UNIVERSIDADE FEDERAL FLUMINENSE

BRUNO BARBOSA DE MEDEIROS

MATHEUS BERNARDES COSTA DO NASCIMENTO

WORDPRESS: UMA FERRAMENTA PARA EXTRAÇÃO DE DADOS DE BLOGS DE INTERESSE

Niterói

2022

BRUNO BARBOSA DE MEDEIROS
Matheus Bernardes Costa Do Nascimento

WORDPRESS: UMA FERRAMENTA PARA EXTRAÇÃO DE DADOS DE BLOGS DE INTERESSE

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

Orientador(a):
ALTOBELLI DE BRITO MANTUAN

NITERÓI
2022

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

M488w Medeiros, Bruno Barbosa de
WORDPRESS: UMA FERRAMENTA PARA EXTRAÇÃO DE DADOS DE BLOGS DE INTERESSE / Bruno Barbosa de Medeiros, Matheus Bernardes Costa do Nascimento ; Altobelli de Brito Mantuan, orientador. Niterói, 2022.
49 f. : il.

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

1. Web Scraping. 2. WordPress. 3. Extração automática de dados. 4. Produção intelectual. I. Nascimento, Matheus Bernardes Costa do. II. Mantuan, Altobelli de Brito, orientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

BRUNO BARBOSA DE MEDEIROS
MATHEUS BERNARDES COSTA DO NASCIMENTO

WORDPRESS: UMA FERRAMENTA PARA EXTRAÇÃO DE DADOS DE BLOGS DE INTERESSE

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, 12 de julho de 2022.

Banca Examinadora:

_____

Prof. Altobelli de Brito Mantuan, MSc. – Orientador
UFF – Universidade Federal Fluminense

_____

Profª. Josiane Coelho de Oliveira, Esp. – Avaliador
UNESA – Universidade Estácio de Sá

BRUNO

Dedico este trabalho a todos que, assim como eu, fascinam-se com a obtenção de novos conhecimentos e com a beleza de boas soluções.

A todos que estiveram ao meu lado e que, verdadeiramente, torceram pela conclusão de mais um passo e me impulsionaram a alçar voos cada vez mais altos.

MATHEUS

Dedico este trabalho a todos que usam o conhecimento para uma sociedade mais justa e democrática por meio da tecnologia.

Àqueles que veem nas dificuldades grandes oportunidades em tornar pessoas melhores para transformar uma sociedade melhor.

AGRADECIMENTOS

BRUNO

À minha esposa, que sempre esteve ao meu lado, me dando apoio, incentivo, forças para nunca desistir e esperança de tempos melhores.

Ao Orientador Altobelli Mantuan, que, além de nos guiar para a conclusão desse trabalho, me deu grande incentivo para que eu conseguisse chegar até o final.

Ao Matheus Bernardes, meu companheiro de trabalho, pelo incentivo recíproco, pela paciência e troca de experiências durante todo o curso e culminando na realização deste trabalho.

A todos os meus familiares e amigos pelo apoio e colaboração, em especial minha mãe e meu irmão, que sempre acreditaram em mim e me deram todo apoio que puderam.

AGRADECIMENTOS

MATHEUS

A Deus primeiramente, se não fosse pela graça Dele não estaria aqui hoje terminando o TCC. Também dedico aos meus pais (José e Dulce) e minha irmã Natália que foram minha base nos momentos bons e ruins durante a faculdade quando mais precisei.

Ao Orientador Altobelli Mantuan, que foi crucial desde o início até o fim da jornada em dar suporte sempre que necessário para finalização do projeto.

Ao Bruno Barbosa, parceiro de trabalho, pelo apoio até o fim do TCC, mesmo depois de muitos altos e baixos não desistiu e estamos finalizando mais uma jornada.

A todos os meus familiares e amigos pelo apoio e colaboração, em especial ao meu pai José Antônio que com seu conhecimento em TI me deu muito suporte para o aprendizado contínuo nesta área.

“Uma mente que se abre a uma nova ideia, jamais voltará ao seu tamanho original”.
Albert Einstein

RESUMO

Com o advento de novas tecnologias e o aumento expressivo na velocidade de navegação, a geração de informações na rede mundial de computadores, e com isso o consumo das mesmas, acompanhou esse crescimento. Como consequência, uma infinidade de dados fica espalhados pela rede e se torna cada vez mais difícil de se tomar conhecimento. Isso fomentou a criação de ideias e métodos de se explorar e manipular toda essa gama de informações. Uma delas é o Web Scraping, que consiste em automatizar a obtenção desses dados. O presente trabalho tem como objetivo mostrar que blogs WordPress são uma excelente fonte de informação, tendo em vista assuntos de interesse, além de contar com opiniões do público interessado. Para tanto, foi elaborada uma ferramenta, utilizando-se a linguagem Python, a qual possibilitou a extração e o armazenamento de dados de alguns blogs selecionados para a pesquisa.

Palavras-chaves: Web Scraping, WordPress, extração automática de dados.

ABSTRACT

As new technologies emerges and the browsing speed gets more expressive, the generation of information on the world wide web and its consumption, followed this growth. As a consequence of it, an infinity of data is scattered across the network and becomes increasingly difficult to stay informed of and the creation of ideas and methods of exploring and manipulating this entire range of information has been driven by it. One of them is Web Scraping, which consists of automating the acquisition of this data. The present work aims to show that WordPress blogs are an excellent source of information, considering matters of interest, in addition to counting on the opinions of the interested public. For this purpose, a tool has been developed, using the Python language, which made it possible to extract and store data from some blogs selected for the research.

Key words: Web Scraping, WordPress, automatic data extraction.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface
CERN – Conseil Européen pour la Recherche Nucléaire
CSS – Cascading Style Sheets
CNN – Convolutional Neural Network
DNS – Domain Name System
DOM – Document Object Model
HTML – Hypertext Markup Language
HTTP – Hypertext Transfer Protocol
IDE – Integrated Development Environment
IoT – Internet of Things
IP – Internet Protocol
JSON – JavaScript Object Notation
KNN – K-Nearest Neighbor
REST – Representational State Transfer
SVM – Support Vector Machines
TCP – Transmission Control Protocol
TXT – Em inglês, text
URI – Uniform Resource Identifier
URL – Uniform Resource Locator
WEB – World Wide Web
WP – WordPress
WWW – World Wide Web
XML – Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO

Atualmente há um grande volume de dados que são armazenados e acessados diariamente na internet. Com base nisso, muitas empresas "exploram" os dados brutos e os tornam interpretáveis usando ferramentas de programação, gráficos, dentre outros recursos, para que possam ser divulgados publicamente por meio de artigos científicos ou privados nos projetos.

O WordPress (WP) é um *Content Management System (*CMS) desenvolvido em 2003 na linguagem PHP com banco de dados MySQL que tem como objetivo permitir que desenvolvedores criem sites dinâmicos de forma rápida e acessível com pouco uso de código para quem não domina a área. Dessa forma, blogs são criados em WP em maior escala que o Blogger, que pertence ao Google. Assim sendo, preferimos usar o WP para dar continuidade ao projeto usando API REST para obter dados JSON das páginas de interesse de cada site de interesse.

Para obter essa gama de informações, foi utilizada a linguagem de programação Python para facilitar nosso trabalho. Desta maneira, utilizamos a seguinte estratégia:

- Desenvolvimento de script para automatização e coleta dos dados brutos das páginas exploradas, a partir de endpoints definidos.
- Armazenamento das informações extraídas em arquivos de texto (TXT), para uma futura análise e classificação.

2. TRABALHOS RELACIONADOS

O avanço tecnológico trouxe consigo um volume imensurável de dados, cujas fontes podem ser heterogêneas e das mais diversas. Dessa forma, a validação da relevância e confiabilidade de tais informações se faz bastante necessária, bem como uma forma apropriada de tratamento e classificação delas para quem tenha interesse.

Existem diversos métodos para a extração de dados de sites da internet. Um deles é o *Web Scraping* (do inglês, raspagem da rede), que consiste em coletar e combinar dados que se tem interesse, com o uso de algoritmos que são capazes de reestruturar e armazenar essas informações, o que torna o processo otimizado em relação aos métodos mais convencionais de coleta, nos quais as pesquisas e a organização de dados é feita de forma manual.

De forma a corroborar com o abordado acima, podemos encontrar diversos trabalhos na Web que discorrem sobre esse assunto. No trabalho [1] observamos que o interesse é encontrar ferramentas de NLP (Natural Language Processing), mineração de dados, algoritmos de aprendizagem para que possam ser feitas análises de media social, por exemplo, sintomas de depressão em lugares onde pessoas comentam, como blogs ou sites específicos para retratar a situação em que o indivíduo se encontra. Dessa forma, a extração desses sintomas expostos pelos usuários nas mídias sociais ajuda a ter uma referência dos principais sintomas de depressão clínica em humanos. O trabalho se resume a uma abordagem de aprendizagem não supervisionada que pode ser usada para dados não estruturados. Ele se divide em 4 sessões: métodos de coleta de dados, pré-processamento de dados brutos, experimento na análise dos resultados e conclusão do trabalho com pesquisas futuras.

Já no trabalho [2], o objetivo é identificar, através do texto escrito, o sentimento relacionado à experiência da viagem, expresso em textos contidos em blogs de turismo, de forma automatizada e unificada, utilizando os métodos computacionais conhecidos como web scraping que são capazes de qualificar e quantificar e, desta forma, mensurar e analisar o que está sendo dito, tendo em vista que este trabalho se torna difícil de ser feito com APIs e outras ferramentas existentes. O processo consiste em fazer o download dos dados e utilizar alguns métodos de filtragem

e tratamento para selecionar as informações realmente relevantes sobre o tema. Os métodos utilizados neste artigo foram: KNN (K-Nearest Neighbor), SVM (Support Vector Machines), Random Forest, CNN (Convolutional Neural Network), Hybrid SVM-CNN. Este trabalho concluiu que os métodos são eficientes e com desempenho aproximado entre si. Foi apresentada também uma forma nova de cruzamentos de informações (Hybrid SVM-CNN).

Podemos ver ainda, no trabalho [3], um artigo que é motivado pelos desafios da aplicação de uma abordagem de mineração de texto automatizada que visa reconhecer, através de uma análise, os sentimentos e opiniões de turistas em blogs. Este estudo tem a intenção de mensurar o quanto os métodos computacionais avançados podem melhorar a aquisição de dados e a análise de dados não estruturados retirados de fóruns e blogs. Além disso, objetiva-se saber até onde esse experimento pode estender-se em relação à análise de sentimentos e sua capacidade de tornar resultados qualitativos mais objetivos. O texto traz também uma nova proposta de abordagem que combina análise de sentimentos e aprendizado supervisionado de forma a identificar termos que possam carregar melhor o sentimento apresentado. O resultado é que métodos computacionais, apesar de terem algumas restrições específicas, podem fornecer análises mais profundas, quando se leva em conta a análise quantitativa.

Já no trabalho [4], o foco é revisar diferentes metodologias relacionadas a análises políticas buscando diversas fontes na internet. A ferramenta mais comum e mais utilizada atualmente são as redes sociais que têm grande impacto no campo político. Dessa forma, o artigo tem como objetivo entender as metodologias para assim usar como estratégia política da melhor decisão a ser tomada.

2.1 DISCUSSÃO

Observando os trabalhos apresentados, podemos perceber a quantidade de informações relevantes que podem ser extraídas de veículos como mídias sociais e outras fontes de interação social como blogs e fóruns de discussão e que, quando

tratadas e classificadas, elas podem se tornar fonte de pesquisa e informação importante nas áreas de interesse desejadas.

Somado a isso, percebemos que as técnicas aplicadas, além de poderem ser aprimoradas em relação à acurácia na classificação e eficiência no processamento, elas também funcionam em diversas áreas de interesse.

Dessa forma, usamos ferramentas para auxiliar no nosso objetivo final como o uso de linguagem de programação, acesso a blogs e API para gerar o resultado desejado.

Para esse tipo de abordagem, podemos citar como vantagem a possibilidade de automatização, tendo em vista que é uma prática em crescimento e que existe uma comunidade interessada nos resultados que podem ser apresentados através dessa técnica.

3. FUNDAMENTAÇÃO TEÓRICA

É indiscutível que os tempos modernos trouxeram consigo uma capacidade notória de conectividade. Dessa forma, é quase impossível estar desconectado do mundo virtual com toda essa tecnologia disponível. Assim sendo, uma abundância de dados vem sendo gerada a partir de recursos computacionais como IoT (Internet das Coisas), Cloud Computing (Computação em Nuvem), aumento da velocidade da internet, dentre outros.

Em concordância com isso, se faz necessária a capacidade de armazenar toda essa informação gerada, bem como de se classificá-la, de modo que esse processo seja preciso e eficiente. A seguir, serão apresentados alguns conceitos que auxiliam na abordagem supracitada, como coleta de dados, armazenamento e organização.

3.1. COLETA DE DADOS

A coleta de dados é uma ferramenta utilizada por pessoas (ou processos automatizados) para obter informações relevantes sobre um determinado negócio com as estratégias necessárias para chegar ao objetivo final. São utilizados para tarefas de pesquisa, planejamento, estudo, desenvolvimento e experimentações. Os dados podem ser coletados de diversas formas. As principais são por meio de formulários, sites e plataformas específicas para coleta [5].

A importância de coletar os dados é crucial para obter informações importantes para abordar as melhores estratégias com os recursos que tem para atingir um objetivo ou vários. A coleta de dados se baseia por meio de três fontes: primária, secundária e terciária [6].

A *fonte primária* tem a possibilidade de ser definida como fonte original, ou seja, refere-se àquela que não houve nenhuma análise ou dados anteriores a ela. Exemplos a serem citados são: correspondências, pesquisa DataFolha, pergaminhos, dentre outros.

A *fonte secundária* tem a premissa de que os dados já foram modificados ou apurados para o mercado. Como exemplo: sínteses, enciclopédias, dicionários, etc.

Por fim, a *fonte terciária* conecta as fontes primárias e secundárias ou uma soma das duas fontes de pesquisa. Exemplos seriam os guias de leituras, catálogos e manuais [7].

3.2. TIPOS DE DADOS

Na atual era digital na qual nos encontramos, todo resultado de pesquisa é facilitado pela tecnologia disponível. Consequentemente, é gerada uma quantidade expressiva de dados, os quais são heterogêneos e, dessa forma, implicam na necessidade de classificação. [8]

Quando se coleta e manipula informações, temos que ter em mente que o ponto focal do nosso trabalho é o dado coletado. Porém, é notório que existem diferentes tipos de informações, os quais representam os vários aspectos da amostra que temos. Isso tende a aumentar a complexidade de análise, uma vez que deve-se determinar as técnicas de classificação a partir do tipo de dado considerado. Desta forma, temos algumas categorias e subcategorias que agrupam esses tipos distintos, conforme ilustra a figura 1, abaixo. [9]

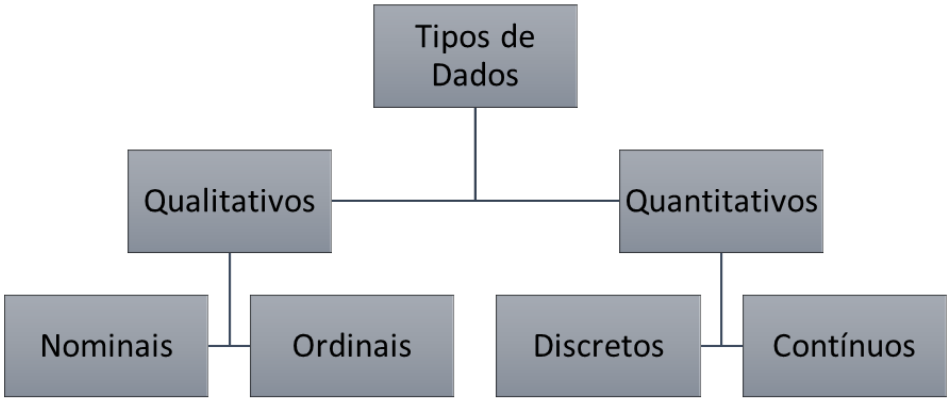

Figura 1: Tipos de Dados

3.2.1. QUALITATIVOS

Os dados qualitativos são aqueles que não podem ser definidos numericamente. Desta forma, eles exprimem a ideia de categorias, ou seja, classificam as características que a amostra define. Eles se dividem em Nominais e Ordinais.

Nominais são os dados que não possuem relação de ordem dentre suas categorias, como sexo, cor, especialidade médica.

- Sexo: Masculino, feminino, etc.
- Cor: Azul, Vermelho, Verde, etc.
- Especialidade Médica: Cardiologia, Pediatria, Endocrinologia, etc.

Ordinais são aqueles que possuem a ideia de ordem entre suas categorias, como por exemplo escolaridade, mês observado, porcentagem concluída.

- Escolaridade: 1º Grau, 2º Grau, 3º Grau, etc.
- Mês Observado: janeiro, fevereiro, março, etc.
- Porcentagem Concluída: 10%, 25%, 80%, etc. [10]

3.2.2. QUANTITATIVOS

Os dados quantitativos, como o próprio nome já diz, expressam a ideia de número, como idade, distância, peso, saldo, temperatura, dentre outros. Eles podem se subdividir em Discretos e Ordinais.

Discretos são aqueles que quantificam valores limitados e inteiros, ou seja, que podem ser enumerados, como o número de carros em um estacionamento ou a idade de uma pessoa.

- Idade: 34 anos
- Vagas do Estacionamento: 300 vagas
- Quantidade em estoque: 80 peças

Ordinais são aqueles que podem assumir quaisquer valores em um intervalo infinito no conjunto dos Reais, como peso ou altura.

- Altura: 1,75 m

- Saldo Bancário: R$ 3210,56
- Peso: 84,65 Kg

Um fato importante a ser considerado é que os valores observados serão sempre números nesse tipo de dado [11].

3.3. ORGANIZAÇÃO E ARMAZENAMENTO

Conforme os dados são extraídos, eles devem ser armazenados e organizados de modo a terem uma melhor apresentação, principalmente quando se trata de análise de dados. Dessa forma, existem três tipos de organização desse fluxo de informações: Estruturados, Semiestruturados e Não-estruturados, ilustrados da figura 2, a seguir:

Figura 2: Tipos de organizações de dados [12]

Os **Dados Estruturados** podem ser exemplificados como os bancos de dados relacionais, os quais possuem tabelas organizadas a partir dos atributos de suas linhas e colunas. Dessa forma, elas possuem regras e, assim sendo, permitem pouca (ou nenhuma) mutabilidade no seu esquema de organização. Portanto, tomando como base um veículo, os atributos representam as diversas características que ele possui e obedecem às regras pré-estabelecidas para cada um deles. A seguir, uma breve descrição de alguns deles.

- **Ano**: representa o ano que o veículo foi fabricado, logo esse atributo pode ser mutável, desde que represente um ano válido, ou seja, segue uma lei de formação.
- **Placa**: este atributo possui regras específicas e, desta forma, permite um conjunto mais seleto de caracteres, uma vez que se deve ter necessariamente uma combinação de alfanuméricos em posições específicas.
- **Cor**: já este atributo representa cores, cujos nomes já estão definidos.
- **Chassi**: este é um atributo único para cada veículo, portanto, ele é imutável. Além disso, cada chassi representa um único veículo, ou seja, cada instância de um banco de dados relacional. [10]

Os **Dados Semiestruturados,** apesar de não seguirem uma lei de formação para seu conteúdo, eles tendem a tomar forma a partir de um padrão construtivo, por consequência, permitem grande mutabilidade. Podemos citar como exemplo arquivos XML, JSON, dentre outros, os quais são fáceis de visualizar, além de marcações que auxiliam a delimitação, identificação e manipulação das informações [13].

Os **Dados Não-Estruturados**, diferentemente dos anteriores, não seguem nenhuma lei de formação e, portanto, não oferecem nenhum tipo de organização ou ordem para serem armazenados. Como exemplo, podemos observar arquivos como os TXT, JPEG ou MKV. Os TXT podem conter quaisquer tipos de textos, inclusive em diversos idiomas, enquanto os outros dois formatos citados podem assumir quaisquer tamanhos, conteúdo, resolução, etc. Dessa forma, esses dados são mais difíceis de serem analisados e processados. [14]

3.4. PÁGINAS WEB

As páginas webs tiveram e tem como objetivo formalizar informações de uma maneira global onde todo mundo possa entender o que está sendo compartilhado e assim fazer as interpretações e ajustes necessários. O surgimento das páginas se deve ao Tim Berners-Lee que desenvolveu a World Wide Web em 1989 para uma iniciativa de hipermídia baseada na internet para compartilhar informações globais no

CERN (Conseil Européen pour la Recherche Nucléaire). Foi criado o primeiro cliente web e servidor em 1990, usando as especificações de URIs, HTTP e HTML [15].

O HTML significa *Hyper Text Markup Language* que, em tradução livre para o português, seria algo como *Linguagem para Marcação de Hipertexto*. Ele já passou por diversas atualizações desde HTML, HTML 2.0, HTML 3.0 e o atual, que é conhecido como HTML5 e é usado desde 2014. Possui a premissa de facilitar a manipulação dos elementos e, assim, permitir desenvolver modificações dos objetos de maneira que não atrapalhe a formatação e transparência final da página para o usuário final [16].

3.4.1. DOCUMENTOS HTML, TAGS E ATRIBUTOS

O HTML funciona a partir das Tags que direcionam o navegador onde um elemento começa e finaliza, já os atributos descrevem os detalhes de cada um dos elementos. Esses elementos são divididos em três partes:

- **Tag de abertura -** Diz onde um elemento começa. Sendo representada por colchetes abertos e fechados.
  Exemplo: *(início de um parágrafo)*
- **Conteúdo -** Onde o conteúdo fica visível ao público.
- **Tag de fechamento -** Segue a mesma lógica da tag de abertura, porém com uma barra anterior ao nome do elemento.
  Exemplo: (finaliza um parágrafo). [17]

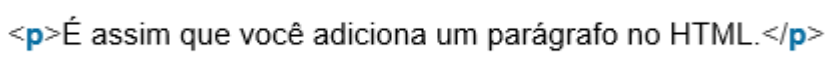

Figura 3: Modelo de Tag no HTML

A estrutura do HTML possui uma estrutura padrão como mostra a figura 4, abaixo:

```
1<!DOCTYPE html>
2<html>
3<head>
4        <meta charset="utf-8" />
5        <title>Aula 1 - PHP</title>
6</head>
7<body>
8
9</body>
10</html>
```

Figura 4: Estrutura padrão do documento HTML [18]

A tag <html> representa a estrutura principal do documento. Ela desenvolve-se em forma de árvore, tendo elementos filhos dentro do elemento pai.

Enquanto a tag <head> particiona as informações do documento HTML que não são visíveis para o usuário/leitor do documento. Tratam-se de dados implícitos, de uso e controle do documento, onde fica toda a parte inteligente da página que ficam os metadados que são informações sobre a página e o conteúdo ali publicado.

A tag <body> representa todo o conteúdo que é visível na página. Tem como seguintes elementos das categorias “Sectioning”, “Phrasing”, “Embedded”, “Interactive”. Os elementos que estão dentro da tag <body> Elemento </body> são visíveis para o usuário. Assim como a tag <head>, ela também deve estar dentro das tags <html ></html>. [17]

Os usos dessas tags são ilustrados na figura 5.

index.php | meuestilo.css | index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5
6          <link rel="stylesheet" type="text/css" href="meuestilo.css">
7
8          <title>Título</title>
9      </head>
10
11     <body>
12         <-- NOSSAS TAGS VIRÃO AQUI -->
13     </body>
14 </html>
```

Figura 5: Exemplos de Tags em HTML [19]

Por fim, temos os atributos, que são palavras que contém características especiais que o site necessita dentro da tag de abertura para assim gerar uma função aos elementos. Como exemplos de atributos podemos citar *class, href, id, src, alt, target* [20].

```
<h1 class="titulo">Mergulhe em Tecnologia!</p>
```

Figura 6: Exemplo de um atributo classe

3.4.2. DOCUMENT OBJECT MODEL (DOM)

O *Document Object Model* (DOM) é uma amostragem de dados dos objetos que compõem a estrutura e o conteúdo de um documento na web. Ele representa o documento como nós e objetos e, dessa forma, as linguagens de programação podem interagir com a página. Assim sendo, uma página da Web será exibida na janela do navegador ou como fonte HTML. Em ambas as situações, será o mesmo documento, mas a representação do *DOM* permite que ele seja manipulado. Além disso, este modelo pode ser exibido como uma representação orientada a objetos da página da Web e, portanto, ela pode ser modificada com uma linguagem de script, como JavaScript [21].

As vantagens da utilização do DOM é que se pode atualizar os dados das aplicações criadas, além de permitir também que os usuários customizem a página sem necessidade de atualização [22].

Os principais objetos a serem manipulados são “document”, “element”, “atribute” e “text”. A estrutura do DOM é mostrada na figura 7 abaixo:

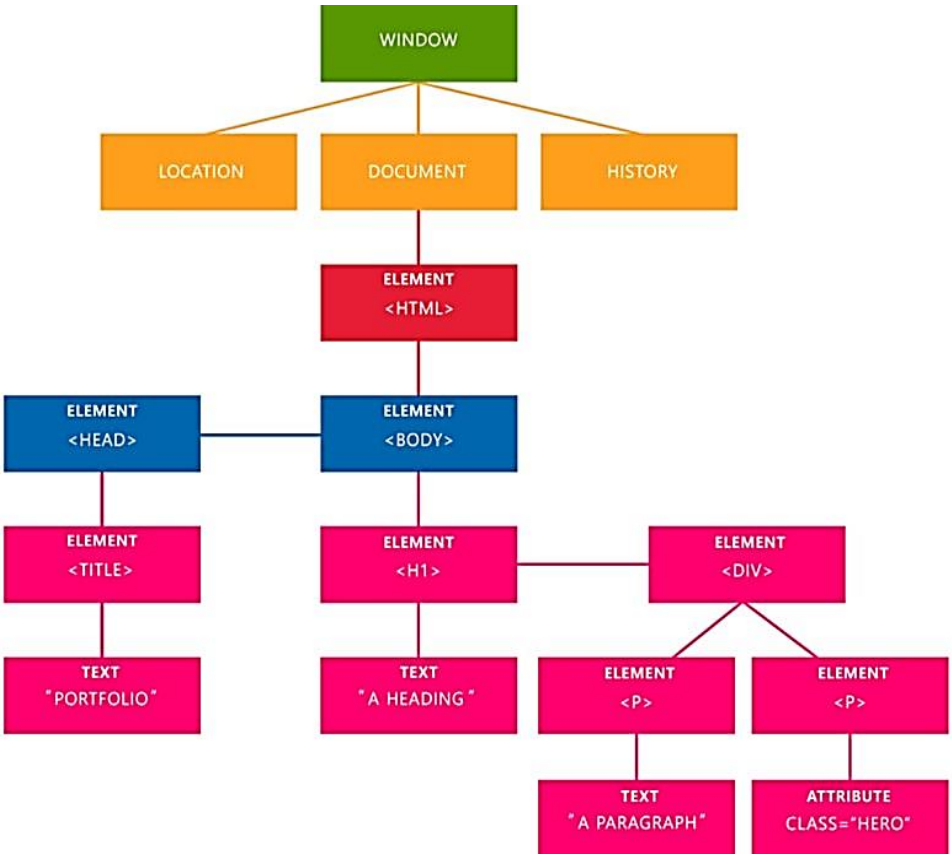

Figura 7: Exemplo da estrutura de uma página com a estrutura DOM [23]

Podemos dividir a estrutura do DOM da seguinte forma de acordo com suas finalidades:

- **Document**: cuida de documentos HTML.
- **Elements**: são todas as tags que estão em arquivos HTML ou XML e se transformam em elementos da árvore DOM.
- **Texts**: É o texto que se localiza entre os elementos, todo o conteúdo das tags.
- **Attributes**: É a junção de todos atributos para um nó específico.

Alguns métodos mais conhecidos do objeto “document” são: getElementByID(), getElemenstByClassName(), getElementsByTagName(). Todavia, o DOM também disponibiliza métodos que executam seleções utilizando os seletores CSS, como: querySelector() e o querySelectorAll(). [21]

3.5. NAVEGADORES WEB

Para a navegação na web, os usuários necessitam de uma interface que faça o acesso às páginas que estes desejam. Esta interface é o Navegador Web, que trabalha traduzindo as informações que trafegam entre o servidor e o usuário exibindo-as na tela. Como exemplos, podemos citar os famosos Microsoft Edge, Google Chrome, Mozilla Firefox, Opera Web Browser.

A Rede Mundial de Computadores (ou WWW - Wide World Web) armazena todo conteúdo navegável, portanto cada acesso a cada recurso na web, que podem ser diversos tipos, é identificado com um identificador único chamado de URI (Uniform Resource Identifier). Já os arquivos e documentos acessados são escritos em HTML, que possibilita ao autor mesclar hiperlinks de outros documentos e fontes no seu próprio. Esses dados são transferidos por um tipo de protocolo chamado HTTP (*Hyper-Text Transfer Protocol*), um meio de comunicação e troca de informações. Porém, para que um processo seja efetuado, é necessário que seja estabelecida uma conexão do tipo TCP/IP.

Além disso, para traduzir os endereços entre cliente e servidor, a WEB utiliza o sistema de DNS, que traduz, de fato, os endereços de IP em URLs e vice-versa. Dessa forma, o usuário não precisa saber cada IP que irá acessar. A seguir, a figura 8 ilustra como ocorre o processo de requisição HTTP feito pelo navegador WEB.

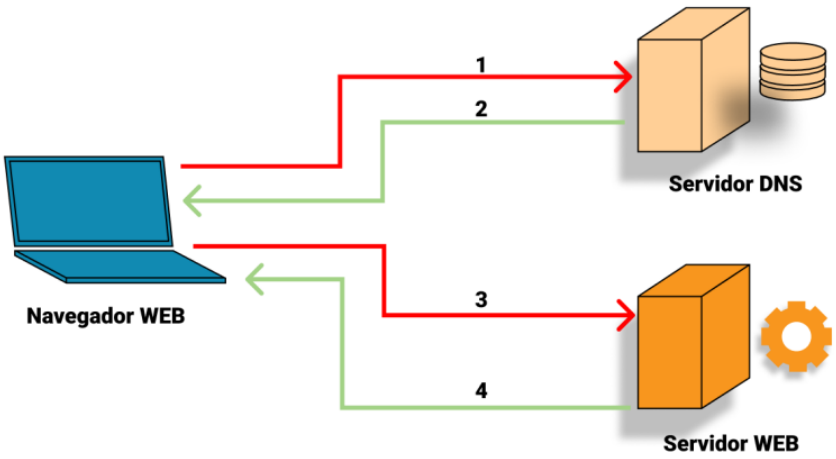

Figura 8: Requisição HTTP

No estado 1, ao digitar o endereço, o navegador solicita o respectivo IP para um servidor DNS. Em 2, o servidor DNS responde a solicitação com o endereço respectivo. Já em posse do IP, no estado 3, o navegador WEB se conecta a um servidor na web e solicita o recurso. Por fim, no estado 4, o servidor na WEB devolve a requisição com o recurso. [24]

3.6. WEB SCRAPING

O Web Scraping tornou-se uma ferramenta excelente nos dias de hoje. Ela consiste em realizar uma coleta de dados, a partir da internet, de forma automatizada através de técnicas que requisitam dados de servidores web para que estes possam ser extraídos e, até mesmo, armazenados de seus respectivos documentos HTML. É possível, também, analisar e estruturar as informações adquiridas. A este procedimento chamamos de *Parsing* (do inglês, análise). O contrário acontece no acesso à web através dos navegadores, onde é utilizada a formatação HTML, CSS e JavaScript para que os dados sejam apresentados. Mas, apesar disso, podemos lançar mão desses recursos para a aquisição desses dados.

De acordo com a definição de Ryan (2019), “Na prática, Web Scraping engloba uma grande variedade de técnicas de programação e de tecnologias, por exemplo, análise de dados, parsing de idiomas naturais e segurança da informação.” [25]

Diversos autores definem o web scraping como um procedimento que acontece em fases diferentes. A primeira fase é onde são recuperadas informações do servidor Web, através de requisições que são cedidas pelos próprios servidores. Logo após, estes dados são armazenados, usualmente XML e JSON, para que possam ser analisados posteriormente. Ilustrado na figura 9, a seguir:

Figura 9: Extração de dados por Web Scraping [26]

A segunda fase consiste na extração propriamente dita, além da análise (parsing) destes dados extraídos. Nesse momento são utilizadas diversas ferramentas, como bibliotecas e funções específicas, que possibilitam a estruturação das informações obtidas [27].

4. DESENVOLVIMENTO

Neste capítulo será abordado o uso do WordPress (WP) e API REST. A aplicação como um todo foi utilizar a linguagem Python para coleta de dados automaticamente. Onde será demonstrado os casos de uso e regra de negócio para melhor compreensão do trabalho.

4.1. API REST - CONCEITOS DE API REST E API RESTFUL

Devido ao grande aumento de dados de forma pública e privada permitindo que usuários pudessem ter acesso no mesmo. Surgiu a necessidade de uma ferramenta que auxiliasse essa coleta de forma segura, rápida e eficiente. Dessa forma, surgiram APIs como API REST e API RESTFUL para auxiliar nisso.

O acrônimo API é a abreviação de ***Application Programming Interface,*** que significa **"Interface de Programação de Aplicações".** Ou seja, um conjunto de rotinas e padrões já documentados para que um determinado software tenha acesso acessível para utilizar as funcionalidades da API, sem necessidade de conhecer o back-end do mesmo [28].

Dessa forma, garante segurança de código e, principalmente, das regras de negócio do software, onde a comunicação ocorre utilizando as requisições HTTP que é responsável por manipular dados [29].

API REST e API RESTFUL têm funções diferentes, mas são complementares conforme é descrito abaixo. Sendo que a sigla REST ao juntar-se com o termo API tem como significado (Representational State Transfer ou transferência representacional de estado), utilizando o protocolo HTTP e os arquivos JSON e XML.

A arquitetura REST API definida abaixo:

- Não possui criptografia;
- Não possui sessão;
- Utiliza apenas o protocolo HTTP;

- O acesso dos recursos devem ser somente usando o protocolo de internet URI (Uniform Resource Identifier);
- Os dados retornados são por: JSON e XML;
- API RESTFUL segue todos os dados citados anteriormente. Ou seja, implementa toda arquitetura API REST [30].

A arquitetura API REST possui um conjunto de regras e princípios que devem ser seguidos conforme mostrado na tabela 1.

Tabela 1: Arquitetura REST

Princípios Fundamentais	Descrição
Cliente-Servidor	Responsável por separar responsabilidades, ou seja, separar usuários (User Interface) do banco de dados, obtendo a dependência entre os lados cliente/servidor.
Interface Uniforme	Interação entre os componentes cliente e servidor. Onde estes compartilham a mesma interface, e assim, é necessário estabelecer um meio para comunicação entre eles. Relação correta dos verbos HTTP: **GET, POST, PUT, DELETE,** e etc.
Stateless	A requisição de cada uma é adicionada entre cliente-servidor possui toda informação necessária e compreensível para realizar a requisição. Devido a isso, pode gerar alto tráfego de dados e impactar no desempenho da aplicação utilizando recursos de cache para resolver esses problemas.
Cache	Tem objetivo de melhorar a performance de comunicação entre aplicações, diminuindo o tempo de resposta na comunicação entre cliente-servidor.

Camadas	Nas boas práticas da arquitetura e design de um projeto, recomendam a construção de camadas independentes e auto gerenciadas, onde as camadas entre elas não se reconhecem. Assim, uma mudança em uma camada não interfere nas demais. Neste modelo, o cliente não poderá conectar-se diretamente ao servidor de aplicação, porém uma camada de balanceamento de carga resolve este problema ao ser adicionado.

Essa tabela demonstra os 5 princípios fundamentais do REST com suas devidas descrições e assim é denominada de RESTFul por usar todas suas funcionalidades.

A figura 10 resume todo o apanhado geral sobre API REST.

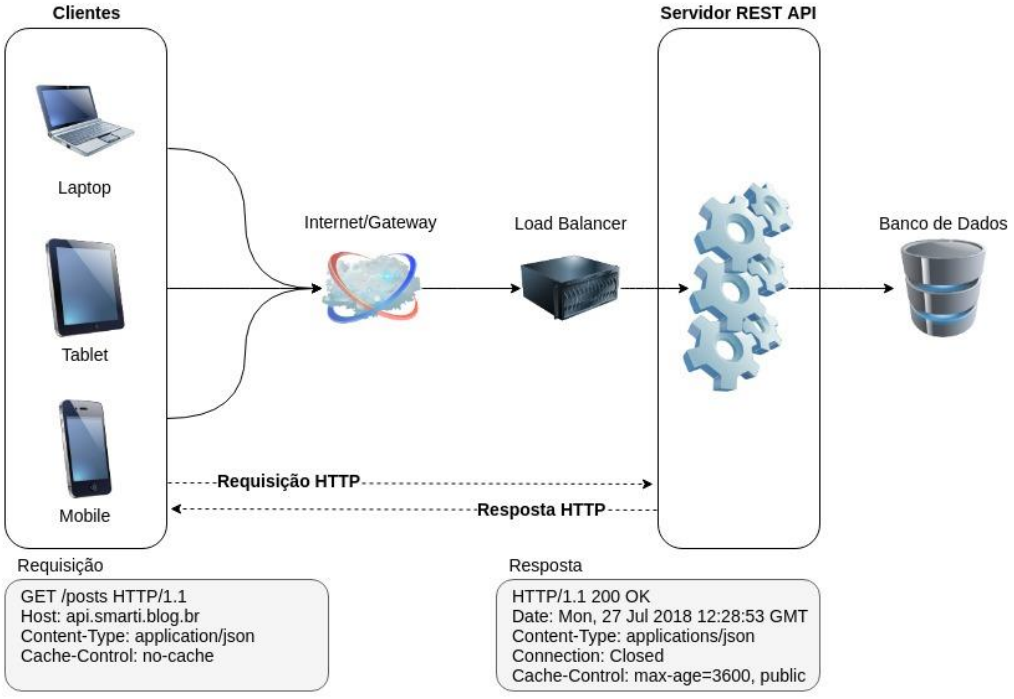

Figura 10: Arquitetura REST [31]

4.2. WORDPRESS

A API REST WordPress é baseada em torno do REST e foi construída para ter URLs previsíveis e orientada a recursos que usam HTTP como resposta para apontar erros de APIs. A API usa ferramentas HTTP integrados, como **autenticação e verbos HTTP**, que podem ser entendidos por clientes HTTP preparados para serem usados e suporta o compartilhamento de recursos de origem cruzada para permitir que tenha interação com segurança com a API da aplicação do client-side web.

Os métodos HTTP mais usados são mostrados na tabela 2.

Tabela 2: Métodos HTTP

Verbo HTTP	Descrição	Resposta
GET	Trazer informações que estão no banco de dados.	Status Code 200 – retorna os registros no formato solicitado.
POST	Adicionar ou criar informações no banco de dados.	Status Code 201 – registro criado.
PUT	Atualizar informações no Banco de dados	Status Code 200 – registro atualizado.
DELETE	Deletar informações no banco de dados.	Status Code 204 – registro deletado

A API REST fornece dados públicos que qualquer pessoa pode acessar e bem como dados privados disponíveis apenas após a autenticação [32].

A distribuição APIs REST WordPress tem sua distribuição gratuita e disponível para cada site que a suporta. Ou seja, não há outra conexão igual da API para conectar entre eles; no entanto, existe um processo que permite interagir com sites que não tenham conexão prévia.

No entanto, se desejar usar plugins, temas ou aplicativos externos com uma aplicação Javascript do lado cliente, ou um programa na linguagem diferente do PHP irá precisar usar uma ferramenta estruturada e flexível que assim, permite ao programador ou "leigos" criar plataformas com menos programação e com mais

plugins de alto desempenho e de forma segura em qualquer linguagem de programação back-end (Java, Python, Swift, Kotlin e muitos outros) [33].

4.2.1. ENDPOINTS DO WORDPRESS API

Este item apresenta os endpoints que foram utilizados no trabalho, eles estão disponíveis no WordPress API conforme mostrado na tabela 3 [34]. Os endpoints foram liberados em 2015 pela equipe de desenvolvimento do WordPress para o público geral criar seu site de interesse utilizando recursos de API.

Tabela 3: Referência de Endpoint da API REST

Resource	Base Route
Posts	**/wp/v2/posts**
Comments	**/wp/v2/comments**

A seguir, uma breve abordagem dos endpoints citados.

- **POST:** O endpoint “Posts” recupera uma coleção de posts. A resposta que recebida pode ser controlada e filtrada usando os parâmetros de consulta de URL, seguindo alguns argumentos, como *page* e *per_page*, por exemplo [35].

  Exemplo Request:

  https://example.com/wp-json/wp/v2/posts
- **COMMENTS:** O endpoint “Comments” recupera uma coleção de comentários. Da mesma forma do endpoint anterior, a resposta obtida pode ser controlada e filtrada usando parâmetros similares de consulta de URL [36].

  Exemplo Request:

  https://example.com/wp-json/wp/v2/comments

Neste trabalho foram usados somente dois endpoints para testes de sites escolhidos que atendem os recursos citados acima. Em projetos futuros, podem ser

utilizados outros endpoints e até mesmo exploradas melhores formas de serem utilizados estes endpoints.

4.3. CASOS DE USO

Os endpoints neste trabalho terão uma breve descrição de toda arquitetura de software desenvolvida em Python para aplicação em sites criados em WP. A demonstração será por meio de casos de uso.

4.3.1. DIAGRAMA DE CASOS DE USO "WordPress API"

Para melhor compreensão dos requisitos em WordPress, demonstramos o fluxo do diagrama de caso de uso na figura 11.

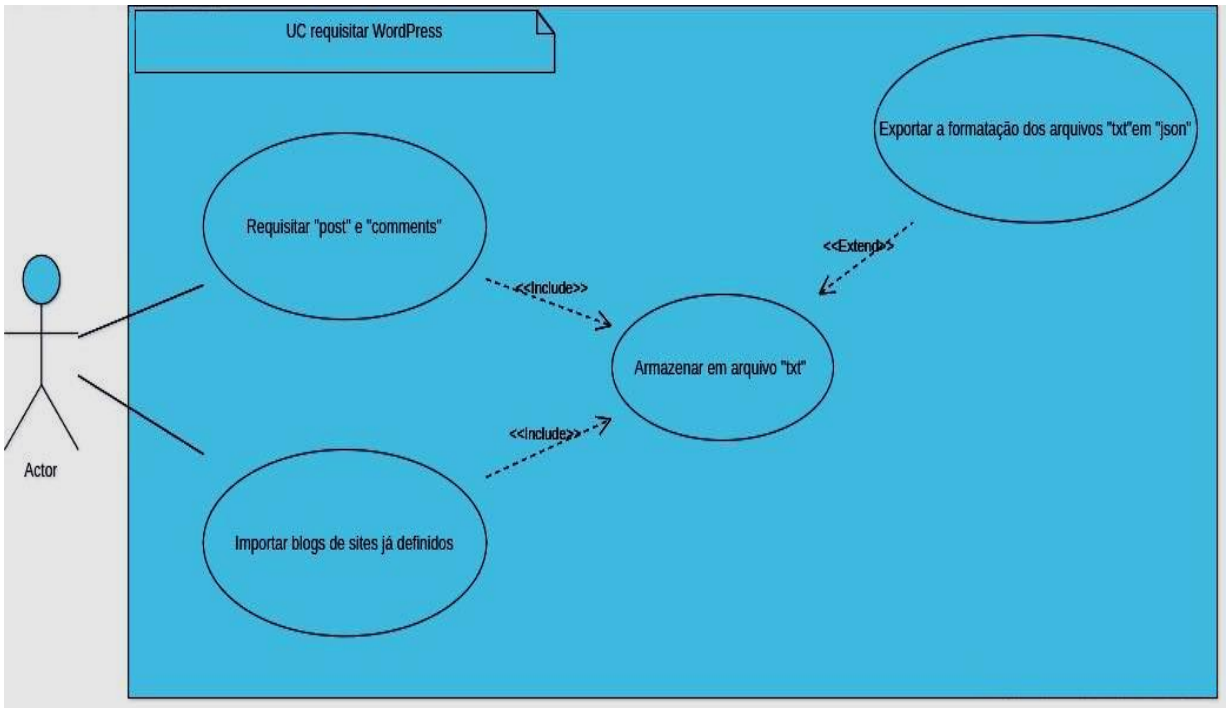

Figura 11: Caso de Uso

Para facilitar o entendimento sobre os fluxos e regras de negócios do caso de uso, exibimos na Tabela 4, a descrição textual deste caso de uso.

Tabela 4: Descrição de Caso de uso "WordPress API"

ID:	**UC01**
Objetivo:	Extrair, de uma determinada página, a partir de um endpoint de interesse.
Requisitos:	Conexão com a internet e IDE que interprete a linguagem Python na versão 3.
Atores:	Usuário
Pré-Condições:	Os pacotes e linguagens deverão ser instalados no ambiente de desenvolvimento.
Pós-Condições:	As informações serão armazenadas em arquivo texto (*.txt)
Fluxo Principal:	1. O Usuário cria os objetos *Site* para cada site de interesse. 2. O Usuário cria os objetos *Scraping* para cada endpoint de interesse. 3. A ferramenta faz as requisições para cada endpoint. 4. Cada busca no blog é salva em um arquivo texto.
Erros/Exceções:	Os erros não são tratados pelo usuário.
Regras de negócio:	**[RN01] –** Os dados de entrada já estão definidos nas classes Site e Scraping para serem utilizados no programa main.py. **[RN02] –** O programa irá executar as requisições. [**RN03] –** O programa recebe os dados e salva em um arquivo (*.txt) para cada site e endpoint configurado.

4.4. DIAGRAMA DE CLASSES

Foi utilizada a técnica de Programação Orientada a Objetos para modelar os dados das Classes. Foram utilizadas as classes *Scraping* e *Site*, ficando a primeira

a cargo dos métodos que realizam as requisições nas fontes a serem exploradas e a saída para um arquivo de texto (TXT) e a segunda armazenando as informações necessárias das fontes a serem pesquisadas.

O diagrama de classes está apresentado a seguir, na figura 10.

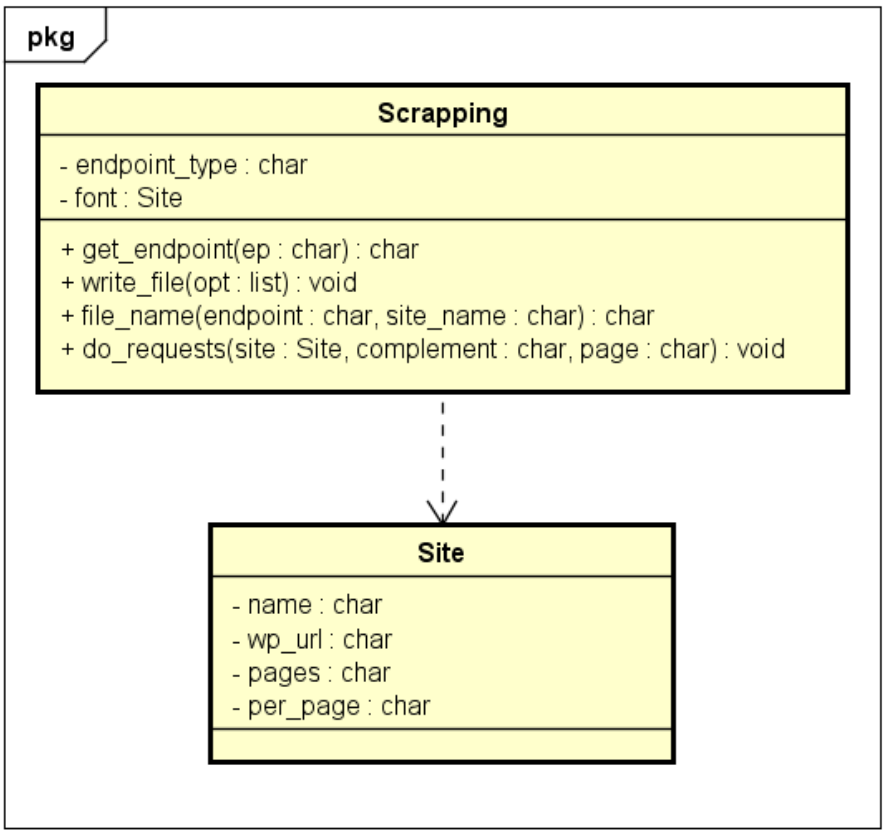

Figura 12: Diagrama de Classes

A classe *Scrapping* possui dois atributos:

- endpoint_type: armazena o tipo do endpoint que será utilizado no web scraping.
- font: armazena os dados das instâncias de cada uma das fontes.

Além disso, essa classe possui também os seguintes métodos:

- get_endpoint: este método é o responsável por buscar, em um dicionário pré-definido, o complemento da url que fará o scrapping.
- write_file: sua função é escrever os dados, recebidos da requisição, em um arquivo de texto no formato TXT.

- file_name: responsável pela formatação do nome do arquivo de saída de dados.
- do_requests: é o método principal, que realiza as requisições do end-point escolhido.

Já a Classe Site, possui quatro atributos referentes a cada site que será utilizado como fonte no desempenho das tarefas:

- name: armazena o nome identificador da página.
- wp_url: possui a url da página.
- pages: determina a quantidade total de páginas.
- per_page: retém a quantidade de posts por página que a fonte possui.

4.5. EXTRAÇÃO DE WORDPRESS

A extração de dados nos sites WP se dá de forma direta, sem a necessidade de uma conta ou token. Desta forma, é possível utilizar recursos nativos do WordPress para recuperar informações relevantes de interesse em blogs selecionados [34].

Assim sendo, para cada endpoint que se almeja fazer a requisição temos um complemento, que será utilizado para complementar a url selecionada. Os dados recebidos serão no formato JSON e poderão ser utilizados para posteriores análises e classificações.

5. TESTES

Este capítulo será destinado a uma breve avaliação da capacidade da ferramenta desenvolvida através dos resultados obtidos com sua execução e em função dos recursos disponíveis de hardware.

Para executar a ferramenta, foi utilizado um microcomputador com a configuração apresentada na tabela abaixo.

Tabela 5: Configuração do Hardware utilizado

CPU	Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz / 2.60 GHz
Memória RAM Instalada	8,00 GB
Tipo de Sistema	Sistema operacional de 64 bits, processador baseado em x64
SO Instalado	Windows 10 Pro
Edição	21H2
Instalado em	23/07/2020
Compilação do SO	19044.1766
Experiência	Windows Feature Experience Pack 120.2212.4180.0

Para a execução dos testes foram utilizados três sites WordPress de assuntos variados, apresentados na Tabela 6. As informações de cada site foram armazenadas em objetos do tipo Site e foram utilizados para serem feitas as requisições necessárias. Cabe ressaltar que a ferramenta desenvolvida para blogs cuja estrutura de paginação não utiliza mecanismos de atualização de páginas, como Javascript ou similar.

Tabela 6: Nomes e URLs dos sites consultados

Nome	URL
Hospício Nerd	https://www.hospicionerd.com.br
World Education Blog	https://world-education-blog.org
Blog do AFTM	https://blogdoaftm.com.br

Utilizando-se identificadores para POSTS e COMMENTS na estrutura dos dados recebidos, foi possível quantificá-los, conforme mostrado no Gráfico 1, a seguir. Nota-se que não há relação direta entre quantidade de Posts e Comments, sendo estes independentes um do outro.

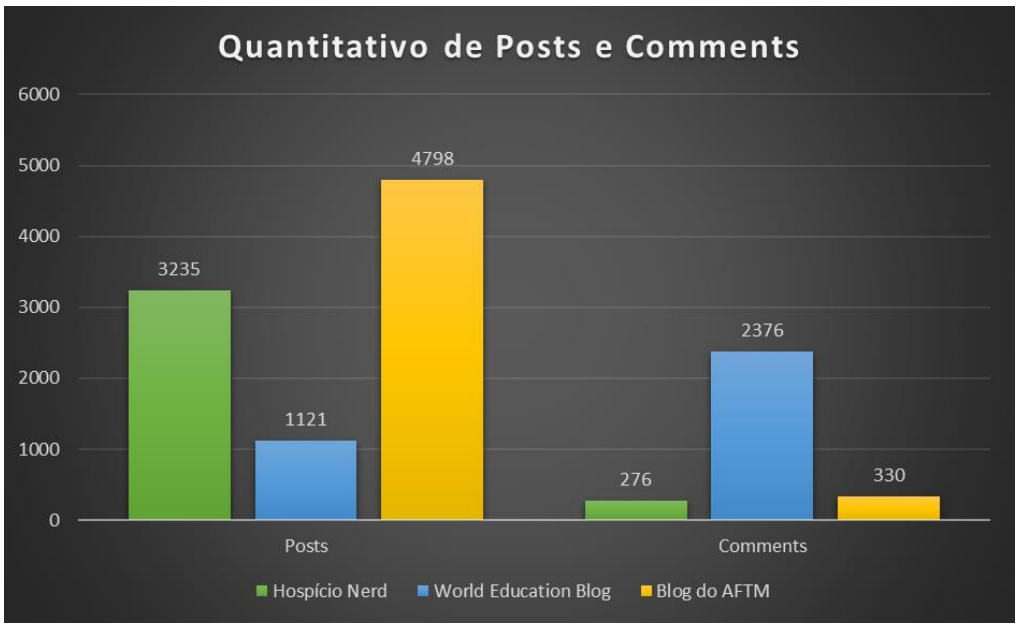

Gráfico 1: Quantitativo de Posts e Comments

Além disso, foi possível mensurar o tempo de execução de cada requisição e endpoint, mostrado no Gráfico 2. Nele percebemos que o tempo não está diretamente ligado ao número de Posts ou de Comments, tendo em vista que mesmo com um volume grande de dados recebido o tempo verificado foi variável para cada requisição.

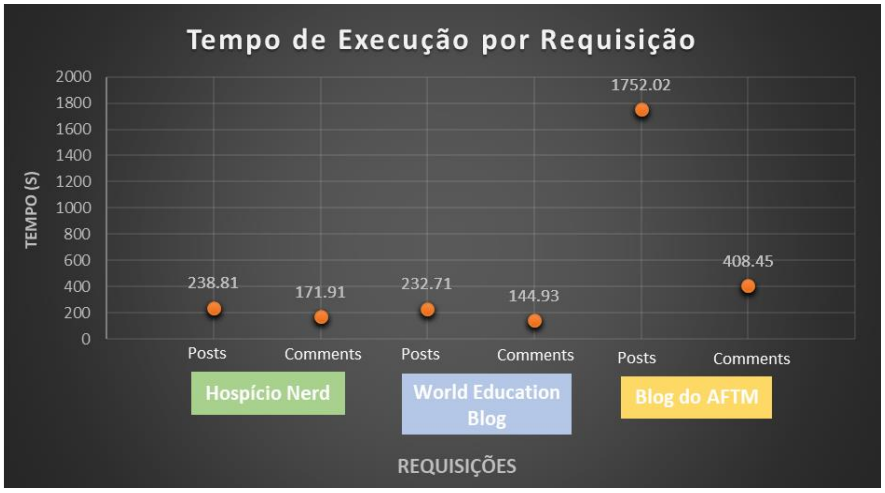

Gráfico 2: Tempo de Execução por Requisição

Todos os dados gerados ficam armazenados em arquivos de texto que podem ser analisados e classificados posteriormente em estudos futuros. É importante observar que para alguns casos de blogs pesquisados foi encontrado um bloqueio como medida de proteção aos blogs consultados, com isso serão gerados erros que não serão tratados por não estarem no contexto do trabalho.

6. CONCLUSÕES E TRABALHOS FUTUROS

A utilização de blogs WP para assuntos de interesse geral são ferramentas de uso intensivo na Internet, haja vista que quase 40% dos sites do mundo são escritos em WP [37]. Com base nisso, o objetivo central do trabalho foi coletar um grande volume de dados sobre algum tema específico a partir desses blogs WP, mostrando sua performance de forma rápida usando os métodos de programação em Python.

Após a obtenção e averiguação das informações, foi possível também confirmar a importância da criação de métodos e procedimentos para a extração e Web Scraping, de forma a agilizar e automatizar uma tarefa que tende a ser massiva se feita manualmente. Além do mais, pode-se garantir uma maior acurácia no desenrolar das atividades.

Não foi desenvolvida uma interface gráfica neste trabalho, tendo em vista que não faz parte do escopo atual. Além disso, podem ser explorados mais endpoints e seus argumentos em trabalhos futuros, de forma a complementar a pesquisa e dar mais robustez à proposta original.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ma, L., Wang, Z., Zhang, Y. (2017). **Extracting Depression Symptoms from Social Networks and Web Blogs via Text Mining**. In: Cai, Z., Daescu, O., Li, M. (eds) Bioinformatics Research and Applications. ISBRA 2017. Lecture Notes in Computer Science(), vol 10330. Springer, Cham.

[2] Suganya, E., Vijayarani, S. (2020). **Sentiment Analysis for Scraping of Product Reviews from Multiple Web Pages Using Machine Learning Algorithms**. In: Abraham, A., Cherukuri, A., Melin, P., Gandhi, N. (eds) Intelligent Systems Design and Applications. ISDA 2018 2018. Advances in Intelligent Systems and Computing, vol 941. Springer, Cham.

[3] Szepannek, Gero, Westphal, Laila, Gronau, Werner and Lehmann, Tine. "**Using a sentiment analysis for the examination of tourism blogs – a step by step methodological reflection process**" *Zeitschrift für Tourismuswissenschaft*, vol. 13, no. 2, 2021, pp. 167-190.

[4] Varela, N., Lezama, O.B.P., Charris, M. (2021). **Web Scraping and Naïve Bayes Classification for Political Analysis**. In: Pandian, A.P., Palanisamy, R., Ntalianis, K. (eds) Proceedings of International Conference on Intelligent Computing, Information and Control Systems. Advances in Intelligent Systems and Computing, vol 1272. Springer, Singapore.

[5] Barbosa, Eduardo F., **Instrumento de Coleta de Dados em Pesquisa**, Belo Horizonte, CEFET-MG, 1999.

[6] Salsa, Ivone da Silva; Moreira, Jeanete Alves. **Probabilidade e estatística** – 2ª ed. – Natal: EDUFRN, 2014.

[7] Magalhães, Marcos N.; Lima, Antônio Carlos P. de. **Noções de probabilidade e estatística**, São Paulo: Editora da Universidade de São Paulo, 2002.

[8] Sales, Luana Farias; Sayão, Luís Fernando. **Uma proposta de taxonomia para dados de pesquisa**, Conhecimento em Ação, v. 4, n.1 Rio de Janeiro, 2019.

[9] Filho, Abrantes Araújo Silva. **Tipos de dados e escalas de mensuração: explicando a confusão semântica**, 2021.

[10] Zozus, Meredith. **The Data Book: Collection and Management of Research Data**, New York: Chapman and Hall/CRC, Abril, 2017.

[11] Baruffi, Aline; Borges, Janaina Brum Gularte; Tozetto, Ricardo Schleder. **Mensuração e Escalas**. Abril, 2014.

[12] Disponível em: https://universidadedatecnologia.com.br/dados-estruturados-e-nao-estruturados/
Acessado em: 07/03/2022.

[13] Abiteboul, S., Buneman, P., & Suciu, D. (1999). **Data on the Web: From Relations to Semistructured Data and XML**. Morgan Kaufmann.

[14] Amaral, Fernando. Introdução à Ciência de Dados: Mineração de Dados e Big Data. Rio de Janeiro: Alta Books, 2016.

[15] Berners-Lee, Tim; Cailliau, Robert; Luotonen, Ari; Nielsen, Henrik Frystyk; Secret, Arthur. **The World-Wide Web**, Communication of the ACM, v. 37, nº 8, agosto, 1994.

[16] Ferreira, Elcio; Eis, Diego. **HTML5 - Curso W3C Escritório Brasil**

[17] Silva, Maurício Samy. **Fundamentos de HTML5 e CSS3**, Novatec Editora, junho, 2015.

[18] Disponível em: https://dmx3002a.wordpress.com/2015/07/07/estrutura-padrao-html-arquivo-php/
Acessado em: 06/06/2022.

[19] Disponível em: https://www.todoespacoonline.com/w/2014/04/site-com-html-e-css/
Acessado em 06/06/2022.

[20] Costa, Carlos J. **Desenvolvimento para WEB**, ITML Press/Lusocredito, 2007.

[21] Rascia, Tania. **Understanding the DOM – Document Object Model**, Digital-Ocean, New York, outubro, 2020.

[22] Marini, Joe. **The Document Object Model – Processing Structured Documents**, McGraw-Hill Education, julho, 2002.

[23] Disponível em: https://tableless.com.br/entendendo-o-dom-document-object-model/
Acessado em 06/06/2022.

[24] Grosskurth, A., Godfrey, M. W. **"A reference architecture for Web browsers"**, 21st IEEE International Conference on Software Maintenance (ICSM'05), 2005, pp. 661-664.

[25] Mitchell, Ryan. **Web Scraping com Python: Coletando Mais Dados da web Moderna**, Novatec Editora, 2019.

[26] Disponível em: https://www.linkedin.com/pulse/web-scraping-para-an%C3%A1lise-de-dados-grimaldo-oliveira/?originalSubdomain=pt
Acessado em: 10/04/2022.

[27] Disponível em: https://azati.ai/how-much-does-web-scraping-cost-in-2019/
Acessado em: 10/04/2022.

[28] Masse, Mark. **Rest API Design Rulebook: Designing Consistent Restful Web Service Interfaces**. Editora O'Reilly Media, 2011.

[29] Woods, Dan; Jacobson, Daniel; Brai, Gregory. **APIs: A Strategy Guide: Creating Channels with Application Programming Interfaces.** Editora O'Reilly Media, 2011.

[30] Disponível em: https://blog.betrybe.com/desenvolvimento-web/api-rest-tudo-sobre/
Acessado em: 02/06/2022.

[31] Disponível em: https://smarti.blog.br/api-rest-principios-boas-praticas-para-arquiteturas-restfu/
Acessado em: 02/06/2022.

[32] Disponível em: https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api
Acessado em: 02/06/2022.

[33] Disponível em: https://developer.wordpress.org/rest-api/
Acessado em: 02/06/2022.

[34] Disponível em: https://developer.wordpress.org/rest-api/reference/
Acessado em: 02/06/2022.

[35] Disponível em: https://developer.wordpress.org/rest-api/reference/posts/
Acessado em: 20/06/2022.

[36] Disponível em: https://developer.wordpress.org/rest-api/reference/comments/
Acessado em: 20/06/2022.

[37] Disponível em: https://www.yogh.com.br/blog/wordpress-e-o-cms-mais-popular-da-web/
Acessado em: 30/06/2022