Gmail **Alex M B de Souza <alexkrypto.ti@gmail.com>**

Engenheiro de dados Júnior

NGI SECOGE <ngi.secoge.sesau@gmail.com> 27 de janeiro de 2025 às 11:44
Para: Alex M B de Souza <alexkrypto.ti@gmail.com>

Olá, Alex!

Analisamos o seu currículo enviado para a vaga aberta de Engenheiro de Dados Jr na Secretaria de Saúde do Recife. **Você passou à etapa seguinte do processo seletivo, que consiste no desafio descrito abaixo!**

Atente-se para o prazo de envio abaixo.

Aqueles que passarem desta etapa farão uma conversa com a gestora para avaliação de competências, histórico e alinhamento de expectativas sobre a vaga até o final da primeira semana de fevereiro.

Objetivo

O objetivo geral desse desafio é a **construção de um pipeline de dados** utilizando **Python** e **SQL**. Mais especificamente, você vai:

1. Construir um pipeline para **extrair**, **transformar**, e **carregar** dados à sua escolha, usando Python e/ou SQL.
2. Elaborar duas consultas sobre esses dados, usando SQL.
3. Escrever um breve relatório sobre esses dados.

Vamos entrar em detalhes sobre cada um desses passos:

Passos

1. Extração de dados usando Python

Dados

Escolha um conjunto de dados *abertos ao público* para utilizar nesse desafio.

- Recomendamos que você escolha dados que você ache interessantes ou que você conheça bem. Não precisa se limitar a dados de saúde!
- Você pode pegar os dados já prontos (em arquivos .csv, por exemplo), ou pode criá-los você mesmo (usando web scraping, por exemplo).

Abaixo, seguem algumas sugestões de onde encontrar dados:

- Base dos dados
- Kaggle
- openDataSUS
- Redes sociais (Letterboxd, Reddit, etc)

Extração

Usando Python, extraia os dados que você escolheu. Por exemplo, se a sua fonte disponibiliza os dados em arquivos .csv, use Python para baixar esses arquivos.

O seu código deve extrair os dados. Não esperamos, por exemplo, que você baixe ou crie os arquivos manualmente e use Python apenas para lê-los e carregá-los.

Fique a vontade para usar bibliotecas Python fora da standard library (como requests), desde que essas bibliotecas sejam instaláveis usando pip. Isso vale também para o próximo passo. Você pode assumir que instalaremos essas bibliotecas antes de rodar seu código

- Você é *fortemente encorajado* a usar *alguma* ferramenta de gerenciamento de dependências para Python. Qualquer ferramenta ajuda aqui, desde simples arquivos requirements.txt e .python-version, até ferramentas como uv ou Poetry.

2. Transformação e carregamento dos dados num banco SQL

Transformação

Realize qualquer transformação necessária nesses dados. Você pode fazer isso em Python, antes de carregar os dados (ou seja, um processo ETL), ou em SQL, depois de carregar os dados (ou seja, um processo ELT).

Por exemplo, se a sua fonte disponibiliza os dados em diversos arquivos .csv, você usar Python para ler esses arquivos e transformá-los em um só dataframe, renomear e normalizar colunas, e criar uma coluna de chave primária a partir de um índice numérico.

Carregamento

Usando Python, carregue esses dados num banco SQL.

- Você pode usar qualquer banco SQL aqui, como Postgres ou SQLite, desde que este seja open source.
  - Se você usar Postgres ou qualquer outro banco que rode a partir de um servidor local, você pode assumir que nós criaremos um servidor localhost antes de rodar seu código (só não se esqueça de especificar o nome da db, porta, e credenciais que você usou, mesmo que estas sejam as defaults).
    - Você pode, mas não necessariamente precisa, rodar o banco (ou o seu projeto inteiro) a partir de um container Docker.
  - Se você usar SQLite, seu código precisa criar o arquivo .db.

3. Consultas SQL

Escreva duas consultas ao banco de dados que você carregou. O objetivo geral é gerar dados que você considere interessantes, seja fazendo recortes, filtragem, ou agrupamento nos dados.

Por exemplo, suponha que os seus dados mostram registros de vacinação contra COVID em Pernambuco, onde cada linha corresponde a uma aplicação de uma dose da vacina.

- Você pode filtrar os dados para mostrar apenas vacinas aplicadas em Recife entre 2021 e 2022.
- Você pode agrupar os dados por cidadão, de maneira que cada linha agora corresponda a um único cidadão, com uma nova coluna informando quais vacinas e doses o cidadão tomou.

Note que você não necessariamente vai fazer uma análise em cima desses dados! Por exemplo, você não precisa nos dizer qual foi a vacina mais aplicada em 2021, ou quantos cidadãos tiveram cobertura vacinal completa em 2022. Seu objetivo aqui é apenas gerar os dados para que estes possam ser utilizados pelos analistas.

4. Relatório

Escreva um *breve* relatório nos contando o processo de elaboração desse desafio. Seu relatório precisa conter, no mínimo, as seguintes informações:

- **Dados:**
  - Uma breve descrição dos dados que você escolheu.
  - Link(s) para a(s) fonte(s) de dados
  - Porque você escolheu esses dados: o que você acha interessante sobre eles?
- **Extração e Transformação:**
  - Um apanhado geral sobre o seu projeto: como você extraiu esses dados? Qual banco de dados você utilizou?
- **Consultas:**
  - Uma breve descrição do que fazem cada uma de suas consultas
  - Porque você escolheu essas consultas: o que você acha que pode ser interessante sobre esses recortes?

Fique a vontade, também, para trazer outras informações ou reflexões acerca do projeto. Por exemplo, houve alguma parte que foi particularmente difícil? Como você lidou com esse obstáculo? Há algo que você faria diferente?

Avaliação

Seu projeto vai ser avaliado segundo os seguintes critérios:

- O projeto obedece as instruções descritas nesse documento;
- O código do seu projeto roda sem erros e conforme descrito;
- O projeto demonstra que você entende os dados e as ferramentas que você utilizou, seja por meio de comentários/documentação no seu código ou pelo relatório.

Encorajamos você a preferir qualidade ao invés de quantidade ou complexidade neste desafio! Nós preferimos um projeto de menor escopo, mas que demonstra claramente que você sabe o que está fazendo, do que um projeto gigante copiado e colado.

Prazo e instruções de envio

Prazo

O prazo final para o desafio é **sexta, 31 de janeiro, às 23:59.**

- Você *não* vai ser penalizado por mandar o projeto próximo ao prazo final.
- No entanto, projetos enviados após esse prazo não serão aceitos, mesmo diante de imprevistos.

Envio

Você pode mandar seus materiais zipados por email para a gente, ou subir os dados no repositório no GitHub ou numa pasta no Google Drive.

- O repositório no GitHub pode ser público ou privado. Se privado, por favor adicione a conta do NGI ao repositório (ngi.secoge.sesau@gmail.com).
- Se você optar por GitHub ou Google Drive e compartilhar o link antes do prazo final, você pode fazer qualquer alteração no projeto antes do prazo final. No entanto, alterações feitas depois do prazo final não serão consideradas.

Dúvidas e contato

Você pode falar conosco através do nosso email: ngi.secoge.sesau@gmail.com. Fique à vontade para fazer qualquer pergunta ou tirar qualquer dúvida a respeito do desafio. Boa sorte!
Atenciosamente,

Núcleo de Gestão da Informação | SECOGE

Em ter., 21 de jan. de 2025 às 12:14, Alex M B de Souza <alexkrypto.ti@gmail.com> escreveu:

> Segue meu currículo em anexo para a vaga de de engenheiro de dados Júnior. Possuo as competências necessárias requisitadas na vaga para de desempenhar meu papel profissional. Para mais informações, estou aberto para contato. Desde já agradeço!