Projeto:

Pipeline de Dados de Vagas - Global System

Data de Emissão:

12/02/2023

Elaborado por:

Gabriel Ribeiro de Araújo

Sobre o autor

Olá, meu nome é Gabriel Ribeiro. Sou formado em Ciência da Computação e estou finalizando a Pós-graduação em Ciência de Dados e Machine Learning. Sou um entusiasta pelo mundo da tecnologia e computação. Admiro bastante o poder que nós cientistas da computação temos de causar grandes impactos na sociedade por meio da aplicação do conhecimento computacional.

Linkedin: https://www.linkedin.com/in/gabrielscientist/
Github: https://github.com/gableisure

"***O que você vê na sua mente é o que vai ter na mão.***"

Bob Proctor

LISTA DE SIGLAS

API	*Application Programming Interface*
ETL	*Extract, transform and load*
PDI	*Pentaho Data Integration*

LISTA DE FIGURAS

1. Objetivos

1.1. Objetivos do projeto

Desenvolver uma pipeline de dados que contemple a aquisição dos dados de vagas da Global System, o ETL, uma API e um interface gráfica de acesso a esses dados.

1.2. Objetivos do documento

O documento tem como objetivo apresentar e descrever o projeto desenvolvido bem como sua implantação em sistemas Windows.

2. Considerações iniciais

O projeto em questão foi desenvolvido para ser executado localmente em uma máquina. O escopo do projeto não contempla implementação de acesso a servidores remotos. Todos os dados utilizados aqui foram adquiridos no site da *Global System* no painel de vagas.

O repositório do projeto encontra-se em: https://github.com/gableisure/prj_sg.

2.1. Tecnologias utilizadas

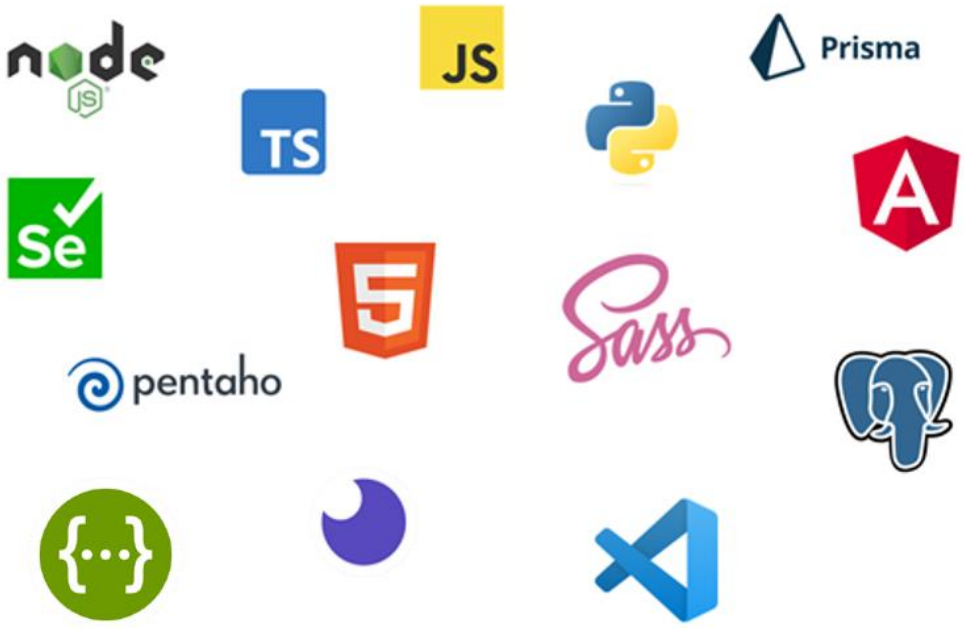

Figura 1: Tecnologias utilizadas

Fonte: Elaborada pelo autor

3. Projeto

3.1. Descrição do projeto

O projeto de implementação da pipeline de dados de vagas da Global System é sustentado por 4 pilares centrais de interoperabilidade:

- Web Scraping;
- Processo de ETL;
- API;
- Interface gráfica.

O processo inicia-se com o módulo de scraping realizando o acesso a página de vagas. O uso da biblioteca Selenium do Python nos possibilita acessar o HTML e extrair as informações das páginas e trabalhar com o Python. A partir daí, as informações de vagas são salvas em um arquivo csv **jobs.csv** no diretório "**file_to_process**" do projeto. É nesse diretório que o PDI vai procurar o arquivo para iniciar o processo de ETL.

Com o arquivo devidamente salvos no diretório, o PDI realiza a leitura, aplica um pré-processamento em cima desses dados e carrega em uma tabela (tb_jobs) no banco de dados PostgreSQL. Após o pré-processamento dos dados temos como output uma tabela com as seguintes colunas:

Tabela: tb_jobs	
Coluna	**Descrição**
id_job	ID da vaga
title_job	Título da vaga
date_publication	Data de publicação da vaga
link_job	Link da vaga
update_at	Data de atualização dos dados

Depois do ETL, o backend, construído com o ORM Prisma e Typescript, é responsável por consumir os dados desse banco de disponibilizá-los por meio de uma API. A API possui apenas um end point, **/api/jobs** que retorna todos os dados da tabela.

Com a API disponível, o frontend construído com o framework Angular, é o responsável por consumir essa API e exibir as vagas em cards com barra de pesquisa na parte superior da página para filtrar as vagas.

A figura abaixo ilustra de forma genérica os módulos do projeto.

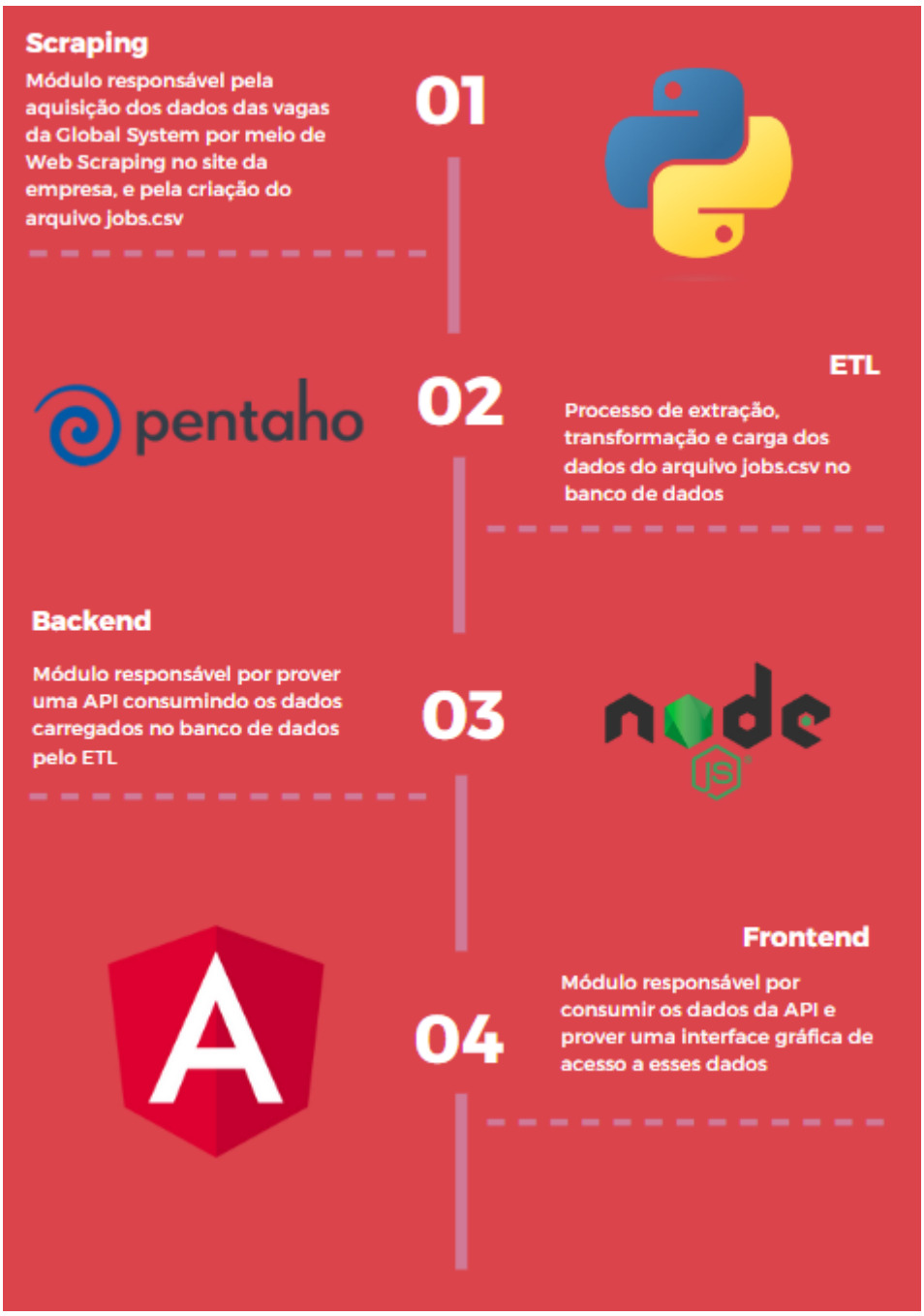

Figura 2: Pilares do projeto

Fonte: Elaborada pelo autor

3.2. Estrutura do projeto

O projeto está estruturado como na figura abaixo.

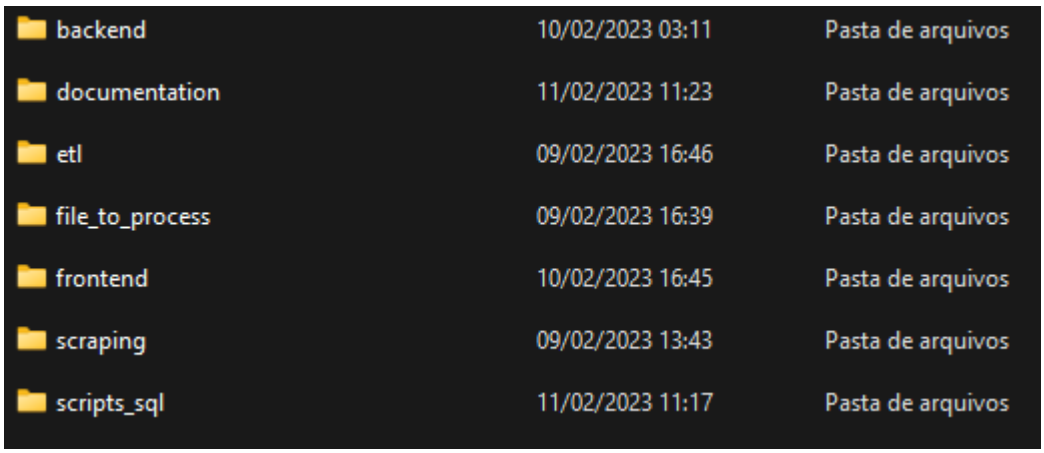

Figura 3: Estrutura do projeto

Fonte: Elaborada pelo autor

- **backend**: Diretório responsável por armazenar o código fonte da API.
- **documentation**: Diretório responsável por armazenar documentos relativos ao projeto.
- **elt**: Diretório responsável por armazenar os *Jobs* e *Transformations* do PDI.
- **file_to_process**: Diretório responsável por armazenar o arquivo csv gerados pelo módulo de scraping.
- **frontend**: Diretório responsável por armazenar o código fonte do app.
- **scraping**: Diretório responsável por armazenar o código fonte do scraping.
- **scripts_sql**: Diretório responsável por armazenar os scripts SQL utilizados no projeto.