# Candidato: Gustavo da Silva Fortunato

# Respostas do exercício

# Ajuste nos dados feito no excell

Havia duas partes nos dados que não seguiam algum padrão visível e estas foram alteradas manualmente usando excell.





Essa alteração gerou a planilha amostras\_raw.xlsx



Nessa planilha todas as amostras tornaram-se os índices e as colunas são as variáveis do laudo. Algumas variáveis foram removidas, pois não faziam sentido no objetivo do exercício. Entre as variáveis removidas estão: ‘Lims Codigo’, ‘id amostra lab’ e as profundidades.

As datas foram alteradas para o padrão correto utilizando o formatador de valor do próprio excell.

# Primeiro ajuste usando Power BI

No Power Query houve a transformação do amostras\_raw em amostras\_transformadas. Os principais eventos foram a seleção das colunas apartir da variável alumínio e conversão dessas colunas em linhas. Com isso houve a repetição dos índices de amostras, datas e pontos de acordo com a variável da coluna. A coluna contendo os analitos agora se chama Atributo.



# Convertendo as unidades µg/L em mg/L

Para que os dados estejam comparáveis com os valores de referência, houve a conversão das unidades para o padrão mg/L. Algumas unidades não foram convertidas, pois não era necessário, como as unidades de pH e condutividade elétrica.

import pandas as pd

import re

import numpy as np

df = pd.read\_excel('amostras\_transformadas.xlsx')

def conversor(valor):

    numeros = re.search(r'[\d.,]+', valor).group().replace(',', '.')  # Extraindo somente os números

    if "µg/L" in valor:

        valor\_convertido = float(numeros) / 1000  # Convertendo as unidades µg/L para mg/L

        valor\_convertido\_str = str(valor\_convertido).replace('.', ',')  # Gerando o padrão numerico do código

        return valor\_convertido\_str + ' mg/L'  # Adicionando a unidade mg/L

    return valor  # retornando o valor original, se o valor não foi convertido de µg/L para mg/L

# convertendo todos os valores para strings

df["Valor"] = df["Valor"].astype(str)

# aplicando a função na coluna Valor

df["Valor"] = df["Valor"].apply(conversor)

# realizando o split baseado no espaço entre valor e unidade

df[['Valor', 'Unidades']] = df['Valor'].str.split(' ', expand=True)

df.to\_excel('Dados\_.xlsx', index=False)

# Gerando a planilha Dados\_final

Utilizando os valores referência, houve então a formação das colunas Resultado e Limite de quantificação. Acreditando que a coluna Resultado seja uma comparação com os dados de quantificação da referência. Considerei que a referência resultaria em conformidade, ou inconformidade, de acordo com a legislação da ANVISA.

refs = pd.read\_excel('20230701\_valores\_referencia.xlsx')

#removendo as unidades

refs['codigo\_parametro'] = refs['codigo\_parametro'].str.split('\(').str[0].str.strip()

#convertendo para o padrão numeríco do python

df["Valor"] = df["Valor"].str.replace("<", "").str.replace(",", ".").astype(float)

# Mesclando os DataFrames com base no atributo

df\_refs = pd.merge(df, refs, left\_on="Atributo", right\_on="codigo\_parametro", how="left")

# Função para comparar os valores das análises e quantificação dos elementos químicos

def comparar\_elementos(row):

    if pd.isnull(row["valor\_maximo"]):

        return 'Sem referência'

    elif row["Atributo"] == 'pH':

        if 6 <= row["Valor"] <= 9:

            return "Conforme"

        else:

            return "Inconforme"

    elif row["Atributo"] == 'Saturação de OD':

        if 0 <= row["Valor"] <= 4:

            return "Conforme"

        else:

            return "Inconforme"

    elif row["Valor"] < row["valor\_maximo"]:

        return "Conforme" #considerando que são análises para fiscalização de água, segundo a legislação da ANVISA

    elif row["Valor"] > row["valor\_maximo"]:

        return "Inconforme"

    else:

        return "Inconforme"

# Aplicando a função

df\_refs["Resultado"] = df\_refs.apply(comparar\_elementos, axis=1)

#removendo os códigos do dataframe

df\_refs.drop(['codigo\_parametro'],axis=1, inplace=True)

#criando a coluna limite de quantificação

df\_refs["Limite de quantificação"] = np.where(

    df\_refs["valor\_minimo"].notnull() & df\_refs["valor\_maximo"].notnull(),

    df\_refs["valor\_minimo"].astype(str) + " a " + df\_refs["valor\_maximo"].astype(str),

    np.where(

        df\_refs["valor\_minimo"].notnull(),

        df\_refs["valor\_minimo"].astype(str),

        df\_refs["valor\_maximo"].astype(str),

    ),

)

#Limpando dados não solicitados pelo exercicio

df\_refs.drop(['valor\_minimo','valor\_maximo'],axis=1, inplace=True)

#convertendo para o padrão brasileiro de decimais

df\_refs.replace('.',',', inplace=True)

#substituindo os nan por Não quantificado

df\_refs.replace('nan','Não quantificado', inplace=True)

#salvando o dataframe como excel

df\_refs.to\_excel('Dados\_Final.xlsx', index=False)

# Resposta para o exercício 1 Extra

Exercício 1 - Extra:

A tabela produzida pode ser interpretada como um modelo de dados OBT (*one big table*). Esta é uma modelagem válida para o relatório proposto? Como o modelo poderia ser otimizado?

Sim, pois os OBT permitem visualização de uma estrutura plana de dados e são mais simples. Para realizar uma consulta em análises que exigem combinações o modelo OBT é viável. Um exemplo seria quantidade de Alumínio em todas as amostras, bastaria ordenar a coluna Atributo de A até Z e todos os valores de Alumínio para cada amostra estariam no início da planilha.

Um dos principais problemas desse tipo de modelo é a redundância das amostras por exemplo. Vários data points para todas as amostras aumentaram o número de linhas na tabela. As 3 primeiras colunas são repetidas a cada 43 índices, pois são o total de análises realizadas nas amostras de água. Um pré-processamento para Machine Learning seria complicado também, pois a normalização dos dados seria difícil de ser feita, sem comprometer a integridade dos dados. Para este caso em específico a escalabilidade seria um problema também, pois para cada nova amostra 43 novos índices seriam feitos e se houver uma outra análise, como coliformes termotolerantes totais, por exemplo, mais 11 índices (quantidade total de amostras) seriam necessários.

Para otimizar um OBT seria necessário normalizar todas as colunas que possuem mesma unidade de medida e o Ph, por exemplo. Uma mudança de índices de coluna também poderia ser feito, mas isso teria que ficar claro de acordo com o objetivo de quem for realizar as consultas na OBT. Por estarem codificadas, seria interessante particionar os OBT para cada região, ou até mesmo para cada data, se forem análises em série temporal por exemplo. Claramente que otimização é um processo que sempre vai ser feito, pois as consultas podem mudar em função da legislação ou até mesmo na forma dos clientes solicitarem serviços da empresa.

# Exercício 2

Para formar o dashboard houve a obtenção dos dados, seleção das colunas de interesse para plotagem da série temporal, gráfico de pizza e tabela com as métricas mínimo, máximo, média e mediana. A mediana foi realizada por meio da aplicação da função MEDIANX no próprio console do PowerQuery usando DAX, conforme a fórmula abaixo:

MEDIANX(

VALUES('Dados\_final'[Nome da amostra]),

CALCULATE(

SUM('Dados\_final'[Valor]),

ALL('Dados\_final'[Atributo])

)

)

Utilizando a medida rápida para máximo, mínimo e média não possibilitou um correto resultado. Para que essas medidas tivesse um resultado correto as formulas utilizadas foram as seguintes:

Máximo =

MAXX(

    VALUES('Dados\_final'[Nome da amostra]),

    CALCULATE(

        SUM('Dados\_final'[Valor]),

        ALL('Dados\_final'[Atributo])

    )

)

Média =

AVERAGEX(

     VALUES('Dados\_final'[Nome da amostra]),

    CALCULATE(

        SUM('Dados\_final'[Valor]),

        ALL('Dados\_final'[Atributo])

    )

)

Mínimo =

MINX(

    VALUES('Dados\_final'[Nome da amostra]),

    CALCULATE(

        SUM('Dados\_final'[Valor]),

        ALL('Dados\_final'[Atributo])

    )

)

O gráfico de pizza possui as definições de Conforme, para valores dentro dos limites mínimos (quando aplicado) e máximos, atendendo assim os valores de referência. Quando fora do padrão eles recebiam Inconforme e quando não possuíam um valor de referência eles recebiam Sem referência. Assim é possível visualizar somente quais amostras que possuem dados Inconformes, Conformes e quais precisam de alguma referência.

Exercício 2 – Extra:

Utilizando a planilha (B), apresente no relatório uma análise de conformidade do monitoramento com os valores orientadores da legislação, destacando os **parâmetros e pontos onde o resultado excedeu o valor orientador**. Utilize os visuais que julgar adequados.

Considerando que a legislação a ser comparada com os dados seja: <http://antigo.anvisa.gov.br/documents/10181/2718376/RDC_717_2022_.pdf/46974199-1976-43d8-8a0d-565152cbeada>  
  
Houve uma alteração na planilha Dados\_final, conforme o script abaixo demonstra.

df = pd.read\_excel('amostras\_transformadas.xlsx')

df["Valor"] = df["Valor"].astype(str)

# aplicando a função na coluna Valor

df["Valor"] = df["Valor"].apply(conversor)

# realizando o split baseado no espaço entre valor e unidade

df[['Valor', 'Unidades']] = df['Valor'].str.split(' ', expand=True)

refs = pd.read\_excel("ANVISA\_valores\_referencia - Copia.xlsx")

refs['codigo\_parametro'] = refs['codigo\_parametro'].str.split('\(').str[0].str.strip()

#convertendo para o padrão numeríco do python

df["Valor"] = df["Valor"].str.replace("<", "").str.replace(",", ".").astype(float)

# Mesclando os DataFrames com base no atributo

df\_refs = pd.merge(df, refs, left\_on="Atributo", right\_on="codigo\_parametro", how="left")

df\_refs["valor\_maximo"] = pd.to\_numeric(df\_refs["valor\_maximo"], errors="coerce")

# Convert "Valor" column to numeric type

df\_refs["Valor"] = pd.to\_numeric(df\_refs["Valor"], errors="coerce")

# Aplicando a função

df\_refs["Resultado"] = df\_refs.apply(comparar\_elementos, axis=1)

#removendo os códigos do dataframe

df\_refs.drop(['codigo\_parametro'],axis=1, inplace=True)

#criando a coluna limite de quantificação

df\_refs["Limite de quantificação"] = np.where(

    df\_refs["valor\_minimo"].notnull() & df\_refs["valor\_maximo"].notnull(),

    df\_refs["valor\_minimo"].astype(str) + " a " + df\_refs["valor\_maximo"].astype(str),

    np.where(

        df\_refs["valor\_minimo"].notnull(),

        df\_refs["valor\_minimo"].astype(str),

        df\_refs["valor\_maximo"].astype(str),

    ),

)

#Limpando dados não solicitados pelo exercicio

df\_refs.drop(['valor\_minimo','valor\_maximo'],axis=1, inplace=True)

#convertendo para o padrão brasileiro de decimais

df\_refs.replace('.',',', inplace=True)

#substituindo os nan por Não quantificado

df\_refs.replace('nan','Não quantificado', inplace=True)

#salvando o dataframe como excel

df\_refs.to\_excel('Dados\_Final\_anvisa.xlsx', index=False)

O script refez todo o processo citado anteriormente, mas agora numa versão atualizada dos valores de referência, de acordo com a ANVISA. Entre os novos valores de referência adicionados e presentes no laudo estão:

|  |
| --- |
| Antimônio |
| Arsênio |
| Bromato |
| Cloro livre |
| Nitrato |
| Nitrito |

Existem outros elementos e moléculas que não estão no Laudo, mas estão na ANVISA, estes não foram quantificados no laudo.

|  |
| --- |
| 2,4 D |
| Acrilamida |
| Alaclor |
| Atrazina |
| Bentazona |
| Benzeno |
| Clordano (isômeros) |
| Cloreto de Vinila |
| Clorito |
| DDT (isômeros) |
| Diclorometano |
| Endossulfan |
| Endrin |
| Estireno |
| Glifosato |
| Heptacloro e Heptacloro epóxido |
| Metoxicloro |
| Microcistinas |
| Molinato |
| Monocloramina |
| Pendimetalina |
| Pentaclorofenol |
| Permetrina |
| Propanil |
| Simazina |
| Tetracloreto de Carbono |
| 1,1 Dicloroeteno |
| 1,2 Dicloroetano |
| Aldrin e Dieldrin |
| Benzopireno |
| Hexaclorobenzeno |
| Metolacloro |
| Lindano (gama-BHC) |
| Tetracloroeteno |
| Triclorobenzenos |
| Tricloroeteno |
| Trifluralina |

Além disso o laudo emitido possui dados, que não estão na ANVISA e nem nos valores de referência, que são:

|  |
| --- |
| Aluminio |
| Calcio |
| Carbonato (como CO3) |
| CO2 total |
| Condutividade Eletrica |
| Magnesio |
| Nitrogenio Amoniacal Total |
| Oxigenio Dissolvido |
| Potassio |
| Sodio |
| Sulfeto como S2- |

Os novos valores de referência foram utilizados para construir um novo dashboard na página 2 do mesmo arquivo (Dashboard exercício.pbix) com o dashboard para o exercício 2. Este novo dashboard possui como fonte de dados a planilha dados\_final\_anvisa.xlsx. Suas fórmulas de métricas foram alteradas para atenderem a nova fonte de dados.