

# Exercício para Cargo de Engenheiro de Software (versão Python)

---

Health Analytics, Funcional Health Tech

## Introdução

---

Neste exercício, pedimos para você implementar uma aplicação muito simples, de modo que possamos avaliar os seguintes pontos:

- Estilo de programação de modo geral, incluindo organização do código e comentários;
- Noções de Programação Orientada a Objetos;
- Reuso de software;
- Práticas de qualidade de software (e.g., testes);
- Práticas de automação;
- Entendimento e elaboração de soluções para requisitos de usuário.

Este exercício deve ser feito na linguagem Python (versão 3 em diante). Estamos supondo que isso foi acordado com você e está de acordo com suas habilidades atuais, caso contrário por favor nos avise. Faz parte do exercício aprender a usar bibliotecas novas para sua realização, como o Dash (<https://plot.ly/dash/>) que iremos indicar mais adiante. Se você se sentir confortável, por favor escreva sua solução inteiramente em inglês (caso não se sinta confortável, é aceitável escrever em português).

Entendemos que talvez você não tenha tempo de concluir todos os pontos pedidos neste exercício. Assim, por favor lembre-se dos nossos objetivos acima, para que aquilo que você concluir possa demonstrar representativamente suas habilidades nesses aspectos.

## Enunciado do Exercício

---

Considere o seguinte problema. A área da empresa responsável por fármacos necessita fazer previsões sobre consumo de medicamentos. Para isso, um Cientista de Dados criou um modelo de previsão de séries temporais de consumo, porém não definiu como o usuário final irá interagir com esse modelo. Ademais, outras áreas de negócio também decidiram influenciar na previsão de consumo, acrescentando modificações às projeções feitas pelo modelo do Cientista de Dados. Ficou a cargo do **Engenheiro de Software**, ou seja, **você**, construir um modo de permitir que o resultado do modelo de consumo e as alterações das áreas de negócio sejam combinadas e que o usuário final possa interagir com essas informações todas. Definiu-se que essa solução de interação com o usuário terá as características colocadas a seguir.

## Fonte de Dados

A solução tomará como entrada alguns arquivos de dados, um deles gerado pelo próprio Cientista de Dados, e outros gerados pelas área de negócio auxiliares, com as seguintes características. O arquivo gerado pelo Cientista de Dados contém os valores mensais da série histórica e da projeção decorrente (em milhões de Reais), que vamos chamar de "cenário base". A área de negócio, por sua vez, envia arquivos com modificações da série criada pelo Cientista de Dados. Essas modificações representam um "cenário otimista" e um "cenário pessimista" (também em milhões de Reais). *A projeção final sempre será ou o cenário base sozinho, ou o cenário base somado a um dos outros cenários, que o modificam aditivamente:*

```
final = base

# ou
final = base + otimista

# ou
final = base + pessimista
```

O cenário base sempre vem em um arquivo CSV gerado automaticamente por um programa. Porém, os arquivos da área de negócio podem vir tanto em formato CSV quanto Excel, pois depende do analista que fizer (manualmente) e ainda podem conter erros ou dados faltantes. Neste exercício, temos dois conjuntos de dados, refletindo dois produtos diferentes, que estão nos seguintes diretórios:

- `data/product_1/`
- `data/product_2/`

Para cada um desses produtos, temos os seguintes arquivos:

- `config.json`: Arquivo JSON com o nome e a marca do produto em questão.
- `base_scenario.csv`: O arquivo gerado pelo Cientista de Dados, que contém os valores mensais da série histórica e da projeção decorrente. Este arquivo é um CSV.
- `best_case_scenario.csv` ou `best_case_scenario.xlsx`: Um arquivo enviado pela área de negócios refletindo um cenário otimista, que contém modificações aditivas ao cenário base. Pode estar em CSV ou Excel, e pode conter erros e problemas variados.
- `worst_case_scenario.csv` ou `worst_case_scenario.xlsx`: Um arquivo enviado pela área de negócios refletindo um cenário pessimista, que contém modificações aditivas ao cenário base. Pode estar em CSV ou Excel, e pode conter erros e problemas variados.

A solução proposta deve ser robusta o suficiente para lidar com dados problemáticos, como por exemplo dados que faltem para datas específicas. Nesses casos, você deve propor a solução que julgar mais adequada (e.g., alertar o usuário, abortar a operação, etc.).

Sugerimos empregar a biblioteca Pandas (<https://pandas.pydata.org/>) para leitura e manipulação desses dados tabulares.

## Visualização e Interação

Deve haver uma interface de usuário gráfica (GUI) para a visualização e interação com os dados fornecidos, voltada aos usuários de negócio. Por favor use a biblioteca Dash (<https://plot.ly/dash/>) para esse fim se possível (i.e., se você tiver o tempo de aprender a usá-la, que deve ser algo rápido).

A área de negócio estabeleceu os seguintes requisitos para essa aplicação:

- O usuário deve poder escolher qual produto deseja analisar, dentre os disponíveis (neste exercício, são dois produtos com dados disponíveis).
- Você pode supor que os dados já estão disponíveis em um diretório junto aos arquivos executáveis da aplicação. Como os dados são colocados nesse local não está no escopo deste problema.
- O usuário deve poder escolher qual cenário quer ver: base, otimista, pessimista. Para o cenário escolhido, os demais pontos devem se aplicar.

- Deve haver a opção de eliminar os outliers da série temporal. Um outlier é definido como sendo um valor que difira da média dos demais valores por uma certa quantidade de desvios padrões (a quantidade pode ser escolhida pelo usuário).
- Devem ser mostradas as seguintes métricas:
  - Crescimento percentual projetado 12 meses no futuro (e.g., se estamos em janeiro/2019 e temos o valor 100, e em janeiro/2020 projeta-se 125, deve-se mostrar o valor de +25%).
  - Média dos crescimentos percentuais anuais.
- Finalmente, deve ser possível exportar a série sendo visualizada (com eventuais modificações aplicadas pelo usuário) para um arquivo Excel.

## Entregáveis

---

Pedimos que a solução para o problema proposto seja entregue com as seguintes características:

- Deve conter um breve arquivo de texto explicando como executar a aplicação. *Você não precisa documentar seu programa externamente; a única documentação necessária é a que faz parte dos comentários do próprio programa (e.g., docstrings).*
- O código fonte deve estar organizado seguindo boas práticas de localização de pastas e arquivos.
- É esperado que seu programa, uma vez em execução, seja auto-explicativo e operável por um usuário sensato. Ou seja, esperamos uma interface gráfica de fácil utilização.
- Você pode supor que o usuário precisa digitar um comando na linha de comando para iniciar a sua aplicação. No futuro, claro, isso poderia ser transformado em uma aplicação clicável, ou em um link online, mas para este exercício você pode ignorar essas complexidades adicionais.
- **Bonus (opcional):** Adicionalmente, se você puder fazer a aplicação rodar via Docker (<https://www.docker.com/>), será ainda mais apreciado por nós. Nesse caso, você pode supor que a máquina do usuário está equipada com Docker e você só precisa explicar como executar o container específico da sua aplicação.
- **Bonus (opcional):** caso você se sinta confortável, preferimos que sua solução seja escrita completamente em inglês (tanto o programa, sua interface gráfica e seus comentários, quanto qualquer documentação acessória).

## Avaliação

---

Após a entrega da solução, marcaremos um telefonema para que você possa explicar alguns pontos da sua solução. O mais importante é a qualidade do que for feito, não a quantidade. Naturalmente, porém, quanto mais requisitos você for capaz de implementar, melhor.