**Relatório *Command Line Interface***

Índice

[1.Introdução 1](#_Toc186580584)

[2. Funcionalidades 2](#_Toc186580585)

[2.1. Limpeza de HTML 2](#_Toc186580586)

[2.2. Listagem de N Trabalhos Recentes 2](#_Toc186580587)

[2.3. Filtragem de Trabalhos Full-Time por Empresa e Localização 2](#_Toc186580588)

[2.4. Extração de Salário 3](#_Toc186580589)

[2.5. Filtragem por Habilidades e Intervalo de Datas 4](#_Toc186580590)

[2.6. Informações Adicionais com AmbitionBox 4](#_Toc186580591)

[2.7. Estatísticas de Vagas 5](#_Toc186580592)

[2.8. Listagem das Habilidades mais comuns 6](#_Toc186580593)

[2.9. Informações Adicionais com Indeed 6](#_Toc186580594)

[3. Conclusão 7](#_Toc186580595)

# 1.Introdução

Este código foi desenvolvido para interagir com a API do site **itjobs.pt** e extrair informações detalhadas sobre vagas de emprego, podendo até enriquece-las com dados adicionais obtidos do site **AmbitionBox**.

O código possui funcionalidades que incluem filtragem de empregos por parâmetros como empresa, localização, e tipo de contrato, bem como exportação dos resultados para arquivos CSV. Este documento detalha as funcionalidades, escolhas de implementação, e uso de expressões regulares no processamento.

# 2. Funcionalidades

## 2.1. Limpeza de HTML

A primeira funcionalidade implementada foi a limpeza das tags html do campo body, de forma a ter uma descrição limpa para melhor exibição, quer no terminal, quer no ficheiro csv.

Para encontrar as tags, usamos a seguinte expressão regular r"<[^>]+>", que encontra todas as tags HTML (abrangendo <tag> e </tag>)

## 2.2. Listagem de N Trabalhos Recentes

A funcionalidade ***listar\_n\_trabalhos*** foi desenvolvida para apresentar as *n* vagas de emprego mais recentes disponíveis na API, ordenadas por data de publicação. A ideia central é fornecer uma visualização das oportunidades mais recentes para o usuário.

Os dados são ordenados com base no campo *publishedAt*, que contém a data de publicação de cada vaga. Essa ordenação assegura que a lista seja cronologicamente precisa.

**Em seguida,** a função *limpar\_html* é usada para remover qualquer conteúdo HTML do campo body antes de exibi-lo ou exportá-lo. Isso torna a saída mais legível e acessível.

Caso o parâmetro *export\_csv* seja fornecido, a função exporta os resultados para um arquivo CSV. Caso contrário, as vagas são exibidas diretamente no terminal.

A funcionalidade foi projetada para ser acessível a diferentes tipos de usuários, desde quem precisa de uma lista rápida no terminal até aqueles que preferem salvar os dados para análises posteriores.

## 2.3. Filtragem de Trabalhos Full-Time por Empresa e Localização

A funcionalidade ***full\_time\_emp*** foi projetada para encontrar vagas de trabalho que sejam:

* **Full-time** (tempo integral).
* Relativas a uma **empresa específica**.
* Disponíveis numa **localização específica**.

Ela permite a listagem dessas vagas, com a opção de limitar o número de resultados exibidos ou exportar os dados filtrados para um arquivo CSV.

**Para tal definimos os seguintes parâmetros:**

* + company: Nome da empresa para o filtro (campo company.name).
  + location: Localização desejada para a vaga (campo locations).
  + limit: Quantidade máxima de vagas a serem exibidas.
  + export\_csv: Nome do arquivo para salvar os resultados no formato CSV (opcional).

A função realiza a filtragem com base em critérios bem definidos e usa listas compreensivas para simplificar o código, ou seja verifica três condições para incluir a vaga na nova lista:

* + O nome da empresa *(vaga['company']['name'])* é igual ao nome informado *(company).*
  + A localização da vaga *(vaga['locations'])* contém a cidade ou região especificada *(location).*
  + O tipo de trabalho *(vaga['types'])* inclui *"Full-time".*

Cada vaga que atende às condições será incluída na nova lista.

## 2.4. Extração de Salário

Esta função é composta por duas funcionalidades principais: uma função para extração de informações salariais *(extract\_wage)* e um comando principal que integra essa funcionalidade ao fluxo de execução do programa *(salar*y).

A função extract\_wage é responsável por processar os dados de uma oferta de trabalho específica, identificando informações salariais que podem estar presentes no campo "wage" ou descritas textualmente no campo "body".

Primeiramente, a função verifica se o campo "wage" contém um valor explícito, se assim for retorna esse valor com uma mensagem clara indicando que o salário foi localizado diretamente.

Caso contrário, a função recorre ao campo "body" para tentar identificar padrões salariais utilizando expressões regulares. Essas expressões permitem localizar valores monetários em diferentes formatos, como “€3000”, “1000 USD” ou “$5000”.

Se nenhuma informação for encontrada, a função retorna uma mensagem informando a ausência de dados salariais.

A utilização de expressões regulares na função *extract\_wage* foi essencial para capturar padrões variados de representação de salários. A regex utilizada permite identificar valores monetários com símbolos no início ou no final (como “€1000” e “1000 USD”), bem como números simples seguidos ou precedidos de abreviações. Essa flexibilidade é necessária para processar informações salariais que não seguem um padrão fixo.

## 2.5. Filtragem por Habilidades e Intervalo de Datas

O comando *skills* foi desenvolvido para filtrar ofertas de trabalho com base em habilidades específicas fornecidas pelo usuário, dentro de um intervalo de datas definido, permitindo também a exportação dos resultados em formato CSV, se necessário.

O processo começa com a obtenção da lista de ofertas de trabalho por meio de uma requisição HTTP à API. Os dados retornados são extraídos do campo "*results*", contendo informações como título, empresa, descrição, data de publicação e outras.

A procura pelas *skills* é feita ao verificar, de forma sequencial e insensível a capitalização, se cada habilidade fornecida pelo usuário está presente no campo *"body"* da descrição da vaga. Esse processo garante que apenas as ofertas que mencionam todas as habilidades desejadas sejam incluídas nos resultados finais.

Para filtrar as ofertas, a função converte as datas de entrada fornecidas pelo usuário (inicial e final) para objetos *datetime*, usando *datetime.strptime*. Em seguida, verifica se a data de publicação de cada oferta está dentro do intervalo especificado.

Apenas as ofertas que satisfazem ambas as condições – publicação no intervalo de datas e presença de todas as habilidades no texto – são consideradas no resultado final.

## 2.6. Informações Adicionais com AmbitionBox

Esta funcionalidade obtém dados de duas fontes principais: a API do itjobs.pt, que contém informações sobre as vagas, e o site AmbitionBox, que fornece detalhes sobre as empresas.

A função *get\_job\_info* é a função principal, sendo chamada quando o usuário quer obter informações sobre uma vaga específica. Ela começa por chamar a função *itjobs\_data*, que é responsável por fazer uma requisição para a API do itjobs.pt, procurando as vagas paginadas até encontrar a vaga com o *job\_id* fornecido. Se a vaga for encontrada, os seus dados são retornados.

Em seguida, o nome da empresa responsável pela vaga é extraído dos dados retornados da API do itjobs.pt. Esse nome da empresa é então passado para a função *ambitionbox\_data*, que acede ao site AmbitionBox para pesquisar informações adicionais sobre a empresa, como a sua classificação, descrição e benefícios oferecidos. Para isso, o nome da empresa é formatado de acordo com o padrão utilizado no AmbitionBox (minúsculas, espaços substituídos por hífens).

Os dados retornados pela função ambitionbox\_data são combinados com os dados da vaga, a função faz uma requisição para a página de visão geral da empresa e usa a biblioteca BeautifulSoup para extrair as informações de interesse: classificação, descrição e benefícios.

Se algum desses dados não estiver disponível, a função retorna um valor padrão, como "N/A", para garantir que a ausência de dados não cause erros.

Esses dados combinados são então exibidos no formato JSON usando a função typer.echo, o que facilita a visualização e leitura.

Além disso, a função permite a exportação desses dados para um arquivo CSV, caso o usuário forneça o parâmetro export\_csv.

## 2.7. Estatísticas de Vagas

Esta funcionalidade tem como objetivo reunir informações sobre o número de vagas disponíveis para diferentes títulos e localizações a partir de uma API paginada, e exportar os dados agregados para um arquivo CSV.

A função principal, *statistics*, começa por iniciar um dicionário *job\_counts* que será usado para contar quantas vagas existem para cada combinação de título e localização.

Em seguida, o código começa a iterar pelas páginas de resultados da API, a variável *has\_more\_results* controla a iteração e continua até que não haja mais resultados.

Para cada página, a função *obter\_dados* é chamada para fazer uma requisição à API.

O código percorre os dados de cada trabalho, extrai o título e as localizações associadas, e cria uma chave única para cada combinação título-localização. Se a chave já existir no dicionário *job\_counts,* o contador é incrementado; caso contrário, uma nova chave é adicionada com o valor 1.

O código continua esse processo de pesquisa e contagem para cada página até que todos os resultados tenham sido processados. Uma vez que a contagem de todas as combinações título-localização tenha sido feita, o próximo passo é criar uma lista *aggregated\_data*, que armazena os dados no formato de dicionários com o título, a localização e o número de vagas.

O código, portanto, permite compilar informações sobre as vagas por título e localização, agregá-las e exportá-las para um arquivo CSV, tornando o processo de análise de vagas mais organizado e acessível.

## 2.8. ****Listagem das Habilidades mais comuns****

A funcionalidade principal é a extração de informações sobre vagas de emprego a partir do site AmbitionBox.

O processo envolve três etapas principais: obtenção de URLs de vagas, extração das habilidades dessas vagas e a contagem das habilidades mais comuns.

Primeiramente, a função *job\_urls(job\_title: str)* é responsável por pegar no título de um emprego, formatá-lo para a URL correta do AmbitionBox, realizar uma requisição HTTP para aceder a essa página e extrair os links das vagas de emprego disponíveis para aquele cargo. Esses links são retornados como uma lista de URLs.

Em seguida, a função *skills\_from\_job(job\_url: str)* acede cada URL de vaga, faz uma requisição HTTP e usa a biblioteca BeautifulSoup para analisar o HTML da página da vaga. Ela procura por todas as habilidades listadas nessa vaga, localizando as tags HTML específicas que indicam essas habilidades. A função então retorna uma lista com as habilidades encontradas, convertidas para minúsculas e sem espaços extras.

A função principal, *list\_skills(job\_title: str, export\_csv: Optional[str] = None),* usa a lista de URLs das vagas de trabalho, obtidas pela função *job\_urls*, e, para cada URL, chama a função *skills\_from\_job* para extrair as habilidades. Todas as habilidades extraídas são armazenadas numa lista, que é depois processada para contar a frequência de cada habilidade usando o Counter da biblioteca collections.

**2.9. Informações Adicionais com Indeed**

Nesta funcionalidade é possível extrair informações relacionadas a um trabalho específico, identificado pelo seu ID, combinando dados de duas fontes. Os dados extraídos incluem avaliações gerais de empresas e descrições, além de detalhes específicos sobre a vaga.

O código utiliza a função *indeed\_data*para procurar informações sobre uma empresa no site Indeed.

Primeiro, o nome da empresa é obtido por meio da função *find\_company\_name(job\_id).* Esse nome é formatado substituindo espaços por hífens, adequando-o ao padrão de URLs do Indeed.

Em seguida, é feita uma requisição HTTP para aceder à página da empresa no site. Se a requisição for bem-sucedida, o conteúdo HTML da página é extraído utilizando a biblioteca BeautifulSoup e retornado para análise posterior.

A função *indeed\_data2* é responsável por processar o HTML obtido e extrair informações relevantes sobre a empresa. O código pesquisa a avaliação geral da empresa usando uma classe CSS específica. A descrição da empresa é extraída de um bloco de texto identificado por atributos específicos. As informações extraídas, como a avaliação geral e a descrição da empresa, são organizadas num dicionário para serem combinadas com outros dados posteriormente.

A função principal, *get\_job\_indeed*, realiza a integração entre os dados extraídos da API e os dados do Indeed. Os dados das duas fontes são então combinados num único dicionário, exibido no terminal em formato JSON. Opcionalmente, o usuário pode exportar esses dados para um arquivo CSV, fornecendo um nome de arquivo por meio do parâmetro *export\_csv.*

# **3. Conclusão**

Este código fornece uma solução abrangente para extração e processamento de informações de vagas. Ele utiliza técnicas modernas de ***scraping***, ***APIs*** e ***expressões regulares*** para atender a diferentes necessidades, desde a análise até à exportação. Com ajustes e extensões, pode ser ampliado para lidar com mais fontes de dados e funcionalidades.

**Realizado por:**

Maria Eduarda Ferreira A106829

João Pedro Duarte A103621

Rafael Ferreira A106819