



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO



**Centro de
Informática**
UFPE

**Universidade Federal de Pernambuco
Centro de Informática - CIn**

Contabilidade e Gerenciamento de Custos

Humberto Hugo da Silva Campos

Recife, 03 de Maio de 2023

Resumo: O projeto tem como objetivo alinhar conhecimentos ministrados durante o semestre para desenvolver um algoritmo capaz de filtrar as informações (base de dados fornecida pelo docente) e com base em linguagem de programação de alto nível escolhida pela equipe (Python) seja possível extrair alguns os dados requeridos pelo usuário, é importante condicionar a busca e seleção de dados de acordo com as áreas de segurança pública, assistência social, previdência social, saúde e educação de cada município por estado brasileiro, e com base nisso descrever despesas empenhadas, liquidadas, pagas, restos a pagar processados e não processados.

Detalhamento da implementação do código:

```
import pandas as pd
import openpyxl
```

Importação das bibliotecas necessárias para habilitar o acesso ao arquivo da planilha localizado no explorador de arquivos do windows, bem como utilizar o pandas para auxiliar no código.

```
def filtro(estado, conta, coluna):
    filtro = (tabela['UF'] == estado) & (tabela['Conta'] == conta
) & (tabela['Coluna'] == coluna)
    return filtro
```

Função responsável por acessar a lista a qual contém a unidade de federação digitada pelo usuário, as contas que vai ser percorrida e a coluna que vai percorrer.

```

tabela = pd.read_excel(r"C:/Users/Pauli\Downloads/municipios.xlsx"
, engine='openpyxl')

contas = ['06 - Segurança Pública', '08 - Assistência Social',
'09 - Previdência Social', '10 - Saúde', '12 - Educação']

estados = ['AC', 'AL', 'AP', 'AM', 'BA', 'CE', 'DF', 'ES', 'GO',
'MA', 'MT', 'MS', 'MG', 'PA', 'PB', 'PR', 'PE', 'PI', 'RJ', 'RN',
'RS', 'RO', 'RR', 'SC', 'SP', 'SE', 'TO']

despesas = ['Despesas Empenhadas', 'Despesas Liquidadas',
'Despesas Pagas', 'Inscrição de Restos a Pagar Não Processados']

```

Tem o arquivo “Tabela” para ser aberto e as contas conforme foi pedida que serão acessadas, juntamente com os estados que contêm na planilha e as despesas exigidas.

```

if __name__ == '__main__':
    print("MENU:")
    print("1. ESTADO")
    print("2. MUNICIPIO")
    print("3. TABELA COMPLETA")
    print("4. Sair")

    while True:
        escolha = input("Escolha uma opção: ")

```

É apresentado um menu de opções para o usuário escolher qual opção ele deseja filtrar as despesas da tabela que podem ser por : Estados, Municípios ou a Tabela completa.

```

if escolha == "1":
    print("Opção 1 selecionada")

    conta = input('Qual conta você quer filtrar: ')
    despesa = input('Qual despesa você quer filtrar: ')

    for i in range(len(estados)):
        valor = filtro(estados[i], conta, despesa)
        linha = tabela.loc[valor]
        print(f'Valores de {despesa} em {conta}
no estado de {estados[i]}')
        print(linha)
        soma += linha['Valor'].sum()
        print(f'A soma total dessa despesa é {round(soma,
2)})')

    soma = 0

```

Na opção 1 que é por estados, ele vai pedir qual conta vai ser filtrada e qual despesa você deseja ver e vai mostrar como saída do código todos os estados e suas respectivas despesas relacionada a conta escolhida.

```

elif escolha == "2":
    print("Opção 2 selecionada")
    conta = input('Qual Conta: ')
    despesa = input('Qual Despesa: ')
    municipio = input('Qual Municipio: ')

    valor = filtro2(municipio, conta, despesa)
    linha = tabela.loc[valor]
    custo = linha['Valor']
    print(f'Valores de {despesa} em {conta} na {municipio}
')

    print(linha['Valor'])

```

A segunda opção é filtrar por municípios que é um filtro mais aprofundado, onde vai mostrar a conta que você deseja saber, referente a qual despesa e de qual município. Fazendo assim, uma busca mais específica.

```
elif escolha == "3":
    print("Opção 3 selecionada")

    conta = input('Qual conta: ')
    despesa = input('Qual despesa: ')
    estado = input('Qual estado: ')

    valor = filtro(estado, conta, despesa)
    linha = tabela.loc[valor]
    print(f'Valores de {despesa} em {conta} no estado de {estado}')
    print(linha)
    soma += linha['Valor'].sum()
    print(f'A soma total dessa despesa é {round(soma, 2)}')
    soma = 0
```

Na opção 3 que é por estado, ele vai pedir qual conta vai ser filtrada e qual despesa você deseja ver e vai mostrar como saída do código do estado e suas respectivas despesas relacionada a conta escolhida.

```
elif escolha == "4":
    print("Saindo...")
    break
else:
    print("Escolha inválida. Tente novamente.")
```

Essa opção sai das opções do filtro. Caso seja digitado algo diferente das opções irá aparecer uma mensagem de erro.