

Este é o manual de instalação e configuração do Projeto Vacina, trabalho de WEB de implementação de agenda de vacinação. O projeto foi desenvolvido no SO Windows 11.

## 1. Instalações

Instale normalmente no seu sistema os seguintes programas, a versão mais recente:

- DBeaver Community: <https://dbeaver.io/download/>
- Docker Desktop: <https://www.docker.com/products/docker-desktop/>
- Postman: <https://www.postman.com/downloads/>
- Github Desktop: <https://desktop.github.com/>
- Navegador. Sugiro Chrome ou Brave.
- IDE. Sugiro o VScode.

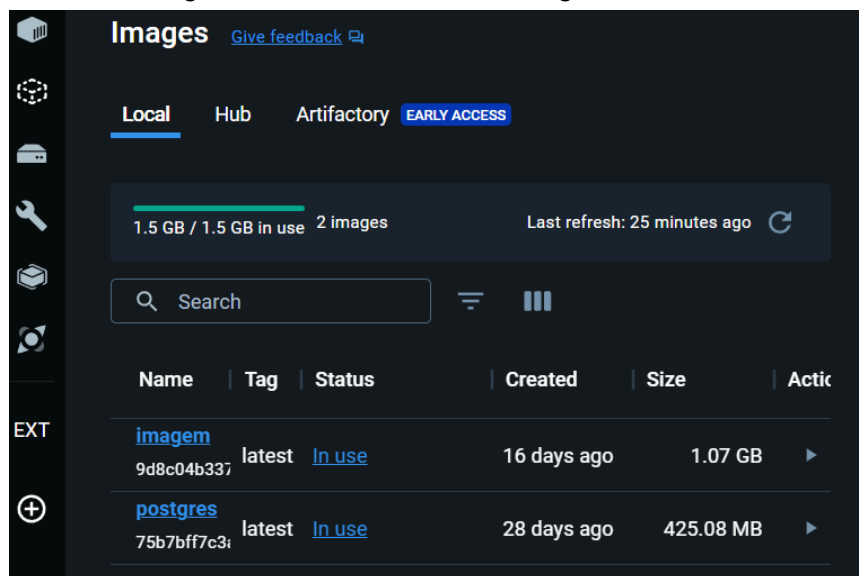
## 2. Configurações

### a. Docker Desktop

O sistema todo foi containerizado usando o Docker. Tanto o servidor de aplicação, quanto o de persistência possuem seus próprios contêineres, de forma que todas as dependências já estarão neles. Procedamos com a configuração dos contêineres.

#### Contêiner de Aplicação

1. Crie uma pasta em qualquer lugar chamada **projeto\_vacina**
2. Coloque dentro dela o seguinte arquivo Dockerfile: [link](#)
3. Abra o Powershell dentro da pasta e execute o seguinte comando:  
"docker build -t imagem ."
4. Criada a imagem do webserver PHP.
5. Abra o Docker Desktop, e vá no menu Images, como mostrado abaixo. A imagem criada terá o nome "imagem".



6. Clique em Run na coluna Actions. Uma janela se abrirá. Clique em Optional Settings.

7. Preencha a janela igual ao mostrado abaixo, e dê **Run**.

**Run a new container**  
imagem:latest

**Optional settings** ^

Container name

A random name is generated if you do not provide one.

**Ports**  
Enter "0" to assign randomly generated host ports.

Host port  
 :80/tcp

**Volumes**

Host path

Container path

**Environment variables**

Variable

Value

**Buttons:** Cancel, Run

Atenção: em **Volumes**, selecione a pasta **projeto\_vacina** que criamos no início.

8. Criado o contêiner de aplicação.

### Contêiner de Persistência

1. Abra o Powershell dentro da pasta **projeto\_vacina**.
2. Execute o seguinte comando:

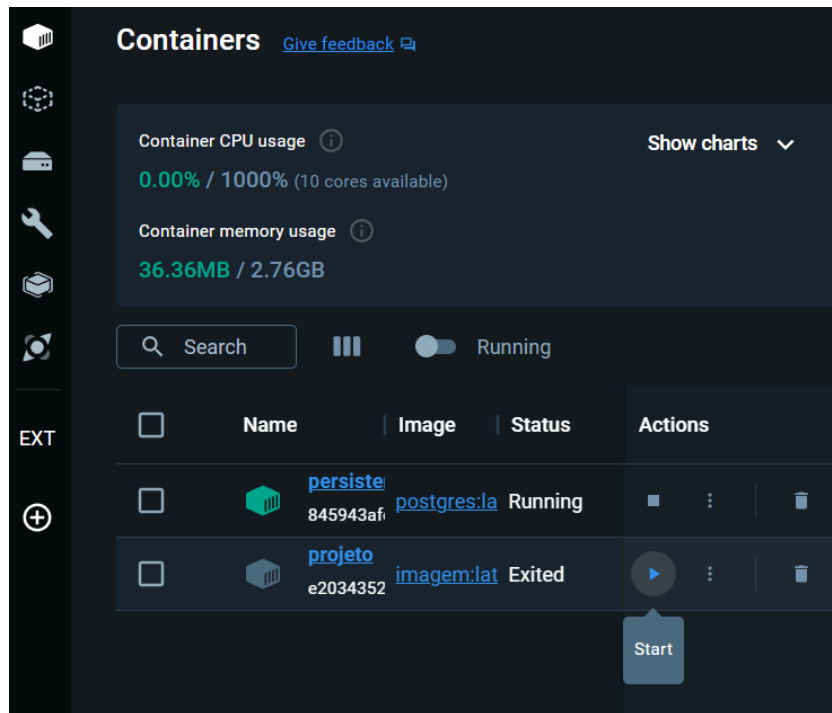
```
"docker run -d --name persistencia -p 5432:5432 -e  
POSTGRES_PASSWORD=12345 -e POSTGRES_USER=postgres -e  
POSTGRES_DB=postgres postgres:latest"
```

Aguarde até a criação do contêiner PostgreSQL. Após finalizar, execute os seguintes comandos:

```
"docker network create minharede"  
"docker network connect minharede persistencia"  
"docker network connect minharede projeto"
```

3. Agora o contêiner **persistencia** está criado, e colocado na mesma rede do contêiner **projeto**.

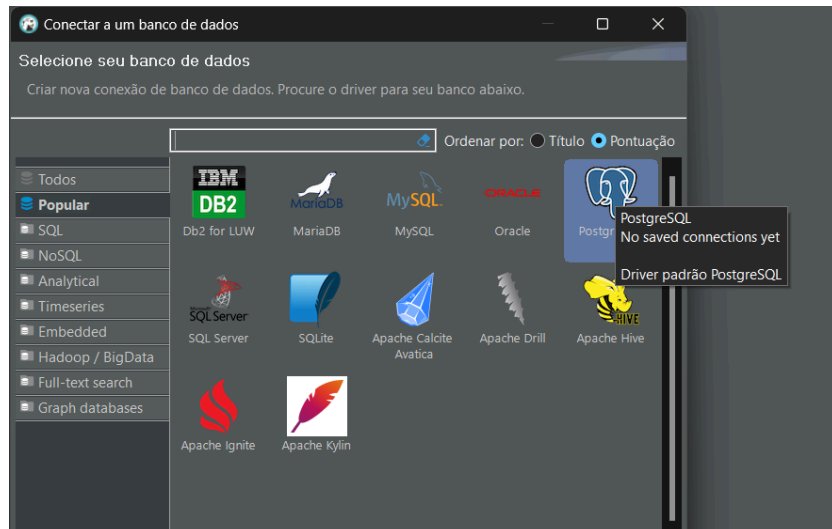
4. Abra o Docker Desktop, navegue até a aba Containers, e dê Start em ambos:



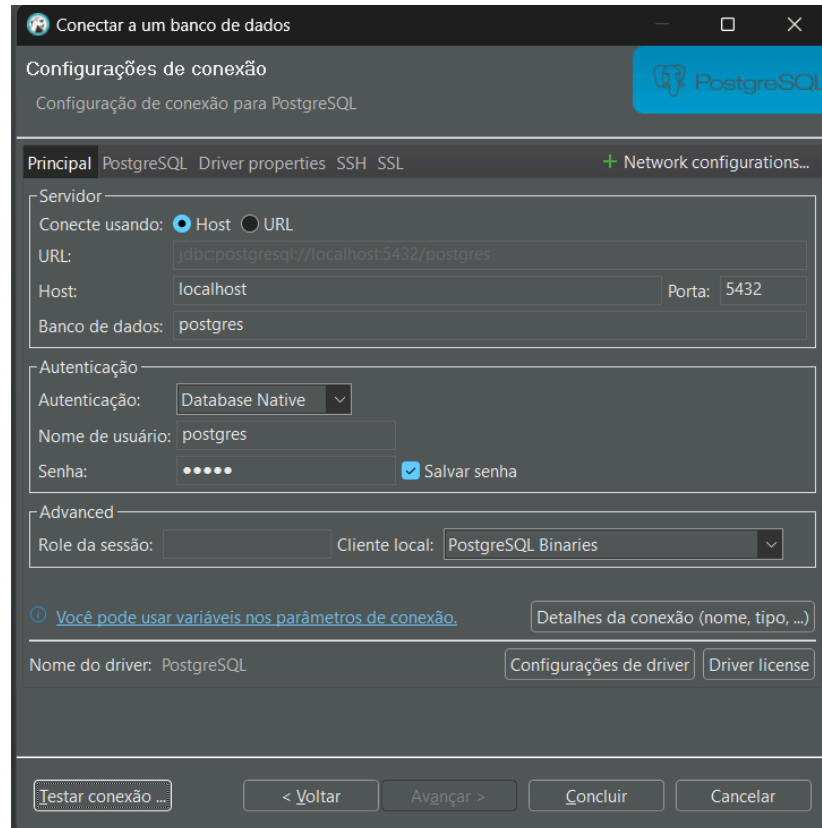
5. Executados os contêineres, deixe o Docker executando em segundo plano.

## b. DBeaver

- Clique em **Nova Conexão**, selecione **PostgreSQL**, e clique em **Avançar**:



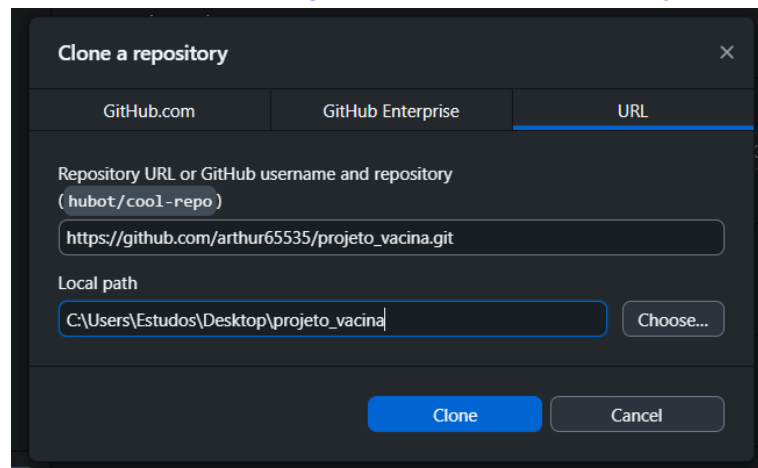
- Preencha de acordo com a imagem. No campo Senha, forneça “12345”, sem aspas, e clique em **Concluir**.



- Criada a conexão **postgres**, vamos agora clonar o projeto.

### c. Github Desktop e clonagem do projeto

- Abra o Github desktop, clique em File, e depois em Clone repository.
- Clone por URL: [https://github.com/arthur65535/projeto\\_vacina.git](https://github.com/arthur65535/projeto_vacina.git)

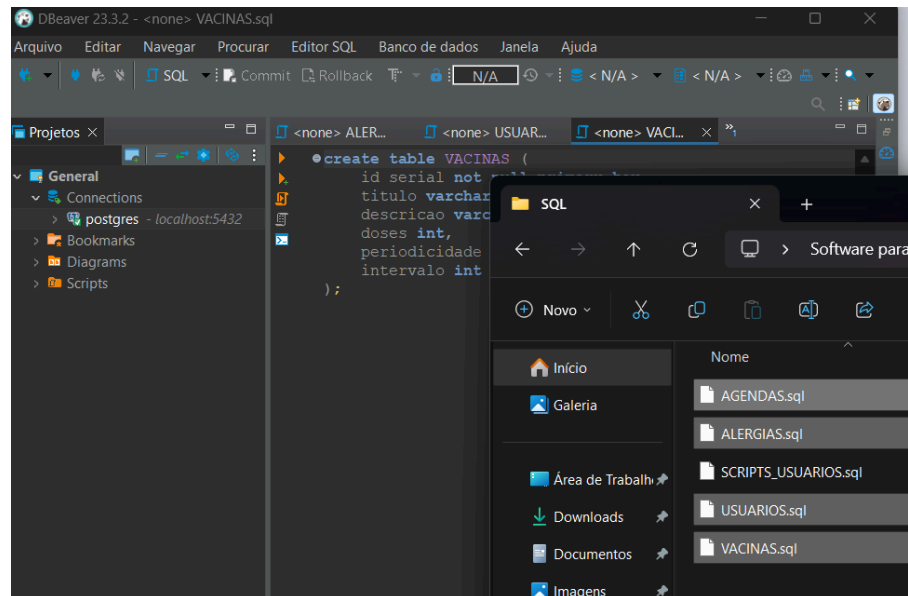


- Em Local Path escolha a pasta **projeto\_vacina** que criamos no início, e dê Clone.

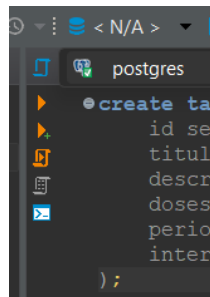
### 3. Execução

#### a. Criando as tabelas

- Antes entrar no sistema, temos que criar as tabelas que ele usará.
- Abra o DBeaver, entre na pasta SQL, dentro de **projeto\_vacina**, e arraste os 4 arquivos SQL para dentro do DBeaver, como ilustrado a seguir:



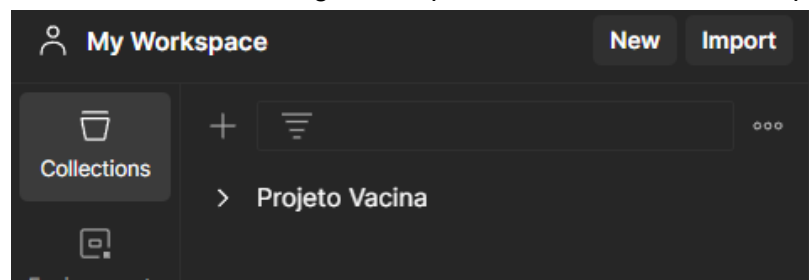
- Em seguida, eles se abrirão. Para cada um deles, clique em “N/A” e em seguida em **postgres**, para atribuímos uma conexão.



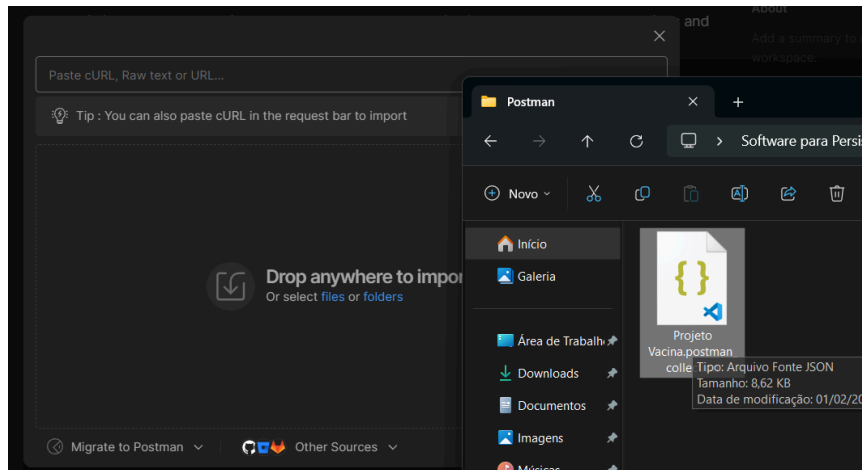
- Por fim, execute os quatro scripts. As tabelas serão criadas.

#### b. API e Backend

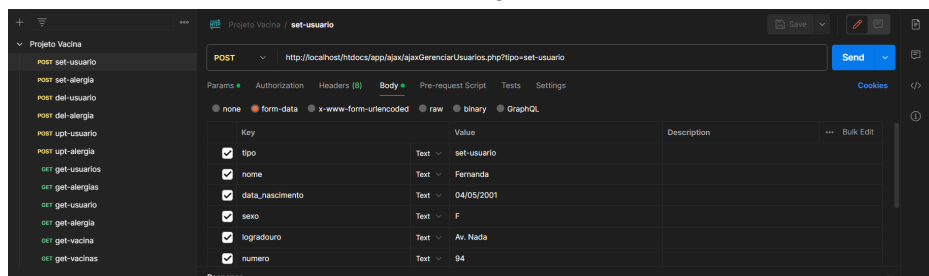
- Podemos experimentar as rotas do sistema pelo Postman API. Incluí uma pasta chamada “Postman” na raiz do projeto com um arquivo de coleção de requests que escrevi.
- Abra o Postman, em seguida clique em Collections, e em Import:



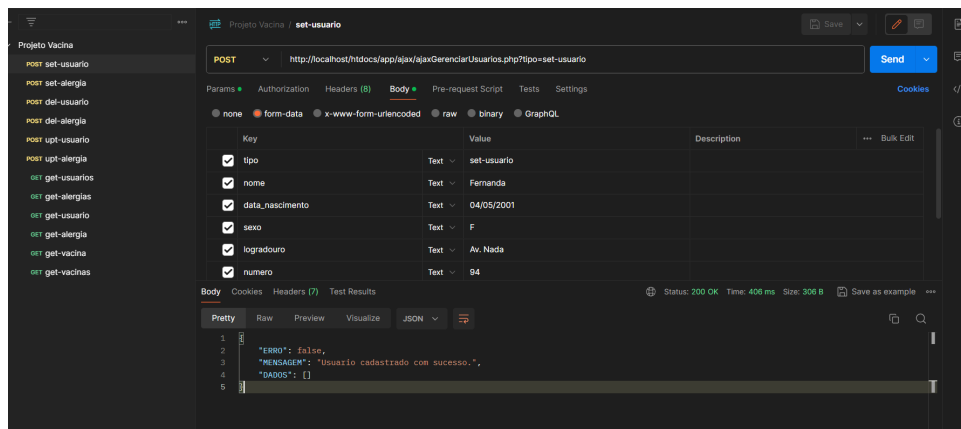
- Na janela que se abriu, arraste para ela o arquivo collection:



- O Postman carregará nossa coletânea de requests, e a partir daí, experimentar todas as rotas da aplicação.



- Basta selecionar a request desejada, e clicar em Send. O resultado do backend será impresso no console logo abaixo.
- Deixei todas as requisições salvas com dados pré-preenchidos para conveniência, mas a modificação dos valores é livre!
- Demonstração de cadastro de usuário:



## c. Frontend

The image displays the frontend of the 'Projeto Vacina' application. The top navigation bar includes the logo, the name 'Projeto Vacina', and links to 'Home', 'Gerenciar Agendas', 'Gerenciar Alergias', 'Gerenciar Usuários', and 'Gerenciar Vacinas'. A 'Sair' button is located in the top right corner. The main content area is divided into four columns, each with a form:

- Cadastrar Usuário:** Fields for Nome, Data de Nascimento, Sexo, Logradouro, Número, Setor, Cidade, and UF. A 'Cadastrar' button is at the bottom.
- Agendar Vacinação:** Fields for Data da Vacinação (with a calendar icon) and a dropdown for 'Vacina' with the label 'Selecione uma vacina'. A 'Cadastrar' button is at the bottom.
- Cadastrar Alergia:** A single 'Título' field and a 'Cadastrar' button.
- Cadastrar Vacina:** Fields for 'Título', 'Descrição', 'Doses', 'Periodicidade', and 'Intervalo'. A 'Cadastrar' button is at the bottom.

Below the main content area, there is a large green rectangular block containing a white login form. The login form has the 'Projeto Vacina' logo and name at the top, followed by the title 'Login'. It includes two input fields labeled 'Email' and 'Senha', and a 'Login' button at the bottom.

- No navegador, cole “<http://localhost/htdocs/app/>” na barra de pesquisa.
- Insira um login e senha fictícios, e clique em Login.
- Na página Home, pode-se cadastrar usuários, agendas, alergias e vacinas à vontade.
- Nas páginas Gerenciar Agendas, Gerenciar Vacinas, Gerenciar Usuários, e Gerenciar Alergias eram para aparecer tabelas dinâmicas com todos os dados do BD, porém..
- **Eu apanhei do front-end até o prazo final e não consegui renderizar as datatables, peço sinceras desculpas.**