundo KRAEMER [5], ao final do século XX, estabeleceu-se a tríade: informação, educação e entretenimento na web, porém, novas possibilidades de uso ainda seguem sendo exploradas até os dias de hoje.

### NAVEGADORES E PÁGINAS WEB

É de suma importância a compreensão da forma básica das páginas estruturadoras dos websites que compõe a web, além da responsabilidade dos navegadores na interpretação do conteúdo das mesmas. Todo site, independente da natureza de seu conteúdo, é composto de uma ou mais páginas que são relacionadas entre si através de *links* (referências) [11]. Estejam aplicadas as devidas práticas de desenvolvimento *web*, frequentemente a navegação entre as páginas é fluida o suficiente para que os usuários percebam um único contexto em todas as páginas que compõe o mesmo site [11].

Toda as páginas *web* são arquivos compostos por dados semiestruturados, mais precisamente através de uma linguagem de marcação denominada *HyperText Markup Language* – HTML [8]. A estrutura do HTML é reconhecível através das marcações realizadas através das *tags* que, por sua vez, podem agrupar outras *tags* diversas. Os dois principais agrupamentos representam o cabeçalho e o corpo do documento (figura 2).



Figura 2: Estrutura básica de um documento HTML

Dadas as características das páginas que compõe a internet, entra em evidência outro elemento também essencial ao funcionamento da web: os navegadores. Os navegadores são interpretadores dos diversos elementos que podem compor as páginas, como imagens, folhas de estilo CSS [13], scripts em Javascript [14] e a estrutura HTML previamente citada. Dentre os navegadores mais conhecidos estão o Google Chrome [15], Mozilla Firefox [16] e *Internet Explorer* [17].

O resultado da estruturação das páginas após a interpretação do navegador (Figura 3) a partir da estrutura básica de um documento (Figura 2) mostra alguns comentários adicionados no documento que foram ignorados pelo navegador, conforme citado anteriormente.



Figura 3: Visualização de um documento HTML em um navegador

Apesar dos navegadores serem a forma mais conhecida e utilizada pelos usuários para acessar páginas *web*, outros programas de computador também podem requisitar e interpretar as mesmas páginas, como é o caso de algumas bibliotecas e *frameworks* de linguagens de programação. A partir destas implementações, é possível extrair o código HTML de uma página web através de poucas linhas de programação.

Dadas as características de páginas *web* previamente citadas, além das possibilidades relacionadas ao seu consumo, seja através de navegadores ou outros meios, existem diversas técnicas empregadas na análise, interpretação, busca e recuperação de dados contidos em páginas *web*.

## WEB SCRAPING

A coleta de dados da *web* moderna, conhecida como *Web Scraping* [18], é uma prática que utiliza técnicas de programação e tecnologias. É baseada no desenvolvimento de um *software* capaz de se comunicar com navegadores e simular, de forma automatizada, a interação de busca de dados de um usuário. O programa, também conhecido como *bot* ou *web crawler*, consulta servidores *web* fazendo uma requisição HTTP e, em seguida, faz uma análise minuciosa pela estrutura do documento HTML recebido afim de retornar o fragmento onde o dado de interesse foi encontrado.

### BUSCANDO INFORMAÇÕES EM UM DOCUMENTO HTML

Um documento HTML é um arquivo de texto simples e possui uma estrutura definida, portanto é possível aplicar técnicas de busca no mesmo. Há várias formas de fazer a busca, como por expressões regulares que formam um padrão de texto, por palavras-chave encontradas no documento e utilizando a estrutura do documento.



Figura 4: Estrutura básica de um documento HTML representado como texto

A partir da estrutura do HTML é viável a utilização de ferramentas que auxiliem a extração de dados no documento como a biblioteca *Beautiful Soup* [19]. Essa biblioteca fornece métodos que facilitam a navegação e pesquisa do conteúdo fazendo o *parsing* dos sites.

O *parsing*, que pode ser compreendido como um tradutor, é uma análise sintática onde lê-se uma entrada e monta-se uma estrutura de árvore da sua composição. Inicialmente o *parsing* faz uma análise léxica do documento tornando sequência de caracteres em palavras-chaves que podem ser analisadas futuramente. Em seguida é feito uma análise semântica conferindo se está em conformidade com as regras de composição de um arquivo HTML. Por fim, é gerado um código que pode ser utilizado para execução de comandos levando em consideração a árvore obtida (Figura 5) ou para responder uma saída de interesse, sendo possível extrair o título do documento representado na Figura 4 a partir do seguinte caminho: html; head; title.



Figura 5: Representação da estrutura básica de um documento HTML

# IMPLEMENTAÇÃO

Conforme definições realizadas em reunião que contou com a presença de um farmacêutico pesquisador na área de interesse do presente trabalho (detalhada no Anexo A), foram levantados os requisitos necessários para a implementação da técnica de *web scraping*, visando a automatização da busca e coleta de dados, em fontes específicas, relacionados aos princípios ativos de determinados medicamentos e seus demais constituintes.

O assunto deste capítulo serão os detalhes técnicos da implementação aplicada, incluindo: diagrama e descrição dos casos de uso observados, informações sobre a tecnologia optada, um diagrama de classes provendo uma visão geral da arquitetura implementada e a definição das fontes de dados a serem exploradas, conforme orientação do especialista farmacêutico consultado.

## CASOS DE USO

Os casos de uso apresentam as principais interações dos usuários com o sistema afim de facilitar a comunicação entre desenvolvedor e cliente. Nesta seção serão apresentados os diagramas e as descrições dos casos de uso para cada fonte utilizada na implementação.

### DIAGRAMA DO CASO DE USO SCRAP ANVISA



Figura 6: Diagrama do caso de uso Scrap Anvisa

### DESCRIÇÃO DO CASO DE USO SCRAP ANVISA

Tabela 1: Descrição do caso de uso Scrap Anvisa

|  |  |
| --- | --- |
| **ID:** | **UC01** |
| **Objetivo:** | Extrair bulas dos profissionais do bulário Anvisa. |
| **Requisitos:** | Ter uma conexão ativa à Internet. |
| **Atores:** | Usuário. |
| **Pré-condições:** | As dependências (compilador/interpretador da linguagem de programação, *framework* e bibliotecas auxiliares) devem estar instaladas no ambiente. |
| **Pós-condições:** | Os arquivos obtidos deverão estar salvos em disco. |
| **Fluxo principal:** | ***Scrap* Anvisa**   1. O ator executa a ferramenta de linha de comando, fornecendo os parâmetros de busca para a fonte Anvisa; 2. A ferramenta recupera os resultados da base de dados Anvisa, conforme o filtro passado como parâmetro; 3. A ferramenta gera um arquivo contendo o histórico da execução e os arquivos de bulas dos resultados da busca no disco; |
| **Fluxo alternativo:** | Não há. |
| **Erros/Exceções:** | O tratamento de erros (como requisições inválidas e timeouts) é feito pelo *framework* utilizado (mensagens de erro podem ser exibidas para o usuário).  Em caso de falha(s) na execução, as pós-condições não serão verdadeiras (os arquivos de resultado não serão obtidos). |
| **Mensagens:** | *Logs* de saída contendo detalhes de execução da ferramenta. |
| **Regras de negócio:** | **[RN01]** O endereço <http://www.anvisa.gov.br/datavisa/fila_bula/frmResultado.asp> deve aceitar requisições com o método HTTP POST, contendo os seguintes parâmetros e seus respectivos significados:   * “txtMedicamento”: nome do princípio ativo * “txtEmpresa“: empresa registrante * “txtNuExpediente”: número de expediente * “txtDataPublicacaoI”: filtro de data inicial de publicação * “txtDataPublicacaoF”: filtro de data final de publicação * “txtPageSize”: número de registros por página   Nenhum dos parâmetros listados acima é obrigatório. Caso o parâmetro “txtPageSize” não seja especificado, o resultado trará 10 registros por página.  **[RN02]** O resultado da requisição descrita em **RN01** deverá ser uma página HTML válida, contendo um elemento do tipo “*table*” com ID “tblResultado”, que representa a listagem de resultados da busca pelo medicamento. Além disso, a tabela citada deverá conter um elemento do tipo cabeçalho de tabela (“th”) contendo o texto “Bula do Profissional”. A célula (“td”) posicionada na mesma coluna do cabeçalho citado, deverá conter um elemento de *hyperlink* (“a”) com o seu atributo “*onclick*” fazendo uma chamada de função Javascript denominada “fVisualizarBula” e que recebe dois parâmetros no formato de texto, mas que são valores numéricos válidos.  **[RN03]** O endereço <http://www.anvisa.gov.br/datavisa/fila_bula/frmVisualizarBula.asp> deve aceitar requisições com o método HTTP POST, contendo os seguintes parâmetros e seus respectivos significados:   * “pNuTransacao”: número de transação * “pIdAnexo”: identificador numérico do anexo   **[RN04]** O resultado da requisição descrita em **RN03** deverá ser um arquivo PDF válido, que representa a bula do profissional. |

### DIAGRAMA DO CASO DE USO TEXT MINING ANVISA



Figura 7: Diagrama do caso de uso Text Mining Anvisa

### DESCRIÇÃO DO CASO DE USO TEXT MINING ANVISA

Tabela 2: Descrição do caso de uso Text Mining Anvisa

|  |  |
| --- | --- |
| **ID:** | **UC02** |
| **Objetivo:** | Extrair as informações de formulação e excipientes das bulas dos profissionais. |
| **Requisitos:** | Ter ao menos um arquivo de bula do profissional salvo no diretório esperado do disco local do usuário. |
| **Atores:** | Usuário. |
| **Pré-condições:** | As dependências (compilador/interpretador da linguagem de programação, *framework* e bibliotecas auxiliares) devem estar instaladas no ambiente. |
| **Pós-condições:** | Os arquivos com os resultados da interpretação deverão estar salvos em disco. |
| **Fluxo principal:** | ***Text Mining* Anvisa**   1. O ator executa a ferramenta de linha de comando; 2. A ferramenta busca nas bulas existentes pelos pontos de interesse delimitados;   A ferramenta gera os arquivos de resultados da busca por pontos de interesse nas bulas no disco; |
| **Fluxo alternativo:** | Não há. |
| **Erros/Exceções:** | O tratamento de erros (como requisições inválidas e timeouts) foi implementado (mensagens de erro podem ser exibidas para o usuário).  Em caso de falha(s) na execução, as pós-condições não serão verdadeiras (os arquivos de resultado não serão obtidos). |
| **Mensagens:** | *Logs* de saída contendo detalhes de execução da ferramenta. |
| **Regras de negócio:** | **[RN01]** A bula deve ser um arquivo no formato PDF.  **[RN02]** A bula deve possuir uma ou mais seções delimitadas iniciando na palavra “composição” e terminando em “informações técnicas” ou “indicações”.  **[RN03]** A seção de composição da bula deve possuir os dados de formulação entre as palavras “cada” e “contém” ou “contem  **[RN04]** A seção de composição da bula deve possuir os dados de excipientes compatíveis com os casos atendidos descritos em [#ANÁLISE DAS BULAS E EXTRAÇÃO DOS EXCIPIENTES](#_ANÁLISE_DAS_BULAS) |

### DIAGRAMA DE CASO DE USO HANDBOOK

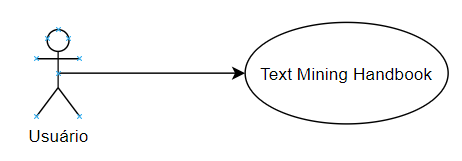


Figura 8: Diagrama do caso de uso Text Mining Handbook

### DESCRIÇÃO DO CASO DE USO HANDBOOK

Tabela 3: Descrição do caso de uso Handbook

|  |  |
| --- | --- |
| **ID:** | **UC03** |
| **Objetivo:** | Extrair as informações de proporções e cenários de uso dos excipientes do livro especializado. |
| **Requisitos:** | Ter o arquivo PDF do livro especializado salvo em disco no diretório esperado, ter uma conexão à internet ativa e ter ao menos um arquivo de resultado de interpretação da bula no diretório esperado. |
| **Atores:** | Usuário. |
| **Pré-condições:** | As dependências (compilador/interpretador da linguagem de programação, *framework* e bibliotecas auxiliares) devem estar instaladas no ambiente. |
| **Pós-condições:** | Os arquivos com os resultados da interpretação deverão estar salvos em disco. |
| **Fluxo principal:** | ***Text Mining Handbook***   1. O ator executa a ferramenta de linha de comando; 2. A ferramenta busca no sumário do arquivo do livro especializado em que página deve encontrar cada excipiente; 3. A ferramenta traduz os nomes dos excipientes salvos nos arquivos de insumo; 4. A ferramenta busca pelo nome dos excipientes traduzidos no livro com base no sumário resultante;   A ferramenta gera os arquivos de resultados da busca por pontos de interesse no livro no disco; |
| **Fluxo alternativo:** | Não há. |
| **Erros/Exceções:** | O tratamento de erros (como requisições inválidas e timeouts) foi implementado (mensagens de erro podem ser exibidas para o usuário).  Em caso de falha(s) na execução, as pós-condições não serão verdadeiras (os arquivos de resultado não serão obtidos). |
| **Mensagens:** | *Logs* de saída contendo detalhes de execução da ferramenta. |
| **Regras de negócio:** | **[RN01]** O arquivo do livro deve ser um arquivo no formato PDF.  **[RN02]** O sumário do livro especializado deve estar contido nas páginas 5 a 9 do arquivo PDF.  **[RN03]** Todo excipiente a ser pesquisado no livro deve conter uma seção iniciando no termo *"7 Applications in Pharmaceutical Formulation or\nTechnology”* e terminando em *“8 Description”*. |

## TECNOLOGIAS

A biblioteca Python Scrapy [20] é um *framework* desenvolvido para extrair dados de páginas web de forma sistematizada. São disponibilizadas diversas funcionalidades para coordenar as requisições HTTP, controlar navegação, fazer o parse em HTML e representar os dados em arquivos com formatos JSON [21], CSV e XML [22].

Um projeto Scrapy tem a arquitetura construída em classes de capturas de dados, conhecidas como *spiders* [24]. Essas classes possuem funcionalidades básicas sobre como extrair informações estruturadas da web, e é nelas que definimos o comportamento específico sobre como capturar e converter as páginas de um site em particular.

Para a tradução dos termos farmacêuticos a serem pesquisados no livro especializado utilizamos a biblioteca Python googletrans [26] e, para suportar a colheita e estruturação dos metadados de execução do projeto, utilizamos a biblioteca Pandas [25].

## DIAGRAMA DE CLASSES

O diagrama de classes do projeto (Figura 9) traduz a arquitetura aplicada, demonstrando a organização em módulos e, quando há, o relacionamento entre as classes utilizadas, com base no framework Scrapy e na biblioteca Pandas [25].

O projeto foi projetado para funcionar através de módulos independentes, de forma que não há comunicação direta entre eles e, portanto, não há relacionamento entre as suas respectivas classes. A sequência de execução se dá através da troca de mensagens e os insumos são recebidos e lidos através de arquivos gravados no próprio sistema de arquivos do computador hospedeiro.

No caso das classes responsáveis por capturar as informações da fonte web, utilizamos o conceito de herança, presente no paradigma de Programação Orientada à Objetos, para herdar todos os atributos e métodos básicos da classe primária e implementamos as especificidades de cada fonte em questão.

### DESCRIÇÃO DAS CLASSES

* **Spider**: componente do *framework* *Scrapy* [20] que já implementa funcionalidades básicas para a captura e conversão de dados oriundos de páginas web.
* **CrawlerRunner**: componente do *framework* *Scrapy* [20] responsável por gerenciar e manter rastreabilidade da execução de diferentes *spiders*.
* **AnvisaAutocompleteSpider:** especificação da classe base *Spider*, cuja responsabilidade é a de capturar as todas as sugestões possíveis de nomes de medicamentos fornecidas pela Anvisa e salvar o resultado em um arquivo texto.
* **AnvisaBularioSpider:** especificação da classe base *Spider*, que recebe um parâmetro de busca em texto do usuário, faz a busca no bulário Anvisa, verificando as sugestões de medicamentos correspondentes, e percorre a lista completa de resultados, baixando todos os PDFs de bula do profissional disponíveis.
* **Stats:** classe que concentra a lógica de verificação dos produtos de processamento das demais classes e estrutura metadados sobre a execução do projeto.
* **BulaParser:** classe responsável por interpretar o contéudo das bulas extraídas do bulário Anvisa, identificar as informações de interesse (formulação e excipientes) e salvar os resultados em arquivos JSON.
* **TranslationManager:** classe responsável por gerenciar as traduções realizadas a partir do conteúdo extraído das bulas, fazendo a interface com a biblioteca googletrans [26].



Figura 9: Diagrama de classes do projeto

## FONTES

Afim de desenvolver a extração do dado corretamente, é executada uma sequência de operações em três módulos (figura 10) que estarão detalhados nesta seção separados pelas fontes utilizadas.



Figura 10: Fluxograma dos módulos da extração do dado

### ANVISA

#### DOWNLOAD DAS BULAS DO PROFISSIONAL

A implementação inicia com uma sucessão de atividades (Figura 11) que possibilitam realizar o download de bulas do profissional do medicamento disponíveis no bulário eletrônico da Anvisa.



Figura 11: Fluxograma de atividades para obter as bulas do profissional.

Detectou-se que para a busca das bulas do medicamento, o site do bulário eletrônico da Anvisa retorna sugestões (Figura 12) conforme o usuário insere o nome do medicamento. Contudo, a correlação entre o nome fornecido e os registros da base de dados do órgão é feita apenas pelas primeiras letras dos medicamentos da base e então não são retornadas as bulas de todas as sugestões, somente são retornadas as bulas das sugestões que apresentam as iniciais descritas no trecho pesquisado.



Figura 12: Exemplo de lista de sugestões de nomes completos do fármaco.

Devido ao cenário citado e para atender ao objetivo de recuperar todos os resultados de bulas conforme o parâmetro fornecido, o script inicia fazendo uma requisição GET para a URL <http://www.anvisa.gov.br/datavisa/fila_bula/funcoes/ajax.asp?opcao=getsuggestion&ptipo=1>, que contém a lista com todas as sugestões de nomes de medicamentos, trata os dados recebidos e salva-os no arquivo “medicamentos.txt”. Assim que o nome do medicamento desejado é inserido, todas as sugestões que o contém são salvas, para garantir que nenhum resultado será ignorado.

Em seguida para cada sugestão é feito uma requisição ao endereço <http://www.anvisa.gov.br/datavisa/fila_bula/frmResultado.asp> passando os seguintes parâmetros através do método POST:

* “txtMedicamento” com uma das sugestões de nomes do medicamento;
* “hddPageSize” contendo o número de registros por página desejado. Conforme alguns testes realizados, e levando em conta relações como o equilíbrio entre o tempo necessário para a consulta e riscos como a qualidade da internet do usuário, optou-se por manter o padrão de 10 registros por página.
* “hddPageAbsolute” contendo o valor 1 para a primeira página e sendo incrementado conforme a necessidade do script de avançar para as próximas páginas.

É retornada a página de resultados que exibe uma página HTML contendo uma tabela que armazena as bulas. Vale ressaltar que a tabela também pode estar vazia. Por fim, com o objetivo de fazer o download de todas as bulas do profissional exibidas na tabela “tblResultado”, faz-se uma busca pela coluna “Bula do Profissional” no cabeçalho das colunas da tabela. Em cada célula da coluna existe a tag <a>, que representa um link HTML, que possui o atributo “onclick” chamando uma função JavaScript denominada “fVisualizarBula”, que fornece dois parâmetros numéricos que são capturados e armazenados para uso posterior. Finalmente é feita uma requisição para <http://www.anvisa.gov.br/datavisa/fila_bula/frmVisualizarBula.asp> passando os parâmetros obtidos no passo anterior, e recebendo o arquivo PDF da bula do profissional como resposta.

#### ANÁLISE DAS BULAS E EXTRAÇÃO DOS EXCIPIENTES

A análise dos PDFs das bulas do profissional requer uma série de etapas, descritas no fluxograma (Figura 13), com objetivo de extrair as formulações e os nomes dos excipientes do medicamento.



Figura 13: Fluxograma de atividades para extração dos excipientes.

A partir das bulas obtidas no bulário eletrônico da Anvisa, inicia-se o processo percorrendo e abrindo cada uma afim de convertê-las de PDF para o arquivo de texto. Para isso, foi utilizada a biblioteca PDFMiner [27] que, em resumo, auxilia na extração e análise dos texto de documentos PDF. Foi necessário a utilização de ferramentas de configuração da biblioteca para manter o *layout* do arquivo original.

Verificou-se que a estrutura do documento das bulas segue um padrão e os excipientes desejados para esse estudo são encontrados no tópico “COMPOSIÇÃO”, portanto foi feita uma busca, com o método find() em Python, pela palavra-chave afim e delimitar o início do texto que contém os dados almejados. E para determinar o fim da seção de composição e otimizar a pesquisa, foi necessário buscar também pelas palavras-chave: “INFORMAÇÕES TÉCNICAS” ou “INDICAÇÕES”, que correspondem aos possíveis títulos utilizados para o próximo tópico da bula.

Com o texto do tópico “COMPOSIÇÃO” em memória, foi possível encontrar a formulação do medicamento uma vez que, analisando o padrão da bula, foi identificado que a mesma aparece entre as palavras “cada” e “contém” ou “contem”.

Apesar de existir um padrão na estrutura dos documentos, foram identificadas diferentes formas de especificar os excipientes nas bulas. Então, por último, foi necessário tratar cada caso de uma maneira e acordado contemplar os seguintes modelos de apresentação da composição dos excipientes nas bulas:

**CASO 1**: Busca-se no intervalo entre a palavra “excipientes:” e o próximo ponto final (Figura 14).



Figura 14: Exemplo de bula contemplada no CASO 1.

**CASO 2**: Busca-se no intervalo entre a palavra “excipiente:” e o próximo ponto final (Figura 15).



Figura 15: Exemplo de bula contemplada no CASO 2.

**CASO 3**: Busca-se a palavra “excipientes\*” e em seguida no intervalo entre “\*” e o próximo ponto final (Figura 16).



Figura 16: Exemplo de bula contemplada no CASO 3.

**CASO 4**: Busca-se no intervalo entre a palavra “veículos:” e o próximo ponto final (Figura 17).



Figura 17: Exemplo de bula contemplada no CASO 4.

**CASO 5**: Busca-se a palavra “excipiente\*\*” e em seguida no intervalo entre “\*\*” e o próximo ponto final (Figura 18).



Figura 18: Exemplo de bula contemplada no CASO 5.

**CASO 6**: Busca-se a palavra “excipientes” e em seguida no intervalo entre “(” e “)” (Figura 19).



Figura 19: Exemplo de bula contemplada no CASO 6.

**CASO 7**: Busca-se a palavra “excipiente” e em seguida no intervalo entre “(” e “)” (Figura 20).



Figura 20: Exemplo de bula contemplada no CASO 7.

**CASO 8**: Busca-se a palavra “veículo” e em seguida no intervalo entre “(” e “)” (Figura 21).



Figura 21: Exemplo de bula contemplada no CASO 8.

**CASO 9**: Busca-se no intervalo entre “quantidade suficiente de” e “como” (Figura 22).



Figura 22: Exemplo de bula contemplada no CASO 9.

Conforme os dados são extraídos, os mesmos são guardados em um arquivo JSON que relaciona as formulações aos excipientes. Os casos que não são contemplados (Figura 23) são retornados como vazio pelo sistema.



Figura 23: Exemplo de bula não contemplada.

### HANDBOOK OF PHARMACEUTICAL EXCIPIENTS

Para concluir a implementação e alcançar o objetivo proposto, foi necessário elaborar a sequência de ações representadas no fluxograma (Figura 24).



Figura 24: Fluxograma de atividades para extração de informações do excipiente.

Utilizou-se o PDF do *Handbook of Pharmaceutical Excipients* [28] que trata-se de um livro tido como referência mundial pela área farmacêutica em seu conteúdo sobre formulações de medicamentos, incluindo detalhes sobre forma de utilização de cada excipiente, como casos de uso, proporções, patentes e outras informações.

A princípio, já que as páginas do documento onde encontra-se o sumário são conhecidas, foi necessário busca-las, converte-las para um arquivo texto e trata-las para que fosse possível criar um arquivo JSON relacionando o nome do excipiente a página do início da sua seção afim de facilitar a busca posterior.

Uma vez que muitos medicamentos possuem a mesma substância como veículo nas bulas, na lista de excipientes extraídos anteriormente havia nomes replicados, portanto, precisou tratar e eliminar as repetições. Em sequência, a nova lista foi traduzida para o inglês, usando a biblioteca googletrans em Python, em razão do livro estar nesse idioma. E então foi possível pesquisar os excipientes da lista no arquivo do sumário e encontrar a página com o início da sua seção no livro.

Logo em seguida é feito a busca pela página de interesse e extraído o texto referente ao tópico “7 *Applications in Pharmaceutical Formulation or Tecnology*”, tendo como início o título citado e como marcador do fim o título do próximo tópico “8 *Decription*”. Sendo assim, busca-se pelas ocorrências dos seguintes caracteres: “%”, “w/w”, “w/v” ou “mL”, que indicam unidades de medida, e ao serem encontrados, extraímos toda a frase que os contém afim de obter as funções e quantidades utilizadas dessas substâncias no medicamento de interesse.

# TESTES

Para averiguação e validação da qualidade dos resultados produzidos pela implementação completa deste projeto, foi utilizada a biblioteca pandas [25] para a manipulação dos dados relacionados a interpretação dos medicamentos. Na tabela 4 abaixo temos um detalhamento sobre as especificações do computador, da versão do interpretador Python e da conexão à internet utilizados para a execução dos testes:

Tabela 4: Especificações utilizadas nos testes

|  |  |
| --- | --- |
| Sistema operacional | Windows 10 Pro build 1909 |
| Arquitetura | 64 bits |
| Processador | Intel Core i7-8565U 1,8Ghz |
| Memória RAM | 16GB |
| Interpretador | Python 3.6.9 |
| Conexão à internet | Banda larga 50Mb |

O projeto foi estruturado em módulos sequenciais que atendem desde a tarefa primária de realizar a busca por bulas na base de dados Anvisa até a efetiva captura e armazenamento dos dados de proporções de excipientes oriundos do *Handbook* [28]. A configuração do ambiente e execução foi realizada em um sistema operacional Linux, distribuição Ubuntu 18.04, embarcado no Windows com as especificações exibidas na tabela 4, através da funcionalidade WSL – *Windows Subsystems for Linux* [33]*.*

Atualmente o projeto é executado em duas partes, sendo que a primeira compreende: o download das sugestões de nomes de medicamentos atualizadas, a interpretação do parâmetro do medicamento recebido do usuário, a verificação de sugestões, busca no bulário Anvisa e download das bulas encontradas. Já a segunda parte compreende a interpretação das bulas, tradução dos excipientes identificados, interpretação do sumário do livro e a busca e armazenamento dos resultados encontrados de excipientes.

A execução do primeiro módulo do projeto é feita conforme abaixo:



Figura 25: Comando para execução do primeiro módulo

Através do comando detalhado (Figura 25), nós chamamos o script Python localizado no arquivo main.py, presente na sequência de diretórios “excipiente\_scraper” e “scrapy” a partir da raiz do repositório Git [34], e fornecemos, por exemplo, o parâmetro “paracetamol”. O script faz a leitura do parâmetro, atualiza as sugestões e fez a busca com base nas mesmas, conforme o *log* abaixo (Figura 26):

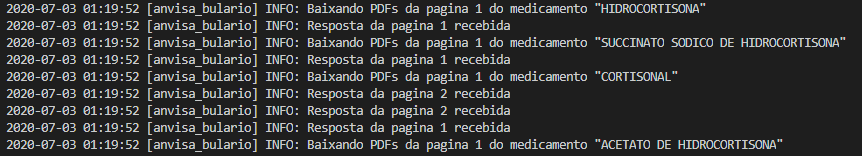


Figura 26: *Log* de execução do primeiro módulo

Em seguida, executamos o segundo e maior módulo do projeto, cuja chamada também foi simplificada através de um script Python único, nos diretórios “excipiente\_scraper” e “pdfreader” (Figura 27):



Figura 27: Comando para execução do segundo módulo

Como pode ser observado, para este módulo não é necessário o fornecimento de nenhum parâmetro extra. Igualmente ao primeiro módulo, o segundo módulo também emite mensagens que permitem ao usuário o acompanhamento de sua execução, como nas três figuras que se seguem:

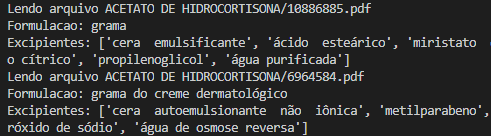


Figura 28: *Log* de execução do segundo módulo 1

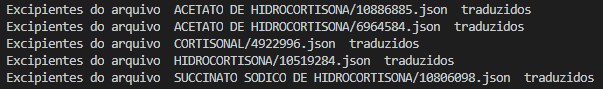


Figura 29: *Log* de execução do segundo módulo 2

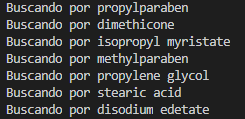


Figura 30: *Log* de execução do segundo módulo 3

O módulo que utilize a implementação da biblioteca pandas [25] possui interface com os arquivos intermediários que servem como troca de mensagens entre os módulos do projeto, e é através desses arquivos que estruturamos as informações colhidas e as exportamos para planilhas Excel. A execução deste módulo se dá através do arquivo denominado stats.py:



Figura 31: Comando para execução do módulo de metadados

O resultado da execução do comando (Figura 31) são dois arquivos no formato xlsx, sendo um contendo a visão de captura de dados das bulas (bulas.xlsx) e o outro exibindo a visão da captura de dados do livro (excipientes.xlsx). Abaixo (Figura 32) temos um exemplo da visão de bulas contendo os metadados de uma execução cujo parâmetro fornecido foi o do princípio ativo dipirona:

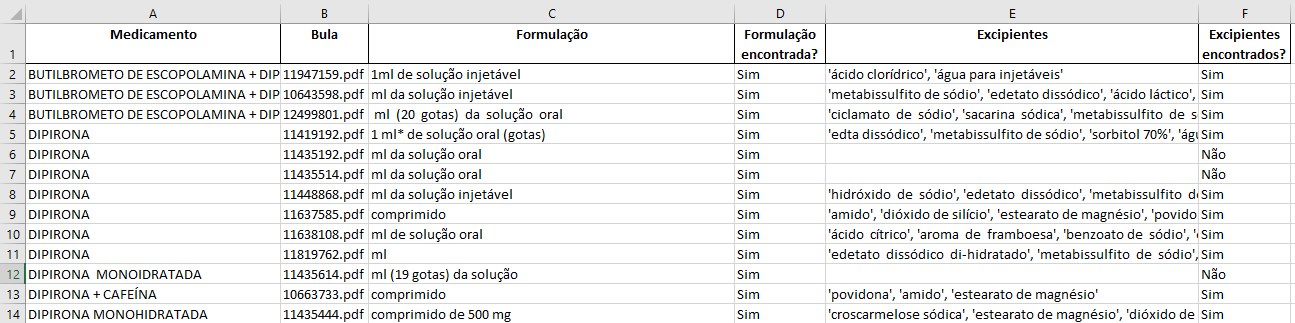


Figura 32: Visão de bulas dos metadados de execução

Nesta visão é exibida o nome da sugestão de medicamento utilizada, o identificador do arquivo resultante da busca, a formulação capturada do arquivo, a lista de excipientes capturada do arquivo, além de duas colunas binárias para indicar a captura correta ou não das informações citadas.

Já na visão de metadados da busca no livro, conforme o exemplo do princípio ativo amoxicilina abaixo, apresentamos o nome do excipiente, sua tradução e os registros de ocorrência das unidades de medida buscadas na seção de aplicações farmacêuticas:

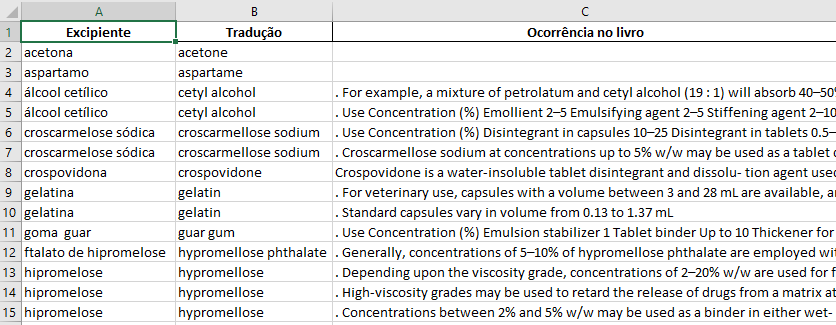


Figura 33: Visão de excipientes dos metadados de execução

Os resultados apresentados neste capítulo foram discutidos com o especialista farmacêutico e aprovados pelo mesmo. A evidência do documento de aceitação encontra-se no anexo B do presente trabalho.

# CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi desenvolvido um sistema com intuito de apoiar e facilitar pesquisas de excipientes para novos medicamentos com base em substâncias já utilizadas e catalogadas. Foi implementado um robô com as técnicas *web scraping* e *text mining* para extrair de forma automatizada os dados necessários para a análise.

Por se tratar de um estudo inicial, o principal foco foi no direcionamento da extração, uma vez que, foram utilizadas duas fontes de dados: bulário eletrônico da Anvisa e *Handbook of Pharmaceutical Excipients* [28]. Para a primeira fonte foram implementadas as duas técnicas, pois inicialmente foi necessário simular a navegação na web para obter as bulas disponíveis no site e em seguida, buscar a informação desejada nos arquivos obtidos. A segunda fonte foi disponibilizada como arquivo PDF e, portanto, a técnica *text mining* auxiliou na obtenção da informação pretendida.

O procedimento de extração automática mostrou-se um recurso de extrema importância e valor para auxiliar na agilidade, precisão e análise das pesquisas de novos medicamentos. Os resultados foram discutidos com o especialista farmacêutico que validou (Anexo B) e reafirmou os ganhos obtidos com a coleta automatizada quando comparado ao procedimento manual. A partir da identificação das zonas de interesse expostas neste estudo, é viável propor ajustes na implementação afim de abranger uma maior parcela de bulas.

Por fim, como sugestões de complementos futuros ao presente trabalho, e visando conferir otimizações ao projeto nos aspectos de refinamento dos resultados, ampliação da abrangência de aceitação das fontes já identificadas, ampliação do número de fontes e melhoria na usabilidade da ferramenta, temos: a ampliação da aceitação dos modelos de bulas existentes no catálogo da Anvisa, a estruturação do projeto como um pacote Python para agilizar a configuração e execução do mesmo, o refinamento no tratamento dos textos finais capturados das fontes de interesse, a pesquisa por ocorrências de excipientes em todo o conteúdo do *Handbook* e a inclusão de fontes de bulários estrangeiros como o da *Food and Drugs Administration (*FDA) [32].

REFERÊNCIAS BIBLIOGRÁFICAS

1. ZANON, Carolina Franco. **Aplicação de um programa de extração de dados através da técnica de web scraping para determinação do perfil epidemiológico dos pacientes da clínica de estomatologia da faculdade de odontologia da ufrj cadastrados no software estomato web**, 2017. Dissertação (Mestrado em Odontologia) – Curso de Pós-graduação em Odontologia, UFRJ, Rio de Janeiro, RJ.
2. MAZINI, Dhaniel Nunes; SATO, Renato Cesar. **Extração de dados financeiros com um web scraper: um estudo sobre a rentabilidade dos dividendos**, 2019. Dissertaação (Artigo na WAIAF) - Workshop of Artificial Intelligence Applied to Finance, ITA, São José dos Campos, SP.
3. BORGES, Thiago da Cunha; GANIMI, Zeus Olenchuk. **Extração de dados com web scraping para análise da variação de preço de veículos automotores**, 2018. Dissertação (Tecnólogo em Sistemas de Computação) - Curso de Tecnologia em Sistemas de Computação, UFF, Niterói, RJ.
4. **Yahoo! Finanças** <<https://br.financas.yahoo.com/>> Acesso em 09 de Maio de 2020
5. ABREU, Karen Kraemer. **História e usos da Internet**, 2009. Biblioteca On-line de Ciências da Comunicação, p. 2-4
6. KABIR MUHAMMAD SAJJAD, Syed. **Basic Guidelines for Research: An Introductory Approach for All Disciplines**, Edition: First, Chapter: 9, Publisher: Book Zone Publication, Chittagong-4203, Bangladesh, p. 201-275.
7. ZOZUS NAHM, Meredith. **The data book collection and management of research data**-Chapman and Hall CRC Press, 2017, p. 36.
8. BERNERS-LEE, Tim. **WWW: past, present, and future**, 1996. Publicação acadêmica – Computer, vol. 29, no. 10, pp. 69-77.
9. LONGMAN WESLEY, Addison. **A History of HTML**, 1998. <https://www.w3.org/People/Raggett/book4/ch02.html> Acesso em 24 mai. 2020.
10. BERNERS-LEE, Tim. **Information Management: A Proposal**, 1989. <https://www.w3.org/History/1989/proposal.html> Acesso em 24 mai. 2020.
11. LAWSON, Bruce; SHARP, Remy. **Introducing HTML5, 2nd Edition**, 2011. Livro. New Riders Press, Berkeley, Canadá.
12. **What is the difference between webpage, website, web server and search engine?** <<https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Pages_sites_servers_and_search_engines>> Acesso em 24 mai. 2020.
13. **CSS: Cascading Style Sheets** <<https://developer.mozilla.org/en-US/docs/Web/CSS> > Acesso em 24 mai. 2020.
14. **Javascript** <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>> Acesso em 24 mai. 2020.
15. **Google Chrome** <https://www.google.com/chrome> Acesso em 24 mai. 2020.
16. **Mozilla Firefox** <https://www.mozilla.org/en-US/firefox/new> Acesso em 24 mai. 2020.
17. **Microsoft Internet Explorer** <https://www.microsoft.com/pt-br/download/internet-explorer.aspx> Acesso em 24 mai. 2020.
18. OLSEN, Wendy; **Coleta de Dados**: Debates e Métodos Fundamentais em Pesquisa Social. 1. ed. Porto Alegre: Penso, 2015. p. 15.
19. RICHARDSON, Leonard. Documentação Beautiful Soup. <<https://www.crummy.com/software/BeautifulSoup/bs4/doc.ptbr/>> Acesso em 08 jun. 2020.
20. **Scrapy** < <https://scrapy.org/>> Acesso em 12 jun. 2020.
21. **JSON – JavaScript Object Notation**. <http://www.json.org/> Acesso em 12 de jun. 2020.
22. **XML – Extensible Markup Language**. Acesso em 21 de jun. 2020.
23. **PDFMiner** <<https://pypi.org/project/pdfminer/>> Acesso em 29 jun. 2020.
24. **Scrapy Spiders** <<https://docs.scrapy.org/en/latest/topics/spiders.html>>. Acesso em 29 de jun. 2020.
25. **Pandas** <<https://pandas.pydata.org/>>. Acesso em 29 de jun. 2020.
26. **Googletrans** < <https://pypi.org/project/googletrans/>>. Acesso em 29 de jun. 2020.
27. **PDFMiner** <<https://pypi.org/project/pdfminer/>> Acesso em 29 jun. 2020.
28. ROWE, Raymond C; SHESKEY, Paul J; QUINN, Mirian E. **Handbook of Pharmaceutical Excipientes**, 6th edition. 2009 London, UK: Pharmaceutical Press e Washington, DC: American Pharmacists Association.
29. **Python** <<https://www.python.org/>>. Acesso em 02 jul. 2020.
30. **PostgreSQL** <<https://www.postgresql.org/>>. Acesso em 02 jul. 2020.
31. **Tableau** <<https://www.tableau.com/pt-br>>. Acesso em 02 jul. 2020.
32. **FDA** <<https://www.fda.gov/>>. Acesso em 03 jul. 2020.
33. **WSL** <<https://docs.microsoft.com/pt-br/windows/wsl/about>>. Acesso em 03 jul. 2020.
34. **Git** <<https://git-scm.com/>>. Acesso em 03 jul. 2020.
35. **GitHub** < [https://github.com//](%20https://github.com//)>. Acesso em 03 jul. 2020.

ANEXOS

ANEXO A – CONTEÚDO DA REUNIÃO COM O ESPECIALISTA

O objetivo do projeto envolve a utilização de engenharia reversa para, a partir dos medicamentos já produzidos, possamos extrair estatísticas de utilização dos seus constituintes e produzir novos fármacos com essas formulações.

O documento abaixo apresenta o conteúdo da reunião realizada no dia 21/05/2020, que envolveu Altobelli de Brito (orientador), Lúcio Cabral (especialista farmacêutico), Marcello Telles (aluno) e Yasmim Martins (aluna).

***Manual da busca por bulas***

A primeira fonte de informação a ser consultada é a base de dados da Anvisa já que o órgão deve publicar as bulas de todos os medicamentos registrados em seu bulário.

Abaixo temos demonstradas as etapas de busca e coleta das bulas.

1. Ao acessar a página principal de consulta ao bulário (<http://www.anvisa.gov.br/datavisa/fila_bula/index.asp>), é apresentada a tela representada (Figura suplementar 1).



Figura suplementar 1: Tela da página principal do Bulário Eletrônico da Anvisa

1. Neste momento são preenchidos os campos desejados para filtrar a busca. Caso nenhum campo seja preenchido, o resultado listará todas as bulas de todos os medicamentos, contando com 10 registros por página. No exemplo abaixo (Figura suplementar 2), buscamos por “dipirona”.



Figura suplementar 2: Tela de resposta da busca por “dipirona”.

1. O resultado exibe informações como nome do medicamento, empresa responsável pela produção, expediente, data de publicação e os arquivos no formato PDF com versões da bula do paciente e do profissional. Nosso alvo para este projeto é a bula do profissional para todos os registros que compõe o resultado, e que pode ser baixada com um simples clique no símbolo do (Figura suplementar 3).



Figura suplementar 3: Localização da Bula do Profissional na tela de resposta da busca.

ANEXO B – DOCUMENTO DE ACEITAÇÃO ASSINADO PELO ESPECIALISTA

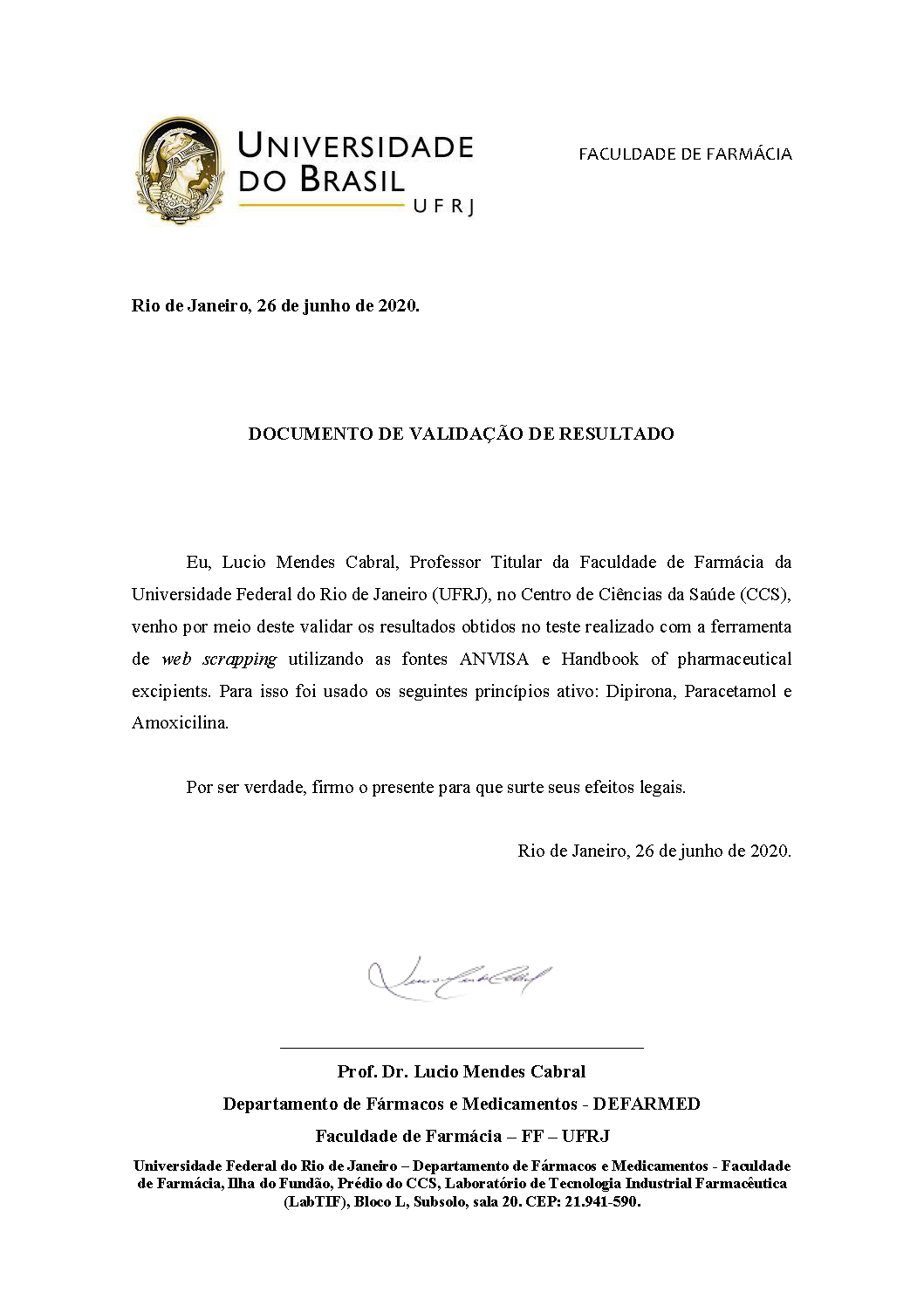


Figura suplementar 4: Documento de aceitação por parte do especialista.