**Projeto Integrador – Desenvolver um site que utilizar o Web-scraping para mineração de dados, e auxiliar pessoas a conseguirem emprego.**

**Engenharia de Computação - 5° Semestre – Maria Auxiliadora – UNISAL – 2022 – Americana/SP**

**Engenheiros do T.I**

Matheus Caldas Menesio, RA: 210017233

Luciano De Araujo Junior, RA: 210017894

Vitor Moscardi, RA: 210017849

***Resumo*** *– Um grupo de estudantes enfrentou o desafio de desenvolver um projeto de web scraping para o navegador com o objetivo de auxiliar indivíduos. Eles conceberam a ideia de oferecer suporte às pessoas em sua primeira experiência profissional, filtrando a área de interesse desejada e coletando oportunidades disponíveis em sua região.*

***Palavras-chave****: Grupo de estudantes, desafio, desenvolver, web scraping, auxiliar, primeira experiência profissional, filtrar, área de interesse, oportunidades disponíveis, região.*

***Abstract*** *- A group of students faced the challenge of developing a web scraping project for the browser in order to help individuals. They conceived the idea of ​​supporting people in their first professional experience, filtering the desired area of ​​interest and collecting opportunities available in their region.*

1. Introdução

Na era digital em que vivemos, o acesso à informação e às oportunidades profissionais desempenha um papel fundamental no desenvolvimento das pessoas. Reconhecendo essa realidade, um grupo de estudantes decidiu enfrentar um desafio desafiador: desenvolver um projeto de extensão para navegadores que visa auxiliar os indivíduos em suas trajetórias profissionais. Com dedicação e criatividade, eles conceberam a ideia de criar uma ferramenta capaz de oferecer suporte às pessoas em seu primeiro emprego, por meio da filtragem criteriosa da área de interesse desejada e da apresentação de oportunidades disponíveis em suas regiões. Esse projeto inovador promete ser uma verdadeira ponte entre as ambições profissionais dos usuários e as informações relevantes e atualizadas sobre vagas e carreiras em suas localidades. Com isso, esse grupo de estudantes busca contribuir para o crescimento e o sucesso dos indivíduos, proporcionando-lhes um caminho mais acessível e direcionado em sua busca por oportunidades profissionais.

1. Referencial Teórico

Este capítulo tem por objetivo apresentar o referencial teórico da pesquisa e fundamentar as contribuições para. A presente pesquisa baseia-se em conceitos e estudos relevantes no campo do desenvolvimento de web scraping e na área de auxílio à inserção profissional de indivíduos. A seguir, serão apresentados alguns dos principais tópicos que fundamentam a compreensão e a abordagem adotada neste estudo.

1. Web scraping para navegadores: O desenvolvimento de web scraping para navegadores tem se tornado uma prática comum e promissora. Essas ferramentas são programas que coletam automaticamente informações de páginas da web, permitindo extrair dados relevantes de forma estruturada. O web scraping visa suprir demandas específicas da sociedade, oferecendo soluções inovadoras para a obtenção de dados do cotidiano.

2. Auxílio à inserção profissional: A inserção no mercado de trabalho é um desafio significativo para muitos indivíduos, especialmente aqueles que estão vivenciando sua primeira experiência profissional. Nesse contexto, o apoio e o suporte adequados podem fazer a diferença na trajetória de carreira desses indivíduos. O auxílio à inserção profissional envolve o fornecimento de informações relevantes sobre vagas, orientações sobre currículos e entrevistas, além de recursos para aprimorar habilidades e competências necessárias para o mundo do trabalho.

3. Filtragem da área de interesse: A filtragem da área de interesse desejada é um aspecto crucial no desenvolvimento de ferramentas de auxílio à inserção profissional. Ao permitir que os usuários indiquem sua área de interesse, é possível personalizar as informações apresentadas, direcionando as oportunidades que se alinham aos seus objetivos e preferências. Essa filtragem aumenta a eficácia do sistema, reduzindo a sobrecarga de informações e proporcionando uma experiência mais relevante e direcionada.

4. Oportunidades regionais: A disponibilização de oportunidades profissionais específicas de cada região é um elemento fundamental na busca por emprego. As características econômicas, sociais e culturais de uma região podem influenciar diretamente nas oportunidades de trabalho disponíveis. Portanto, apresentar informações atualizadas sobre as oportunidades regionais é essencial para auxiliar os usuários na identificação de vagas próximas a eles e aumentar suas chances de sucesso na busca por emprego.

Ao embasar-se nessas perspectivas teóricas, a presente pesquisa tem como objetivo desenvolver um site para navegadores que ofereça suporte às pessoas em sua primeira experiência profissional, por meio da filtragem da área de interesse desejada e da apresentação de oportunidades disponíveis em suas regiões. Acredita-se que essa iniciativa possa contribuir de forma significativa para a inserção e o crescimento profissional dos indivíduos, fornecendo-lhes informações relevantes e direcionadas para auxiliá-los em suas trajetórias profissionais.

1. Desenvolvimento do projeto

O projeto em questão está sendo desenvolvido com a utilização de diversas ferramentas e tecnologias, como HTML, CSS, PHP, Python, JavaScript e Bootstrap. Essas linguagens e frameworks são amplamente reconhecidos e empregados na construção de aplicações web robustas e interativas.

Uma das etapas fundamentais do projeto é a implementação do web scraping. O web scraping é uma técnica de extração de dados de páginas da web de forma automatizada. Essa técnica permite que o sistema colete informações relevantes de fontes externas, como sites de empresas, agências de emprego e plataformas de recrutamento. Com o uso de bibliotecas e frameworks específicos em Python, como BeautifulSoup e Scrapy, é possível realizar a extração dos dados necessários de forma estruturada e eficiente.

Ao utilizar o web scraping, o sistema será capaz de buscar, filtrar e armazenar informações sobre oportunidades de emprego disponíveis em diferentes sites. Esses dados coletados serão processados e apresentados de forma organizada aos usuários da extensão para navegador. Dessa forma, os usuários poderão encontrar oportunidades relevantes em sua área de interesse de maneira rápida e conveniente.

O desenvolvimento do projeto contempla a criação de uma interface amigável e responsiva, utilizando HTML, CSS e Bootstrap. Essas tecnologias possibilitam a construção de um layout atraente e adaptável a diferentes dispositivos, garantindo uma experiência de usuário consistente e intuitiva.

Além disso, a programação em PHP será utilizada para o processamento dos dados coletados e a realização de consultas personalizadas com base nas preferências e filtros dos usuários. Essa linguagem de programação permite a interação com bancos de dados e o gerenciamento dinâmico das informações apresentadas aos usuários. Por fim, a implementação de funcionalidades interativas será realizada com o auxílio do JavaScript. Essa linguagem de programação é amplamente utilizada para tornar as páginas web mais dinâmicas e interativas, possibilitando a inclusão de recursos como filtros avançados, sugestões de palavras-chave, classificação de resultados e outras interações que aprimoram a usabilidade do sistema.

Em resumo, o projeto de desenvolvimento da extensão para navegador envolve a combinação de diversas tecnologias, como HTML, CSS, PHP, Python, JavaScript e Bootstrap. O uso do web scraping permitirá a coleta automatizada de informações sobre oportunidades de emprego, enquanto as demais tecnologias serão aplicadas para criar uma interface intuitiva, apresentar os dados de forma organizada e fornecer funcionalidades interativas aos usuários. O objetivo final é fornecer um recurso eficiente e prático para auxiliar indivíduos em sua busca por oportunidades profissionais, facilitando sua inserção no mercado de trabalho.

***Mockup das telas***

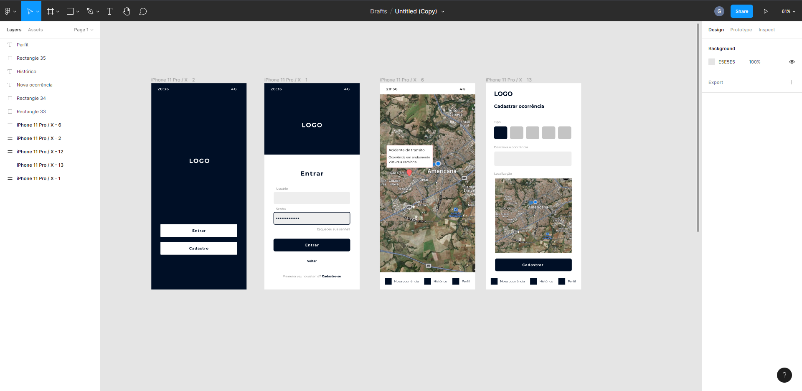
Para facilitar o modo de desenvolver o produto, e pensar em como devem ficar as telas, foram criadas as telas protótipo do projeto utilizando o Figma

Ilustração 2 – Logo Figma.



Fonte: vetores.org

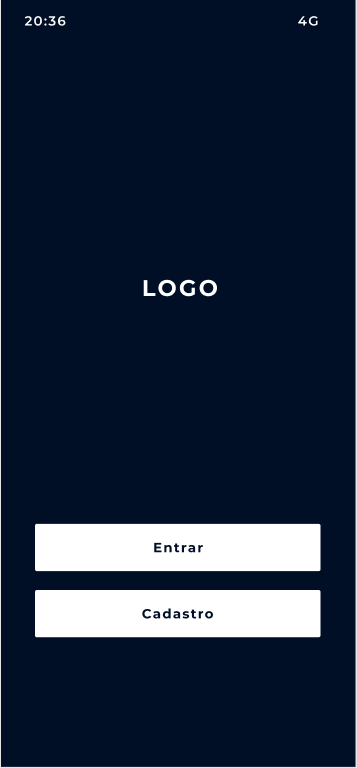
Ilustração 3 –Área de trabalho do Figma com mockup



Fonte: Acervo dos autores

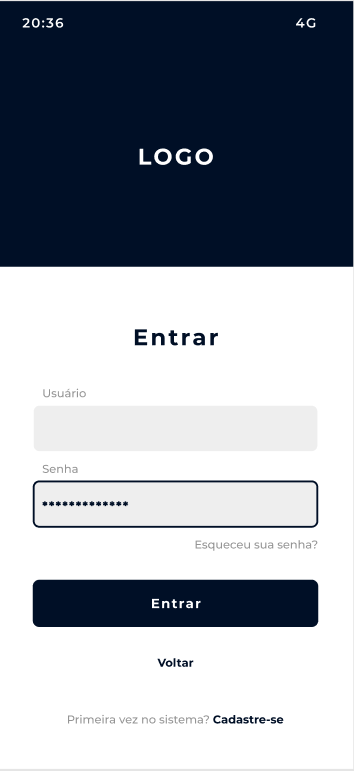
Com isso, foram geradas as seguintes telas para protótipo:

Ilustração 4 – Mockup tela.



Fonte: Acervo dos autores

Ilustração 5 – Mockup tela.



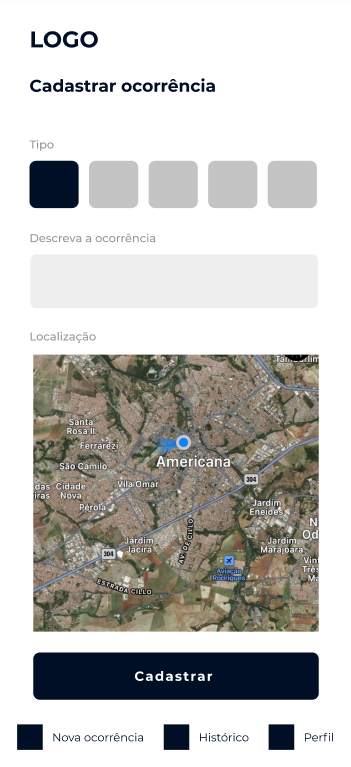
Fonte: Acervo dos autores

Ilustração 6 – Mockup mapa



Fonte: Acervo dos autores

Ilustração 7 – Mockup cadastro de ocorrência



Fonte: Acervo dos autores

***Desenvolvimento***

*Busca e coleta de vagas de emprego*

A busca por vagas de emprego pode ser um processo lento e cansativo. Porém, usando a automação e mineração de dados, é possível simplificar essa tarefa e obter resultados com mais eficiência. Neste artigo, vamos descrever um código em Python que utiliza a biblioteca Selenium e a API Flask para a coleta de vagas de emprego.

O código utiliza a biblioteca Selenium para interagir com o navegador da web. O objetivo é acessar um site de empregos, realizar uma pesquisa com base em um termo fornecido pelo usuário, ou pelo perfil do usuário, e extrair informações relevantes sobre as vagas encontradas. Para isso, são necessárias algumas etapas.

Primeiramente, importamos as bibliotecas necessárias, como o requests, Flask e Seleniudm. As opções do navegador são configuradas para que a automação e a mineração dos dados sejam executada de forma invisível, desta forma, o usuário não verá as interações com o navegador.

Em seguida, criamos uma rota na API Flask para receber uma requisição GET com o termo de busca fornecido pelo usuário. Esse termo é armazenado na variável "vaga". A URL do site de empregos é definida, e o navegador é inicializado utilizando o driver do Chrome.

@appi.route('/get/<string:msg>', methods=['GET', 'POST'])

A automação prossegue com o acesso ao site, o aceite dos cookies e a inserção do termo de busca no campo apropriado. Em seguida, o botão de busca é pressionado e aguardamos o carregamento dos resultados.

 Através da biblioteca Selenium, as vagas são localizadas na página e informações como título, empresa, salário, descrição, experiência desejada, tipo de contrato e jornada de trabalho são extraídas. As informações da vaga são armazenadas em um dicionário e adicionadas a uma lista chamada "vagas\_json".

Após coletar as vagas, enviamos a lista de dados coletados para um servidor local através do método POST da biblioteca requests.

response = requests.post(url, json=vagas\_json)

Em seguida, o status da resposta é verificado para garantir que os dados foram enviados. Por fim, o navegador retorna à página anterior e a lista de vagas é convertida para o formato JSON antes de ser retornada como resposta da requisição GET.

return json.dumps(vagas\_json)

Com essa automação e raspagem de dados, é possível coletar informações sobre vagas de emprego de forma rápida e eficiente.

*Classificador de avaliações:*

A análise de avaliações de empresas é um aspecto importante e decisório na busca por um ambiente de trabalho satisfatório. No entanto, buscar, ler e classificar manualmente um grande número de avaliações pode ser uma tarefa demorada. Nesta seção, apresentaremos um código em Python que utiliza técnicas de processamento de linguagem natural (NLP) e aprendizado de máquina para buscar e classificar avaliações de empresas, que são mostradas ao usuário, quando ele solicita ver as avaliações da empresa de uma determinada vaga.

O código do modelo classificador de avaliações começa importando as bibliotecas, como nltk pandas, e sklearn. Em seguida é carregado um arquivo Excel que contém algumas avaliações de empresas, o arquivo possui três colunas, sendo elas o título, a avaliação e a classificação da avaliação, se é bom, regular ou ruim.

O pré-processamento do texto é realizado removendo stop words e convertendo as palavras para letras minúsculas.

stop\_words = set(stopwords.words('portuguese'))

def preprocess\_text(text):

As avaliações são vetorizadas usando a técnica TF-IDF (Term Frequency-Inverse Document Frequency), que atribui pesos às palavras com base em sua importância no documento.

vectorizer = TfidfVectorizer()

X\_vectorized = vectorizer.fit\_transform(df['titulo'] + ' ' + df['avaliacao'])

O classificador SVM (Support Vector Machine) é treinado utilizando os dados vetorizados e as classificações fornecidas no arquivo Excel. Definimos uma função para classificar uma avaliação com base no modelo treinado.

def classify\_review(review):

*Busca de avaliações de empresa:*

Para coletar avaliações de empresas de um site, foi utilizado as mesmas bibliotecas do código que coleta as vagas, porém as avaliações são classificadas utilizando o modelo classificador citado anteriormente.

Importamos as bibliotecas, incluindo a função "classify\_review" do modelo classificador.

from classificador import classify\_review

Criamos uma rota na API Flask para receber uma requisição GET com o nome da empresa desejada. Utilizamos a biblioteca Selenium para acessar um site onde há avaliações de empresas. E através da biblioteca, as interações necessárias no site para buscar as avaliações da empresa que está anunciando a vaga.

Coletamos as informações relevantes das avaliações, como título, avaliação e o funcionário. Durante esse processo, as avaliações são classificadas a partir do modelo classificador.

classificacao = classify\_review(texto\_avaliacao)

As informações são armazenadas em um formato de dicionário e adicionadas a uma lista. A lista de avaliações classificadas é enviada para um servidor local utilizando o método POST da biblioteca requests.

Esses códigos em Python permitem automatizar a coleta e classificação de avaliações de empresas de forma eficiente. Com eles, é possível economizar tempo e obter insights valiosos sobre as experiências dos funcionários em diferentes empresas.

Em resumo, a combinação de técnicas de processamento de linguagem natural, aprendizado de máquina e automação pode proporcionar uma análise rápida e precisa das avaliações de empresas. Essas abordagens podem ser aplicadas em várias áreas, como recursos humanos e tomada de decisões empresariais. Ao utilizar esses códigos como base, é possível personalizar e adaptar o sistema para atender às necessidades específicas de análise de avaliações de empresas.

1. Resultados e Discussões

Discutindo sobre os resultados gerados durante o semestre, percebemos que esse novo processo criado pelo grupo irá facilitar muito o trabalho da GAMA já que o cadastro da ocorrência será feito com localização exata do usuário e com a utilização de fotos, pode aumentar muito o entendimento do problema. Também facilitará o processo de cadastro, não sendo mais necessário ligar para a central de atendimento. Acreditamos, também, que com o crescimento do projeto e a implementação da verificação de segurança da Idwall, o aplicativo irá ficar mais seguro ainda, diminuindo muito o risco de fraudes

*Missão*

Nossa missão é fazer com que o processo de ocorrências seja facilitado, aumentando a agilidade de seus processos e a segurança da cidade de Americana SP

*Visão*

Ser referência nacional no desenvolvimento de aplicativos de cadastro de ocorrências, pensando em expansão do aplicativo.

*Valores*

Nossos valores consistem em entregar um aplicativo condizente com o proposto com grande eficiência e segurança para os usuários administrativos e clientes.

*Sustentabilidade*

Todas as corporações da Polícia que começarem a utilizar esse sistema que estamos desenvolvendo eles deixaram de fora o uso de papéis, elásticos, clips entre outros aparatos que agridem a natureza para serem gerados e depois viram lixo que vão até os aterros sanitários. Com isso o desuso desses aparatos deixa o planeta menos explorado e mais limpo.

*Ação Social*

Nossa ação social consiste em todos os softwares desenvolvidos sejam de código aberto para que qualquer corporação por meio de um profissional consiga implementar esse sistema em sua corporação.

***Conclusão***

1. Conclusões

Em virtude dos fatos mencionados no nosso artigo, acreditamos que, de fato, essa mudança possa realmente revolucionar a forma de cadastro de ocorrências em todo o município, já que foi criado um processo inteligente, seguro e eficaz para a segurança da cidade. Podendo, futuramente, melhorar todo o processo de segurança nacional.

Referências

DOCUMENTAÇÃO FLUTTER. Disponível em: < <https://docs.flutter.dev/>>

PESQUISA FORBES TRANFORMAÇÃO DIGITAL Disponível em: <https://forbes.com.br/forbes-tech/2020/11/pandemia-faz-875-das-empresas-no-brasil-aceleraram-projetos-de-transformacao-digital/>

PESQUISA G1 DEMORA NO ATENDIMENTO DO 190 Disponível em: < https://g1.globo.com/pr/parana/noticia/2011/05/demora-no-atendimento-do-190-preocupa-quem-precisa-da-policia.html >

DOCUMENTAÇÃO MAPBOX. Disponível em: < https://docs.mapbox.com/ >

DOCUMENTAÇÃO FIREBASE. Disponível em: < https://firebase.google.com/docs >.

SITE GAMA. Disponível em: < http://gama.sp.gov.br/>.

DOCUMENTAÇÃO IDWALL. Disponível em: < https://docs.idwall.co/docs >.