

Guia Completo: Instalação e Configuração do PostgreSQL para a API GestorPropostas

Versão: 1.0

Data: Novembro 2025

Objetivo: Instruções passo a passo para instalar PostgreSQL, criar banco de dados e configurar a API.

Índice

1. Pré-requisitos
2. Instalar PostgreSQL e pgAdmin
3. Criar o Banco de Dados
4. Configurar Connection String
5. Aplicar Migrations
6. Executar e Testar a API
7. Troubleshooting

Pré-requisitos

Antes de começar, certifique-se de ter instalado:

- Windows 10/11 (ou outro SO compatível)
- .NET 8 SDK - Download: [dotnet.microsoft.com](#)
- Visual Studio 2022 (Community, Professional ou Enterprise) - Download: [visualstudio.microsoft.com](#)
- Git (para clonar o repositório) - Download: [git-scm.com](#)
- Projeto clonado da API via GitHub

Instalar PostgreSQL e pgAdmin

Passo 1: Baixar o PostgreSQL

1. Acesse o site oficial: [www.postgresql.org/download/windows](#)
2. Clique em **Download the installer** para Windows.
3. Escolha a versão mais recente (ex: PostgreSQL 16 ou superior).
4. Salve o arquivo **.exe** em seu computador.

Passo 2: Executar o Instalador

1. Abra o arquivo baixado (duplo clique).
2. Clique em **Next** para prosseguir pela tela inicial.
3. Escolha a pasta de destino (deixe padrão se não souber) e clique **Next**.
4. Selecione componentes:
 - PostgreSQL Server (obrigatório)
 - pgAdmin 4 (interface gráfica - recomendado)
 - Command Line Tools (opcional)
5. Clique **Next** e escolha a pasta de dados (deixe padrão).
6. Clique **Next** até chegar à **Tela de Senha**.

Passo 3: Definir Senha do Usuário PostgreSQL

1. Na tela **Password**, defina uma senha para o usuário padrão **postgres**.
 - Exemplo: Jhoue9883* (anote bem, será necessária depois)
2. **Confirme a senha** digitando novamente.
3. Clique **Next**.

Passo 4: Confirmar Porta

1. A porta padrão é **5432** - deixe assim, a menos que tenha conflito.
2. Clique **Next** e depois **Next** novamente nas telas de configuração.
3. Clique **Finish** para concluir a instalação.

Passo 5: Verificar Instalação

1. Abra o **pgAdmin 4** (deve ter atalho no desktop ou no menu Iniciar).
2. Na primeira execução, ele abre um navegador e solicita criar uma senha master (para acessar o pgAdmin).
3. Defina a senha master e clique **OK**.
4. Pronto! O pgAdmin está pronto para usar.

Criar o Banco de Dados

Passo 1: Conectar ao Servidor no pgAdmin

1. Abra o **pgAdmin 4** novamente (ou acesse `http://localhost:5050` no navegador se já estiver fechado).
2. No painel à esquerda, você verá **Servers** - expanda clicando na seta.
3. Você verá **PostgreSQL 16** (ou outra versão).
4. Clique duas vezes em **PostgreSQL 16** para conectar usando o usuário `postgres` e a senha definida na instalação.

Passo 2: Criar Novo Banco

1. No painel esquerdo, clique com o botão **direito** em **Databases**.
2. Selecione **Create → Database...** (ou apenas **Database**).
3. Uma janela ou abas aparecem. Na aba **General**, preencha:
 - **Database name:** `gestorpropostas`
 - **Owner:** `postgres` (deixe padrão)
4. Clique na aba **Definition** (opcional - não precisa alterar nada).
5. Clique em **Save** ou **Create**.

Passo 3: Confirmar Criação

1. Volte ao painel esquerdo e expanda **Databases**.
2. Você deve ver o banco `gestorpropostas` listado.
3. Banco criado com sucesso! ☺

Configurar Connection String

Passo 1: Abrir o Projeto no Visual Studio

1. Abra o Visual Studio.
2. Abra o projeto da API (File → Open → Folder / Solution).
3. Localize o arquivo `appsettings.json` na raiz do projeto Web API.

Passo 2: Editar a Connection String

1. Abra o arquivo `appsettings.json` (duplo clique).
2. Procure pela seção "ConnectionStrings".
3. Dentro dessa seção, altere ou adicione a chave "PostgresConnection":

```
"ConnectionStrings": {  
    "PostgresConnection": "Host=localhost;Port=5432;Database=gestorpropostas;Username=postgres;Password=SUAENHA"  
}
```

Substituições necessárias:

- `SUAENHA` → senha definida na instalação do PostgreSQL (ex: `Jhoue9883*`)
- `localhost` → IP do servidor (se for remoto, mude para o IP real)
- `5432` → porta (não alterar, a menos que tenha mudado na instalação)

Passo 3: Salvar

1. Pressione **Ctrl+S** para salvar o arquivo.
2. Pronto! A conexão está configurada.

Aplicar Migrations

As migrations criam as tabelas no banco vazio. Existem dois métodos:

Método 1: Usando Package Manager Console (Visual Studio) - Recomendado

1. No Visual Studio, vá ao menu **Tools → NuGet Package Manager → Package Manager Console**.
2. Uma aba inferior abre com um campo de entrada (linha de comando).
3. No topo da aba, há um campo **Default project** - certifique-se de que está selecionado o projeto **Web API** (onde está o `GestorPropostasDbContext`).
4. Na linha de comando, digite:

Update-Database

5. Pressione **Enter** e aguarde.
6. Você verá mensagens como:
 - Build started...
 - Applying migration '...._InitialCreate'
 - Done.

Se tudo funcionar, as tabelas foram criadas no PostgreSQL. ☺

Método 2: Usando .NET CLI (Terminal)

1. Abra um terminal (Command Prompt ou PowerShell).
2. Navegue até a pasta raiz do projeto Web API:

```
cd C:\caminho\para\seu\projeto\GestorProposta
```

3. Digite:

```
dotnet ef database update
```

4. Pressione **Enter** e aguarde o resultado.

Executar e Testar a API

Passo 1: Definir Startup Project

1. No Visual Studio, clique com o **botão direito** no projeto Web API.
2. Selecione **Set as Startup Project**.
3. O nome do projeto deve aparecer em **negrito** na árvore de soluções.

Passo 2: Executar a API

1. Pressione **F5** (ou Debug → Start Debugging).
2. O Visual Studio compila e executa a API.
3. Uma janela do navegador abre automaticamente com o **Swagger UI**.
 - URL será algo como: <https://localhost:7197/swagger> (porta varia)

Passo 3: Testar um Endpoint

1. No Swagger UI, procure pela seção **Propostas**.
2. Clique em **POST /api/Propostas** (verde).
3. Clique em **Try it out**.
4. Preencha o JSON com dados de teste:

```
{
  "clienteNome": "João Silva",
  "clienteEmpresa": "Tech Solutions",
  "clienteEmail": "joao@email.com",
  "valor": 5000,
  "prazo": "2025-12-31",
  "descricao": "Projeto de automação"
}
```

5. Clique em **Execute**.
6. Se aparecer **201 Created** ou **200 OK**, a API e banco funcionam! ☺

Troubleshooting

Erro: "relação 'Propostas' não existe"

Causa: As migrations não foram aplicadas.

Solução: Execute `Update-Database` novamente no Package Manager Console.

Erro: "Connection refused" ou "host is down"

Causa: PostgreSQL não está rodando ou porta está bloqueada.

Solução:

1. Abra o **Services** (services.msc no Windows).
 2. Procure por **postgresql-x64-16** (ou versão similar).
 3. Clique com botão direito e escolha **Start** (iniciar).
-

Erro: "Password authentication failed"

Causa: Senha incorreta na connection string.

Solução:

1. Verifique a senha definida na instalação.
 2. Atualize o `appsettings.json` com a senha correta.
 3. Salve e tente novamente.
-

Erro: "Database already exists"

Causa: Banco já foi criado.

Solução: Use o banco existente, ou delete-o e recrie (via pgAdmin) antes de rodar `Update-Database`.

Próximas Etapas

Após confirmar que a API está funcionando:

1. **Documentação da API:** Explore o Swagger para entender todos os endpoints.
 2. **Testes Avançados:** Teste todos os CRUD (Create, Read, Update, Delete).
 3. **Deploy:** Se precisar colocar em produção, consulte a documentação de deploy.
-

Contato e Suporte

Se surgir dúvida durante o processo, consulte:

- [PostgreSQL Documentation](#)
 - [Entity Framework Core Migrations](#)
 - Documentação do projeto no repositório GitHub
-

Fim do Guia

Versão 1.0 | Novembro 2025