

Processo de Seleção:

Desafio Prático

A **First Decision** agradece sinceramente sua participação. Este desafio foi cuidadosamente elaborado para avaliar suas competências em programação Python para uma posição de nível sênior, o que requer uma base sólida de conhecimento técnico e boas práticas de desenvolvimento.

Instruções Gerais:

- **Confidencialidade:** O conteúdo e questões deste documento serão tratados de forma sigilosa pelo candidato, ficando expressamente proibido divulgação das questões ou detalhes sobre o processo.
- **Duração:** O tempo máximo para a realização dos exercícios 1 e 2 deste desafio é de **1 hora**, a contar a partir da leitura deste documento. O prazo de término não será postergado, busque ser produtivo e realizar o máximo possível. Questões 3, 4 e 5 poderão ser realizadas de forma assíncronas com prazo de entrega do código em até no máximo 24 horas.
- **Aproveite ao máximo:** Desenvolva com base em seus conhecimentos e experiências. Não se preocupe em resolver tudo, o mais importante é demonstrar seu raciocínio, capacidade técnica e boa organização do código.
- **Consultas:** Buscas pontuais na internet em sites confiáveis e bibliografias reconhecidas são permitidas. Contudo, não é permitido o uso de IAs generativas (como OpenAI, Gemini, Llama, etc).
- **Comunicação:** Somente com a equipe de seleção. **Não** é permitido o uso de celular, Teams e/ou comunicação com outras pessoas.
- **Fique tranquilo:** Não se sinta pressionado a saber tudo. O objetivo é avaliar o seu raciocínio e a forma como enfrenta desafios reais do cotidiano.
- **Boas práticas:** Utilize padrões de código, boas práticas de programação e organização de código. Isso será um diferencial importante.
- **Ferramentas:** Utilize a IDE/framework de sua preferência (VSCode, PyCharm, Jupyter Notebook, Databricks, Google Colab, etc.). Escolha o ambiente com o qual você se sente mais confortável.
- **Versionamento de código:** Todo o código deve ser versionado. Utilize o controlador de versão de sua escolha, como Git, GitHub, GitLab, Azure Repo, etc.
- **Entrega:** Ao concluir o desafio, disponibilize o código-fonte conforme as orientações fornecidas.

Critérios de avaliação:

- Estrutura e organização do Código
- Capacidade técnica e fluência com Python
- Tratamento de exceção
- Conhecimento de Banco de Dados
- Consumo e construção de API RESTful
- Desenvolvimento Web
- Qualidade do Código, boas práticas, estilo e padrões

Vamos lá, mãos à obra!

Crie um projeto e/ou notebook, chamado **FD_Selecao_SEUNOME**. Monte seu ambiente com variáveis e bibliotecas necessárias para construção do projeto.

1 Realize leitura do arquivo CSV disponível em: <https://raw.githubusercontent.com/datasets/finance-vix/main/data/vix-daily.csv> e atribua a um dataframe.

- 1.1 Liste as 6 primeiras linhas.
- 1.2 Liste informações básicas sobre o dataset.
- 1.3 Liste as estatísticas básicas do dataframe.
- 1.4 Remova linhas com valores ausentes.
- 1.5 Converta o tipo da data para datetime.
- 1.6 Crie uma coluna 'Volatility Range' que é derivada das colunas 'HIGH' menos 'LOW'.
- 1.7 Crie um novo dataframe contendo a média da coluna 'CLOSE' agrupada e ordenada por Ano_Mes
- 1.8 Crie um gráfico (plot) visual demonstrando os dados.
- 1.9 Exporte para formato .parquet incluindo no nome do arquivo a data e horário do dia de hoje. Formato prefixo_diamesano_horaminuto Exemplo: finance_251224_1235.parquet

2 Consuma a API disponível <https://makeup-api.herokuapp.com/api/v1/products.json>, armazene os dados em um DataFrame e faça as seguintes operações:

- 2.1 Filtre apenas os produtos com a categoria lipstick.
- 2.2 Liste as categorias (category) exclusivas presentes no DataFrame.
- 2.3 Crie uma função chamada buscar_produtos, passando como parâmetro qual a categoria e retorne a lista de produtos daquela categoria. Utilize a função para categoria pencil.
- 2.4 Utilize expressão regular Regex para trabalhar o formato do campo "created_at" exemplo de dado: "2018-07-08T23:45:08.056Z" e transformar para o formato dd/mm/yyyy hh:mm ou utilize um outro exemplo com expressão regular.
- 2.5 Exporte o DataFrame filtrado para o formato .json.
- 2.6 Crie um gráfico de barras mostrando as 10 marcas ('brand') com mais produtos do tipo lipstick

3 Escolha/utilize um banco de dados de sua preferência, local ou nuvem (SQL Server, MySQL, PostgreSQL, Oracle, MongoDB, etc), crie uma tabela tb_produto.

- 3.1 Armazene os dados consumidos pela API e tratados no dataframe nela.
- 3.2 Consuma a informação do banco de dados e inclua em um dataframe.

4 Crie uma API com a biblioteca de sua preferência apenas com método get, para consumir a informação da tabela tb_produto.

5 Crie uma aplicação web simples com biblioteca de sua preferência, mostrando os dados da API na interface Web.

- Lembre-se de publicar seu código em um controlador de versão privado de sua preferência (exemplo git, github, gitlab, etc) e envie o link, juntamente com o código fonte do seu projeto em formato .zip contendo todos os fontes do projeto para avaliação.