Comparador de Preços – Web Crawler

Ágata Raiza Lima, Renato Ribeiro

Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie
São Paulo – SP – Brasil

Abstract. *Web Crawler is a set of procedures and rules that filter data from web pages using HTML tags. In this article, we will show how a system was developed using a design science methodology for the development of a system capable of, using a web crawler, entering the pharmacy pages and searching for the remedies to create a comparator of prices.*

Resumo. *Web Crawler é um conjunto de procedimentos e regras que filtram os dados das páginas web utilizando-se de tags HTML. Neste artigo iremos mostrar como foi desenvolvido um sistema utilizando-se de uma metodologia de design science para o desenvolvimento de um sistema capaz de, utilizando-se de um web crawler, entrar nas páginas das farmácias e buscar os remédios para ser criado um comparador de preços.*

1. Introdução

Este artigo consiste na pesquisa realizada sobre o tema " Web Crawler - Comparador de Preços" que visa o mundo atual com pandemia e todas as facilidades para o comércio virtual. O desenvolvimento deste comparador de preço vai ajudar as pessoas a encontrarem remédios tarja preta que normalmente são os mais caros por preços mais acessíveis.

Normalmente os medicamentos é subdivididos por tarjas como: amarela, vermelha e preta, neste trabalho focamos na tarja preta pois é um tipo de medicamento controlado, muitas vezes não tem a pronta entrega e os farmacêuticos tem que encomendar antes, ou seja, são remédios que tem poucas unidades e que atualmente está tendo alta procura(obesidade, depressão) tendo a subir o preço.

O Web Crawler é um rastreador que tem por objetivo indexar páginas, ou seja, trabalhar de forma programada e sistemática navegando em páginas web, procurando e rastreando links para outras páginas e dados.

Localizar informações que atendem as necessidades do usuário vem se tornando uma tarefa difícil, grandes motores de busca tendem a ter 3,5 milhões de busca por dia [Flávio.N.2020]. Sabemos que a internet é usada por mais de 4 bilhões de pessoas em todo o mundo, e a quantidade de dados que vem sendo adicionado nela é cada dia maior, minerar essa quantidade gigantesca de dados tem se tornado algo cada vez mais importante e caro, pois desenvolver sistemas automatizados para cuidar disso tem exigido cada vez mais dos especialistas na área.

Grandes motores de busca como Google, Bing e Yahoo têm utilizado o Web Crawler conhecido também como "Boot", "Spider" que é um recurso inovador que ajuda na extração de dados para manter sua base de dados atualizada. Sua principal função é

examinar links, verificar códigos HTML, além de poder ser usado para o benefício do marketing digital gerando insights.

Neste trabalho será mostrado a utilização de uma arquitetura para o desenvolvimento de um sistema capaz de filtrar preços de remédios em sites de farmácias.

1.1. Contextualização do problema e da pesquisa

Web Crawler nada mais é que um conjunto de procedimentos e regras, ou seja, um algoritmo que permite verificar o código fonte de vários sites e identificar os dados mais relevantes.

A utilização do crawler é muito comum e está presente no nosso dia a dia por exemplo grandes motores de busca como o google usa para realizar a função de busca de sites, precisa ter vários sites indexados, ou seja, eles têm uma web crawler que varre toda web procurando links e vai indexando.

Web Crawler funciona na seguinte forma: O sistema tem uma URL e ele faz uma requisição para o servidor dessa URL e o servidor irá retornar uma página HTML, depois ele irá indexar essa página e dentro dela será procurado uma referência, as tags HTML em busca de informações e hiperlinks para outras páginas e assim começa um processo recursivo, pois ele irá entrar em cada uma dessas páginas encontradas e repetir todo o processo novamente.

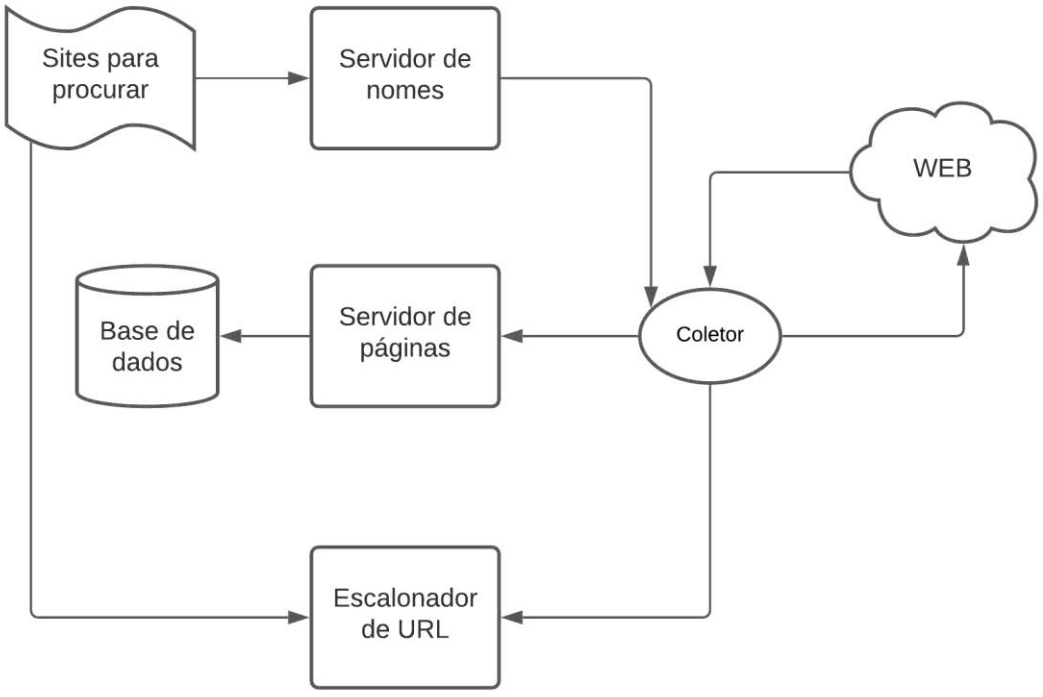


Figura 1 - Arquitetura Padrão de Web Crawler

O desafio do comparador de preço é localizar/filtrar medicamentos (tarja preta) com preços mais baixos mas também manter a sua base de dados sempre atualizada, pois o

crawler permite que as informações de sites sejam capturadas, e encontrando outras páginas similares.

1.2. Objetivo

O objetivo é desenvolver um sistema de crawler a partir de uma metodologia de design science tendo como resultado final um sistema capaz de comparar preços de remédios.

Para isso, seguindo a metodologia de design science mostrada por [Wieringa 2014], no livro Design Science ,os objetivos deste trabalho são: Para isso, os objetivos deste trabalho são:

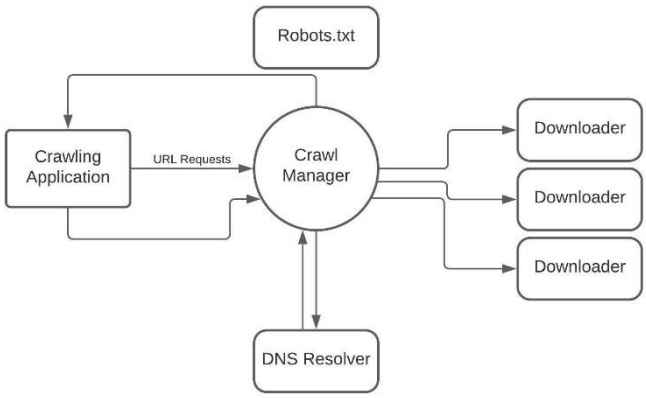
- Entender o contexto do problema proposto.
- Levantar requisitos (Funcionais e Não Funcionais).
- Desenvolver uma arquitetura para solucionar o nosso problema.
- Desenvolver diagrama de classe e extrair atributos e métodos.
- Desenvolver o sistema.
- Realizar os testes.

2. Referencial Teórico

Segundo [Kapor (2010], um bom sistema deveria ser como um prédio bem-feito. Eles exibem 2 características: Comodidade: Um programa não deve ter nenhum bug que impeça seu funcionamento e Prazerosa: A experiência de usar o sistema deve ser uma boa experiência.

Segundo [A.Bushara et al.2013] Web Crawler que é um sistema que extrai dados, conceito apresentado por ICOSST: “Um rastreador é um programa que visita sites e lê suas páginas e outras informações para criar entradas para um índice de mecanismo de pesquisa”. Os principais motores de busca da web têm esse programa, que também é conhecido como "spider" ou um "robô." Os rastreadores são geralmente programados para visitar sites que foram enviados por seus proprietários como novos ou atualizados. Aparentemente, os rastreadores ganharam o nome porque rastreiam através de um site e uma página por vez.

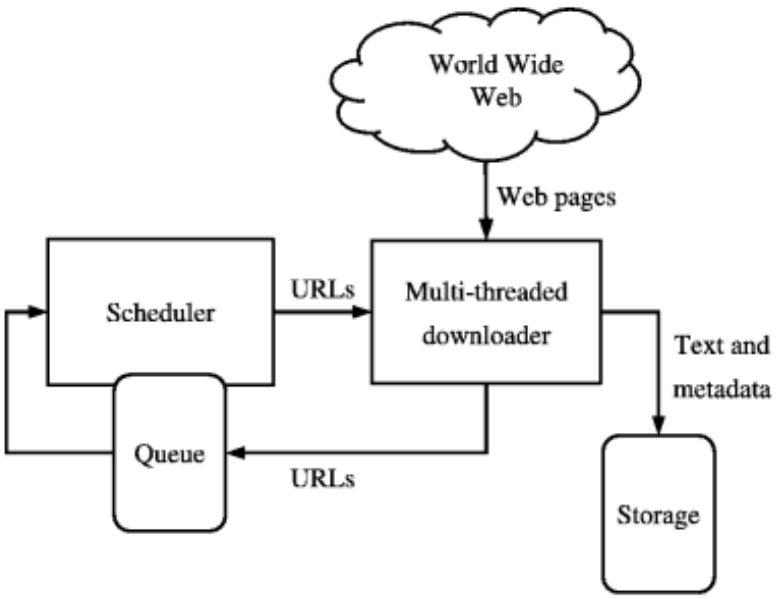
Figura 2 - Modelo de Crawler desenvolvido por Bushara



Fonte: ICOSST, 2013

Segundo [Castillo 2004] o funcionamento de um web crawler acontece quando um conjunto de URLs iniciais são enfileiradas e um URL daquele conjunto é obtido em alguma ordem da fila por um rastreador e em seguida, o rastreador baixa a página(dados). Isso é seguido pela extração das URLs e da página baixada e enfileirando-os o processo só interrompido quando um dos rastreadores para completamente.Um loop de rastreamento consiste em obter um URL da fila, baixar o arquivo correspondente com a ajuda de HTTP, percorrer a página para novos URLs e incluir as URLs não visitados para a fila além disso o um crawler deve ter um boa estratégia de rastreamento e uma arquitetura otimizada".

Figura 3 - Arquitetura de Alto Nível apresentada por Castillo



Fonte: Carlos Castillo, 2004

Segundo [Dhenakaran et.al. 2011], crawler identifica todos os links das páginas e adiciona-os na lista de URL que serão visitadas.

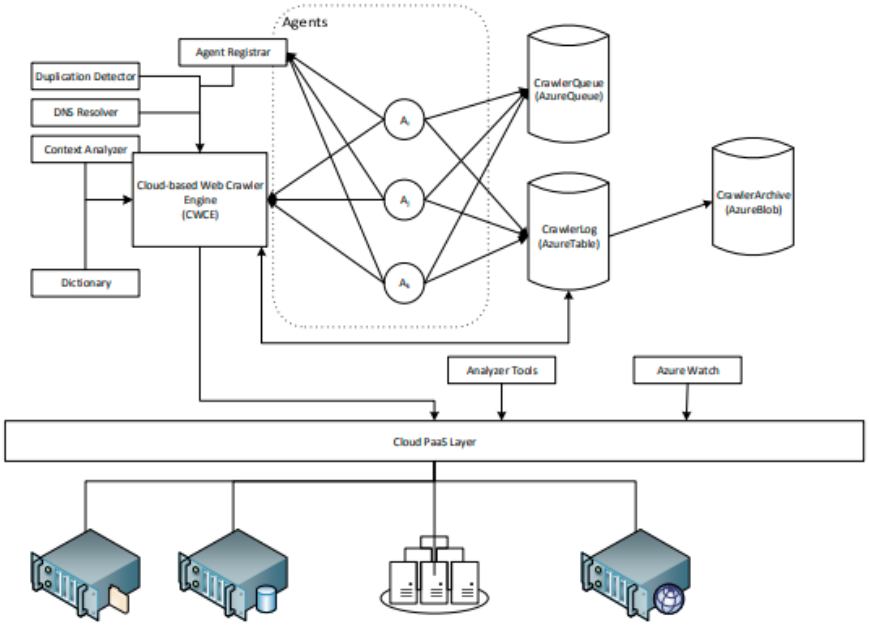
Segundo [Laender et al.2002] a abordagem tradicional para a extração de dados de páginas Web é a escrita de programas especializados, chamados de wrappers, que identificam os dados de interesse e mapeiam eles para um formato adequado".

Segundo [Bahrami et al.2015] O primeiro requisito para um rastreador distribuído da web é a seleção de um esquema de particionamento de página da web apropriado.

O primeiro esquema é baseado em hash de URL, que apresenta as páginas da Web de partição com base no valor de hash do URL. Cada hash é atribuído a um agente. O segundo esquema é a função baseada em hash de site e atribui páginas no mesmo site ao mesmo agente. O terceiro esquema é o esquema hierárquico e atribui páginas com base em algum recurso, como idioma e região. Em nosso rastreador da web proposto, primeiramente particionado a web por site hash com base (por exemplo, subdomain.domain) e atribuídas a um agente. Em segundo lugar, cada site é particionado com base no hash de URL.

Bahrami também nos mostra um web crawler baseado em cloud que utiliza Azure Cloud Queue para manter uma lista de URLs recuperados de uma página. A fila de URLs é temporária e aguardam um busca correspondente ao agente da sua zona e para a tabela foi utilizado o Azure Cloud Table para armazenar informações permanentes sobre as páginas rastreadas, já a tabela é baseado no NoSQL o que nos permite definir um novo campo instantaneamente quando inserimos um registro com um novo campo.

Figura 4 - Arquitetura Web Crawler Cloud desenvolvido por Bahrami



Fonte: International Conference on Intelligence in Next Generation Networks, 2015

Foi utilizado o artigo Relatório Web Crawlers et al. (2013) para desenvolver a arquitetura utilizada na figura 1.0.

Segundo [Brin et al.1999], para construir o índice, o motor de busca deve visitar cada documento a ser incluído no índice, analisar o documento e adicionar os dados relevantes para o arquivo.

Segundo [Halil et.al 2008], o índice é uma lista de URLs e de palavras – chaves escritas pelo usuário.

3. Metodologia

Segundo [Wieringa 2014], para fazer um projeto de design science, é necessário compreender seus componentes principais, ou seja, seu objeto de estudo e suas duas atividades principais. O objeto de estudo é um artefato no contexto, e suas duas atividades principais são projetar e investigar esse artefato no contexto. Para a atividade de design, é importante conhecer o contexto social das partes interessadas e os objetivos do projeto, visto que esta é a fonte do orçamento da pesquisa e o destino de resultados úteis da pesquisa.

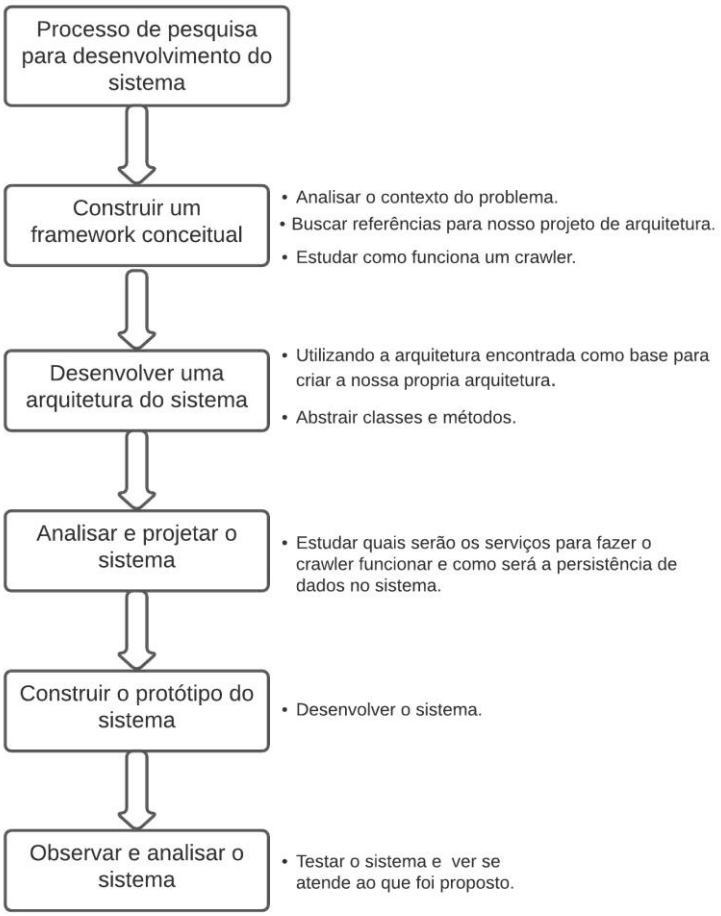


Figura 5 - Processo para a pesquisa em desenvolvimento de sistemas

No que tange a Metodologia empregada neste trabalho teve início com a percepção de um contexto onde foi notado que as pessoas não estão muito acostumadas a fazer compras de remédios online assim como fazem com outros produtos, então será desenvolvido um sistema capaz de procurar dentro dos sites das farmácias selecionadas e buscar dentro deles os preços dos remédios, para o cliente conseguir fazer a compra com o menor preço.

Analisando o contexto existente começamos a montar qual seria o nosso método de resolvê-lo, estudando livros e artigos já publicados na área, então encontramos qual era a arquitetura base de um web crawler e a partir dela observamos o que precisamos para o nosso projeto.

Após entendermos quais eram nossas entidades e como elas se relacionam, então adaptamos a arquitetura que encontramos para a nossa própria arquitetura.

Com nossa arquitetura desenvolvida começamos então a estudar como se desenvolvia na prática um crawler, quais as linguagens mais usadas e quais as bibliotecas mais usadas, e chegamos a conclusão de que Python era a melhor para o que precisávamos.

Finalizado o desenvolvimento então começaram os testes de esforço, para pegar o nome dos remédios que seriam crawleados utilizamos o próprio site da ANVISA que disponibiliza uma lista de remédios tarja preta ,adicionamos os remédios e então rodamos o programa.

Ao final do processo tínhamos uma lista de vários remédios de algumas farmácias diferentes.

4. Resultados

Para o desenvolvimento deste trabalho inicialmente começamos a estudar sobre os crawlers, como funcionam, qual a arquitetura de um crawler básico e comparando com os nossos próprios desafios para poder chegar então a uma arquitetura em camadas vista na figura 6.

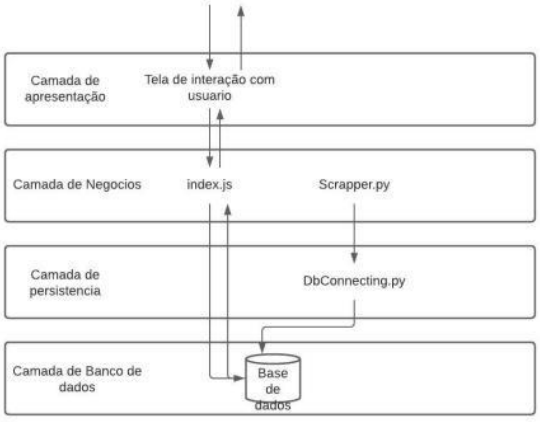


Figura 6 - Arquitetura em Camadas

Nessa proposta o sistema de crawler fica separado da interface com o usuário, tendo este apenas acesso a uma busca dos valores que o crawler coloca no banco de dados quando o mesmo é rodado.

Para a interface com o usuário foi desenvolvida uma aplicação em Node Js que carrega as páginas onde o cliente irá se relacionar com o sistema escrevendo o nome do remédio que quer procurar e então o sistema utilizando-se do que foi escrito anteriormente faz uma busca no banco de dados para pegar quais tipos de remédios se relacionam com o que foi procurado(por exemplo o usuário busca por “Rivotril” e o sistema mostra quais tipos de Rivotril com diferente quantidade de comprimidos).

Para o desenvolvimento do crawler foi desenvolvido inicialmente um diagrama de classes como apresentado na figura 7 e para a codificação foi utilizada a linguagem Python, que já é uma linguagem bastante utilizada no mercado para desenvolver Crawlers, utilizando-se da biblioteca do Beautiful Soup e Selenium, duas bibliotecas para a automação de processos web. O sistema inicializa carregando os nomes dos remédios e farmácias que irão ser procurados, e então se faz uma requisição HTTP para o site, utilizando-se do Selenium iremos procurar na barra de pesquisa pelo nome dos remédios, para cada remédio buscado se retorna um código HTML da página, com o Beautiful Soup é possível utilizando as tags navegar pelo site e pegar apenas os dados necessários que serão utilizados para criar um objeto do Tipo Remédio que posteriormente será gravado num banco de dados SQL.

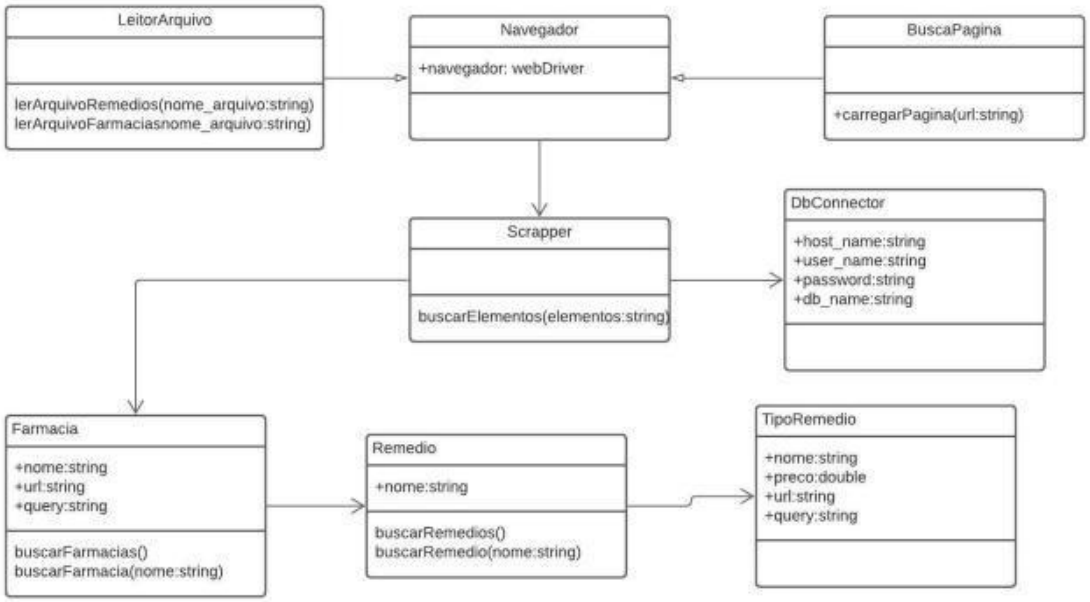


Figura 7 - Diagrama de Classe

Como resultado final obtivemos um sistema capaz de mostrar o melhor preço e em qual farmácia ele se encontra e um link para acesso rápido. Um exemplo do seu funcionamento encontra-se nas figuras 8 e 9.

Na figura 8 se encontra a tela inicial do sistema, nela pode-se ver o usuário escrevendo o nome do remédio que está procurando.

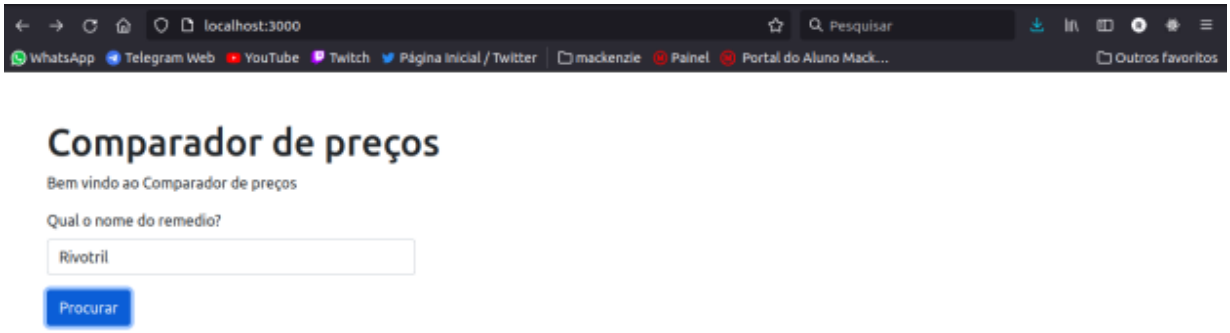


Figura 8 - Tela Inicial

Já na figura 9 pode-se encontrar os resultados obtidos dessa busca, mostrando todos os tipos de remédios que têm relacionado ao que foi buscado.

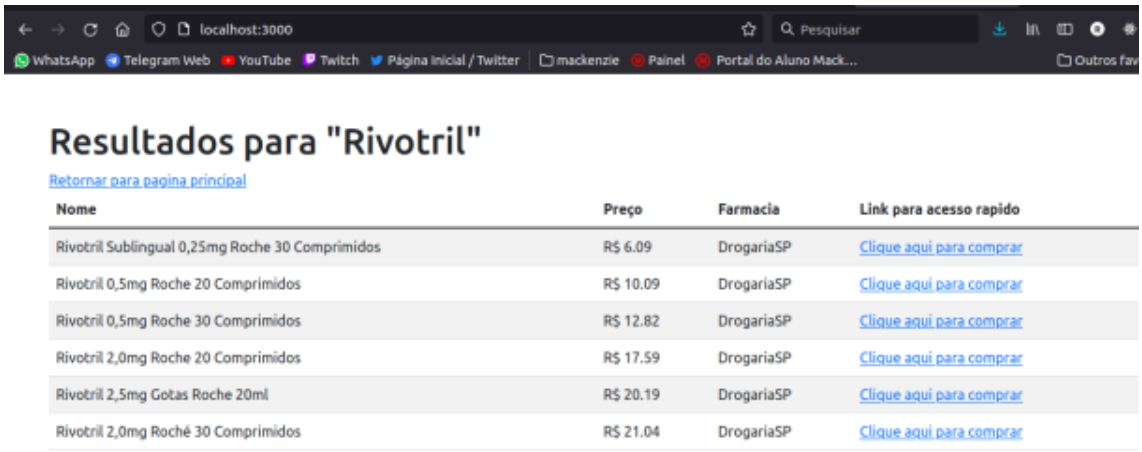


Figura 9 - Tela Resultado

5. Conclusão

Com o desenvolvimento acelerado das vendas online gerado principalmente pela pandemia, nenhum ramo do comércio ficou de fora das vendas online, farmácias não foram exceção. Com a grande quantidade de farmácias surgindo, os clientes começaram a procurar melhores preços para remédios que costumam tomar regularmente. Para facilitar esse processo surgem os comparadores de preço, onde em apenas um único lugar pode-se procurar o mesmo remédio em várias farmácias diferentes e ver qual está com o menor preço.

Com isso selecionamos os remédios tarja preta que normalmente são os mais caros e as farmácias mais populares e mostramos ao cliente onde aquela opção de remédio escolhida vai estar com o preço mais acessível.

No futuro pretendemos pesquisar um número maior de remédios e farmácias e para isso conseguimos deixar nossa arquitetura mais escalável possível, ou seja, todas as funcionalidades serão definidas num mesmo bloco e assim poderemos expandir o sistema de maneira eficiente, pois, separando cada classe com seus atributos e funções é possível apenas passando a url das farmácias novas e as tags necessárias para encontrar no HTML o elemento que procuramos buscar as informações de diferentes origens.

6. Referências

ARIF,Bushra;NISA,Aroojun;SHAFI,Qasim;HAZAIMA,Naheed;SIDDIQUI,Um;HABIBA,Tariq;*." "Web Crawlers to Detect Security Holes"*. Paquistão: ICOSST,2013.

BAHRAMI,Mehdi;SINGHAL,Mukesh;ZHUANG, Zixuan*." A Cloud-based Web Crawler Architecture"*. California: IEEE,2015.

CASTILHO, C; *"Effective Web Crawling"*. Chile: Universidade do Chile,2004.

DHENAKARAN.S; SAMBANTHAN, K.*" Web Crawler - An Overview"*. California:2011.

LAENDER,L;RIBEIRO,Neto .*" A brief survey of web data extraction tools"*.Sigmod Record,2002.

VERZBICKAS,Augusto:MOCELIN,Eduardo:SIEGA,Renata:NETO,Milton*;"Relátorio Web Crawlers"*,2013.

HEVNER,Alan:CHATTERJEE*: "Design Research in Information Systems",* 2010.

HALIL,Ali*:"Effective Web Crawlers",*2008

ALINE,D.*"Design Science Research":*Porto Alegre*,* Grupo A, 2015.

FLÁVIO,N *"Quantas pesquisas são feitas por dia no Google",*2020.Disponivel em: https://eco.sapo.pt/2020/08/15/quantas-pesquisas-sao-feitas-por-dia-no-google/