

Explicação do Código - Dashboard de Gestão de Problemas

****Objetivo do Script**:** Criar um painel visual interativo com filtros e gráficos para visualização de dados relacionados a problemas gerenciados em um arquivo Excel ou CSV.

1. IMPORTAÇÕES

```
import pandas as pd
import plotly.express as px
import dash
from dash import dcc, html, dash_table, Input, Output, State
import os
from datetime import datetime
import io
```

- Bibliotecas necessárias para manipular dados, criar gráficos e construir a interface do Dash.

2. CAMINHOS DOS ARQUIVOS

```
caminho_arquivo = r"C:...xlsx.xlsx"
caminho_logo = "assets/logo.png"
```

Define os caminhos do arquivo de dados padrão e da imagem da logo.

3. CARREGAMENTO INICIAL DOS DADOS

```
if os.path.exists(caminho_arquivo):
```

```
df = pd.read_excel(...)
```

```
df['Data Criação'] = pd.to_datetime(...)
```

else:

```
df = pd.DataFrame()
```

Verifica se o Excel existe e carrega os dados. Converte a coluna de datas para datetime.

4. CONFIGURAÇÃO DO APP DASH

```
app = dash.Dash(__name__)
```

```
app.title = "Dashboard de Problemas"
```

Inicializa a aplicação Dash e define o título da aba.

5. LAYOUT DO DASHBOARD

```
app.layout = html.Div([...])
```

Cria a estrutura visual contendo:

- Logo e título
- Componente para upload de arquivos
- Filtros de Status, Prioridade, Módulo e Período
- Gráficos (barra, pizza e linha)
- Tabela de registros
- Botões para exportar dados

6. FUNÇÃO DE FILTRAGEM

```
def filtrar_dados(dataframe, status, prioridade, modulo, data_ini, data_fim):
```

...

Filtra o DataFrame com base nos filtros selecionados pelo usuário.

7. CALLBACK PARA ATUALIZAR CONTEÚDO

```
@app.callback(...)
```

```
def atualizar_conteudo(contents, filename):
```

```
    ...
```

- Recebe o arquivo importado (CSV ou Excel)
- Atualiza filtros, gráficos e tabela de forma dinâmica

8. GERAÇÃO DE GRÁFICOS

- **Barra**: Quantidade por Status
- **Pizza**: Distribuição por Prioridade
- **Linha**: Evolução temporal dos registros

Cada gráfico tem cor personalizada e plano de fundo roxo.

9. EXECUÇÃO DO APP

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

Inicia o servidor local para acessar o dashboard.