

Ambientes e Linguagens de Programação para Ciência de Dados

TP2

Licenciatura em ciência de dados

Universidade do Minho

Docente: Tiago João Fernandes Baptista

**Trabalho realizado por:**

David Costa – A102397

Joao Alves – A102395

Ioannis Kokkinovrachos – A106820

Índice

[Introdução 3](#_Toc185008446)

[Questão a) 4](#_Toc185008447)

[Código usado: 4](#_Toc185008448)

[Análise de Output: 5](#_Toc185008449)

[Resultados Obtidos: 5](#_Toc185008450)

[Interpretação dos Resultados: 6](#_Toc185008451)

[Questão b) 7](#_Toc185008456)

[Código usado: 7](#_Toc185008448)

[Análise de Output: 8](#_Toc185008449)

[Resultados Obtidos: 8](#_Toc185008450)

[Interpretação dos Resultados: 9](#_Toc185008451)

[Questão c) 10](#_Toc185008462)

[Código usado: 4](#_Toc185008448)

[Análise de Output: 6](#_Toc185008449)

[Resultados Obtidos: 6](#_Toc185008450)

[Interpretação dos Resultados: 6](#_Toc185008451)

[Questão d) 11](#_Toc185008467)

[Código usado: 4](#_Toc185008448)

[Análise de Output: 6](#_Toc185008449)

[Resultados Obtidos: 6](#_Toc185008450)

[Interpretação dos Resultados: 6](#_Toc185008451)

[Comparação com a TP1 12](#_Toc185008471)

[Dificuldades nesta TP 4](#_Toc185008448)

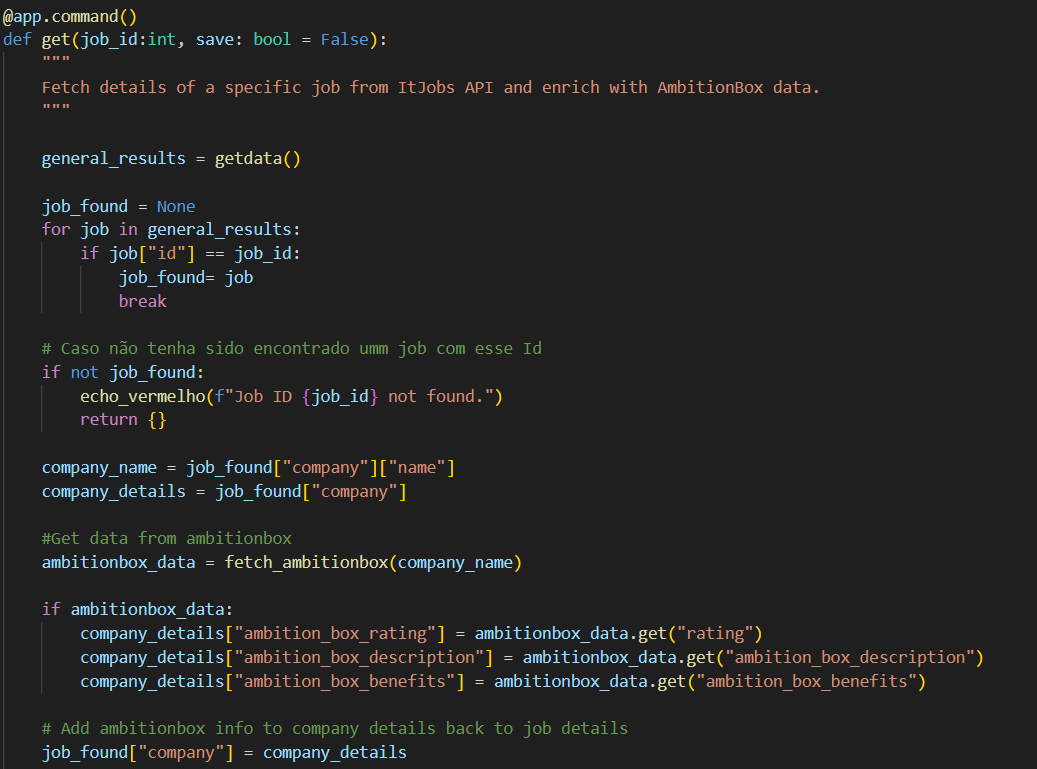
[Conclusão 19](#_Toc185008495)

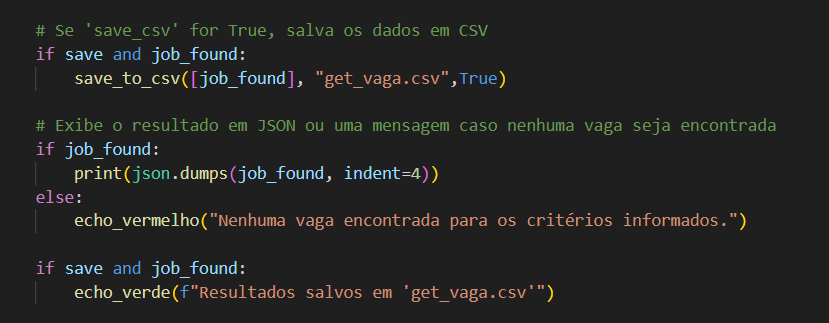
Introdução

Este relatório apresenta o desenvolvimento do Trabalho Prático (TP) realizado por nós, cujo principal objetivo foi a aplicação dos conhecimentos adquiridos em Web Scraping, processamento de dados e criação de interfaces de linha de comandos (CLI) utilizando a linguagem Python. O projeto teve como base a construção de uma solução funcional e interativa que permite aos utilizadores obter e enriquecer dados provenientes de uma REST API e de websites externos. Para tal, explorámos módulos como beautifulsoup4, requests e re, bem como a biblioteca typer, que simplifica a criação de CLIs eficientes e intuitivas. Adicionalmente, o trabalho integrou funcionalidades como a consulta de informações sobre ofertas de emprego e respetivas empresas, a análise estatística de vagas, e a recolha de competências profissionais mais solicitadas, com a possibilidade de exportação dos dados processados para ficheiros CSV. Explorámos também fontes de dados alternativas, reforçando o uso de abordagens criativas e inovadoras na recolha e enriquecimento de informação. Este relatório está organizado de forma a detalhar os processos implementados, as escolhas realizadas e os resultados obtidos, garantindo uma análise clara e objetiva de todas as etapas do desenvolvimento.

Questao a)

Código usado:





Análise do output:

O comando implementado permite ao utilizador consultar informações detalhadas sobre um determinado job, identificado pelo seu job\_id, recorrendo à API do ItJobs e enriquecendo os dados com informações adicionais extraídas do site AmbitionBox. O processo inicia-se com a pesquisa do job na lista de resultados gerais obtidos da API, verificando se o ID fornecido pelo utilizador corresponde a algum trabalho. Caso o trabalho seja encontrado, o nome da empresa associada é utilizado para realizar a extração de informações adicionais do site AmbitionBox, que incluem: Avaliação geral da empresa (ambition\_box\_rating); Descrição da empresa (ambition\_box\_description); Principais benefícios de trabalhar na empresa (ambition\_box\_benefits). Essas informações são integradas no objeto JSON correspondente ao trabalho e podem ser exibidas no terminal ou exportadas para um ficheiro CSV, caso o parâmetro save seja definido como True.

Resultados Obtidos:

A execução do comando get com um job\_id válido produz um objeto JSON enriquecido com informações sobre a empresa obtidas de fontes externas. Por exemplo:

{

"id": 125378,

"title": "Software Engineer",

"company": {

"name": "Tech Solutions",

"ambition\_box\_rating": 4.5,

"ambition\_box\_description": "Tech Solutions é uma empresa líder em inovação...",

"ambition\_box\_benefits": ["Horários flexíveis", "Seguro de saúde", "Formação contínua"]

},

...

}

Caso o save seja ativado, um ficheiro CSV denominado get\_vaga.csv será criado contendo os detalhes do trabalho e as informações enriquecidas.

Se o job\_id fornecido não for encontrado, o programa exibe a mensagem de erro:

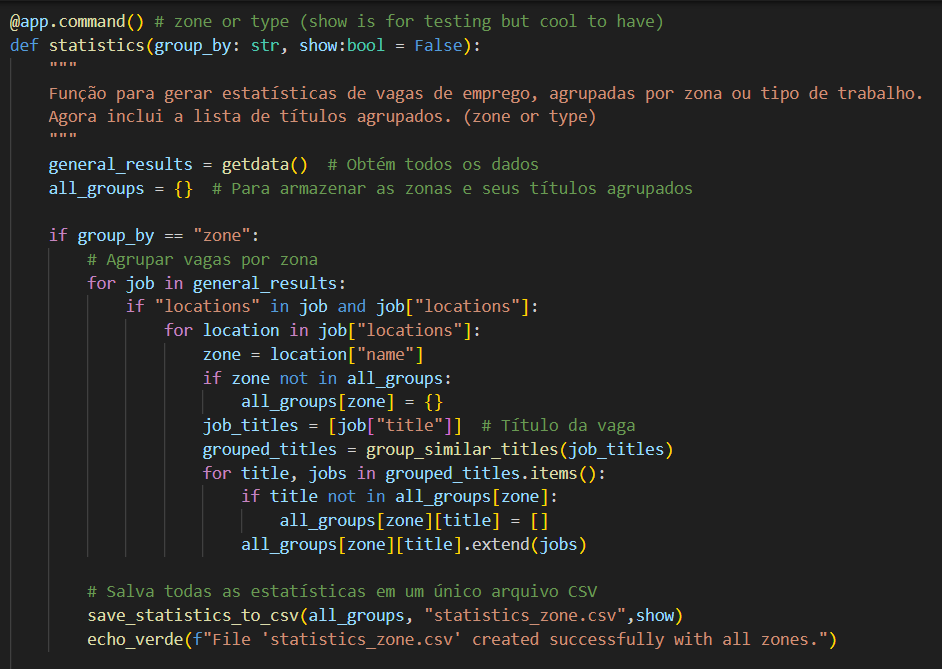
**Job ID {job\_id} not found.**

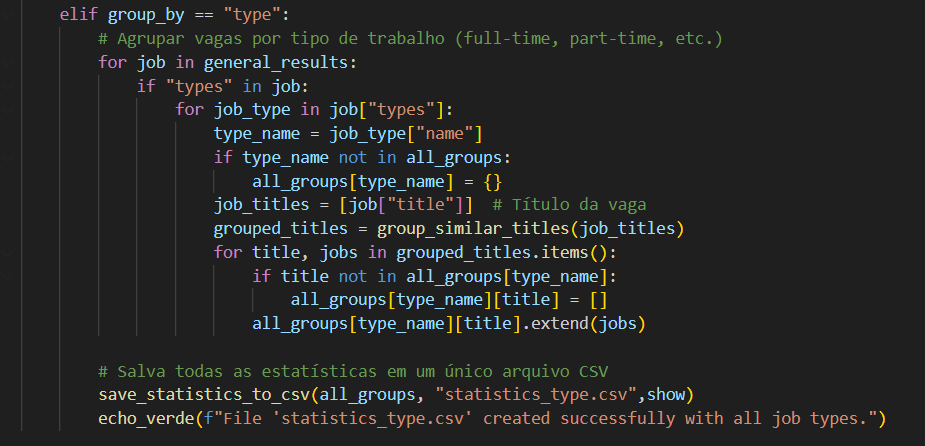
Interpretação dos Resultados:

Os resultados demonstram a capacidade do comando em combinar dados de múltiplas fontes de forma eficiente. Ao enriquecer as informações do trabalho com detalhes provenientes do AmbitionBox, o utilizador tem acesso a um conjunto mais completo de dados que pode ser utilizado para tomada de decisão ou análise de mercado. Além disso, a funcionalidade de exportação para CSV amplia as possibilidades de utilização dos dados, permitindo análises posteriores em ferramentas externas. O comando é particularmente útil para identificar oportunidades de emprego e avaliar as empresas que oferecem as vagas, proporcionando ao utilizador uma visão mais detalhada sobre o mercado de trabalho. A implementação foi desenvolvida de forma modular e flexível, facilitando futuras extensões ou adaptações. A integração com a biblioteca typer garante uma interface amigável e eficiente, adequada a diferentes utilizadores.

Questao b)

Código usado:





Análise do Output:

O comando implementado permite agrupar vagas de emprego com base em dois critérios:

1. **Zona geográfica** (zone): Agrupa as vagas de acordo com as localizações disponíveis no dataset.
2. **Tipo de trabalho** (type): Agrupa as vagas com base na categoria do trabalho (ex.: full-time, part-time).

A funcionalidade percorre todos os trabalhos disponíveis no dataset e organiza-os em grupos, onde cada grupo contém listas de títulos de vagas relacionados a uma zona ou a um tipo de trabalho. Para facilitar a análise e visualização, os dados processados podem ser exportados para um ficheiro CSV, cujo nome é definido dinamicamente com base no critério de agrupamento:

* **statistics\_zone.csv** (quando agrupado por zona);
* **statistics\_type.csv** (quando agrupado por tipo).

Além disso, há a opção de visualizar os dados no terminal ao definir o parâmetro show como True, permitindo testar a funcionalidade antes de salvar os resultados.

**Resultados Obtidos:**

**Agrupamento por Zona**

Exemplo de dados gerados para agrupamento por zona geográfica:

Zona,Título de Trabalho,Número de Vagas

Lisboa,Software Engineer,10

Lisboa,Data Scientist,5

Porto,Frontend Developer,7

Estes dados são exportados para o ficheiro **statistics\_zone.csv** e indicam o número de vagas disponíveis para cada tipo de trabalho em diferentes localizações.

**Agrupamento por Tipo**

Exemplo de dados gerados para agrupamento por tipo de trabalho:

Tipo de Trabalho,Título de Trabalho,Número de Vagas

Full-time,Software Engineer,20

Part-time,Web Developer,8

Internship,Data Analyst,4

Estes dados são exportados para o ficheiro **statistics\_type.csv** e oferecem insights sobre a distribuição de vagas por categoria de trabalho.

**Interpretação dos Resultados:**

A funcionalidade de estatísticas é uma ferramenta poderosa para identificar tendências no mercado de trabalho. O agrupamento por **zona geográfica** fornece uma visão clara sobre as localizações mais ativas no recrutamento, permitindo que candidatos priorizem regiões com maior número de vagas ou empresas concentrem esforços em áreas estratégicas.

Por outro lado, o agrupamento por **tipo de trabalho** ajuda a compreender quais categorias de emprego são mais procuradas, como empregos a tempo inteiro ou estágios, oferecendo insights valiosos tanto para candidatos quanto para recrutadores.

A integração com a exportação para CSV adiciona um valor significativo, permitindo que os dados sejam facilmente utilizados em análises externas ou relatórios mais detalhados. Além disso, a modularidade do código facilita futuras extensões, como a inclusão de novos critérios de agrupamento ou melhorias na apresentação dos resultados.

O comando, em conjunto com o dataset robusto da API, destaca-se como uma funcionalidade essencial para a análise e exploração de dados relacionados ao mercado de trabalho.

Questao c)