# Documentação do Módulo pagina2.py

## 1. Introdução

O módulo pagina2.py é um componente essencial do Sistema de Gestão Integrada (SGI), desenvolvido em Python com a biblioteca Streamlit. Sua principal função é fornecer um painel de Business Intelligence (BI) para o **Monitoramento de Realização de RDO (Registro Diário de Ocorrências)**. A página permite aos usuários visualizar a aderência diária, regional, por unidade e por cidade dos RDOs, comparando o total de equipes com o total de RDOs registrados. Inclui também uma tabela detalhada para verificar a realização de RDO por equipe.

## 2. Dependências

Este módulo importa as seguintes bibliotecas e módulos:

* streamlit as st: Para a construção da interface de usuário interativa.
* pandas as pd: Para manipulação e análise de dados tabulares.
* plotly.express as px: Para a criação de gráficos interativos (embora o código fornecido use streamlit\_echarts para os gráficos principais, plotly.express está importado).
* numpy as np: Para operações numéricas.
* datetime, date, timedelta (do módulo datetime): Para manipulação de datas e períodos.
* subprocess: Para executar comandos externos, especificamente para atualizar os dados do banco.
* os: Para interações com o sistema operacional.
* streamlit\_echarts: Para a renderização de gráficos interativos baseados em ECharts.
* io.BytesIO: Para manipulação de dados em memória (não diretamente usado para download nesta página, mas importado).

## 3. Funções Principais

### 3.1. app()

**Propósito:** Esta é a função principal que orquestra a interface de usuário e a lógica de negócios da pagina2.py. Ela gerencia o carregamento de dados, a aplicação de filtros, a geração de gráficos e a exibição da tabela de aderência de RDO.

**Fluxo de Execução:** 1. **Inicialização do session\_state:** Garante que a variável ultima\_atualizacao esteja presente no estado da sessão do Streamlit. 2. **Barra Lateral (Sidebar):** \* **Logotipo:** Exibe o logotipo da empresa. \* **Botão “Atualizar dados”:** Permite ao usuário forçar a atualização dos dados. Ao ser clicado, executa o script banco\_dados\_rdo.py via subprocess, que é responsável por extrair os dados mais recentes do banco e atualizar o arquivo turnos\_eventos\_fim\_rdo.csv. A data e hora da atualização são registradas no session\_state. \* **Informação da Última Atualização:** Exibe a data e hora da última atualização dos dados. \* **Filtros Interativos:** Permite ao usuário filtrar os dados por: \* **Período:** Seleção de um intervalo de datas (data\_inicio, data\_fim). \* **Regional:** Seleção múltipla de regionais predefinidas (Morrinhos, Rio Verde, TO, MT). \* **Unidade:** Seleção múltipla de unidades dentro das regionais selecionadas. \* **Tipo de Prefixo:** Seleção múltipla de tipos de prefixo, filtrando inicialmente por “EPS - EQUIPE PESADA DE SERVIÇOS” e “EPLV - EQUIPE LINHA VIVA”. 3. **Carregamento e Preparação de Dados:** \* Lê o arquivo turnos\_eventos\_fim\_rdo.csv para obter os dados brutos. \* Converte a coluna data\_turno para o formato de data. \* Define um dicionário regionais\_unidades para mapear unidades a regionais. 4. **Aplicação de Filtros:** O DataFrame df é filtrado com base nas seleções do usuário na barra lateral (período, unidades, tipos de prefixo). O resultado é df\_filtrado. 5. **Cálculos para Gráficos:** \* **Aderência Dia Geral:** Calcula o total de equipes únicas e o total de eventos “FIM DO RDO” por dia (data\_turno). \* **Aderência Regional:** Mapeia as unidades para suas respectivas regionais e calcula o total de equipes e RDOs por regional. \* **Aderência Unidade:** Calcula o total de equipes e RDOs por unidade. \* **Aderência Cidade:** Calcula o total de equipes e RDOs por cidade. 6. **Geração de Gráficos (ECharts):** Utiliza streamlit\_echarts para exibir gráficos de barras comparando o “Total Equipes” e “Total RDO” para as visões diária, regional, por unidade e por cidade. Os gráficos são interativos e incluem tooltips e opções de zoom. 7. **Tabela Detalhada:** \* Cria uma coluna “Equipe” concatenando num\_operacional e prefixo. \* Cria uma coluna booleana “Fez RDO” para indicar se um evento “FIM DO RDO” ocorreu para aquela equipe na data. \* Agrupa os dados por Unidade, Equipe, Cidade e Data, e exibe uma tabela formatada com “Sim” ou “Não” para a realização do RDO.

## 4. Fluxo de Dados e Lógica de Negócio

1. **Fonte de Dados:** O módulo depende do arquivo turnos\_eventos\_fim\_rdo.csv, que é gerado e atualizado pelo script banco\_dados\_rdo.py.
2. **Atualização:** A atualização dos dados é manual, acionada pelo usuário através de um botão na barra lateral, que executa um script Python externo para garantir que os dados exibidos sejam os mais recentes.
3. **Classificação Geográfica:** As unidades são agrupadas em regionais predefinidas, permitindo uma análise hierárquica dos dados.
4. **Filtragem:** Os filtros de data, regional, unidade e tipo de prefixo são aplicados sequencialmente para refinar o conjunto de dados antes da visualização.
5. **Cálculo de Aderência:** A aderência é calculada comparando o número de equipes ativas com o número de RDOs finalizados, fornecendo um indicador de conformidade operacional.
6. **Visualização:** Os gráficos de barras ECharts oferecem uma representação visual clara da aderência, enquanto a tabela detalhada permite uma inspeção granular por equipe e data.

## 5. Integração com o Projeto SGI

O pagina2.py é um módulo de página que é importado e chamado pelo main.py quando o usuário seleciona a opção correspondente no menu principal. Ele se integra ao sistema SGI fornecendo uma interface de BI específica para o monitoramento de RDOs, utilizando dados gerenciados por outros módulos (banco\_dados\_rdo.py) e aderindo ao sistema de permissões centralizado no main.py.

## 6. Considerações Finais

Este módulo é crucial para a supervisão da execução das atividades diárias, permitindo que os gestores acompanhem a taxa de realização dos Registros Diários de Ocorrências. A capacidade de filtrar e visualizar dados por diferentes dimensões geográficas e operacionais torna-o uma ferramenta valiosa para identificar gargalos, garantir a conformidade e otimizar a eficiência das equipes.