# 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 e fazer uma análise detalhada do perfil dos pacientes. Concluiu-se que o uso do novo sistema, chamado Estomato Web Scraper, facilitou a extração e qualidade dos dados, e, assim sendo, poderá 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 aborda uma análise da rentabilidade dos dividendos das empresas que compõem o IBrX 50. Foi implementado um Web Scraper a partir da linguagem Python. As informações foram extraídas do site Yahoo!Finanças [[4](#REFERENCIASBIBLIOGRÁFICAS_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. Utilizando a tecnologia proposta, foi possível a automatização da coleta em uma escala de tempo reduzida. Os elementos obtidos foram armazenados em um arquivo que favorece a manipulação, possibilitando a realização de uma análise de regressão múltipla para chegar no 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. 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.

## DISCUSSÃO

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 aos ingredientes mais utilizados pela comunidade para se compor a receita, 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 são extraídas de sites especializados da área.

A aplicação de Web Scraping nesse contexto de extração das informações de sites farmacêuticos especializados através da busca por elementos-chave que compõe a página web proverá agilidade em uma etapa importante da criação de novos medicamentos, principalmente nessa época na qual a nossa sociedade demanda por soluções rápidas e eficientes que combatam os agentes infecciosos recém descobertos.

# 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 sobre tipos de dados, técnicas de coleta e de análise e armazenamento.

## COLETA DE DADOS

Ao iniciar uma pesquisa é necessário um estudo intensivo do conjunto de dados que gere informações relevantes para a análise a ser feita. 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 interessante a mesma, 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 desta será a garantia da precisão, coerência e confiabilidade nas informações obtidas.

## TIPOS DE DADOS

Para compreendermos melhor o desenvolvimento do trabalho, que será discorrido no capítulo a seguir, é importante definirmos com clareza os tipos de dados que existem. Estes 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, há a desvantagem de o esforço de coleta ser alto. São exemplos de dados primários as respostas de entrevistas e questionários dadas diretamente pelo entrevistado. [7]

### SECUNDÁRIOS

Dados secundários são obtidos a partir de coletas previamente realizadas. Por já terem sido publicados, comumente os dados secundários não possuem o formato ou a integridade mais adequada ao estudo de interesse. Desta forma, apesar do menor esforço de obtenção, em comparação aos dados primários, os dados secundários requerem maiores esforços na definição de fontes e na metodologia empregada em seu manejo. [7]

## ORGANIZAÇÃO E ARMAZENAMENTO

Posterior a coleta dos dados, os mesmos têm de ser armazenados de acordo com 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 em 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 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 já não mais para fins necessariamente relacionados à 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 são 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 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].

Na figura 3, podemos visualizar o resultado da página estruturada na figura 2, quando interpretada por um navegador. Vale observar que os comentários adicionados no documento foram ignorados pelo navegador, conforme citado anteriormente.



Figura 3 – Visualização de um documento HTML em um navegador

Apesar de os 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, 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 que também é conhecido como bot, consulta servidores web fazendo uma requisição 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 o código que pode ser utilizado para execução de comandos levando em consideração a árvore obtida ou para responder uma saída de interesse.



Figura 5 – Representação da estrutura básica de um documento HTML como árvore de objetos

Portanto é possível extrair o título do documento representado na Figura 4 a partir do seguinte caminho: html; head; title.

# 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

### 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. |

## 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 a conversão do conteúdo das páginas em formatos conhecidos como texto, XML [22] e HTML, além de facilitar a exportação dos dados em arquivos JSON [21], CSV e XML [22].

Um projeto Scrapy tem a arquitetura construída em classes de capturas de dados, conhecidas como spiders [23]. 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 as traduções de componentes farmacêuticos necessárias antes da busca no livro especializado, que possui conteúdo em inglês, utilizamos a biblioteca googletrans [25], disponível no repositório de pacotes Python.

Para a captura e tratamento de metadados sobre a execução do projeto, incluindo, por exemplo, informações sobre sugestões de medicamentos encontradas, bulas baixadas, conteúdo traduzido de cada bula e posterior busca no livro especialista, utilizamos a biblioteca especializada em análise de dados Pandas [24].

## DIAGRAMA DE CLASSES

O diagrama de classes do projeto 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 [24].

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.

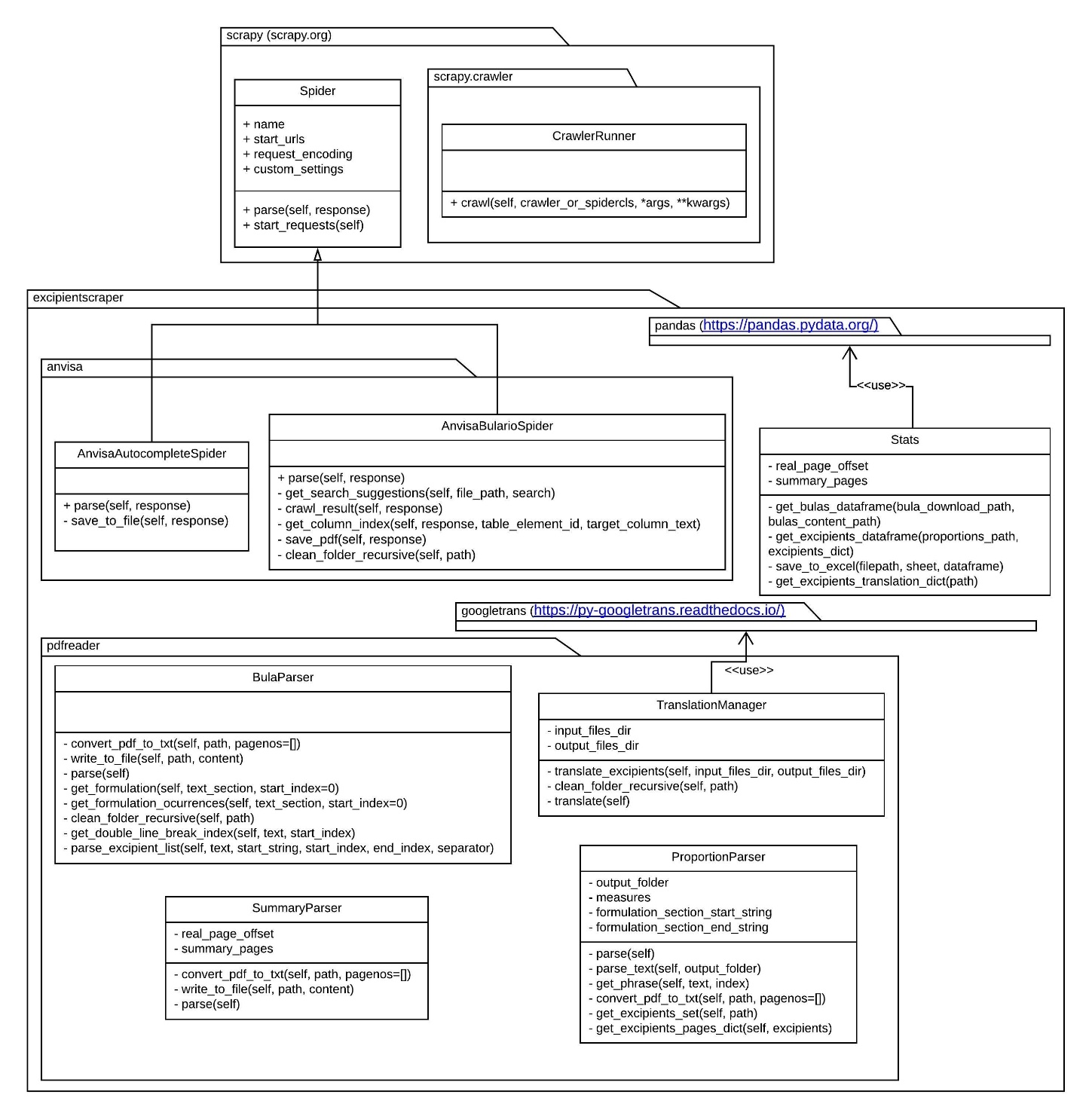


Figura 7 – Diagrama de classes do projeto

## FONTES

Afim de desenvolver a extração do dado corretamente, é executada uma sequência de operações em três módulos (figura 8) que estarão detalhados nesta seção separados pelas fontes utilizadas.



Figura 8 – Fluxograma dos módulos da extração dos dados

### ANVISA

#### Download das bulas do profissional

A implementação inicia com uma série de atividades, figura 9, afim de fazer o download de bulas do profissional do medicamento disponíveis no bulário eletrônico da Anvisa.

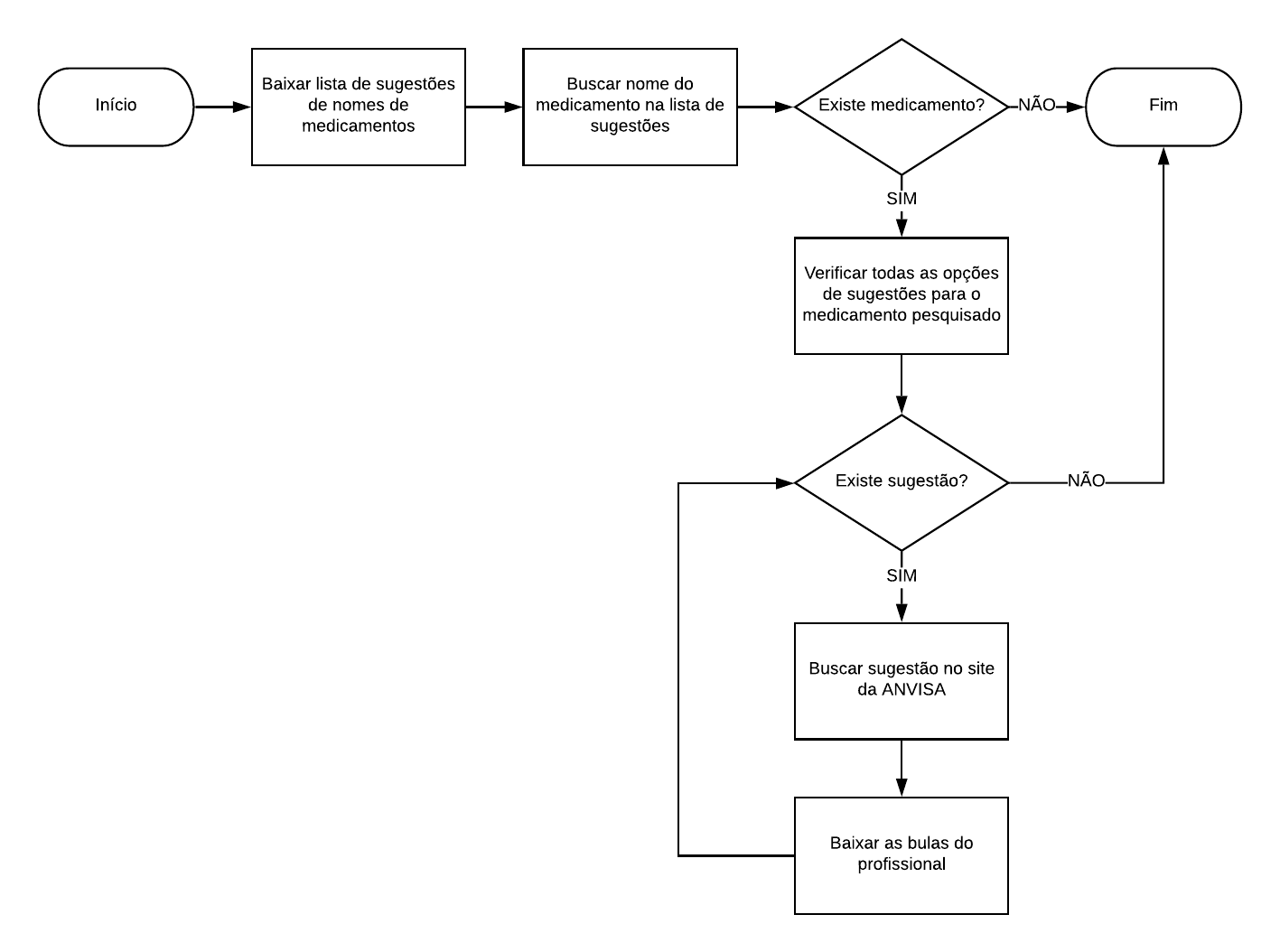


Figura 9 – Fluxograma de atividades para obter as bulas do profissional.

Detectou-se que para a busca das bulas do medicamento, o site do bulário eletrônico da Anvisa retorna sugestões, figura 10, conforme o usuário insere as letras do 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, apenas daquelas que o nome inicia com o que foi pesquisado.



Figura 10 – Exemplo de sugestões de nomes completos do fármaco a ser pesquisado.

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 é inserido o nome do medicamento desejado, 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 nenhuma ou uma lista de bulas que pode ser paginada dependendo da quantidade encontrada. Por fim, com o objetivo de fazer o download de todas as bulas do profissional exibidas na tabela de ID “tblResultado”, faz-se uma busca pela coluna “Bula do Profissional” no cabeçalho das colunas da tabela. Em cada célula dessa coluna existe a tag <a>, que representa um link em HTML, com o atributo “onclick” chamando uma função JavaScript denominada “fVisualizarBula”. O script captura os dois parâmetros numéricos que são enviados na função. Finalmente é feita uma requisição para <http://www.anvisa.gov.br/datavisa/fila_bula/frmVisualizarBula.asp>, fornecendo os parâmetros obtidos no passo anterior, e recebendo o arquivo PDF da bula do profissional como resposta.

#### Análise dos PDFs das bulas e extração dos excipientes

### Handbook of Pharmaceutical Excipients

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. **Scrapy Spiders** <<https://docs.scrapy.org/en/latest/topics/spiders.html>>. Acesso em 29 de jun. 2020.
24. **Pandas** <<https://pandas.pydata.org/>>. Acesso em 29 de jun. 2020.
25. **Googletrans** < <https://pypi.org/project/googletrans/>>. Acesso em 29 de jun. 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), Yasmim Martins (aluna) e Marcello Telles (aluno).

***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 abaixo:



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, buscamos 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 arquivo:



INCLUIR INFORMAÇÕES SOBRE OS BULÁRIOS DO FDA E EUDRA

ANEXO B – DOCUMENTO DE ACEITE ASSINADO PELO ESPECIALISTA

TODO