

# Sistema de Planilhas e Relatórios em Delphi 12 e PostgreSQL

## Objetivo Principal

Desenvolver um sistema robusto (GridFlow) para a criação e gerenciamento de **Planilhas** e **Relatórios**. A aplicação será construída em **Delphi 12** e usará o banco de dados **PostgreSQL** para dados públicos, enquanto dados privados serão armazenados localmente em arquivos **JSON**. A aplicação deve ser modular, segura e oferecer diferentes níveis de acesso.

Ao iniciar o sistema, uma **Planilha** de exemplo e um usuário de demonstração serão carregados automaticamente, sem a necessidade de conexão com um servidor. Isso permitirá que o usuário explore a aplicação de forma privada, com acesso às seguintes funcionalidades:

- **Planilhas**
- **Relatórios**
- **Planilhas com seus relatórios**
- **Seleção de Planilha a compartilhar**
- **Seleção de Relatório a compartilhar**
- **Seleção de Planilhas com relatórios a compartilhar**
- **Editor de Planilhas**
- **Editor de Relatórios**

## 1. Estruturas de Dados

O sistema utilizará a estrutura de dados **record** do Delphi para agrupar variáveis relacionadas.

**TLoginDTO:** Carrega os valores de **Usuario** e **Senha** para validações posteriores, tanto no modo público (comunicação com o servidor) quanto no modo privado (verificação em arquivo JSON local).

**TPlanilha:** Estrutura para representar uma planilha, contendo dados como o nome e o caminho onde o arquivo está salvo.

**TLogEntry:** Representa uma entrada no log de atividades do sistema, contendo o ID do usuário, seu tipo, a ação realizada e o nível de acesso necessário.

## 2. Login e Autenticação

A autenticação de usuários será realizada de forma diferente, dependendo do modo de operação. O usuário escolhe entre **login privado** (para acesso local) ou **público** (para acesso online).

- **Modo Público:** O sistema envia as credenciais para o servidor da empresa (servidor Docker). O servidor valida os dados no banco de dados **PostgreSQL** e, se a conexão e as credenciais forem bem-sucedidas, a view correspondente será retornada para o usuário. Caso o servidor não responda após 5 tentativas, um alerta será exibido informando sobre a falha na conexão.
- **Modo Privado:** As credenciais serão verificadas localmente. A aplicação verificará os dados de usuário e senha em um arquivo **JSON** específico, que não tem conexão com o servidor da empresa.

## 3. Níveis de Acesso e Views

O sistema exibirá diferentes views com base no nível de acesso.

- **view\_admin (View de Administrador, disponível somente em versões internas)**
  - Esta view, bem como seus **controllers, models, views, services e libs**, não estão presentes nas versões disponíveis ao público. Eles existem apenas na versão interna da empresa para fins de gerenciamento e segurança.
  - A **view\_admin** é acessível somente se um login de nível **admin** no banco de dados for bem-sucedido (necessário estar conectado na rede local) no modo público. A validação das credenciais é feita no banco de dados **PostgreSQL**.
  - O administrador pode visualizar e filtrar todas as **Planilhas** e relatórios armazenados no **PostgreSQL**.
  - Após um login de **admin** bem-sucedido, a aplicação enviará um aviso (*ping*) ao servidor da empresa, indicando que a senha de administrador foi acessada.
- **view\_usuario (View de Usuário, disponível em qualquer versão)**
  - Disponível após um login bem-sucedido, seja ele público ou privado.

## 4. Gerenciamento de Tentativas de Login no Modo Público

Se o login falhar, o usuário será instruído a tentar novamente, com um limite de três tentativas. Se o limite for atingido, o programa será fechado.

## 5. Modos de Operação e Dados

O sistema operará em dois modos, com armazenamento de dados distinto.

- **Modo Privado:**
  - Utiliza apenas uma chave privada e armazena os dados em um arquivo **JSON** local e inacessível pelo servidor da empresa.
  - Não há limite de dados, pois o armazenamento é local.

- **Modo Público (Acessível por Link):**
  - Armazena os dados no banco de dados **PostgreSQL** da empresa.
  - Pode ter um limite de 10 **Planilhas** por usuário de acesso público, sendo essa regra gerenciada pela aplicação Delphi e validada pelo servidor.

## 6. Acordos de Usuário e Exportação

Ao logar no modo público, será exibido um termo de responsabilidade legal. O aceite será registrado no banco de dados **PostgreSQL** junto com a data e hora.

- **Funcionalidade para "Não" (Planilhas Públicas):**
  - A versão "Não" não terá acesso direto ao banco de dados da empresa.
  - O sistema fornecerá orientações sobre como exportar e importar dados de **Planilhas** e Relatórios, que serão arquivos **JSON**, permitindo ao usuário gerenciar seus próprios dados localmente.

## 7. Exportação de Dados

O sistema permitirá a exportação de dados de **Planilhas** e Relatórios do banco de dados **PostgreSQL** ou de arquivos **JSON** para um novo arquivo no formato **JSON**.

## 8. Requisitos do Sistema

Os requisitos funcionais e não funcionais permanecem os mesmos, mas com uma adaptação para a utilização de **Delphi 12** e um modelo híbrido de armazenamento de dados (**JSON** local para dados privados e **PostgreSQL** para dados públicos).

- **Requisitos Funcionais e Não Funcionais (RF e RNF):** Todas as operações (Planilhas, relatórios, autenticação, etc.) serão implementadas utilizando a conectividade do **Delphi 12**. A lógica de negócios, validações e regras de acesso serão codificadas na aplicação Delphi. A persistência e a integridade de dados públicos serão garantidas pelo **PostgreSQL**, enquanto a gestão de dados privados será feita por meio de arquivos **JSON**. A segurança de credenciais (hashing com salt) será implementada no Delphi, e as senhas para o modo público serão armazenadas de forma segura no **PostgreSQL**, enquanto as do modo privado estarão no arquivo **JSON** local, protegido por um mecanismo próprio.

## 9. Log de Atividades e Permissões

O sistema de log foi aprimorado para garantir maior segurança e rastreabilidade:

- **Estrutura do Log:** Cada entrada de log terá a seguinte estrutura no banco de dados **PostgreSQL**:
  - **ID de usuário:** Identifica quem realizou a ação.
  - **Tipo de usuário:** Indica se o usuário é **admin** ou **usuario**.

- **Ação:** Descreve a operação realizada (ex.: "Criação de Planilha", "Exclusão de Relatório"). A descrição da ação deve incluir o **nível de acesso necessário** para realizá-la (ex.: "Criação de Planilha (Requer **usuario**)").
- **Regra de Negócio para Permissões:** Antes de salvar qualquer ação no banco de dados, o sistema verificará se o **tipo de usuário** é igual ao **nível de acesso necessário** para a ação. **Se os tipos não corresponderem, a ação não será salva no banco de dados, garantindo que usuários não autorizados não possam registrar atividades que não deveriam ter permissão para realizar.**

## 10. Tipos de Planilhas e Relatórios

### Planilhas

- **Planilha 1: Planilhas:** Planilha que lista todas as planilhas disponíveis no sistema para clicar em cima e realizar ação via botão (editar/excluir/fazer relatório para/associar a relatório/salvar no computador (.json)).
- **Planilha 2: Relatórios:** Planilha que lista todos os relatórios.
- **Planilha 3: Planilhas com seus relatórios:** Uma planilha que mostra quais relatórios (1,2,3,5...) estão associados para quais planilhas (1,2,3,7...).
- **Planilha 4: Seleção de Planilha a compartilhar:** Uma funcionalidade para selecionar uma planilha específica (ou todas, com botão) para baixar junto de seus relatórios associados.
- **Planilha 5: Seleção de Relatório a compartilhar:** Uma funcionalidade para selecionar um relatório específico (ou todos, com botão) para baixar junto de suas planilhas associadas.
- **Outras Planilhas:**
  - **Seleção de Planilhas com relatórios a compartilhar:** Funcionalidade para selecionar um conjunto de planilhas e relatórios para compartilhamento.
  - **Editor de Planilhas:** Interface para edição completa de planilhas.
  - **Editor de Relatórios:** Interface para edição completa de relatórios.

### Relatórios

- **Relatório 1: Nome e tamanho de cada planilha salva no dispositivo:** Exibe o nome e o tamanho de cada planilha salva localmente, com o caminho para fácil acesso.
- **Relatório 2: Quantos dados tem em cada coluna:** Ajuda o usuário a verificar a integridade dos dados, mostrando se todos os campos têm dados preenchidos.
- **Relatório 3: Nome, tamanho, caminho e tipo de relatórios:**
  - **Ordenador:** Permite organizar dados por ordem alfanumérica, por quantidade de colunas ou por mais linhas nulas.
  - **Analítico:** Relatórios focados em análise de dados, como inventário abaixo/acima do estoque mínimo ou desempenho de funcionários.
  - **Gráfico/Visual:** Exemplo de visualização como um Gráfico de Gantt, que mostra a dependência entre tarefas e ajuda a identificar atrasos.

- **Riscos/Alertas:** Identifica riscos, como uma lista de produtos com estoque abaixo do nível mínimo ou contas a receber com pagamentos atrasados há mais de 30 dias.
- **Outros Relatórios:**
  - **Relatório de Auditoria:** Um log detalhado de todas as atividades do sistema para fins de auditoria e segurança.

# Arquivos presentes no projeto com suas explicações

## 1. Estruturas de Dados (DTOs - Data Transfer Objects)

Units que definem os **records** para transportar dados de forma estruturada pelo sistema.

- **TLoginDTO.pas**: Contém a estrutura para armazenar o usuário e a senha durante o processo de autenticação.
- **TUsuarioDTO.pas**: Representa os dados de um usuário logado, como ID, tipo de acesso (admin/usuario) e outras informações relevantes.
- **TPlanilhaDTO.pas**: Define a estrutura para os dados de uma planilha, incluindo seu nome, caminho local e metadados.
- **TRelatorioDTO.pas**: Estrutura para os dados de um relatório, como nome, tipo, e configurações de ordenação ou análise.
- **TLogEntryDTO.pas**: Carrega as informações para uma única entrada de log, como ID do usuário, ação realizada e nível de acesso.
- **TAceiteTermoDTO.pas**: Armazena os dados do aceite do termo de responsabilidade, incluindo o ID do usuário e a data/hora do aceite.

## 2. Camada de Serviço (Lógica de Negócio)

Units responsáveis por orquestrar as regras de negócio e a lógica principal da aplicação.

- **TAuthService.pas**: Centraliza a lógica de autenticação, validando credenciais no modo público (via servidor) ou privado (via JSON) e controlando as tentativas de login.
- **TPlanilhaService.pas**: Gerencia todas as operações de negócio relacionadas a planilhas (CRUD, associações e regras de compartilhamento).
- **TRelatorioService.pas**: Comanda a lógica para geração, gerenciamento e associação de relatórios com as planilhas.
- **TLogService.pas**: Implementa as regras para criação de logs, verificando as permissões do usuário antes de registrar uma atividade.
- **TExportacaoService.pas**: Contém a lógica para exportar planilhas e relatórios do banco de dados ou de arquivos locais para o formato JSON.
- **TConfigService.pas**: Responsável por carregar configurações iniciais, como o usuário de demonstração e a planilha de exemplo para o modo privado.

## 3. Camada de Acesso a Dados (DAO e Handlers)

Units que lidam diretamente com a persistência dos dados, seja no banco de dados ou em arquivos.

- **TDBConnection.pas**: Gerencia a conexão com o banco de dados PostgreSQL, tratando a abertura, fechamento e configuração da comunicação.
- **TPostgresDAO.pas**: Abstrai a comunicação com o PostgreSQL, contendo métodos genéricos para executar consultas de leitura e escrita.

- **TJSONDataHandler.pas:** Especializada na manipulação (leitura e escrita) de arquivos JSON para todas as operações do modo privado.
- **TSecurityDataHandler.pas:** Isola as funções de segurança de dados, como criptografia (hashing com salt) das senhas, tanto para o modo público quanto para o privado.

#### 4. Interface do Usuário (Views/Formulários)

Units que representam as telas (formulários) com as quais o usuário interage.

- **TViewLogin.pas:** Formulário inicial para o usuário inserir suas credenciais e escolher entre o login público ou privado.
- **TViewPrincipal.pas:** Tela principal do sistema após o login, servindo como painel de acesso às funcionalidades de planilhas e relatórios.
- **TViewEditorPlanilha.pas:** Interface visual para o usuário criar, visualizar e editar o conteúdo de uma planilha.
- **TViewEditorRelatorio.pas:** Formulário para a criação e configuração dos diferentes tipos de relatórios.
- **TViewGerenciadorDados.pas:** Tela unificada para listar e gerenciar planilhas, relatórios e suas associações.
- **TViewCompartilhamento.pas:** Interface dedicada para selecionar planilhas e/ou relatórios para a exportação/compartilhamento.
- **TViewVisualizadorRelatorio.pas:** Janela para exibir os relatórios gerados, sejam eles analíticos, de auditoria ou gráficos.
- **TViewModalTermos.pas:** Janela modal que exibe o termo de responsabilidade legal para o aceite do usuário no modo público.

#### 5. Utilitários e Módulos de Suporte

Units com funções de apoio e recursos reutilizáveis em todo o projeto.

- **TAppConstants.pas:** Define constantes globais, como mensagens de erro, limites de tentativas, nomes de arquivos padrão e chaves de configuração.
- **TAppUtils.pas:** Biblioteca com funções utilitárias genéricas, como formatação de datas, validação de campos e conversões de dados.
- **TServerPinger.pas:** Contém a funcionalidade específica para enviar o "ping" ao servidor da empresa quando um login de administrador é bem-sucedido.

