

# Sistema de Planilhas e Relatórios em Delphi 12 e PostgreSQL

## Objetivo Principal

Desenvolver um sistema robusto para a criação e gerenciamento de **Planilhas** (que incluem operações de Create, Read, Update e Delete) e a geração de 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, para permitir que o usuário explore a aplicação de forma privada.

## 1. Estruturas de Dados

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

## 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. 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.**

No log tem que ter ID de usuario, tipo (admin ou usuario), Ação tem que ter nível de acesso necessário pra fazer, se tipo não for igual a nível de acesso necessário, não pode ir pro banco de dados

Pessoa manda pro servidor, servidor valida, vai devolta pro usuario, vai pro banco de dados se dentro das regras de negócio