

Programa IT Academy – Processo Seletivo

Problema

Solução de Problema: Internações em Porto Alegre

Instruções para executar o código

O código foi escrito na linguagem Python na versão `python3`, para ler e tratar o arquivo CSV, utilizei a biblioteca `Pandas` através do `Anaconda`.

A execução do programa será via terminal, conforme opção contida nas instruções.

Iniciando o sistema

Este comando importa a biblioteca `pandas`

```
import pandas as pd
```

Com este laço o menu é exibido ao usuário.

```
# Opções do menu
op = 0
while op != 6:
    print(15 * "_")
    print("[1] Consultar média de idade dos pacientes")
    print("[2] Consultar internações por ano")
    print("[3] Consultar hospitais")
    print("[4] Calcular tempo de internação")
    print("[5] Determinar tempos de espera na fila")
    print("[6] Sair do programa")
    print(15 * "_")

# Entrada do usuário
op = int(input("Escolha uma opção: "))
```

Conforme saída a baixo

```
(base) maroberto@maroberto-ThinkPad-T430 ~/Python_projects/Python/PucRs_Dell Master ± python3
index.py
-----
[1] Consultar média de idade dos pacientes
[2] Consultar internações por ano
[3] Consultar hospitais
[4] Calcular tempo de internação
[5] Determinar tempos de espera na fila
[6] Sair do programa
-----
Escolha uma opção: █
```

Comando para lê o arquivo de dados CSV

```
# lê os dados contidos no CSV
dados = pd.read_csv("gerint_solicitacoes_mod.csv", encoding="utf-8", sep=";")
```

1. Consultar média de idade dos pacientes

Neste primeiro bloco é realizado a verificação de entrada, caso seja válida o sistema passa para o próximo passo.

```
if op == 1:
    # verifica se o município existe
    municipio_very = 0
    while municipio_very == 0:
        municipio = str(input("Digite o município residencial: "))
        cidade_df = dados[["municipio_residencia", "idade", "sexo"]][
            dados["municipio_residencia"] == municipio.upper()
        ]
        municipio_very = cidade_df.shape[0]
        if municipio_very == 0:
            print(
                bcolors.FAIL
                + f"\nNão encontramos {municipio.title()}, digite um
municipio valido!"
                + bcolors.CEND
```

Após a verificação, o sistema filtra e organiza os dados para serem exibidos ao usuário.

```
else:

    # filtra os dados pelo município, idade e sexo
    cidade_df = dados[["municipio_residencia", "idade", "sexo"]][
        dados["municipio_residencia"] == municipio.upper()
    ]

    # total dos pacientes
    total_df = cidade_df.shape[0]

    # separa pacientes por gênero
    sexo_f_df = cidade_df[cidade_df["sexo"] == "FEMININO"]
    sexo_m_df = cidade_df[cidade_df["sexo"] == "MASCULINO"]

    # média das idades por gênero e total
    mean_df_f = sexo_f_df["idade"].mean()
    mean_df_m = sexo_m_df["idade"].mean()
    mean_df_g = cidade_df["idade"].mean()

    # imprime o resultado por gênero

    print(
        bcolors.OKGREEN
        + "\nO número total de pacientes do município do {} é de:
{:.0f}".format(
            municipio.title(), total_df
        )
    )
    print(
        "\nA média de idade:\nmulheres {:.2f} anos.\nHomens {:.2f}
anos.".format(
            mean_df_f, mean_df_m
```

```

    )
    )
    print(
        "\nA média de idade de todos os pacientes é de {:.2f}
anos.".format(
            mean_df_g
        )
    + bcolors.CEND

```

Saída deste bloco:

```

-----
Escolha uma opção: 1
Digite o município residencial: porto alegre

O número total de pacientes do município do Porto Alegre é de: 20947

A média de idade:
mulheres 51 anos.
Homens 49 anos.

A média de idade de todos os pacientes é de 50 anos.

```

2. Consultar internações por ano

No início do segundo bloco refazemos as verificações para validar as entradas de usuário, logo depois convertemos as datas, filtramos os anos escolhidos e geramos a saída para o usuário.

```

if op == 2:
    # verifica se o município existe
    municipio_very = 0
    while municipio_very == 0:
        municipio = str(input("Digite o município residencial: "))
        cidade_df = dados[["municipio_residencia", "idade", "sexo"]][
dados["municipio_residencia"] == municipio.upper()
        ]
        municipio_very = cidade_df.shape[0]
        if municipio_very == 0:
            print(
                bcolors.FAIL
                + f"\nNão encontramos {municipio.title()}, digite um
municipio valido!"
                + bcolors.CEND
            )
        else:

            # converte datas para formato 'datetime64'
            dados["data_internacao"] =
dados["data_internacao"].astype("datetime64")
            # dados.dtypes
            cidade_df = dados[["municipio_residencia", "data_internacao"]][
                dados["municipio_residencia"] == municipio.upper()
            ]
            # filtra as internações por ano
            ano_2018 = cidade_df[["data_internacao"]][
                cidade_df["data_internacao"].dt.year == 2018
            ]
            ano_2019 = cidade_df[["data_internacao"]][
                cidade_df["data_internacao"].dt.year == 2019
            ]

```

```

ano_2020 = cidade_df[["data_internacao"]][
    cidade_df["data_internacao"].dt.year == 2020
]
ano_2021 = cidade_df[["data_internacao"]][
    cidade_df["data_internacao"].dt.year == 2021
]

# imprimir o resultado da consulta
print(
    bcolors.OKGREEN
    + f"\nInternações em {municipio.title()}: \n2018:
{ano_2018.shape[0]} \n2019: {ano_2019.shape[0]} \n2020: {ano_2020. shape[0]}
\n2021: {ano_2021. shape[0]}"
    + bcolors.CEND
)

```

Esta é a saída deste bloco, nesta imagem é possível visualizar o alerta de entrada inválida, após corrigir, a saída é exibida

```

-----
Escolha uma opção: 2
Digite o município residencial: santa marina

Não encontramos Santa Marina, digite um municipio valido!
Digite o município residencial: santa maria

Internações em Santa Maria:
2018: 63
2019: 27
2020: 112
2021: 32

```

3. Consultar hospitais

Neste bloco fizemos novamente a validação e em seguida preparamos a tabela, onde será exibido os dados filtrados

```

if op == 3:
    # verifica se o executante existe
    executante_very = 0
    while executante_very == 0:
        # entrada do usuário
        executante = str(
            input(
                "Digite o nome do executante"
                + bcolors.OKBLUE
                + "\nEx.: Hospital Sao Lucas Da PUCRS ou Sao Lucas: "
                + bcolors.CEND
            )
        )
        executante_df = dados[
            [
                "idade",
                "municipio_residencia",
                "solicitante",
                "data_autorizacao",
                "data_internacao",
                "data_alta",
                "executante",
            ]
        ][dados["executante"].str.contains(executante.upper())]
        executante_very = executante_df.shape[0]

```

```

if executante_very == 0:
    print(
        bcolors.FAIL
        + f"\nNão encontramos {executante.upper()}, digite um
executante valido!"
        + bcolors.CEND
    )
else:

    # converte datas para formato 'datetime64'
    dados["data_autorizacao"] = dados["data_autorizacao"].astype(
        "datetime64"
    )
    dados["data_internacao"] =
dados["data_internacao"].astype("datetime64")
    dados["data_alta"] = dados["data_alta"].astype("datetime64")

    # filtra idade, municipio de residencia, solicitante, data da
autorizacao, internação e alta, executante
    executante_df = dados[
        [
            "idade",
            "municipio_residencia",
            "solicitante",
            "data_autorizacao",
            "data_internacao",
            "data_alta",
            "executante",
        ]
    ] # [dados["executante"].str.contains(executante.upper)]

    # filtra os Hospitais por nomes ou parte dos nomes
    filter_executante_df = executante_df[
        executante_df["executante"].str.contains(executante.upper())
    ]

    # resultado da consulta
    print(
        bcolors.OKGREEN
        + f"\nNo total o {executante.title()} teve
{filter_executante_df.shape[0]} pacientes."
        + bcolors.CEND
    )

    print(filter_executante_df.head())

```

O resultado deste bloco é a tabela abaixo, onde podemos visualizar a busca usando a função **Contains**, que permite filtrar, por parte do nome. Para otimizar a execução, foram exibidos apenas 5 linhas, porem o total é de 2603 linhas. Neste exemplo, o executante é Hopsital Femina.

```

[1] Consultar média de idade dos pacientes
[2] Consultar internações por ano
[3] Consultar hospitais
[4] Calcular tempo de internação
[5] Determinar tempos de espera na fila
[6] Sair do programa

-----
Escolha uma opção: 3
Digite o nome do executante
Ex.: Hospital Sao Lucas Da PUCRS ou Sao Lucas: femi

No total o Femi teve 2603 pacientes.

```

	idade	municipio_residencia	solicitante	...	data_internacao	data_alta	executante
23	41.0	SAO BORJA	HOSPITAL FEMINA	...	2019-01-24	2019-01-28	HOSPITAL FEMINA
24	28.0	PORTO ALEGRE	HOSPITAL FEMINA	...	2018-03-12	2018-03-07	HOSPITAL FEMINA
33	65.0	PORTO ALEGRE	HOSPITAL FEMINA	...	2021-01-06	2021-01-07	HOSPITAL FEMINA
42	46.0	PORTO ALEGRE	HOSPITAL FEMINA	...	2018-03-12	2018-03-02	HOSPITAL FEMINA
54	61.0	NOVA HARTZ	HOSPITAL FEMINA	...	2018-05-02	2018-04-30	HOSPITAL FEMINA

```

[5 rows x 7 columns]

```

4. Calcular tempo de internação

Aqui repetimos a validação, as datas são convertidas para fazer o calculo de tempo de internação

```

if op == 4:
    # verifica se o solicitante existe
    solicitante_very = 0
    while solicitante_very == 0:
        # entrada do usuário
        solicitante = str(
            input(
                "Digite o nome do solicitante"
                + bcolors.OKBLUE
                + "\nEx.: Hospital Sao Lucas Da PUCRS ou Sao Lucas: "
                + bcolors.CEND
            )
        )
        solicitante_df = dados[
            ["executante", "solicitante", "data_solicitacao",
            "data_alta"]
        ][dados["solicitante"].str.contains(solicitante.upper())]
        solicitante_very = solicitante_df.shape[0]
        if solicitante_very == 0:
            print(
                bcolors.FAIL
                + f"\nNão encontramos {solicitante.upper()}, digite um
solicitante valido!"
                + bcolors.CEND
            )
        else:

            # converte datas para formato 'datetime64'
            dados["data_solicitacao"] = dados["data_solicitacao"].astype(
                "datetime64"
            )
            dados["data_autorizacao"] = dados["data_autorizacao"].astype(
                "datetime64"
            )
            dados["data_internacao"] =
dados["data_internacao"].astype("datetime64")
            dados["data_alta"] = dados["data_alta"].astype("datetime64")

```

```

        # filtra as colunas que seram exibidas
        tempo_df = dados[["executante", "data_solicitacao",
"data_alta"]][
            dados["solicitante"].str.contains(solicitante.upper())
        ]
        # calcula dos dias de internação
        dias_internacao = tempo_df["data_alta"] -
tempo_df["data_solicitacao"]
        # inseri os dias de internação na tabela de saida
        tempo_df["dias_internados"] = dias_internacao

        print(
            bcolors.OKGREEN
            + f"\nTempo de internação, total de {tempo_df.shape[0]}
pacientes"
            + bcolors.CEND
        )
        print(tempo_df)

```

Mais uma vez testamos se a entrada é valida, e após a correção, os dados referente, aos dias de internação são exibidos

```

Digite o nome do solicitante
Ex.: Hospital Sao Lucas Da PUCRS ou Sao Lucas: posto alegre

Não encontramos POSTO ALEGRE, digite um solicitante valido!
Digite o nome do solicitante
Ex.: Hospital Sao Lucas Da PUCRS ou Sao Lucas: pucrs

Tempo de internação, total de 6320 pacientes

```

	executante	data_solicitacao	data_alta	dias_internados
0	HOSPITAL SAO LUCAS DA PUCRS	2018-05-02	2018-05-03	1 days
11	HOSPITAL SAO LUCAS DA PUCRS	2019-01-24	2019-02-01	8 days
12	HOSPITAL SAO LUCAS DA PUCRS	2018-04-02	2018-04-03	1 days
52	HOSPITAL SAO LUCAS DA PUCRS	2018-03-13	2018-03-14	1 days
57	HOSPITAL SAO LUCAS DA PUCRS	2018-05-01	2018-05-15	14 days
...
50965	HOSPITAL SAO LUCAS DA PUCRS	2019-03-21	2019-03-22	1 days
50972	HOSPITAL SAO LUCAS DA PUCRS	2019-12-02	2019-12-06	4 days
50984	HOSPITAL SAO LUCAS DA PUCRS	2019-04-09	2019-04-14	5 days
50990	HOSPITAL SAO LUCAS DA PUCRS	2018-09-05	2018-09-06	1 days
50992	HOSPITAL SAO LUCAS DA PUCRS	2019-04-10	2019-04-11	1 days

[6320 rows x 4 columns]

5. Determinar tempo de espera na fila

Neste último bloco, calculamos o tempo de espera na fila, como neste ponto não temos entrada de usuario, apenas exibimos os dados do dataframe geral

```

if op == 5:
    # converte datas para formato 'datetime64'
    dados["data_solicitacao"] =
dados["data_solicitacao"].astype("datetime64")
    dados["data_autorizacao"] =
dados["data_autorizacao"].astype("datetime64")
    dados["data_internacao"] =
dados["data_internacao"].astype("datetime64")
    dados["data_alta"] = dados["data_alta"].astype("datetime64")

    # filtra os dados por data
    espera_df = dados[["data_solicitacao", "data_internacao"]]
    # calcula dias na fila

```

```

    dias_espera = espera_df["data_internacao"] -
espera_df["data_solicitacao"]
    # inseri na tabela a nova coluna com dias na fila
    espera_df.insert(2, "dias_na_fila", dias_espera)
    # separa o maiores tempos de internação imprimi
    print(
        bcolors.OKGREEN
        + "\n0 cinco maiores tempos de internação são"
        + bcolors.CEND
    )
    print(espera_df[espera_df["dias_na_fila"].between("1200 days", "3000
days")])
    else:
        print(
            bcolors.FAIL + "\nDigite uma opção valida, para continuar!" +
bcolors.CEND
        )

```

Aqui é exibido os cinco maiores tempos de espera, porem a busca é feita pelo dataframe geral, pode ocorrer divergencias com que se espera na saída. para chegar a uma busca mais realista, podemos filtrar pelo ano do inicio da pandemia, por exemplo.

```

O cinco maiores tempos de internação são
data_solicitacao data_internacao dias_na_fila
17812      2015-06-18      2019-05-12      1424 days
19307      2017-06-20      2021-01-21      1311 days
25471      2017-09-12      2021-01-25      1231 days
29100      2015-11-05      2019-03-02      1213 days
39557      2014-04-01      2019-04-01      1826 days

```

6. FIM (sair do programa)

```
print("FIM!")
```

aqui temos um simples print("fim") para sair do programa.

Agradecimentos

Gostaria de manifestar a minha satisfação em participar deste processo seletivo, e como estudante em busca de oportunidades, foi um grande desafio executar esta tarefa.

Muito Obrigado, PucRS/Dell e todos os envolvidos neste processo.

contatos:

email: maroberto13@gmail.com

Linkedin: <https://www.linkedin.com/in/marcos-roberto-moreira-418611b2/>

Github: <https://github.com/maroberto>

Fone/WhatsApp: (11) 95134-0113

