# Documentação do Módulo pagina4.py

## 1. Introdução

O módulo pagina4.py é um componente crucial do Sistema de Gestão Integrada (SGI), desenvolvido em Python com a biblioteca Streamlit. Ele funciona como um painel de Business Intelligence (BI) dedicado aos **Indicadores de Análises de Monitoramento das Filmagens**. Esta página oferece uma visão abrangente sobre a aderência das equipes ao processo de filmagem de atividades, a evolução do monitoramento ao longo do tempo, e o desempenho individual de pessoas e equipes monitoradas. Permite a filtragem detalhada por ano, mês, semana, empresa, unidade, tipo de contrato, tipo de equipe, equipe específica, supervisor e monitor.

## 2. Dependências

Este módulo importa as seguintes bibliotecas e módulos:

* mysql.connector: Para conexão e interação com bancos de dados MySQL (embora não diretamente utilizado no código fornecido para carregar dados, sua presença sugere uma origem de dados MySQL).
* pandas as pd: Essencial para manipulação e análise de dados tabulares.
* streamlit as st: Para a construção da interface de usuário interativa.
* matplotlib.pyplot as plt: Para a criação de gráficos estáticos (embora plotly seja preferido para interatividade).
* plotly.graph\_objects as go: Para a criação de gráficos interativos e dinâmicos.
* plotly.express as px: Para a criação rápida de gráficos interativos.
* io.BytesIO: Para manipulação de dados em memória, especialmente para exportação de arquivos.
* os: Para interações com o sistema operacional, como verificar a existência de arquivos e obter datas de modificação.
* subprocess: Para executar comandos externos, como scripts de atualização de dados.
* datetime, timedelta (do módulo datetime): Para manipulação de datas e cálculos de períodos.
* time: Para funcionalidades relacionadas a tempo, como obter o tempo de modificação de arquivos.
* plotly.subplots.make\_subplots: Para criar layouts de gráficos com múltiplos subplots.
* st\_aggrid.AgGrid, GridOptionsBuilder, JsCode: Para exibir tabelas interativas e personalizáveis no Streamlit.
* numpy as np: Para operações numéricas, especialmente na classificação de dados.

## 3. Funções Principais

### 3.1. get\_datas\_semana(ano, num\_semana)

**Propósito:** Calcula as datas de início (segunda-feira) e fim (domingo) de uma semana específica em um determinado ano. É útil para exibir o intervalo de datas correspondente a uma semana selecionada nos filtros.

**Parâmetros:** \* ano (int): O ano de referência. \* num\_semana (int): O número da semana (1 a 52 ou 53).

**Retorno:** Uma tupla (str, str) contendo as datas de início e fim da semana no formato “DD/MM/AAAA”. Retorna “Data inválida” se a semana não for válida para o ano.

### 3.2. get\_data\_mod\_time(file\_path)

**Propósito:** Retorna a data e hora da última modificação de um arquivo especificado. Utilizado para informar ao usuário quando os dados foram atualizados pela última vez.

**Parâmetros:** \* file\_path (str): O caminho completo para o arquivo.

**Retorno:** Uma string com a data e hora da última modificação no formato “DD/MM/AAAA HH:MM:SS”, ou “Arquivo não encontrado” se o arquivo não existir.

### 3.3. conectar\_csv()

**Propósito:** Carrega, limpa e prepara os dados a partir de vários arquivos CSV (turnos\_monitoria.csv, avulsa.csv, pessoas\_monitoria.csv, turnos\_pessoas\_monitoria.csv) localizados no diretório data/. Esta função é decorada com @st.cache\_data para otimizar a performance, armazenando os resultados em cache. Realiza a padronização de formatos de data, classificação de tipos de equipe e prefixos, e mescla DataFrames para enriquecer os dados.

**Retorno:** Uma tupla de DataFrames (df\_turnos, df\_avulsa, df\_pessoas, df\_turnos\_pessoas) processados. Retorna (None, None, None, None) em caso de erro crítico no carregamento.

### 3.4. aplicar\_filtros(...)

**Propósito:** Aplica uma série de filtros aos DataFrames de turnos e avulsa com base nas seleções do usuário na barra lateral. Garante que apenas os dados relevantes sejam utilizados para as análises e visualizações.

**Parâmetros:** Recebe os DataFrames originais e todas as seleções de filtro do usuário (ano, meses, empresa, unidades, tipo de prefixo, equipe, supervisor, monitor, semanas, tipos de equipe).

**Retorno:** Uma tupla (df\_turnos\_filtrado, df\_avulsa\_filtrado) contendo os DataFrames após a aplicação de todos os filtros.

### 3.5. app()

**Propósito:** Esta é a função principal que constrói a interface de usuário da pagina4.py no Streamlit e orquestra todas as operações de carregamento, processamento, filtragem e exibição de dados. É o ponto de entrada lógico para a renderização da página.

**Fluxo de Execução:** 1. **Barra Lateral (Sidebar):** \* **Atualização de Dados:** Exibe a data da última atualização do arquivo turnos\_monitoria.csv e um botão para forçar a atualização dos dados, executando o script exporta\_dados\_monitoria\_csv.py via subprocess. \* **Filtros Interativos:** Permite ao usuário filtrar os dados por: \* **Ano:** Seleção de um ano específico. \* **Meses:** Seleção múltipla de meses dentro do ano selecionado. \* **Semanas do Ano:** Seleção múltipla de semanas, com um expansor para visualizar as datas correspondentes a cada semana. \* **Empresa:** Seleção de uma empresa. \* **Unidades:** Seleção múltipla de unidades associadas à empresa selecionada. \* **Tipo de Contrato:** Seleção de tipo de prefixo (Âncora, Satélite ou Todas). \* **Tipo de Equipe:** Seleção múltipla de tipos de equipe (e.g., Linha Leve, Equipe Pesada). \* **Equipe:** Campo de texto para buscar equipes por nome ou prefixo. \* **Supervisor:** Seleção de um supervisor. \* **Monitor:** Seleção de um monitor. 2. **Carregamento e Filtragem de Dados:** Chama conectar\_csv() para carregar os dados e aplicar\_filtros() para filtrar os DataFrames com base nas seleções do usuário. 3. **Conteúdo Principal (Abas):** A página é organizada em quatro abas para diferentes visualizações: \* **MONITORAMENTO EQUIPE:** \* **Métricas de Aderência:** Exibe cards com o total de equipes que gravaram atividades, que não gravaram, e porcentagens de aderência. \* **Gráfico de Pizza (Aderência Filmagens):** Visualiza a proporção de atividades filmadas versus não filmadas. \* **Gráfico de Barras (Filmagens por Tipo de Equipe):** Compara o número de atividades filmadas e não filmadas por tipo de equipe. \* **Gráfico de Barras (Filmagens por Unidade):** Apresenta a aderência de filmagens por unidade. \* **Gráfico de Barras (Filmagens por Supervisor):** Mostra a aderência de filmagens por supervisor. \* **EVOLUÇÃO EQUIPES MONITORADAS MÊS:** \* **Gráfico de Linha (Evolução Mensal):** Exibe a tendência do número de equipes monitoradas e filmagens realizadas ao longo dos meses. \* **EQUIPES MONITORADAS:** \* **Tabela Detalhada (AgGrid):** Uma tabela interativa que lista as equipes, seus supervisores, monitores, unidades, tipos de equipe e o status de filmagem, com funcionalidades de busca e ordenação. \* **MONITORAMENTO PESSOAS:** \* **Tabela Detalhada (AgGrid):** Uma tabela interativa que lista as pessoas, suas equipes, supervisores, monitores e o status de filmagem, com funcionalidades de busca e ordenação.

## 4. Fluxo de Dados e Lógica de Negócio

1. **Fonte de Dados:** O módulo depende de quatro arquivos CSV (turnos\_monitoria.csv, avulsa.csv, pessoas\_monitoria.csv, turnos\_pessoas\_monitoria.csv) que são atualizados por um script externo (exporta\_dados\_monitoria\_csv.py).
2. **Pré-processamento:** Após o carregamento, os dados passam por um processo de limpeza e enriquecimento, incluindo a criação de colunas para equipe\_display (nome de exibição da equipe), tipo\_equipe (classificação baseada no prefixo numérico) e tipo\_prefixo (Âncora/Satélite).
3. **Filtragem:** Os filtros na barra lateral são aplicados de forma cumulativa, permitindo que o usuário refine a análise em múltiplas dimensões (tempo, geográfico, organizacional).
4. **Cálculos de Métricas:** São calculadas métricas como o total de atividades filmadas/não filmadas, equipes distintas, e porcentagens de aderência, que são a base para os indicadores exibidos.
5. **Visualização:** Os dados processados são apresentados através de uma combinação de gráficos (Plotly e ECharts) e tabelas interativas (AgGrid), proporcionando diferentes perspectivas sobre o monitoramento das filmagens.
6. **Interatividade:** Os gráficos e tabelas são interativos, permitindo que o usuário explore os dados em detalhes.

## 5. Integração com o Projeto SGI

O pagina4.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 filmagens, utilizando dados gerenciados por outros módulos (exporta\_dados\_monitoria\_csv.py) e aderindo ao sistema de permissões centralizado no main.py.

## 6. Considerações Finais

Este módulo é fundamental para a gestão da qualidade e conformidade das operações que exigem registro em vídeo. Ele permite que os gestores monitorem a aderência das equipes às diretrizes de filmagem, identifiquem tendências, avaliem o desempenho de supervisores e monitores, e tomem decisões baseadas em dados para melhorar os processos de trabalho. A riqueza de filtros e as diversas visualizações tornam-no uma ferramenta poderosa para a auditoria e otimização contínua.