

# **Desenvolvimento de Um Analisador Léxico para Avaliação de Currículos com Base em Palavras-chave**

**Instituição: Centro Universitário UNA - Campus Sete Lagoas**

**Curso: Ciência da Computação**

## **Resumo**

Este relatório descreve o desenvolvimento de uma ferramenta computacional com foco na leitura e análise automática de currículos profissionais no formato Word (.docx). A aplicação foi implementada em Python, utilizando bibliotecas de processamento textual para identificar e contar ocorrências de palavras-chave previamente definidas, como "Node", "React" e "Front End". O resultado final fornece uma métrica de compatibilidade entre o currículo e os requisitos desejados.

## **Introdução**

A triagem de currículos é uma etapa fundamental em processos seletivos. Muitas vezes, a análise inicial envolve a verificação de conhecimentos específicos. Este projeto visa automatizar essa etapa através do desenvolvimento de um analisador léxico que leia currículos digitais e identifique a presença de termos técnicos relacionados ao desenvolvimento web.

## **Objetivos**

### **2.1 Objetivo Geral**

Desenvolver uma aplicação capaz de ler arquivos de currículos no formato .docx e retornar um percentual de compatibilidade baseado em palavras-

chave.

## 2.2 Objetivos Específicos

- Ler currículos armazenados em uma pasta específica.
- Extrair texto bruto dos arquivos .docx utilizando Python.
- Identificar palavras-chave com expressões regulares.
- Calcular e exibir a porcentagem de "match" de cada currículo.
- Organizar os resultados em formato interpretável.

## Metodologia

A aplicação foi desenvolvida em Python, utilizando a biblioteca python-docx para leitura de documentos Word. A busca por palavras-chave foi implementada com expressões regulares (re), permitindo análise insensível a maiúsculas e minúsculas.

Estrutura funcional:

- Input: Arquivos .docx em uma pasta local (ex: /curriculos)
- Processamento: Extração de texto e busca de palavras-chave
- Output: Lista de currículos com palavras encontradas e percentual de compatibilidade

## Desenvolvimento

Durante a implementação, foram realizados os seguintes passos:

- Leitura dos arquivos com python-docx.
- Uso de re.search com a flag re.IGNORECASE para busca das palavras-chave.
- Cálculo do percentual de compatibilidade baseado no total de palavras encontradas.
- Apresentação dos resultados em terminal/console.

## Principais trechos do código (resumidamente):

```
def calcular_match(texto, palavras_chave):  
    encontradas = [p for p in palavras_chave if re.search(rf'\b{re.escape(p)}\b', te
```

```
return encontradas, len(encontradas) / len(palavras_chave) * 100
```

## Resultados e Ajustes

A aplicação apresentou desempenho satisfatório para o conjunto de teste composto por três currículos de exemplo. Foram realizados ajustes importantes:

- Adequação para leitura de arquivos Word ao invés de texto puro.
- Substituição de comparação direta de strings por expressões regulares.
- Inclusão de arredondamento de valores e organização dos resultados para melhor compreensão.

## Limitações:

- Não há detecção de sinônimos, variações ortográficas ou contexto semântico.
- Palavras compostas como "Front End" são tratadas literalmente.
- Não há interface gráfica; aplicação ainda é baseada em terminal.

## Conclusão

A aplicação cumpre seu papel como ferramenta auxiliar em processos seletivos. Com a possibilidade de expansão, incluindo o uso de técnicas de NLP, banco de dados ou interface web, ela pode se tornar uma solução robusta para análise automatizada de currículos.

## Referências

- Python Software Foundation. Python 3. <https://www.python.org>
- python-docx: Read and write Microsoft Word files with Python. <https://python-docx.readthedocs.io>
- re — Regular expression operations. <https://docs.python.org/3/library/re.html>