# Documentação Técnica do Projeto SGI

## 1. Introdução

Este documento detalha a arquitetura, funcionalidades e componentes do projeto SGI (Sistema de Gestão), uma aplicação desenvolvida em Python, utilizando a biblioteca Streamlit para a interface de usuário. O sistema é projetado para auxiliar na gestão de operações, monitoramento de equipes e inspeções, com controle de acesso baseado em permissões.

## 2. Estrutura do Projeto

O projeto SGI é organizado em uma estrutura de diretórios que separa os componentes principais, dados e arquivos de configuração. A seguir, uma visão geral da estrutura:

sgi\_project/  
├── sgi/  
│ ├── \_\_pycache\_\_/ # \_Arquivos compilados do Python para otimização de carregamento.\_  
│ ├── data/ # \_Contém arquivos CSV e JSON com dados brutos e configuráveis para a aplicação.\_  
│ │ ├── Contratos.csv # \_Dados relacionados a contratos.\_  
│ │ ├── avulsa.csv # \_Dados avulsos ou temporários.\_  
│ │ ├── blitz.csv # \_Informações sobre operações blitz.\_  
│ │ ├── blitzPessoas.csv # \_Relação de pessoas envolvidas em operações blitz.\_  
│ │ ├── dados\_turnos\_eventos.csv # \_Dados detalhados de turnos e eventos.\_  
│ │ ├── escala\_data.csv # \_Dados de escala de trabalho.\_  
│ │ ├── escala\_nova.csv # \_Nova versão ou dados complementares de escala.\_  
│ │ ├── eventos.csv # \_Registro de eventos diversos.\_  
│ │ ├── funcoes.csv # \_Definição de funções ou cargos.\_  
│ │ ├── log\_acessos.csv # \_Registro de acessos dos usuários ao sistema.\_  
│ │ ├── log\_permissoes.csv # \_Registro de alterações de permissões.\_  
│ │ ├── permissoes.json # \_Configuração das permissões de acesso dos usuários.\_  
│ │ ├── pessoas.csv # \_Cadastro de pessoas ou usuários.\_  
│ │ ├── pessoas\_monitoria.csv # \_Dados de pessoas para monitoramento.\_  
│ │ ├── regras.json # \_Regras de negócio ou de validação.\_  
│ │ ├── respostas.csv # \_Respostas de formulários ou interações.\_  
│ │ ├── turnos.csv # \_Dados gerais de turnos.\_  
│ │ ├── turnosPessoas.csv # \_Relação de pessoas por turno.\_  
│ │ ├── turnos\_data.csv # \_Dados específicos de turnos.\_  
│ │ ├── turnos\_eventos\_fim\_rdo.csv # \_Eventos de fim de RDO (Registro Diário de Ocorrências) por turno.\_  
│ │ ├── turnos\_monitoria.csv # \_Dados de monitoramento de turnos.\_  
│ │ ├── turnos\_newmars.csv # \_Dados de turnos relacionados ao sistema Newmars.\_  
│ │ ├── turnos\_pessoas\_monitoria.csv # \_Pessoas e seus dados de monitoramento em turnos.\_  
│ │ ├── turnos\_pessoas\_pessoas.csv # \_Relação detalhada de pessoas em turnos.\_  
│ │ └── unidades.csv # \_Cadastro de unidades organizacionais.\_  
│ ├── .env # \_Arquivo de variáveis de ambiente, incluindo senhas de acesso ao banco de dados SGD.\_  
│ ├── 01.py # \_Módulo auxiliar com funcionalidades diversas.\_  
│ ├── banco\_dados\_rdo.py # \_Módulo para extração e manipulação de dados de Registro Diário de Ocorrências (RDO).\_  
│ ├── banco\_escala\_turnos.py # \_Módulo para extração e manipulação de dados de escala de turnos.\_  
│ ├── banco\_unidad\_função.py # \_Módulo para extração e manipulação de dados de unidades e funções.\_  
│ ├── cadastro.py # \_Módulo para funcionalidades de cadastro de informações no sistema.\_  
│ ├── exportar\_dados\_para\_csv.py # \_Módulo para exportar dados gerais para arquivos CSV.\_  
│ ├── exporta\_dados\_monitoria\_csv.py # \_Módulo para exportar dados de monitoria para arquivos CSV.\_  
│ ├── exporta\_dados\_para\_pessoasblitz\_csv.py # \_Módulo para exportar dados de pessoas envolvidas em blitz para arquivos CSV.\_  
│ ├── extract\_data\_csv.py # \_Módulo genérico para extração e processamento de dados de arquivos CSV.\_  
│ ├── loginsgd.py # \_Módulo responsável pela autenticação de usuários via API do SGD.\_  
│ ├── main.py # \_Ponto de entrada principal da aplicação Streamlit, gerencia o fluxo e navegação.\_  
│ ├── mm.py # \_Módulo auxiliar com funcionalidades específicas, possivelmente relacionadas a medições ou métricas.\_  
│ ├── pagina1.py # \_BI de Escala - Visão Geral: Exibe dados e análises da escala de trabalho de forma geral.\_  
│ ├── pagina10.py # \_BI Escala Regional Morrinhos: Exibe dados e análises da escala de trabalho para a regional de Morrinhos.\_  
│ ├── pagina11.py # \_BI Escala Regional Rio Verde: Exibe dados e análises da escala de trabalho para a regional de Rio Verde.\_  
│ ├── pagina12.py # \_BI Escala Regional TO: Exibe dados e análises da escala de trabalho para a regional de Tocantins.\_  
│ ├── pagina13.py # \_BI Escala Regional MT: Exibe dados e análises da escala de trabalho para a regional de Mato Grosso.\_  
│ ├── pagina14.py # \_BI Taxa de Contato - Inspeção Regional Morrinhos: Exibe dados e análises da taxa de contato em inspeções para a regional de Morrinhos.\_  
│ ├── pagina15.py # \_BI Taxa de Contato - Inspeção Regional Rio Verde: Exibe dados e análises da taxa de contato em inspeções para a regional de Rio Verde.\_  
│ ├── pagina16.py # \_BI Taxa de Contato - Inspeção Regional TO: Exibe dados e análises da taxa de contato em inspeções para a regional de Tocantins.\_  
│ ├── pagina17.py # \_BI Taxa de Contato - Inspeção Regional MT: Exibe dados e análises da taxa de contato em inspeções para a regional de Mato Grosso.\_  
│ ├── pagina18.py # \_BI Monitoria Não Conformidades: Exibe dados e análises de não conformidades, possivelmente integrando com Power BI.\_  
│ ├── pagina19.py # \_Módulo para funcionalidades específicas.\_  
│ ├── pagina2.py # \_BI de RDO - Visão Geral: Exibe dados e análises do Registro Diário de Ocorrências de forma geral.\_  
│ ├── pagina20.py # \_Módulo de página com funcionalidades específicas.\_  
│ ├── pagina21.py # \_BI Previsto x Realizado: Exibe dados e análises comparando o planejado com o executado.\_  
│ ├── pagina3.py # \_Módulo de página com funcionalidades específicas.\_  
│ ├── pagina30.py # \_Módulo de página com funcionalidades específicas.\_  
│ ├── pagina4.py # \_BI Monitoria Filmagens - Visão Geral: Exibe dados e análises de monitoramento de filmagens de forma geral.\_  
│ ├── pagina40.py # \_BI Inspeção Blitz - Geral por Prefixo: Exibe dados e análises de inspeções blitz, utilizando prefixos para filtragem.\_  
│ ├── pagina7.py # \_BI Inspeção Blitz - Geral por Números Operacionais: Exibe dados e análises de inspeções blitz, focando em números operacionais.\_  
│ ├── pagina8.py # \_BI Contato Pessoas - Visão Geral: Exibe dados e análises de contato com pessoas de forma geral.\_  
│ ├── pagina\_admin.py # \_Página de Administração: Interface para gerenciamento de usuários e liberação de acessos.\_  
│ ├── sgd.py # \_Módulo para teste e interação com a API externa do SGD.\_  
│ ├── teste.py # \_Arquivo de teste ou protótipo.\_  
│ ├── teste01.py # \_Arquivo de teste ou protótipo.\_  
│ ├── testet.py # \_Arquivo de teste ou protótipo.\_  
│ ├── turnos\_data.csv # \_Dados de turnos.\_  
│ ├── turnos\_eventos\_fim\_rdo.csv # \_Eventos de fim de RDO por turno.\_  
│ └── ultima\_atualizacao.txt # \_Registro da última atualização do sistema ou dados.\_

├ ├── .env # - senha do acessos ao Banco do SGD

## 3. Componentes Principais

### 3.1. main.py

Este é o **ponto de entrada principal** da aplicação Streamlit. Ele orquestra o fluxo de execução, desde a autenticação do usuário até a navegação entre as diferentes seções do sistema. Suas responsabilidades incluem:

* main(): A função principal que configura a página da aplicação (título, layout), aplica estilos CSS globais e gerencia o estado de autenticação do usuário. Dependendo do estado de autenticação, ela direciona para a tela de login ou para o menu principal.
* tela\_login(): Apresenta a interface gráfica para o usuário inserir suas credenciais. Após a submissão, tenta autenticar o usuário e, se bem-sucedido, carrega suas permissões de acesso.
* menu\_principal(): Exibe o menu de navegação principal do sistema, organizado em abas (Operacional, SESMT, Monitoria, Admin). A visibilidade de cada aba e dos botões internos é controlada pelas permissões do usuário, garantindo que apenas funcionalidades autorizadas sejam apresentadas.
* show\_page(page\_name): Função responsável por renderizar a página específica solicitada pelo usuário. Antes de exibir o conteúdo, ela verifica se o usuário possui as permissões necessárias para acessar aquela página, exibindo uma mensagem de erro caso contrário. Inclui também um botão de retorno ao menu principal.
* aplicar\_css\_global(): Aplica estilos CSS personalizados para garantir uma identidade visual consistente em toda a aplicação.

### 3.2. sgd.py

Este módulo é dedicado à **comunicação com a API externa do SGD** (https://www.sgddolp.com.br/api\_mars/). Ele atua como um cliente para a API, permitindo que a aplicação SGI se autentique e troque informações com o sistema SGD. A principal função é:

* acessar\_api\_terminal(): Solicita o login e senha do usuário (com a senha oculta para segurança), constrói uma requisição POST com essas credenciais e outros parâmetros (versão, fonte) e a envia para a URL da API. Processa a resposta da API, verificando se o login foi bem-sucedido e extraindo o souser (identificador de sessão do usuário) para uso posterior na aplicação SGI.

### 3.3. Módulos de Página (paginaX.py)

O projeto é modularizado em diversos arquivos paginaX.py (e.g., pagina1.py, pagina2.py, …, pagina40.py, pagina\_admin.py), onde cada arquivo representa uma **seção ou painel específico da interface de usuário**. Essas páginas são carregadas dinamicamente pelo main.py com base nas permissões do usuário. As funcionalidades abrangem diversas áreas do sistema:

* **Módulos Operacionais (e.g., pagina1.py, pagina2.py, pagina3.py, pagina10.py a pagina13.py, pagina21.py):** Focados no monitoramento e gestão de atividades diárias. Incluem painéis para monitoramento de escala de turnos (geral e por regionais como Morrinhos, Rio Verde, Tocantins, Mato Grosso), acompanhamento de Registro Diário de Ocorrências (RDO), configuração de turnos contratados e comparação entre o previsto e o realizado.
* **Módulos SESMT (e.g., pagina7.py, pagina8.py, pagina14.py a pagina17.py, pagina40.py):** Dedicados à Segurança e Saúde no Trabalho. Apresentam análises de inspeções, taxa de contato de equipes e pessoas, com detalhamento por regionais e visões gerais focadas em números operacionais ou prefixos.
* **Módulos de Monitoria (e.g., pagina4.py, pagina18.py, pagina19.py):** Oferecem indicadores e ferramentas para monitoramento de desempenho e qualidade. Incluem análises de monitoramento de filmagens, apontamento de não conformidades e outras funcionalidades de monitoria.
* **Módulo de Administração (pagina\_admin.py):** Fornece uma interface para que administradores gerenciem usuários, atribuam e revoguem permissões de acesso a diferentes partes do sistema.

### 3.4. Módulos de Banco de Dados e Exportação (banco\_\*.py, exportar\_\*.py, extract\_data\_csv.py)

Esses módulos são responsáveis pela **interação com fontes de dados e pela exportação de informações**. Eles atuam como a camada de acesso a dados e de geração de relatórios:

* banco\_dados\_rdo.py: Contém a lógica para consultar, extrair e processar dados relacionados aos Registros Diários de Ocorrências (RDOs).
* banco\_escala\_turnos.py: Gerencia a extração e manipulação dos dados referentes às escalas de turnos dos colaboradores.
* banco\_unidad\_função.py: Lida com a recuperação e organização de dados sobre unidades organizacionais e as funções desempenhadas pelos usuários.
* exportar\_dados\_para\_csv.py: Um módulo genérico para a funcionalidade de exportar conjuntos de dados para o formato CSV, facilitando a análise externa ou o arquivamento.
* exporta\_dados\_monitoria\_csv.py: Especificamente projetado para exportar dados coletados durante as atividades de monitoria para arquivos CSV.
* exporta\_dados\_para\_pessoasblitz\_csv.py: Focado na exportação de dados de pessoas e equipes envolvidas em operações blitz para o formato CSV.
* extract\_data\_csv.py: Provavelmente contém funções utilitárias para ler, parsear e extrair informações de arquivos CSV genéricos, servindo como uma base para outros módulos de dados.

### 3.5. Diretório data/

Este diretório é o **repositório central para os arquivos de dados** utilizados pela aplicação. Ele armazena informações em formatos CSV e JSON, que são carregadas e processadas pelos módulos Python. A organização desses arquivos permite uma gestão flexível dos dados do sistema:

* **Arquivos CSV:** Contêm dados tabulares como Contratos.csv, pessoas.csv, turnos.csv, eventos.csv, escala\_data.csv, entre outros. Estes arquivos são a base para os relatórios e painéis de BI exibidos na aplicação.
* **Arquivos JSON:** permissoes.json e regras.json são particularmente importantes, pois definem as permissões de acesso dos usuários e as regras de negócio do sistema, respectivamente. Eles permitem uma configuração dinâmica do comportamento da aplicação sem a necessidade de alterar o código-fonte.

## 4. Fluxo de Autenticação e Permissões

O sistema SGI implementa um robusto fluxo de autenticação e gerenciamento de permissões para garantir a segurança e o acesso adequado às funcionalidades:

1. **Login Inicial:** Ao iniciar a aplicação, o usuário é direcionado para a tela\_login(), onde deve inserir suas credenciais.
2. **Autenticação Externa:** As credenciais são enviadas para a API externa do SGD através do módulo sgd.py. Esta API valida o usuário e, em caso de sucesso, retorna um identificador de sessão (souser).
3. **Carregamento de Permissões:** Com base no souser ou outras informações de usuário autenticado, o sistema carrega as permissões específicas do usuário, geralmente a partir do arquivo permissoes.json ou de uma fonte de dados similar. Estas permissões determinam quais partes do sistema o usuário pode acessar.
4. **Exibição do Menu Principal:** O menu\_principal() é então renderizado, apresentando ao usuário apenas as abas e botões correspondentes às funcionalidades para as quais ele possui permissão. Isso personaliza a experiência do usuário e restringe o acesso a informações sensíveis.
5. **Verificação de Permissão por Página:** Sempre que o usuário tenta navegar para uma página específica (paginaX.py), a função show\_page() realiza uma nova verificação de permissão. Isso garante que, mesmo que um usuário tente acessar uma URL diretamente, o acesso será negado se ele não tiver a permissão necessária.

## 5. Dependências

As principais bibliotecas e frameworks utilizados no desenvolvimento do projeto SGI incluem:

* **Streamlit:** Framework Python de código aberto para a criação rápida de aplicações web interativas e painéis de dados. É a base para a interface de usuário do SGI.
* **Requests:** Biblioteca HTTP para Python, utilizada para fazer requisições à API externa do SGD, facilitando a comunicação com serviços web.
* **getpass:** Módulo padrão do Python para entrada segura de senhas, utilizado em sgd.py para ocultar a senha digitada no terminal.
* **Pandas:** Biblioteca fundamental para manipulação e análise de dados em Python. É amplamente utilizada para ler, processar e transformar os dados contidos nos arquivos CSV e JSON do diretório data/.

## 6. Considerações Finais

O projeto SGI é uma aplicação robusta e modular, desenvolvida para otimizar a gestão operacional, o monitoramento de equipes e o controle de inspeções. Sua arquitetura baseada em Streamlit, combinada com a integração a uma API externa e o uso extensivo de arquivos de dados (CSV, JSON), oferece uma solução abrangente e flexível. O sistema de autenticação e permissões garante que o acesso às informações e funcionalidades seja seguro e personalizado para cada usuário, tornando-o uma ferramenta eficaz para as necessidades de gestão.

Paginas ultilizadas resulmo

├── pagina1.py - BI DE ESCALA - GERAL  
│ ├── pagina10.py - BI ESCALA REGIONAL MORRINHOS  
│ ├── pagina11.py – BI ESCALA REGIONAL RIO VERDE  
│ ├── pagina12.py - BI ESCALA REGIONAL TO  
│ ├── pagina13.py - BI ESCALA REGIONAL MT  
│ ├── pagina14.py - BI TAXA DE CONTATO - INSPEÇÃO REGIONAL MORRINHOS  
│ ├── pagina15.py - BI TAXA DE CONTATO - INSPEÇÃO REGIONAL RIO VERDE  
│ ├── pagina16.py - BI TAXA DE CONTATO - INSPEÇÃO REGIONAL TO  
│ ├── pagina17.py - BI TAXA DE CONTATO - INSPEÇÃO REGIONAL MT  
│ ├── pagina18.py - BI MONITORIA NÃO CONFORMIDADES - EM POWER BI  
│ ├  
│ ├── pagina2.py – BI DE RDO - GERAL  
│ ├── pagina20.py  
│ ├── pagina21.py - BI PREVISTO X REALIZADO  
│ ├  
│ ├  
│ ├── pagina4.py – BI MONITORIA FILMAGENS - GERAL  
│ ├── pagina40.py - BI INSPESÃO BLITZ - GERAL- UTILIZANDO OS PREFIXOS  
│ ├── pagina7.py - BI INSPESÃO BLITZ - GERAL- NUMERO OPERACIONAIS AINDA  
│ ├── pagina8.py - BI CONTATO PESSOAS - GERAL  
│ ├── pagina\_admin.py – PAGINA DE ADM PARA LIBERAÇÃO PESSOAS DE ACESSOS  
│ ├── sgd.py – TESTE DA API