# Documentação do Módulo pagina1.py

## 1. Introdução

O módulo pagina1.py é um dos principais componentes do Sistema de Gestão Integrada (SGI), desenvolvido em Python com a biblioteca Streamlit. Ele é responsável por apresentar um painel de Business Intelligence (BI) focado no **Monitoramento de Escala x Turnos**, oferecendo uma visão comparativa entre a escala planejada (SGD), os turnos abertos no NewMars e os turnos contratados. A página permite a filtragem de dados por período, regional, unidade, coordenação e tipo de equipe, além de fornecer um controle detalhado da abertura de turnos por equipe.

## 2. Dependências

Este módulo importa as seguintes bibliotecas e módulos internos:

* 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.
* datetime, date, timedelta (do módulo datetime): Para manipulação de datas e períodos.
* subprocess: Para executar comandos externos (não utilizado diretamente no código fornecido, mas pode ser para scripts de atualização).
* os: Para interações com o sistema operacional, como verificar a existência de arquivos.
* streamlit\_echarts: Para a renderização de gráficos ECharts (não utilizado diretamente no código fornecido).
* numpy as np: Para operações numéricas, especialmente na classificação de dados.
* io.BytesIO: Para manipulação de dados em memória, usado na exportação para Excel.
* sys: Para funcionalidades do sistema (não utilizado diretamente no código fornecido).
* banco\_escala\_turnos.exportar\_dados\_para\_csv: Módulo interno para atualização dos dados base.

## 3. Funções Principais

### 3.1. atualizar\_dados()

**Propósito:** Esta função é responsável por acionar a atualização dos dados da aplicação. Ela limpa o cache de dados do Streamlit e executa a função exportar\_dados\_para\_csv do módulo banco\_escala\_turnos.py, que se encarrega de extrair os dados mais recentes do banco de dados e salvá-los em arquivos CSV. Isso garante que o painel sempre exiba informações atualizadas.

**Retorno:** Uma tupla (bool, str) indicando se a atualização foi bem-sucedida e uma mensagem correspondente.

### 3.2. carregar\_dados()

**Propósito:** Carrega, limpa e prepara os dados a partir dos arquivos CSV (escala\_nova.csv, turnos\_newmars.csv, Contratos.csv) localizados no diretório data/. Esta função é decorada com @st.cache\_data, o que significa que seus resultados são armazenados em cache para evitar recarregamentos desnecessários e otimizar a performance da aplicação. Ela padroniza formatos de data, prefixos e realiza merges para enriquecer os DataFrames com informações de unidade e tipo de equipe baseadas no Contratos.csv.

**Retorno:** Uma tupla (df\_escala, df\_turnos, df\_contrato) contendo os DataFrames processados. Retorna (None, None, None) em caso de erro crítico no carregamento.

### 3.3. classificar\_regional(df)

**Propósito:** Classifica os prefixos ou unidades em regionais específicas com base em um conjunto de condições predefinidas. Esta função adiciona uma nova coluna REGIONAIS ao DataFrame, facilitando a análise e filtragem por agrupamentos geográficos ou operacionais.

**Parâmetros:** \* df: O DataFrame a ser classificado (e.g., df\_escala, df\_turnos, df\_contrato).

**Retorno:** O DataFrame com a nova coluna REGIONAIS preenchida.

### 3.4. classificar\_coordenacao(df)

**Propósito:** Classifica as equipes em coordenações específicas (e.g., “STC”, “PERDAS”, “MANUTENÇÃO”, “CONSTRUÇÃO”, “C&M”) com base na regional e no tipo de prefixo (descricao\_tipo\_prefixo). Esta função adiciona uma nova coluna COORDENACAO ao DataFrame, permitindo uma análise mais granular por área de atuação.

**Parâmetros:** \* df: O DataFrame a ser classificado.

**Retorno:** O DataFrame com a nova coluna COORDENACAO preenchida.

### 3.5. app()

**Propósito:** Esta é a função principal que constrói a interface de usuário da página no Streamlit e orquestra todas as operações de carregamento, processamento, filtragem e exibição de dados. Ela é o ponto de entrada lógico para a renderização da página.

**Fluxo de Execução:** 1. **Carregamento e Classificação de Dados:** Chama carregar\_dados() e, em seguida, aplica as funções classificar\_regional() e classificar\_coordenacao() aos DataFrames de escala, turnos e contrato. 2. **Barra Lateral (Sidebar):** \* Exibe o logotipo da empresa. \* Botão “Atualizar Dados”: Permite ao usuário forçar a atualização dos dados, chamando atualizar\_dados(). \* Informação da Última Atualização: Lê e exibe a data e hora da última atualização dos dados, armazenada em ultima\_atualizacao.txt. \* Filtros Interativos: Permite ao usuário filtrar os dados por: \* **Período:** Seleção de um intervalo de datas. \* **Regional:** Seleção de uma ou mais regionais predefinidas. \* **Unidade:** Seleção de uma ou mais unidades dentro das regionais selecionadas. \* **Coordenação:** Seleção de uma ou mais coordenações. \* **Tipo de Equipe:** Seleção de um ou mais tipos de prefixo. \* **Dias na Escala (Contrato):** Filtro opcional baseado na coluna ‘ESCALA’ do df\_contrato. 3. **Lógica de Filtragem Centralizada:** Aplica todos os filtros selecionados pelo usuário aos DataFrames de escala, turnos e contrato, garantindo que os gráficos e métricas exibam apenas os dados relevantes. 4. **Cálculo de Dias no Filtro:** Determina o número total de dias e o número de domingos dentro do período selecionado, informações usadas nos cálculos de turnos contratados. 5. **Exibição de Conteúdo (Abas):** A página é organizada em abas para diferentes visualizações: \* **Aba 1: Análise Geral** \* **Métricas de Resumo:** Exibe cards com o total de equipes na escala, turnos abertos e turnos contratados para o período e filtros selecionados. \* **Gráfico 1: Comparativo Geral:** Um gráfico de barras que compara o total de equipes na escala SGD, turnos abertos no NewMars e turnos contratados para o período geral. \* **Gráfico 2: Comparativo por Coordenação:** Similar ao gráfico geral, mas segmentado por coordenação, mostrando a distribuição e comparação entre os diferentes tipos de equipes. \* **Aba 2: Análise Geográfica** \* **Gráfico 1: Comparativo por Regional:** Compara os dados de escala, turnos abertos e contratados por regional. \* **Gráfico 2: Comparativo por Unidade:** Detalha a comparação por unidade, permitindo uma visão mais específica dentro de cada regional. \* **Gráfico 3: Comparativo por Cidade:** Se os dados de cidade estiverem disponíveis, exibe a comparação por cidade. \* **Aba 3: Controle de Abertura de Turno por Equipe** \* **Métricas Detalhadas:** Apresenta cards com o total de equipes contratadas, na escala (SGD), com turnos (NewMars) e contratos sem turno. \* **Visualização de Status por Prefixo:** Exibe caixas coloridas para cada prefixo, indicando seu status (e.g., “Turno + Escala + Contrato”, “Escala + Contrato (s/ Turno)”, “Na Escala (fora do Contrato)”). Cada caixa possui um tooltip com informações detalhadas (Coordenação, Tipo, Escala). \* **Filtros de Status:** Checkboxes para filtrar os prefixos exibidos com base em seu status. \* **Botão de Download:** Permite baixar um arquivo Excel (.xlsx) com o controle detalhado de turnos, incluindo prefixo, status, coordenação, tipo de equipe e flags de presença em contrato, escala e turno.

## 4. Fluxo de Dados e Lógica de Negócio

1. **Carregamento:** Os dados são carregados de arquivos CSV. O Contratos.csv é crucial para mapear prefixos a unidades e tipos de equipe, e para definir os dias de escala contratados.
2. **Padronização e Enriquecimento:** As colunas de data são convertidas para o formato datetime, e os prefixos são padronizados (maiúsculas, sem espaços). Os DataFrames de escala e turnos são enriquecidos com informações de unidade, regional e coordenação, que são derivadas do Contratos.csv e das funções de classificação.
3. **Filtragem:** A filtragem é aplicada em cascata: primeiro os filtros de data, regional, unidade, coordenação e tipo de equipe são aplicados aos DataFrames de escala e turnos. Se o Contratos.csv estiver presente e filtrado, a lista de prefixos válidos resultante é usada para um filtro final nos DataFrames de escala e turnos, garantindo consistência.
4. **Cálculos:** Cálculos como o número de equipes únicas por dia e a soma total de equipes em escala/turnos são realizados para alimentar os gráficos. O número de turnos contratados é calculado com base nos dias de escala (ESCALA no Contratos.csv) e no período selecionado, considerando o número de dias úteis e domingos.
5. **Visualização:** Os dados processados são apresentados através de gráficos de barras interativos do Plotly e métricas de resumo, oferecendo uma visão clara do desempenho e conformidade da escala de turnos.
6. **Exportação:** A funcionalidade de download permite que os usuários exportem os dados detalhados para análise offline.

## 5. Integração com o Projeto SGI

O pagina1.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 escalas e turnos, utilizando dados gerenciados por outros módulos (banco\_escala\_turnos.py) e aderindo ao sistema de permissões centralizado no main.py.

## 6. Considerações Finais

Este módulo é fundamental para a gestão operacional, permitindo que os usuários visualizem e analisem a conformidade entre o planejamento de escala e a execução dos turnos. A capacidade de filtrar dados por diversas dimensões e a exportação para Excel tornam-no uma ferramenta poderosa para tomada de decisões e auditoria.