

AutoTS

Projeto de Computational Thinking with Python

Azor Tartuce - 563995
Daniel Oliveira de Souza - 566284
Lucas de Almeida Pires - 562757

Sumário

Sumário.....	2
Introdução.....	3
Instalação.....	4
1. Requisitos.....	4
2. Clone o Repositório.....	4
3. Instale as Dependências.....	4
4. Execute o Projeto.....	4
Visão Geral e Uso do Sistema.....	5
1. Menu Inicial.....	5
2. Gerenciamento de Candidatos.....	5
3. Gerenciamento de Vagas e Empresa.....	6
4. Análise por IA de Vaga para Candidato.....	6
5. Visualização de Histórico de Resultados anteriores.....	7
Dicionário de Dados.....	8
Candidato.....	8
Vaga.....	8
Empresa.....	9
Historico Resultados.....	9
Anexos.....	10

Introdução

Sistema de Recrutamento Inteligente com IA

Este projeto é um **Sistema de Rastreamento de Candidatos (ATS)** inteligente, desenvolvido em Python como uma aplicação de console (CLI), que visa modernizar e otimizar o processo de recrutamento e seleção.

O núcleo do sistema permite o gerenciamento completo (CRUD) de dois pilares essenciais: perfis detalhados de **candidatos** (incluindo habilidades, experiências, formação e projetos) e **vagas** em aberto, além do perfil da própria empresa contratante.

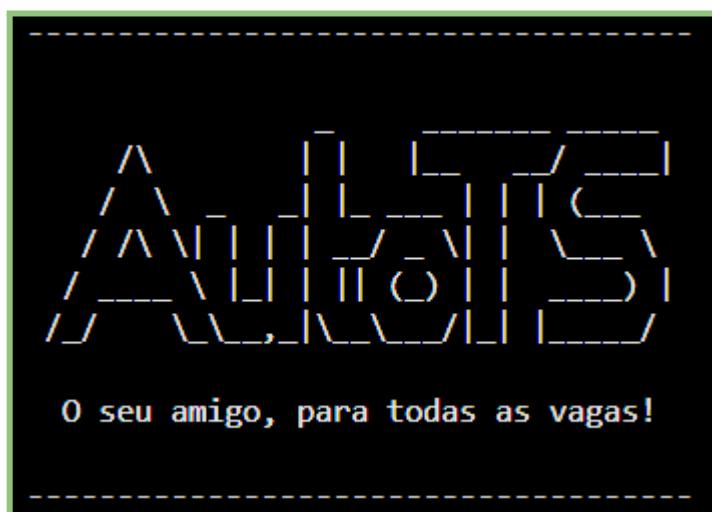
O principal diferencial do projeto é sua integração com a **IA generativa do Google (Gemini)**. Ao iniciar uma análise, o sistema coleta os requisitos da vaga, o contexto da empresa e os perfis de todos os candidatos cadastrados. Esses dados são enviados à IA, que realiza um cruzamento de informações para ranquear e selecionar os candidatos mais qualificados.

O objetivo central é criar um processo de seleção mais **justo, imparcial e eficiente**, focado puramente em dados e méritos, ajudando a **mitigar vieses** humanos inconscientes que podem prejudicar a diversidade e a qualidade das contratações.

Notavelmente, o projeto está alinhado a dois **Objetivos de Desenvolvimento Sustentável (ODS) da ONU**:

1. **ODS 10 (Redução das Desigualdades)**: Ao promover a igualdade de oportunidades no acesso ao emprego, garantindo que os candidatos sejam avaliados de forma justa e imparcial por suas qualificações.
2. **ODS 9 (Inovação e Infraestrutura)**: Ao aplicar tecnologia de ponta (Inteligência Artificial) para inovar e modernizar uma infraestrutura de negócios tradicional (Recursos Humanos).

O sistema também inclui funcionalidades para salvar os resultados da IA em um histórico, permitindo auditoria e consulta futura, e oferece a opção de encerrar o processo seletivo, limpando automaticamente a vaga preenchida e os candidatos contratados da base de dados.



Logo da aplicação - TUI

Instalação

1. Requisitos

- Ter Python 3 instalado no OS.
- Conexão com Internet
- Ter 10MB de espaço disponível

2. Clone o Repositório

Abra seu terminal ou prompt de comando e use o `git clone` para baixar os arquivos do projeto:

```
git clone https://github.com/Global-Solution-Future-At-Work/Python-AutoTS.git  
cd Python-AutoTS
```

3. Instale as Dependências

Este projeto precisa da biblioteca `requests` para funcionar. (Opcional: usar um ambiente virtual (`venv`) para manter as instalações organizadas.). Disponibilizamos também o arquivo `requirements.txt` na raiz do projeto com as bibliotecas utilizadas.

```
pip install requests
```

4. Execute o Projeto

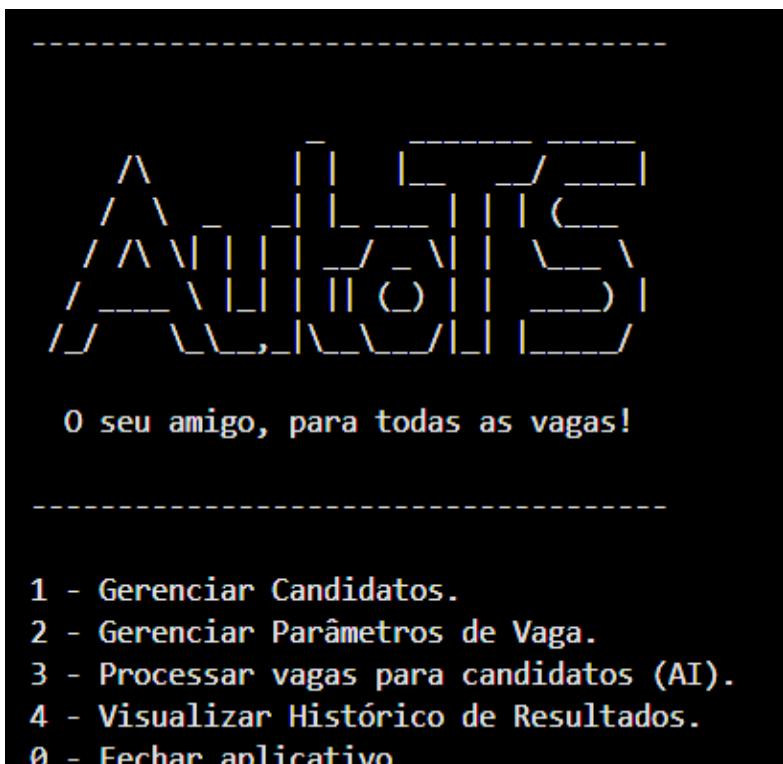
Após instalar as dependências e configurar a chave, você pode iniciar o programa.

```
python main.py
```

```
# Se estiver usando "py" (comum no Windows)  
py main.py  
# Se estiver usando "python3" (comum em sistemas Unix-like)  
python3 main.py
```

Visão Geral e Uso do Sistema

1. Menu Inicial



Este é o Menu Inicial do sistema. Ele disponibiliza os módulos necessários para a configuração central do ambiente. A ativação e o ajuste destes módulos são cruciais, pois tornam o uso da Inteligência Artificial (IA) possível. É nesta etapa que se garante que a IA esteja pronta para realizar a análise de vagas de forma eficiente.

É importante notar que é necessário configurar a **opção 1** e a **opção 2** primeiro, para que assim a **opção 3** funcione corretamente. Caso essa ordem não seja seguida, o

sistema irá avisar que é preciso adicionar os dados básicos antes de prosseguir.

2. Gerenciamento de Candidatos

```
Lista de candidatos registrados
=====
ID: 2
NOME: Pessoa B
RESUMO: Sou um desenvolvedor Frontend com habilidades em ReactJs e Angular
IDIOMAS ->
-----
idioma: Inglês
ano: 2021
-----
ID: 1
NOME: Pessoa A
RESUMO: Sou desenvolvedor Backend de Python com 10 anos de experiência
IDIOMAS ->
-----
idioma: Inglês
ano: 2020
```

```
1 - Ver lista de candidatos registrados.
2 - Ver candidato detalhado por ID.
3 - Registrar um candidato.
4 - Atualizar um candidato.
5 - Deletar um candidato.
0 - Voltar ao menu principal.
Selecione qual opção você deseja usar: |
```

Esta seção é focada no gerenciamento de todos os candidatos do processo seletivo da empresa. Com todas as funções de CRUD (Criar, Ler, Atualizar, Excluir) implementadas, ela permite que as alterações necessárias sejam feitas de forma eficiente.

Isso possibilita uma análise mais precisa e justa, baseada unicamente na capacidade do candidato, ajudando a evitar vieses e preconceitos. Para mais detalhes sobre a estrutura, consulte o dicionário de dados de 'Candidato' no capítulo 'Dicionário de Dados'.

3. Gerenciamento de Vagas e Empresa

```
1 - Ler Dados e Ramos Empresariais.  
2 - Definir Dados e Ramos Empresariais.  
3 - Ver Padrões de Vaga em aberto.  
4 - Criar Padrão de Vaga.  
5 - Atualizar Padrão de Vaga.  
6 - Deletar Padrão de Vaga.  
0 - Voltar ao menu Principal.  
Seleciona qual opção você deseja usar:
```

```
Vagas em aberto no sistema:  
=====  
id: 1  
descricao_requisitos: Precisamos desenvolver um frontend em ReactJS com Bootstrap e PrimeNg  
quantidade_vagas: 1  
=====
```

Esta seção é dedicada a definir as informações essenciais da empresa contratante e os detalhes específicos da vaga. Esses dados são fundamentais, pois servem de base para estruturar o processo seletivo que será analisado pela IA.

O sistema oferece operações de CRUD (Criar, Ler, Atualizar, Excluir) tanto para o cadastro da empresa quanto para o da vaga, garantindo total flexibilidade para que o usuário possa editar os dados conforme necessário.

Para mais detalhes, consulte os modelos de 'Empresa' e 'Vaga' no capítulo 'Dicionário de Dados'.

4. Análise por IA de Vaga para Candidato

```
Verifique se as informações estão corretas:  
Descrição e requisitos da vaga:  
Precisamos desenvolver um frontend em ReactJS com Bootstrap e PrimeNg  
Quantidade de vagas a preencher: 1  
Deseja continuar? Ao digitar 'SIM' a IA irá começar a analisar a base de dados:
```

```
Candidatos Selecionados: [2]  
Resumo Análise:  
A análise considerou a vaga de desenvolvimento frontend em ReactJS com Bootstrap e PrimeNg. O candidato Pessoa B possui habilidades técnicas em entendimento e experiência relevante, alinhando-se melhor aos requisitos da vaga em comparação com Pessoa A, que foca em backend. Como há apenas uma vaga em aberto, Pessoa B é o candidato selecionado.  
Deseja salvar o resultado? Digite 'SIM' para salvar:
```

Esta área é dedicada ao processamento de dados e à geração de resultados com base na vaga selecionada. O processo de análise geral dos candidatos pela IA é automático, iniciando assim que o usuário seleciona a vaga e confirma a operação.

Como resultado, o sistema retorna um relatório indicando os IDs dos candidatos mais indicados, acompanhado de uma descrição detalhada que justifica a decisão.

Você pode salvar o resultado obtido. O sistema também oferece a opção de deletar os candidatos e as vagas do processo, já que a posição pode ter sido preenchida. As respostas são salvas no banco de dados de histórico `resultados.json`.

5. Visualização de Histórico de Resultados anteriores

```
1 - Visualizar Histórico de Resultados.
2 - Deletar Resultados Anteriores.
0 - Voltar ao menu Principal.
Selecione qual opção você deseja usar: [1]

Lista de resultados anteriores:
=====
ID: 1
DESCRIÇÃO VAGA: Contrata-se pessoas com capacidade de prestar suporte a cliente com dúvidas em tecnologia
CANDIDATOS SELECIONADOS: 1;
RESUMO DA ANÁLISE:
A vaga em aberto é para Suporte a Sistemas, com foco em atendimento ao cliente com dúvidas em tecnologia. O candidato Kaiky (ID 1) se destaca por possuir experiência direta na área de Suporte, com habilidades em Resoluções de Problema e soft skills como Comunicação e Amigável, que são essenciais para a função. Sua formação em Informática e interesse em Dados e Estatística também agregam valor. O candidato Daniel (ID 2), apesar de ter habilidades técnicas em Backend e formação em Informática, não possui experiência direta na área de suporte, o que o torna menos alinhado com os requisitos da vaga. Portanto, Kaiky é o candidato mais adequado para a única vaga disponível.
=====
```

Esta seção é dedicada à consulta de resultados anteriores, permitindo análises futuras e um acompanhamento histórico dos processos seletivos.

Caso seja do interesse do usuário, o sistema oferece a funcionalidade de excluir resultados que já são considerados obsoletos, ajudando a manter a base de dados sempre relevante e organizada.

Para visualizar a estrutura completa desses dados, acesse o dicionário de 'histórico resultados' no capítulo 'Dicionário de Dados'.

Dicionário de Dados

Candidato

Nome	Tipo de Dado/Estrutura
id	int
nome	str
foto	str
cargo	str
resumo	str
localizacao	str
area	str
habilidades_tecnicas	list[str]
soft_skills	list[str]
experiencias	dict{empresa: str, cargo: str, inicio: str, fim: str, descricao: str}
formacao	dict{curso: str, instituicao: str, ano: str}
projetos	list[str]
certificacoes	list[str]
idiomas	dict{idioma: str, ano: str}
area_interesses	list[str]

Vaga

Nome	Tipo de Dado/Estrutura
id	int
descricao_requisitos	str
quantidade_vagas	str

Empresa

Nome	Tipo de Dado/Estrutura
nome	str
descricao	str

Historico Resultados

Nome	Tipo de Dado/Estrutura
id	int
candidatos_selecionados	list[int]
resumo_analise	str
descricao_requisitos	str

Anexos

Link do vídeo para o Youtube mostrando a solução:

[Vídeo Demonstração Python WorkQueue](#)