Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Sistemas de Informação

Web Scraping e Análise de Dados no Aperfeiçoamento do Processo Seletivo de Programadores

Pedro Cotta Badaró

JUIZ DE FORA
JULHO, 2023

Web Scraping e Análise de Dados no Aperfeiçoamento do Processo Seletivo de Programadores

Pedro Cotta Badaró

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Sistemas de Informação

Orientador: Edelberto Franco Silva

JUIZ DE FORA

JULHO, 2023

Web Scraping e Análise de Dados no Aperfeiçoamento do Processo Seletivo de Programadores

Pedro Cotta Badaró

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Edelberto Franco Silva
Doutor em Computação

Luciano Jerez Chaves
Doutor em Computação

Luciana Conceição Dias Campos
Doutora em Engenharia Elétrica

JUIZ DE FORA
12 DE JULHO, 2023

Aos meus parentes.

Aos amigos, pelo apoio.

Resumo

Com a crescente utilização de métodos e ferramentas de mineração de dados, empresas estão buscando aprimorar suas estratégias com o uso de *web scraping*, a fim de obter vantagem competitiva. Tal técnica permite a extração automatizada de informações de *websites*, onde um *software* é programado para navegar por páginas da Internet, coletar o conteúdo desejado e armazená-lo em formato estruturado. No contexto do processo seletivo, *web scraping* e o uso de algoritmos têm sido empregados para melhorar diversas etapas do processo que antes eram realizadas manualmente. Este trabalho propõe o desenvolvimento de um *software* baseado em *web scraping* e ranqueamento de usuários conforme perfil de vagas de empregos. Como resultado, tem-se a apresentação do embasamento teórico e prático relacionado à coleta de dados em mídia social, além do estudo de caso da sua aplicação no auxílio ao processo seletivo de candidatos para o cargo de programador.

Palavras-Chave: *Mineração de dados.* Automação. *Software. Web Scraping.*

Abstract

With the increasing use of data mining methods and tools, companies are seeking to enhance their strategies through the use of web scraping in order to gain a competitive advantage. This technique involves the automated extraction of information from websites, where software is programmed to navigate web pages, collect desired content, and store it in a structured format. In the context of the selection process, web scraping and the use of algorithms have been employed to improve various stages of the process that were previously carried out manually. This work proposes the development of software based on web scraping and user ranking according to job vacancy profiles. As a result, it presents the theoretical and practical foundation related to data collection from social media, along with a case study on its application in assisting the candidate selection process for the position of a programmer.

Palavras-Chave: *Mineração de dados.* Automação. *Software. web scraping.*

Agradecimentos

Aos meus pais, amigos e professores envolvidos durante minha trajetória de estudo.

Conteúdo

Lista de Tabelas

Lista de Abreviações

API	*Application Programming Interface*
DCC	Departamento de Ciência da Computação
IA	Inteligência Artificial
RDF	*Resource Description Framework*
RH	Recursos Humanos
UFJF	Universidade Federal de Juiz de Fora
URL	*Uniform Resource Locators*
HTML	*Hypertext Markup Language*
CSS	*Cascade Style Sheet*
TF-ID	*Term Frequency Inverse Document Frequency*

1 Introdução

A seleção de programadores qualificados é uma etapa crucial para a contratação de profissionais de tecnologia da informação. No entanto, o processo seletivo nas empresas enfrenta desafios devido ao grande volume de informações disponíveis na *web*. Para lidar com essa complexidade, técnicas como o *web scraping* e a análise de dados têm se mostrado cada vez mais relevantes, oferecendo uma solução eficiente em relação a seu custo-benefício (NAMOUN et al., 2020).

O *web scraping* é uma técnica automatizada que permite extrair informações de documentos da Internet para posterior análise e armazenamento. Essa técnica tem sido amplamente utilizada em *websites* de *networking*[1], como o LinkedIn, para identificar e filtrar potenciais candidatos para vagas de programadores (ZHAO, 2017). Com base nisso, este trabalho propõe uma abordagem automatizada de coleta de dados por meio de *web scraping*, seguida pela aplicação de análise de dados, visando otimizar o processo de seleção de candidatos para vagas de programadores.

Assim como o LinkedIn, outras plataformas de *networking*, como redes sociais, *e.g.*, Facebook e repositórios, *e.g.*, GitHub, proporcionam uma vasta quantidade de dados sobre profissionais do mercado de trabalho. No entanto, estabelecer uma correspondência precisa entre os candidatos e as empresas que buscam perfis específicos para preencher suas vagas é uma tarefa complexa (PAPOUTSOGLOU; MITTAS; ANGELIS, 2017). A seleção de candidatos para uma vaga de emprego requer a análise cuidadosa de suas habilidades técnicas e comportamentais, garantindo que estejam alinhadas com as necessidades da empresa e do cargo em questão (SCHMIDT; HUNTER, 1998).

Encontrar o candidato ideal pode ser um processo demorado, especialmente para empresas que recebem um grande número de inscrições. A análise de currículos e entrevistas consome tempo e recursos significativos, tornando o processo ainda mais lento e dispendioso. Estudos mostram que essas práticas tradicionais podem ser ineficazes para

[1]Plataformas na internet que facilitam a comunicação e interação entre indivíduos (BALLANTYNE et al., 2010)

identificar candidatos de alta qualidade (HUSELID, 1995). Portanto, é necessário explorar outras abordagens, como testes de habilidades e avaliações de desempenho, a fim de identificar candidatos qualificados de maneira mais eficiente.

Ao revisar a literatura existente, observa-se a escassez de estudos que explorem a abordagem de *web scraping* para seleção de candidatos. Neste trabalho, será apresentada a explicação detalhada sobre as técnicas de *scraping* e utilização da API do LinkedIn, as quais foram adotadas para a coleta de informações relevantes sobre habilidades e experiências dos candidatos. Essas abordagens serão abordadas de forma mais aprofundada em seções posteriores do trabalho. O objetivo principal é extrair as informações necessárias dos perfis dos usuários do LinkedIn, realizar uma análise detalhada e gerar uma análise personalizada, levando em consideração as preferências inseridas. Para alcançar esse objetivo, são estabelecidas as seguintes etapas: coletar um amplo conjunto de perfis de usuários do LinkedIn, desenvolver um *web scraper* capaz de extrair informações detalhadas sobre as experiências de trabalho dos perfis coletados e criar um código de análise personalizada para avaliar as informações coletadas.

Com essa abordagem, espera-se otimizar o processo de seleção de candidatos para vagas de programadores, reduzindo custos e aumentando a eficiência. O trabalho busca explorar o potencial do *web scraping* e da análise de dados para tornar a seleção de candidatos mais eficiente e precisa, oferecendo uma alternativa viável às práticas tradicionais de recrutamento.

2 Fundamentação Teórica

A utilização de técnicas de *web scraping* e análise de dados tem se tornado cada vez mais comum no contexto empresarial, devido à crescente quantidade de informações disponíveis na *web* e à necessidade de embasar decisões em dados (VORDING, 2021). No processo seletivo de programadores, tais técnicas podem ser empregadas para coletar informações relevantes sobre os candidatos e aprimorar o processo de seleção, resultando em redução de custos e aumento da eficiência (SIVARAM; RAMAR, 2010). Nesta capítulo são abordados os conceitos fundamentais de *web scraping*, coleta de dados e análise de dados, com o intuito de apresentar uma base teórica sólida para a compreensão das aplicações dessas técnicas no aperfeiçoamento do processo seletivo de programadores.

2.1 *Web Scraping* e coleta de dados

É possível definir o *web scraping* como o processo de extrair e combinar conteúdo da Internet de maneira sistemática. Nesse sentido, o processo envolve simular a interação humana em um site para percorrê-lo e obter as informações necessárias (GLEZ-PEÑA et al., 2014). Existem três tipos de classificações para *web scrapers*, que são: *web scraping* sintático, *web scraping* semântico e análise da página *web* feita pelo computador (KRIJNEN; BOT; LAMPROPOULOS, 2014).

O método sintático coleta as informações do *website* analisando o HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e outras linguagens *web*. O segundo método é o *web scraping* semântico, que é realizado quando os dados extraídos pelo *scraping* sintático são mapeados para recursos web semânticos, sendo possível a utilização de *frameworks* como o RDF (*Resource Description Framework*) (VILLAMOR et al., 2011). Por último, temos a análise da página *web* feita pelo computador. Esse método utiliza *Machine Learning*[2] para identificar e extrair informações das páginas. O *Diffbot*

[2]Aprendizado de máquina (ou *Machine Learning*) em inglês, consiste em um computador aprender por experiência com base na tarefa em questão e em métricas de desempenho. Com isso, sua atuação pode ser medida e aprimorada com a experiência adquirida (MITCHELL, 1997)

é um ótimo exemplo aplicado ao *web scraping*, pois é um programa capaz de analisar o conteúdo HTML e extrair informações precisas e significativas (UPSHALL et al., 2016).

Conforme os métodos de *web scraping* apresentados, fica clara as diversas possibilidades de aplicá-los. Atualmente, há uma variedade de *softwares* disponíveis que facilitam a automação da coleta de dados. Dentre eles, destacam-se o *Scrapy*, *Selenium Web Driver* e *Screen Scraper*.

O uso de *websites* de redes sociais como ferramenta para coletar informações sobre potenciais funcionários tem se expandido nos últimos anos. Entre os mais utilizados, o LinkedIn se destaca. Os *websites* de redes sociais proporcionam aos usuários a possibilidade de apresentar suas qualificações e fotos de forma mais dinâmica do que o formato tradicional do currículo. Em virtude do acesso público, os recrutadores têm a oportunidade de obter informações sobre os candidatos que anteriormente não eram disponíveis (ZIDE; ELMAN; SHAHANI-DENNING, 2014).

2.2 Classificadores

Um aspecto de suma importância em projetos que envolvem a mineração de dados é encontrar o classificador adequado para realizar a classificação dos dados de forma precisa e confiável. Sem esta etapa, as classificações e estudos comparativos podem se tornar inconclusivos (SALZBERG, 1997). Desta forma, nesta seção, estão apresentados os classificadores mais relevantes para o projeto, juntamente com suas principais características.

2.2.1 Árvore de Decisão

Considerada como uma abordagem simples e amplamente utilizada, a árvore de decisão, (em Inglês, *Decision Tree*), consiste em uma representação gráfica, seja em forma de fluxograma ou diagrama, que ilustra um sistema de classificação ou um modelo preditivo. A estrutura da árvore é composta por uma sequência de perguntas simples, cujas respostas traçam um caminho descendente na hierarquia da árvore. Através dessa sequência de perguntas e respostas, são estabelecidas regras hierárquicas que segmentam os dados em grupos distintos. Para cada grupo formado, uma decisão é tomada, seja ela relacionada à classificação ou à previsão de eventos futuros (MOORE; JESSE; KITTLER, 2001).

A Figura 2.1 apresenta um exemplo de uma árvore de decisão utilizada para determinar se deve-se realizar a compra de um carro. Nesse exemplo, a árvore começa perguntando se o carro é vermelho e, com base na resposta, segue por diferentes ramos. Ao longo do processo, outras perguntas são feitas, como a presença de ar condicionado, preço do veículo e sua idade. Essas perguntas são usadas para classificar as características do carro e, ao final, a árvore de decisão chega a uma conclusão sobre se é recomendado comprar o carro ou não, baseado nos critérios estabelecidos.

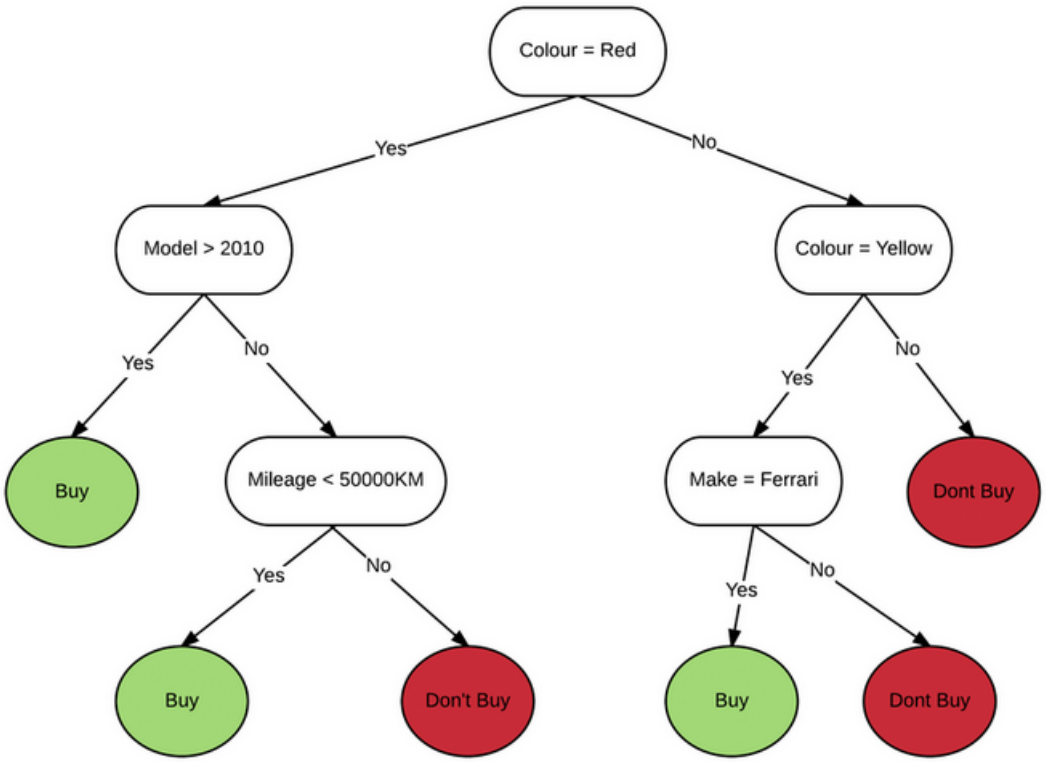

Figura 2.1: Diagrama da visão geral de uma *Decision Tree* (MOODLEY, 2016)

2.2.2 Floresta Aleatória

O classificador Floresta aleatória, *Random Forest*, é composto por conjuntos de árvores de decisão formados pelo algoritmo de *Bagging*, e configura-se como uma floresta de classificadores que realizam votação para determinar uma classe específica. Para treiná-lo, são necessários dois parâmetros: o número de árvores na floresta e o número de características selecionadas aleatoriamente em cada nó da árvore. Além disso, é fundamental dispor de um banco de dados de treinamento com rótulos de classe corretos (PETKOVIC et al., 2018).

A Figura 2.2 representa o classificador Floresta Aleatória, que utiliza um conjunto de dados para alimentar múltiplas árvores de decisão. Cada árvore gera um resultado individualmente e, em seguida, é aplicado um sistema de ranking ou média para calcular o resultado final.

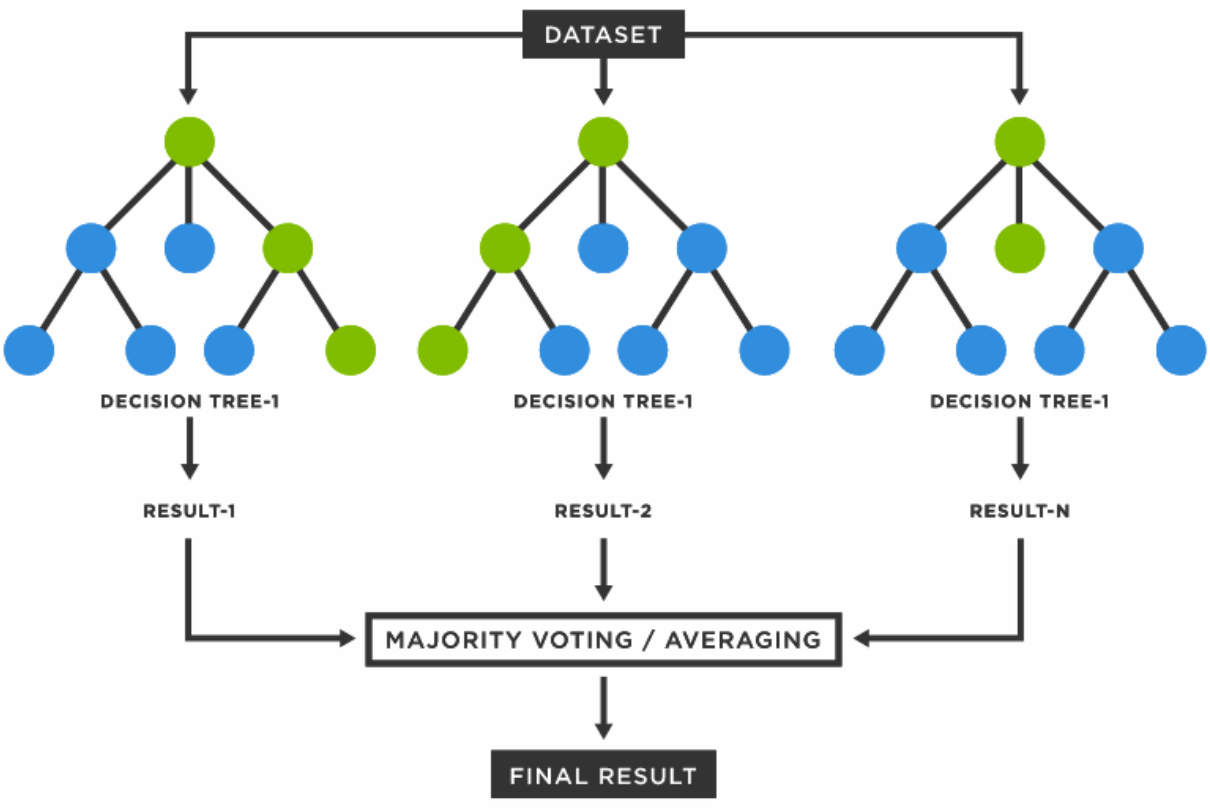

Figura 2.2: Diagrama da visão geral de uma *Random Forest* (TIBC, 2023)

2.2.3 Regressão Logística

A regressão logística, (*Logistic Regression*, em inglês), é uma técnica de modelagem estatística usada para problemas de classificação binária, onde o objetivo é prever a probabilidade de um evento ou a probabilidade de um resultado pertencer a uma das duas categorias. É um tipo de análise de regressão que modela a relação entre uma variável dependente (categórica ou binária) e uma ou mais variáveis independentes (contínuas ou categóricas) (DREISEITL; OHNO-MACHADO, 2002).

A Figura 2.3 ilustra a aplicação da regressão logística na classificação das postagens da EstatMG em categorias positivas e negativas. Essa técnica de aprendizado de máquina é utilizada para analisar e compreender a polaridade das postagens, atribuindo-lhes uma classificação binária.

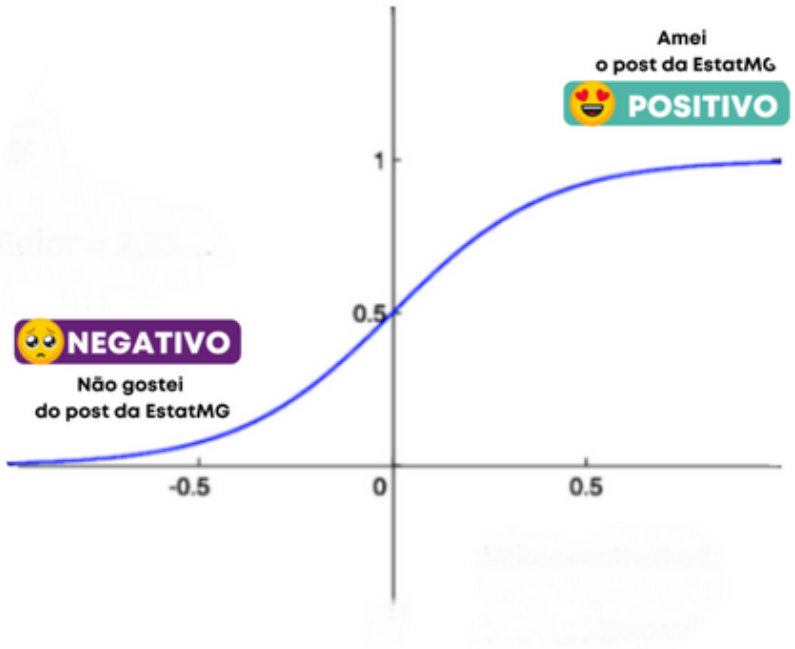

Figura 2.3: Diagrama da visão geral de uma Regressão Logística (ESTATMG, 2021)

3 Trabalhos Relacionados

Este capítulo tem como objetivo realizar uma revisão dos estudos relacionados, a fim de compreender as principais metodologias utilizadas no *web scraping* para extrair dados do LinkedIn e classificá-los.

Quando se trata da extração de dados dos perfis, é evidente que os estudos diferem significativamente em relação à metodologia adotada. Alguns optaram por utilizar a API oficial do LinkedIn como fonte de dados, enquanto outros recorreram ao *web scraping*. Essa distinção metodológica revela duas abordagens distintas para acessar e coletar as informações desejadas, cada uma com suas vantagens e considerações específicas.

A utilização da API, como demonstrado por Caldeira et al. (2017), Garg, Rani e Miglani (2015) e Kothari, Bafna e Pawar (2019), oferece uma estrutura consistente para os dados e possibilita a extração eficiente de diversas informações relevantes. Esses estudos não enfrentaram os desafios comuns do *web scraping*, como perfis incompletos, variações de idioma e restrições impostas pela plataforma do LinkedIn. Isso permitiu que eles concentrassem seus esforços em outras etapas do processo de análise e filtragem dos dados, garantindo a coerência e confiabilidade das informações obtidas.

Em contraste com estudos de Caldeira et al. (2017), Garg, Rani e Miglani (2015) e Kothari, Bafna e Pawar (2019), a pesquisa realizada por Dai et al. (2015) obteve seus dados a partir de todas as informações disponíveis publicamente pelo usuário. Para fazer isso, foi utilizado o *framework* Scrapy em conjunto com a linguagem Python. O objetivo principal do trabalho é extrair as informações sobre o nível de escolaridade dos indivíduos.Para percorrer os diversos perfis, foi criada uma estratégia que consistia em explorar o diretório público dos membros do LinkedIn e extrair os links para os perfis. Em seguida, os dados extraídos passaram por um processo de filtragem, no qual apenas os perfis em inglês que apresentavam algum nível de escolaridade foram considerados e depois por uma classificação que consistia em dividir os perfis de acordo com sua escolaridade.Para classificar os dados, Dai et al. (2015) utilizou um algoritmo de clusterização para agrupar os perfis de acordo com seu histórico de trabalho, atribuindo-os à categoria mais

adequada. Para essa finalidade, foi estabelecida uma metodologia de grupos, resultando em um total de 9 *clusters*[3] para 14.343 perfis. O autor também mencionou a dificuldade de obter um conjunto de dados abrangente do LinkedIn, que incluísse informações dos perfis e suas interações.

Também utilizando a clusterização das informações coletadas, o trabalho de Garg, Rani e Miglani (2015) consistia em obter os dados da API, normalizá-los e realizar uma análise detalhada dos profissionais. Sua metodologia envolvia o tratamento dos dados e a aplicação de três técnicas de clusterização, com critérios diferentes em cada uma delas, resultando em gráficos que auxiliaram na visualização das conexões geográficas e possibilitaram a identificação de correlações entre trabalho, educação e localização.

Por outro lado, Caldeira et al. (2017) utilizou o algoritmo TF-IDF (*Term Frequency-Inverse Document Frequency*) que é uma métrica para avaliar a importância de uma palavra em um documento em relação a um conjunto de documentos e o algoritmo de Apriori, que é aplicado para analisar os requisitos e identificar relacionamentos entre eles.Em seu projeto, divergindo dos trabalhos citados anteriormente, Caldeira et al. (2017) buscou selecionar requisitos relevantes em vagas de emprego na área de tecnologia da informação. O trabalho consistiu em percorrer um conjunto de 7000 vagas de emprego, filtrar os dados e aplicar os algoritmos em cada país. Dessa forma, foi possível identificar as palavras-chave mais comuns em cada país, resultando em grafos que mostram os requisitos técnicos mais exigidos.

Em contrapartida, Kothari, Bafna e Pawar (2019) propôs um sistema de classificação baseado em filtros selecionados pelo usuário. O sistema proposto analisa os perfis de acordo com várias categorias, como histórico de graduação, histórico de trabalho e formação profissional atual. Os dados são coletados pelo sistema de nuvem da Sales Cloud e passaram por um processo de extração, transformação e carregamento para gerar o resultado de acordo com os filtros aplicados. Dessa forma, o usuário tem acesso a relatórios e gráficos personalizados de acordo com suas preferências.

Utilizando como referência a tabela 3.1 para resumir as características dos tra-

[3]grupos de objetos que exibem similaridade ou relacionamento, podendo ser definidos por variáveis, dissimilaridades ou arestas ponderadas, e podem assumir a forma de partições não sobrepostas ou sobrepostas, com pertencimentos definidos ou difusos (HENNIG, 2015).

balhos estudados, enquanto os estudos anteriores abordaram principalmente a extração e análise de dados dos perfis do LinkedIn, a abordagem desse trabalho se diferencia ao considerar a classificação dos usuários com base em sua pontuação, a qual será definida na parte de metodologia.

Tabela 3.1: Análise dos trabalhos e suas principais características

Artigo	**Obtenção**	**Armazenamento**	**Filtragem**	**Motivo**
(DAI et al., 2015)	Scraping	N/A	Perfis em inglês e com pelo menos 1 escolaridade	Agrupamento por Escolaridade
(CALDEIRA et al., 2017)	API	N/A	Remoção de tags HTML e tokenização	Identificação dos requisitos para vagas de trabalho
(GARG; RANI; MIGLANI, 2015)	API	N/A	API do LinkedIn	Agrupamento por emprego
(KOTHARI; BAFNA; PAWAR, 2019)	API	Salesforce Cloud	Salesforce	N/A

4 Metodologia

Este capítulo apresenta uma visão abrangente das etapas percorridas no processo de extração de informações dos perfis, juntamente com a pontuação e classificação dos dados. Além disso, são explorados casos de uso práticos dos dados coletados, evidenciando sua aplicabilidade em diferentes contextos.

4.1 *Web Scraper*

Nesta seção, são expostas as ferramentas empregadas no *web scraper*, juntamente com as configurações do sistema adotadas e o atributo utilizado para extrair as informações da página. A coleta de informações para este trabalho foi realizada por meio do *website* LinkedIn. Essa plataforma foi escolhida devido à sua capacidade de permitir que profissionais compartilhem informações relevantes sobre si e suas experiências de trabalho, tornando-se um recurso excelente para extração de dados. Ao explorarmos perfis públicos na plataforma, teremos a oportunidade de mapear e analisar informações que possuem potencial para obter resultados interessantes (BRADBURY, 2011).

4.1.1 *Framework* e configurações de máquina

Para a implementação do *web scraper*, devido à facilidade de automação e implementação do *framework Selenium Web Driver*, conforme descrito no livro "Selenium WebDriver Recipes in C#" (ZHAN, 2015), o projeto utilizará esta ferramenta. Além disso, para permitir a execução no navegador Google Chrome, utilizou-se o *chrome driver*. Para simplificar a identificação dos elementos na página, foi empregada a extensão *Selector Gadget* do Google Chrome, que oferece uma interface visual para seleção de elementos. Quanto à programação, a IDE Visual Studio foi utilizada, fornecendo um ambiente integrado para o desenvolvimento de software.

Foi necessário criar um ambiente compatível com todas as tecnologias fundamentais para o desenvolvimento das aplicações que constituem o sistema responsável pela

captura e entrega e análise de informações necessárias. Para isso, o computador utilizado possuía as seguintes configurações:

- Processador AMD Ryzen 3 2700x 3700mhz com 8 núcleos.
- Memória RAM de 16 GB DDR4.
- Placa de Vídeo NVIDIA GTX 1050 TI 4 GB.
- Placa Máe ASROCK Steel Legend B450M.
- Sistema Operacional Windows 11 Enterprise 64 Bits.

4.1.2 Extração de dados

Para a realização do *web scraping* sintático, serão empregados os seletores de CSS, os quais constituem uma tecnologia que possibilita a seleção e extração de dados com base nas propriedades de estilo dos componentes da página (SINGRODIA; MITRA; PAUL, 2019). Utilizando a ferramenta *Selector Gadget*, conforme mostrado na Figura 4.1, é possível separar facilmente as informações relevantes para a extração, utilizando os seletores para identificar os elementos desejados. Com isso, evitam-se dados vazios e informações desnecessárias.

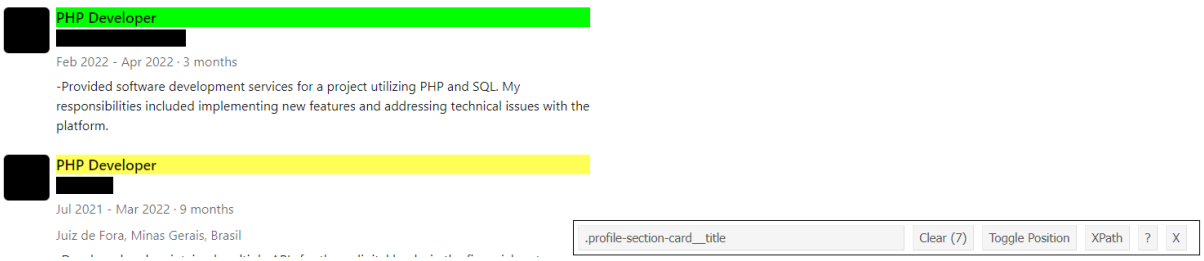

Figura 4.1: Seletor de CSS para extrair o nome de cada emprego

4.2 Etapas

O processo consiste em três etapas. Na primeira etapa, serão coletadas e armazenadas as URLs dos perfis. Na segunda etapa, todas as URLs serão percorridas para extrair as informações dos perfis. Finalmente, na terceira etapa, essas informações serão analisadas

com base em critérios específicos, gerando gráficos que permitirão a visualização dos programadores mais bem pontuados.

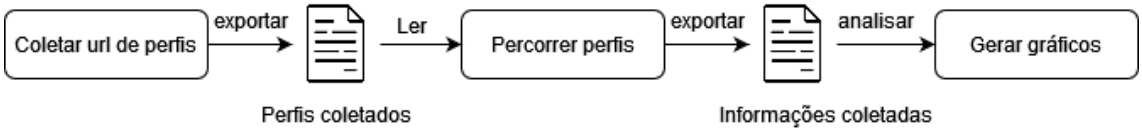

Figura 4.2: Etapas do trabalho

4.3 Obtenção dos Perfis

Para simular um processo seletivo de qualidade, é necessário contar com dados de experiências diferenciadas para uma boa comparação. No entanto, antes de obter as informações dos perfis, é preciso encontrar os perfis em si. Serão percorridos aleatoriamente 50 perfis de programadores para exportar a URL de todos os perfis recomendados a partir de cada um deles, já que os perfis recomendados são similares à vaga e à experiência do perfil atual.

Dentro da página de perfil de um usuário no LinkedIn, são recomendados outros 20 perfis de pessoas. Portanto, durante o processo de extração de dados, são exportadas 1000 URLs de perfis, devido aos 50 perfis de programadores para coletar as URLs de todos os perfis recomendados a partir de cada um deles. Esse procedimento garante uma base abrangente e diversificada de perfis para comparação e identificação dos melhores pontuadores. A Figura 4.3 apresenta a lista de perfis recomendados de um usuário, juntamente com a descrição da vaga de cada perfil.

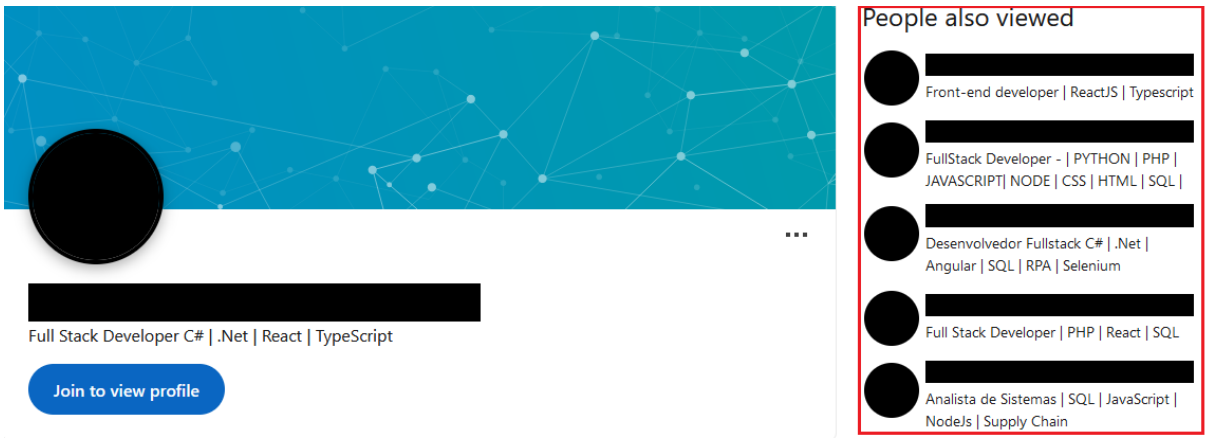

Figura 4.3: Pessoas recomendadas baseadas no perfil atual.

4.4 Percorrer os Perfis

Nessa fase do projeto, o objetivo é coletar todas as informações relevantes dos perfis, com ênfase especial na seção de experiência do usuário. A extração desses dados é feita de acordo com o exemplo apresentado na Figura 4.4, que demonstra as informações específicas que são coletadas de cada perfil.

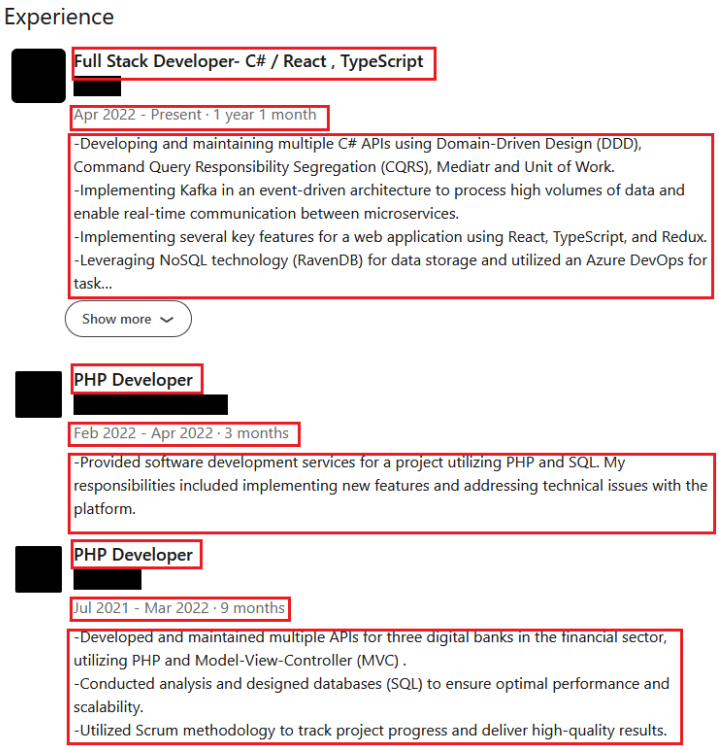

Figura 4.4: Informações coletadas na seção de experiência do perfil

4.5 Análise dos Dados

4.5.1 Classificação e Pontuação

Para garantir uma pontuação justa e equilibrada em todos os casos de uso, é importante considerar que o número de linguagens e *frameworks* pode variar. Portanto, foi adotada uma abordagem em que o valor de 10 é dividido igualmente entre todas as linguagens e *frameworks*. Com base nisso, a tabela 4.1 mostra a divisão de classes que utilizo na avaliação do candidato, ela separa os candidatos baseados na sua pontuação que é calculada pela equação de pontuação. Essa divisão igualitária permite que a classificação seja significativa e aplicável a todos os casos de uso.

Tabela 4.1: Classificação dos candidatos

Classe	Pontuação
A	Acima de 50 pontos
B	Entre 30 e 50 pontos
C	Entre 10 e 30 pontos
D	Abaixo de 10 pontos

A equação 4.1 representa a pontuação é calculada somando o produto do peso da linguagem ou *framework* do candidato (ω_i) pelo tempo em anos de experiência (p_i) da linguagem ou *framework*. O conjunto N representa o número total de linguagens e *frameworks* avaliados.

$$\text{pontuação} = \sum_{i=1}^{N} (\omega_i \times p_i)$$

4.5.2 Classificador

Dada a sua notável capacidade de classificar dados com precisão, robustez e habilidade de *ranking*, o classificador *Random Forest* será utilizado. Seu modelo se baseia em dois parâmetros adotados: as classes A, B, C, D, que representam os níveis de pontuação atribuídos aos candidatos, e os atributos, os quais englobam os anos dedicadas a cada linguagem de programação ou *framework*, assim como seus respectivos pesos. Devido à necessidade de treinamento do modelo, uma parcela de 20% dos dados coletados será utilizada para essa finalidade enquanto o resto será para a classificação.

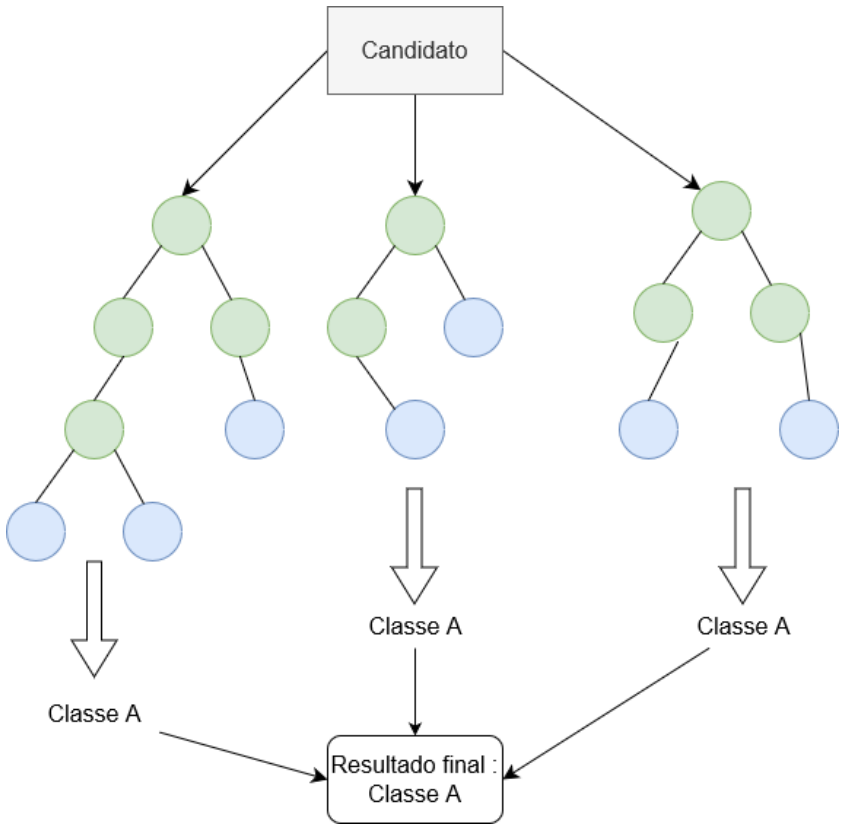

Figura 4.5: Representação do *Random Forest* na aplicação do trabalho

4.5.3 Coleta de Dados

Os dados utilizados para embasar esta parte do trabalho foram obtidos através da pesquisa anual de desenvolvedores do Stack Overflow (Stack Overflow, 2022). Essa pesquisa é considerada uma das mais abrangentes do mercado, com a participação de milhares de desenvolvedores de todo o mundo.

Se considerarmos front-end como a parte de uma aplicação onde o usuário vê e interage e back-end como a parte responsável pelo banco de dados, servidores e sistema (ABDULLAH; ZEKI, 2014). Full-stack, por sua vez, seria uma pessoa capaz de desenvolver o software tanto para o front-end quanto para o back-end de uma aplicação web (TAIVALSAARI et al., 2021). Dito isso, os três casos de uso a serem testados são as vagas mais populares identificadas na pesquisa do Stack Overflow (Stack Overflow, 2022):

Para as linguagens e *frameworks*, decidi selecionar as mais comuns de acordo com a pesquisa realizada, para cada caso de uso. Portanto, as vagas simuladas serão:

- Desenvolvedor *full-stack*: *frameworks* NodeJS (back-end) e ReactJS (*front-end*).
- Desenvolvedor *back-end*: linguagem Python e *framework* Django.

- Desenvolvedor *front-end*: linguagens JavaScript, HTML/CSS e *framework* ReactJS.

5 Estudos de Casos

5.1 Dados gerais

Neste trabalho, o scraper percorreu um conjunto de 1000 perfis de programadores no LinkedIn, totalizando 3804 experiências profissionais registradas. Esses perfis foram submetidos a uma filtragem criteriosa, removendo aqueles que estavam vazios ou que não continham pelo menos uma linguagem ou *framework* relacionado à pesquisa realizada do Stack Overflow (Stack Overflow, 2022). Após essa filtragem, restaram um total de 446 perfis, que foram considerados para análise neste estudo. Além disso, ao considerar o tempo totalizado a partir dessas experiências, foi identificado um tempo médio de 5.8 anos de experiência pelos programadores analisados. Esses dados gerais fornecem uma base sólida para uma análise abrangente das tendências e padrões encontrados nos perfis de programadores do LinkedIn.

A Figura 5.1 apresenta a nuvem de tags das linguagens de programação mais comumente encontradas nos perfis analisados

Figura 5.1: Nuvem de tag das linguagens encontradas.

A seguir, destacam-se as cinco principais linguagens identificadas, juntamente com o número de ocorrências em parênteses:

- JavaScript (175)
- PHP (135)

- Java (121)
- Python (109)
- C (107)

A tabela apresenta o tempo médio das linguagens. O tempo médio foi calculado com base na quantidade de pessoas que tinham a linguagem e quantidade de vezes que a linguagem foi encontrada.

Tabela 5.1: Tempo Médio das top 5 Linguagens de Programação

Linguagem de Programação	**Tempo Médio (Anos)**
JavaScript	1.77
PHP	2.02
Java	1.63
Python	1.52
C	1.98

É igualmente relevante examinar os *frameworks* mais utilizados pelos programadores. A figura apresenta a nuvem de tags dos *frameworks* encontrados, evidenciando aqueles que foram identificados com maior frequência:

Figura 5.2: Nuvem de tag dos *frameworks* encontrados.

Observando a Figura, podemos destacar os cinco principais *frameworks* identificados, juntamente com suas respectivas frequências de ocorrência:

- ReactJs (103)
- NodeJs (68)
- Jquery (64)

- Angular (56)
- Laravel (44)

A Tabela apresenta o tempo médio de uso dos *frameworks*. Assim como nas linguagens, o tempo médio foi calculado com base na quantidade de pessoas que utilizaram o *framework* e na frequência com que ele foi encontrado.

Tabela 5.2: Tempo Médio dos top 5 *Frameworks* encontrados

Framework	**Tempo Médio (Anos)**
ReactJs	1.33
NodeJs	1.83
Jquery	1.80
Angular	1.98
Laravel	1.51

5.2 Casos de uso

5.2.1 Desenvolvedor *Back-end*

O caso de uso 1 foi focado na análise dos resultados obtidos com a linguagem Python no desenvolvimento de aplicações *Back-end*. Ao analisar os dados coletados, observamos que a linguagem Python foi identificada em um total de 109 perfis. Isso indica uma presença significativa e uma preferência entre os profissionais para o desenvolvimento de soluções de *Back-end*.

Além disso, o tempo médio de experiência dos candidatos foi de 1.52 anos, o que demonstra que muitos programadores possuem uma base sólida e habilidades relevantes nessa linguagem.

A Figura apresenta os 10 perfis com a maior pontuação depois da classificação entre aqueles que utilizam a linguagem Python:

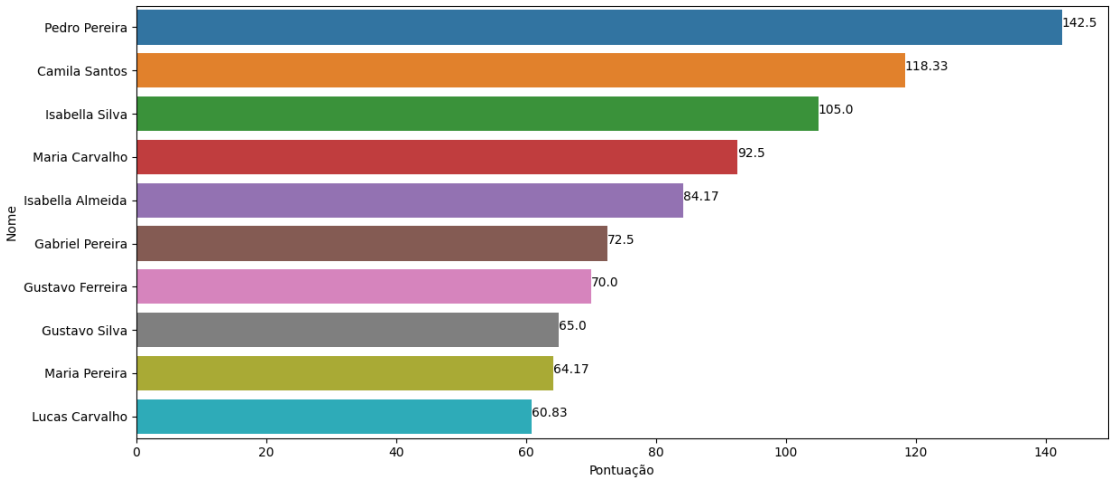

Figura 5.3: Caso de uso 1

5.2.2 Desenvolvedor *Front-end*

O caso de uso 2 foi direcionado para a análise dos resultados obtidos com as linguagens JavaScript, HTML e CSS no desenvolvimento de aplicações *front-end*.

Especificamente, a linguagem JavaScript foi identificada em 175 perfis, seguida de CSS em 85 perfis e HTML em 77. Esses números evidenciam a predominância do JavaScript como uma das principais linguagens para o desenvolvimento *Front-end*,

Além disso, os candidatos apresentaram um tempo médio de experiência de 1.77 anos com JavaScript, 2.1 anos com HTML e 1.94 anos com CSS.

A Figura 5.4 destaca os 10 perfis classificados com a maior pontuação entre aqueles que pelo menos utilizam a linguagem JavaScript, HTML ou CSS:

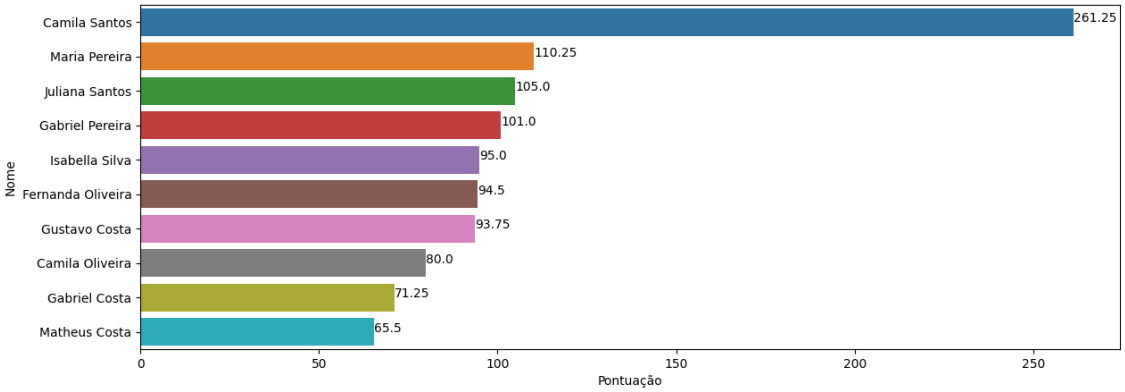

Figura 5.4: Caso de uso 2

5.2.3 Desenvolvedor *Full-stack*

O caso de uso 3 abrange a análise dos resultados obtidos com os frameworks ReactJs e NodeJs no desenvolvimento de aplicações *Full- stack*.

No total, ReactJs foi identificado em 103 perfis analisados, enquanto NodeJs foram 68. Esses números evidenciam a popularidade e o amplo uso dessas linguagens no desenvolvimento de aplicações *full-stack*.

No que diz respeito ao tempo médio de experiência dos candidatos, constatou-se que o tempo de experiência com ReactJs foi de 1.33 anos, enquanto o tempo médio de experiência com NodeJs foi de 1.83 anos. A Figura 5.5 destaca os 10 perfis com a maior pontuação entre aqueles que utilizam ReactJs e NodeJs.

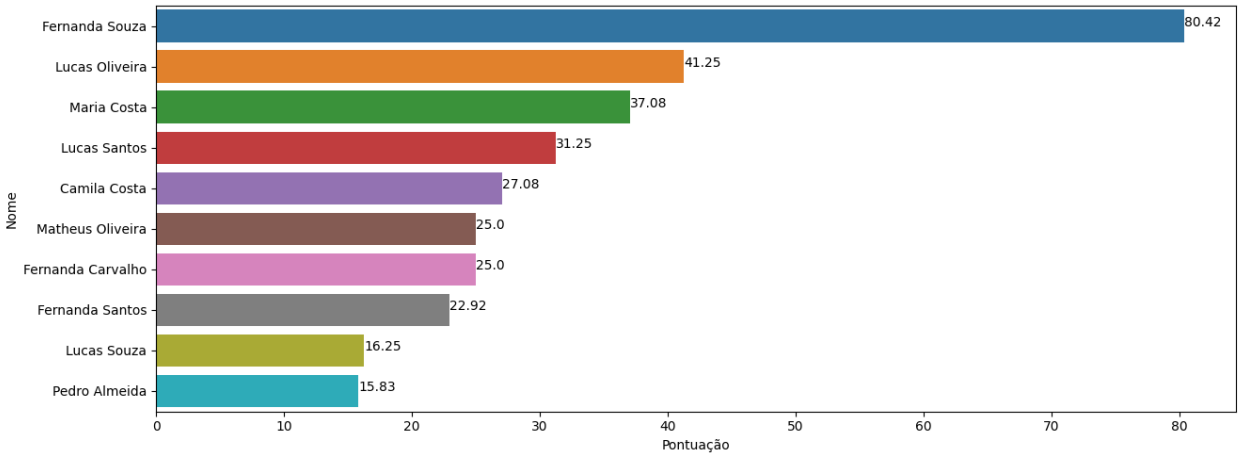

Figura 5.5: Caso de uso 3

5.3 Análise dos Resultados

Foram identificados padrões interessantes nos resultados dos casos de uso. No primeiro caso, observou-se uma quantidade significativa de indivíduos com pontuações elevadas, bem acima da classe B que são todos candidatos com uma pontuação acima de 50 pontos. Essa tendência pode estar relacionada ao fato de que a linguagem Python possui uma média mais alta de tempo de experiência. Além disso, a ampla aplicabilidade dessa linguagem também influencia, já que muitas pessoas a conhecem, mesmo sem estarem vinculadas a um framework específico.

No segundo caso, também foram registradas pontuações altas para os principais

pontuadores da classe A. Isso indica que essas linguagens frequentemente são adquiridas em conjunto ao longo do tempo de experiência, já que as pontuações permaneceram elevadas, apesar dos pesos diferentes atribuídos às linguagens.

No terceiro caso, houve uma mudança drástica nas pontuações, indicando que os maiores pontuadores não possuem uma quantidade tão alta de experiência em comparação com os casos anteriores. Durante a análise, observou-se que muitos candidatos possuíam uma extensa experiência em apenas um dos frameworks, o que resultou em pontuações mais baixas nesse contexto.

Essas observações destacam a importância de considerar a diversidade de habilidades e experiências dos candidatos, bem como compreender o contexto em que essas habilidades foram adquiridas. A análise dos resultados dos casos de uso pode fornecer informações valiosas para tomar decisões informadas na seleção e alocação de profissionais.

6 Conclusões

Vale ressaltar que a importância da coleta de informações pessoais sem o consentimento dos usuários pode violar a privacidade e os direitos individuais, resultando em consequências legais para as empresas que realizam esse tipo de atividade. O caso da disputa jurídica entre o LinkedIn e a empresa hiQ Labs ilustra claramente a importância de uma análise cuidadosa das implicações éticas e legais do *web scraping*, especialmente no que diz respeito à privacidade dos dados coletados (HIQ..., 2019). Nesse sentido, é fundamental que as técnicas de *web scraping* sejam transparentes em relação às informações coletadas, obtenham o consentimento dos usuários quando necessário e estejam em conformidade com as regulamentações e leis vigentes de proteção de dados.

Em suma, o presente trabalho demonstrou a viabilidade e eficácia do desenvolvimento de um scraper para a coleta automatizada de informações relevantes na avaliação de candidatos. Ao extrair URLs de perfis e realizar a raspagem de dados, foi possível obter um conjunto diversificado de informações sobre as experiências profissionais dos candidatos. Através das etapas de filtragem e análise, foi possível selecionar os perfis mais adequados e atribuir pontuações com base em critérios específicos.

Diversas dificuldades foram encontradas durante o processo de raspagem de dados, sendo a principal a dificuldade em encontrar vários perfis com informações semelhantes e estabelecer uma classificação que atendesse a diversos critérios. Para resolver esse problema, foi realizada a coleta de perfis com base nas pessoas recomendadas para cada perfil e a utilização do classificador Floresta Aleatória.

A aplicação do processo de filtragem permitiu a identificação dos perfis que atendiam aos critérios específicos relacionados a linguagens e frameworks, proporcionando uma análise mais precisa e direcionada.

Através do uso dessa abordagem automatizada, foi possível otimizar o processo seletivo, permitindo que os avaliadores concentrassem sua atenção em outros aspectos fundamentais na escolha do candidato ideal, como habilidades interpessoais e compatibilidade com a equipe. Essa combinação de scraping e análise baseada em critérios pro-

porcionou uma avaliação mais eficiente e objetiva, eliminando parte do trabalho manual e possibilitando uma tomada de decisão mais ágil e informada.

Dessa forma, o desenvolvimento do scraper e a aplicação da análise baseada em critérios se mostraram ferramentas poderosas e aliadas no processo de seleção de candidatos, trazendo benefícios tanto para os profissionais que buscam se destacar quanto para as empresas que desejam encontrar o candidato ideal para suas vagas.

7 Trabalhos futuros e Desafios

No contexto das oportunidades futuras, há diversas possibilidades a serem exploradas no âmbito do *web scraping* no processo seletivo. Este trabalho específico concentrou-se na utilização de dados extraídos do LinkedIn para o processo de seleção de programadores. Nessa aplicação em particular, há sugestões que podem ser implementadas visando a ampliação dos resultados obtidos.

Um dos aspectos que pode ser aprimorado na coleta de perfis é a identificação e aquisição de perfis relacionados. Ao possibilitar a obtenção de uma quantidade maior de perfis associados, torna-se viável uma análise mais abrangente dos dados coletados.

Com um maior volume de dados obtidos, seria possível implementar *threads* para percorrer os perfis de maneira mais eficiente. Além disso, poderiam ser incluídos mecanismos de tratamento para lidar com casos em que os perfis não estejam disponíveis para acesso, assim como aqueles que não contenham as informações necessárias para a coleta.

Quanto à filtragem de dados, poderia ser realizada uma tokenização abrangendo todas as possíveis variações na escrita de linguagens de programação e *frameworks*. Essa abordagem permitiria aumentar a quantidade de perfis a serem classificados, independentemente da linguagem empregada nos dados dos perfis.

Bibliografia

ABDULLAH, H. M.; ZEKI, A. M. Frontend and backend web technologies in social networking sites: Facebook as an example. In: IEEE. *2014 3rd international conference on advanced computer science applications and technologies*. [S.l.], 2014. p. 85–89.

BALLANTYNE, A. et al. 'i feel less lonely': what older people say about participating in a social networking website. *Quality in Ageing and Older Adults*, Emerald Group Publishing Limited, v. 11, n. 3, p. 25–35, 2010.

BRADBURY, D. Data mining with linkedin. *Computer Fraud & Security*, Elsevier, v. 2011, n. 10, p. 5–8, 2011.

CALDEIRA, D. C. et al. Data mining on linkedin data to define professional profile via mineraskill methodology. In: IEEE. *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.], 2017. p. 1–6.

DAI, K. et al. Scraping and clustering techniques for the characterization of linkedin profiles. *arXiv preprint arXiv:1505.00989*, 2015.

DREISEITL, S.; OHNO-MACHADO, L. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, Elsevier, v. 35, n. 5-6, p. 352–359, 2002.

ESTATMG. *Guia Rápido sobre Regressão Logística*. 2021. Disponível em: ⟨https://estatmg.com.br/2021/11/05/guia-rapido-sobre-regressao-logistica/⟩.

GARG, P.; RANI, R.; MIGLANI, S. Mining professional's data from linkedin. In: IEEE. *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*. [S.l.], 2015. p. 98–101.

GLEZ-PEÑA, D. et al. Web scraping technologies in an api world. *Briefings in bioinformatics*, Oxford University Press, v. 15, n. 5, p. 788–797, 2014.

HENNIG, C. What are the true clusters? *Pattern Recognition Letters*, Elsevier, v. 64, p. 53–62, 2015.

HIQ Labs, Inc. v. LinkedIn Corp. [S.l.]: Court of Appeals, 9th Circuit, 2019. 985 p.

HUSELID, M. A. The impact of human resource management practices on turnover, productivity, and corporate financial performance. *Academy of management journal*, Academy of Management, v. 38, n. 3, p. 635–672, 1995.

KOTHARI, M. S. S.; BAFNA, M. J. P.; PAWAR, M. D. P. Linkedin profile extractor. *JournalNX*, Novateur Publication, p. 172–174, 2019.

KRIJNEN, D.; BOT, R.; LAMPROPOULOS, G. Automated web scraping apis. *Online: http://mediatechnology. leiden. edu/images/uploads/docs/wt2014_web_scraping. pdf (accessed April 25, 2018). Search in*, 2014.

MITCHELL, T. *Machine Learning*. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: ⟨https://books.google.com.br/books?id=EoYBngEACAAJ⟩.

MOODLEY, A. *Language Identification With Decision Trees: Identification Of Individual Words In The South African Languages*. Tese (Doutorado), 01 2016.

MOORE, T.; JESSE, C.; KITTLER, R. An overview and evaluation of decision tree methodology. In: *American Statistical Association quality and productivity conference papers. University of Texas, Austin*. [S.l.: s.n.], 2001.

NAMOUN, A. et al. Web design scraping: Enabling factors, opportunities and research directions. 2020.

PAPOUTSOGLOU, M.; MITTAS, N.; ANGELIS, L. Mining people analytics from stackoverflow job advertisements. In: IEEE. *2017 43rd Euromicro conference on software engineering and advanced applications (SEAA)*. [S.l.], 2017. p. 108–115.

PETKOVIC, D. et al. Improving the explainability of random forest classifier–user centered approach. In: WORLD SCIENTIFIC. *Pacific symposium on biocomputing 2018: proceedings of the pacific symposium*. [S.l.], 2018. p. 204–215.

SALZBERG, S. L. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery*, Springer, v. 1, p. 317–328, 1997.

SCHMIDT, F. L.; HUNTER, J. E. The validity and utility of selection methods in personnel psychology: Practical and theoretical implications of 85 years of research findings. *Psychological bulletin*, American Psychological Association, v. 124, n. 2, p. 262–274, 1998.

SINGRODIA, V.; MITRA, A.; PAUL, S. A review on web scrapping and its applications. In: IEEE. *2019 international conference on computer communication and informatics (ICCCI)*. [S.l.], 2019. p. 1–6.

SIVARAM, N.; RAMAR, K. Applicability of clustering and classification algorithms for recruitment data mining. *International Journal of Computer Applications*, Citeseer, v. 4, n. 5, p. 23–28, 2010.

Stack Overflow. *Developer Survey Results 2022*. 2022. ⟨https://survey.stackoverflow.co/2022/#overview⟩. Acesso em: 27 de abril de 2023.

TAIVALSAARI, A. et al. Full stack is not what it used to be. In: SPRINGER. *Web Engineering: 21st International Conference, ICWE 2021, Biarritz, France, May 18–21, 2021, Proceedings*. [S.l.], 2021. p. 363–371.

TIBC. *What is a Random Forest?* 2023. Disponível em: ⟨https://www.tibco.com/reference-center/what-is-a-random-forest⟩.

UPSHALL, M. et al. Extracting meaning from web content. *eLucidate*, v. 13, n. 1, 2016.

VILLAMOR, J. I. F. et al. A semantic scraping model for web resources-applying linked data to web page screen scraping. Telecomunicacion, 2011.

VORDING, R. *Harvesting unstructured data in heterogenous business environments; exploring modern web scraping technologies*. Dissertação (B.S. thesis) — University of Twente, 2021.

ZHAN, Z. *Selenium WebDriver Recipes in C#*. [S.l.]: Springer, 2015.

ZHAO, B. Web scraping. *Encyclopedia of big data*, Springer Living ed. Cham, p. 1–3, 2017.

ZIDE, J.; ELMAN, B.; SHAHANI-DENNING, C. Linkedin and recruitment: How profiles differ across occupations. *Employee relations*, Emerald Group Publishing Limited, v. 36, n. 5, p. 583–604, 2014.