

## TUTORIAL DE CONFIGURAÇÃO DO BD

---

### Guia Definitivo do Projeto SBD: Configuração Local com pgAdmin 4



**Objetivo:** Ao final deste guia, cada membro do grupo terá um ambiente de banco de dados PostgreSQL idêntico em seu próprio computador, com a estrutura e os dados do projeto prontos para a análise.

**Regra de Ouro:** Para que o trabalho em grupo funcione, todos devem seguir estes passos exatamente da mesma forma.

---

### Parte 1: Pré-requisitos (Instalação e Organização) - Caso ainda não tenha instalado para as aulas de SBD

#### 1.1 - Instalar o PostgreSQL Server

1. **Download:** Baixe o instalador do PostgreSQL (versão 14 ou mais recente) a partir deste site oficial:  
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>
2. **Instalação:**
  - Execute o instalador.
  - Na tela de "Select Components", deixe marcadas as opções:
    - PostgreSQL Server
    - pgAdmin 4
    - Command Line Tools
  - Na tela de senha, o grupo deve **combinar e usar a mesma senha simples** (ex: **postgres**) para o superusuário **postgres**. Anotem essa senha.
  - Mantenha as outras opções padrão e conclua a instalação.

#### 1.2 - Organizar os Arquivos CSV

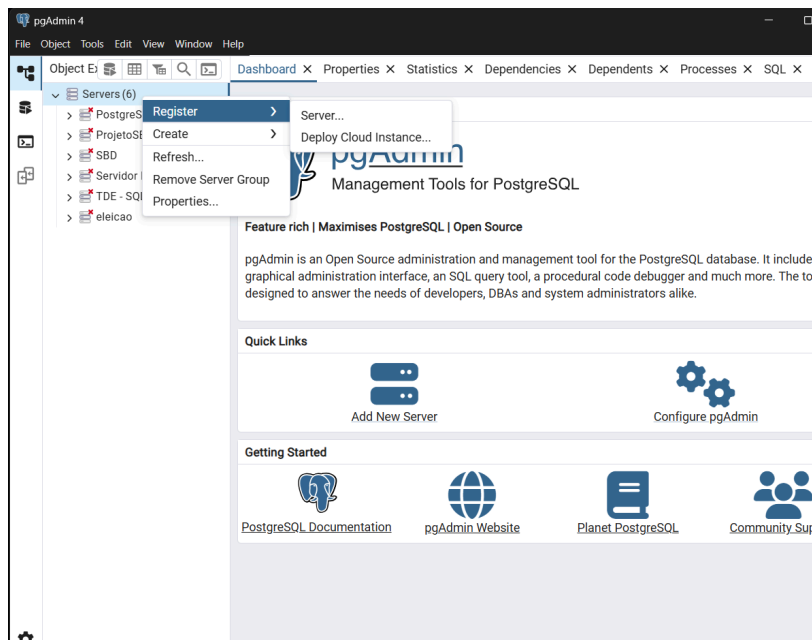
1. No seu computador, vá para o **Disco Local (C:)**.
2. Crie uma nova pasta com o nome **dados**.
3. Coloque todos os seus arquivos **.csv** do Bolsa Família dentro desta pasta **C:\dados**.

**OBS: a pasta contendo as planilhas está no repositório do github**

## Parte 2: Configuração no pgAdmin 4

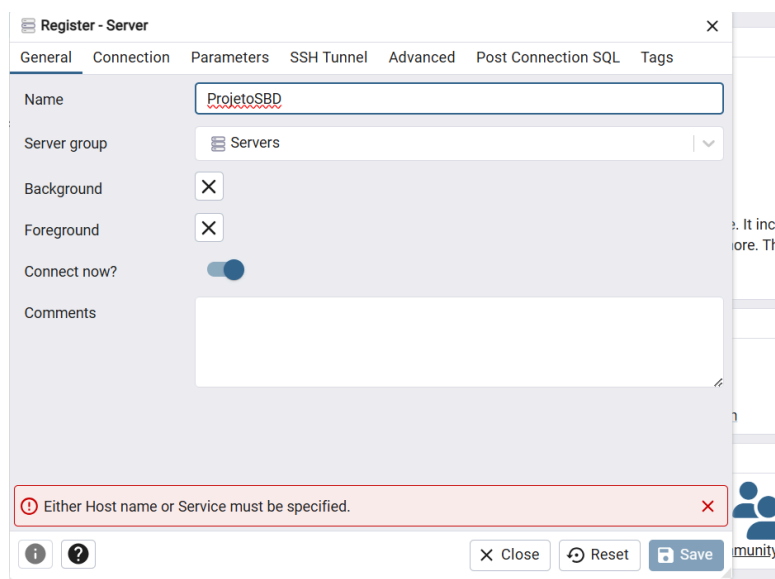
### 2.1 - Conectar ao Servidor Local

1. Abra o **pgAdmin 4** (procure no Menu Iniciar).
2. No painel "Browser" à esquerda, clique com o botão direito em "**Servers**" > "**Register**" > "**Server...**".



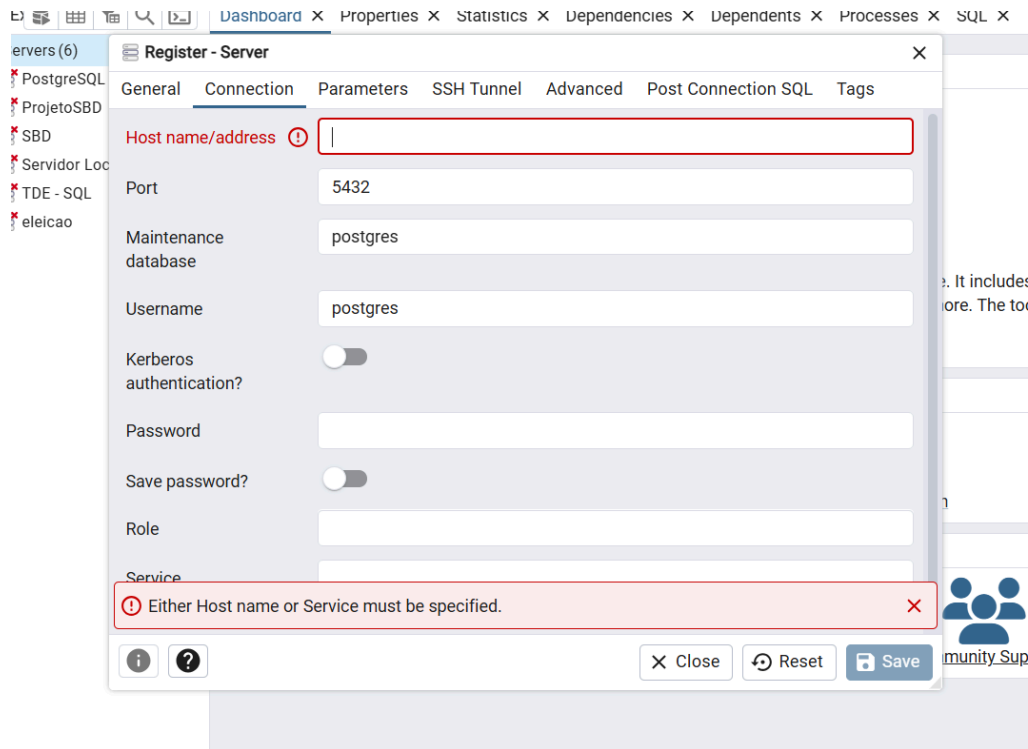
3. Na janela que se abrir, preencha as abas:

- **Aba "General"**: Dê um nome para a conexão, como **ProjetoSBD**.



- **Aba "Connection":**
  - **Host name/address:** localhost
  - **Username:** postgres
  - **Password:** Digite a senha: 628507

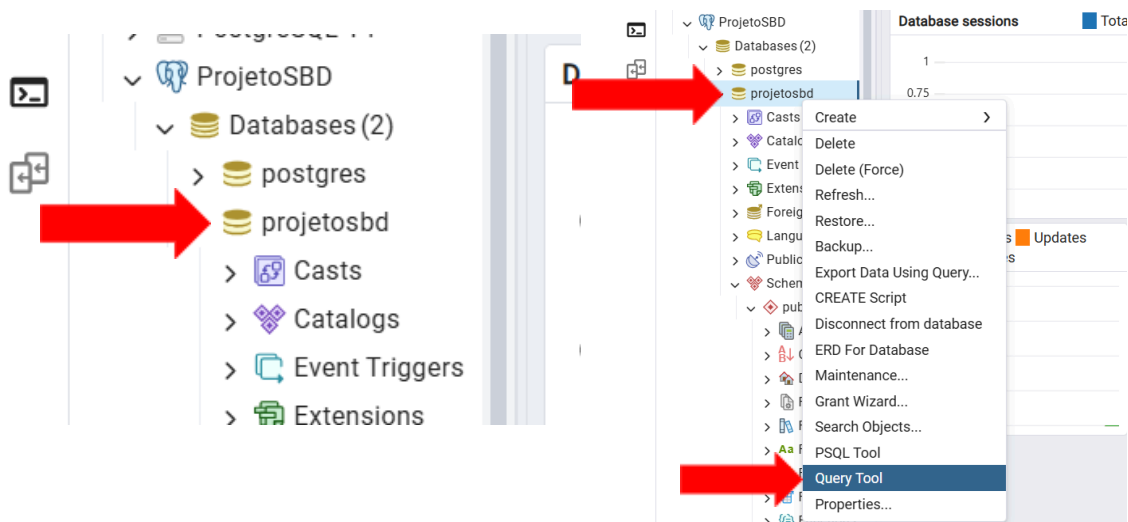
4. Clique em **"Save"**. Sua conexão aparecerá no painel à esquerda.



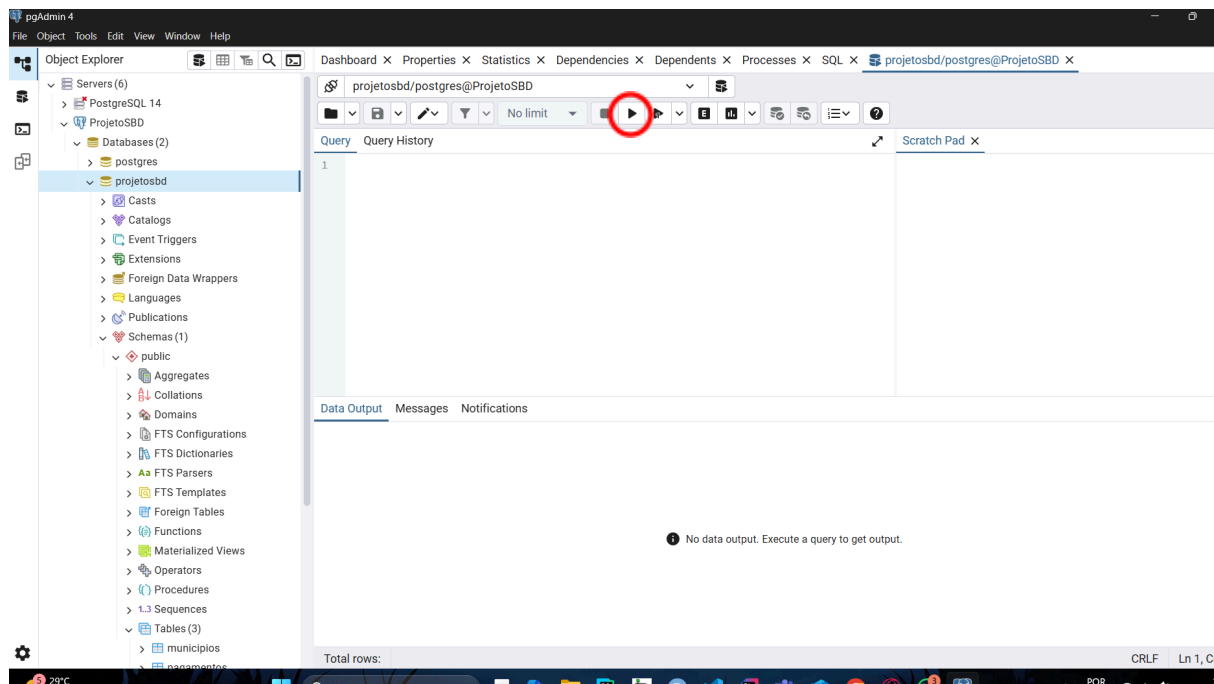
## 2.2 - Script de Criação e Reset do Banco de Dados

Este script cria a base de dados do projeto e todas as tabelas necessárias. Ele pode ser executado várias vezes, pois sempre limpa a estrutura antiga antes de criar a nova.

1. No painel "Browser", clique com o botão direito no seu servidor (**ProjetoSBD Local**) e escolha **"Query Tool"**.



Uma janela de script se abrirá. Cole e execute o script completo abaixo (clique no ícone de raio play ou aperte F5).



**Copie e cole o script em verde abaixo e execute-o nessa página acima**

-- SCRIPT DE RESET TOTAL DO PROJETO SBD

-- Cria a base de dados do projeto, se ela não existir

-- Se este comando falhar porque a base já existe, não tem problema, continue.

CREATE DATABASE projetosbd;

-- A PARTIR DAQUI, EXECUTE CONECTADO À BASE 'projetosbd'

-- Apaga as tabelas antigas na ordem correta, se existirem

DROP TABLE IF EXISTS pagamentos;

DROP TABLE IF EXISTS pessoa;

DROP TABLE IF EXISTS municipios;

DROP TABLE IF EXISTS staging\_pagamentos;

-- Cria as tabelas finais

CREATE TABLE municipios ( codigo\_siafi INTEGER PRIMARY KEY, nome VARCHAR(100) NOT NULL, uf CHAR(2) NOT NULL );

CREATE TABLE pessoa ( nis BIGINT PRIMARY KEY, nome VARCHAR(150) NOT NULL, cpf VARCHAR(14) );

```
CREATE TABLE pagamentos ( id_pagamento SERIAL PRIMARY KEY, mes_competencia
DATE NOT NULL, mes_referencia DATE NOT NULL, municipio_siafi INTEGER NOT
NULL, pessoa_nis BIGINT NOT NULL, valor_parcela DECIMAL(10, 2) NOT NULL,
CONSTRAINT fk_pessoa FOREIGN KEY(pessoa_nis) REFERENCES pessoa(nis) ON
DELETE RESTRICT ON UPDATE CASCADE, CONSTRAINT fk_municipio FOREIGN
KEY(municipio_siafi) REFERENCES municipios(codigo_siafi) ON DELETE RESTRICT
ON UPDATE CASCADE );
```

-- Cria a tabela de passagem com TODAS as colunas como TEXTO

```
CREATE TABLE staging_pagamentos ( mes_competencia TEXT, mes_referencia TEXT, uf
TEXT, codigo_municipio_siafi TEXT, nome_municipio TEXT, cpf_favorecido TEXT,
nis_favorecido TEXT, nome_favorecido TEXT, valor_parcela TEXT );
```

---

### Parte 3: Carga dos Dados dos Arquivos CSV

**Atenção: Neste processo deve ser feito um arquivo de cada vez.**

#### 1. Inicie a Importação:

- No painel "Browser" à esquerda, encontre a tabela **staging\_pagamentos** (dentro de **projetosbd > Schemas > public > Tables**).
- Clique com o botão direito nela e escolha **"Import/Export..."**.

#### 2. Configure o Assistente:

- Mude o botão no topo para **"Import"**.
- **Filename:** Clique no ícone da pasta e selecione o primeiro arquivo (ex: **C:\dados\202401\_NovoBolsaFamilia.csv**).
- **Format:** Escolha **csv**.
- **Encoding:** Escolha **LATIN1**.
- Vá para a aba **"Options"**:
  - Ative o botão **Header**.
  - Escolha o **Delimiter** como **;** (ponto e vírgula).
- Clique em **"OK"**. Aguarde a mensagem de sucesso "Process completed".

#### 3. Repita para os Outros Arquivos: Repita os passos 1 e 2 desta seção para todos os outros arquivos CSV.

---

## Parte 4: Migração Final dos Dados

Este é o último passo, que move os dados da tabela temporária para as tabelas finais.

1. Abra uma nova **"Query Tool"** conectada à base de dados **projetosbd**, que foi mostrado como se faz passos acima.
2. Cole e execute o script completo abaixo.

**ATENÇÃO:** Este é o passo que pode levar **várias horas**. Execute o script e garanta que o computador não suspenda ou desligue, ou seja, coloque um video de chuva de 10 horas no YouTube (essa parte levou 5h47min comigo).

-- Popula as tabelas finais a partir dos dados brutos

```
INSERT INTO municipios (codigo_siafi, nome, uf) SELECT DISTINCT  
CAST(codigo_municipio_siafi AS INTEGER), nome_municipio, uf FROM  
staging_pagamentos ON CONFLICT (codigo_siafi) DO NOTHING;
```

```
INSERT INTO pessoa (nis, nome, cpf) SELECT DISTINCT CAST(nis_favorecido AS  
BIGINT), nome_favorecido, cpf_favorecido FROM staging_pagamentos WHERE  
nis_favorecido IS NOT NULL AND nis_favorecido <> " ON CONFLICT (nis) DO  
NOTHING;
```

```
INSERT INTO pagamentos (mes_competencia, mes_referencia, municipio_siafi, pessoa_nis,  
valor_parcela) SELECT TO_DATE(mes_competencia, 'YYYY-MM-DD'),  
TO_DATE(mes_referencia, 'YYYY-MM-DD'), CAST(codigo_municipio_siafi AS  
INTEGER), CAST(nis_favorecido AS BIGINT), CAST(REPLACE(valor_parcela, ',', '.') AS  
DECIMAL(10, 2)) FROM staging_pagamentos WHERE nis_favorecido IS NOT NULL  
AND nis_favorecido <> ";
```

-- Limpa a tabela de passagem que não é mais necessária

```
DROP TABLE staging_pagamentos;
```

4. Após a conclusão (que pode levar horas), o banco de dados estará pronto.