

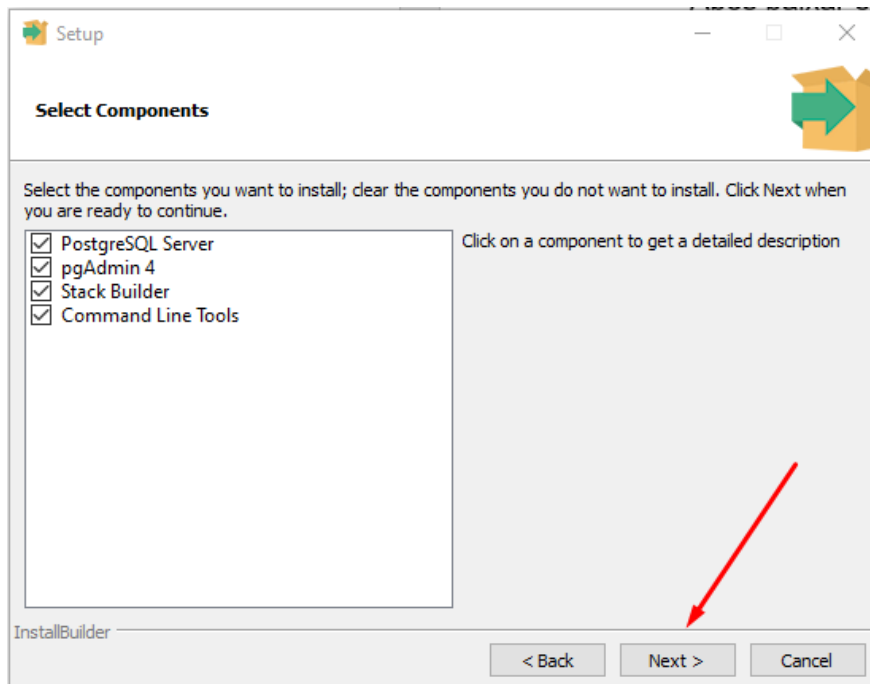
Manual de configuração do ambiente de desenvolvimento

Sumário

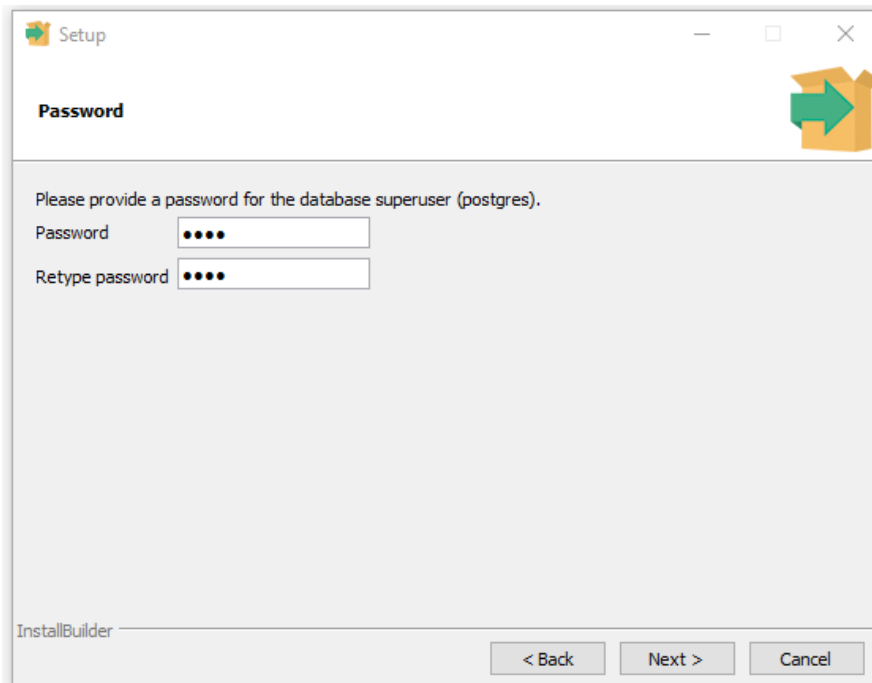
Instalação do postgres	2
Configuração do banco no visual Studio	5
Configuração de execução da primeira vez da aplicação Postman	10
Inserção/Alteração e Seleção por Status	12
Logs	12

Instalação do postgres

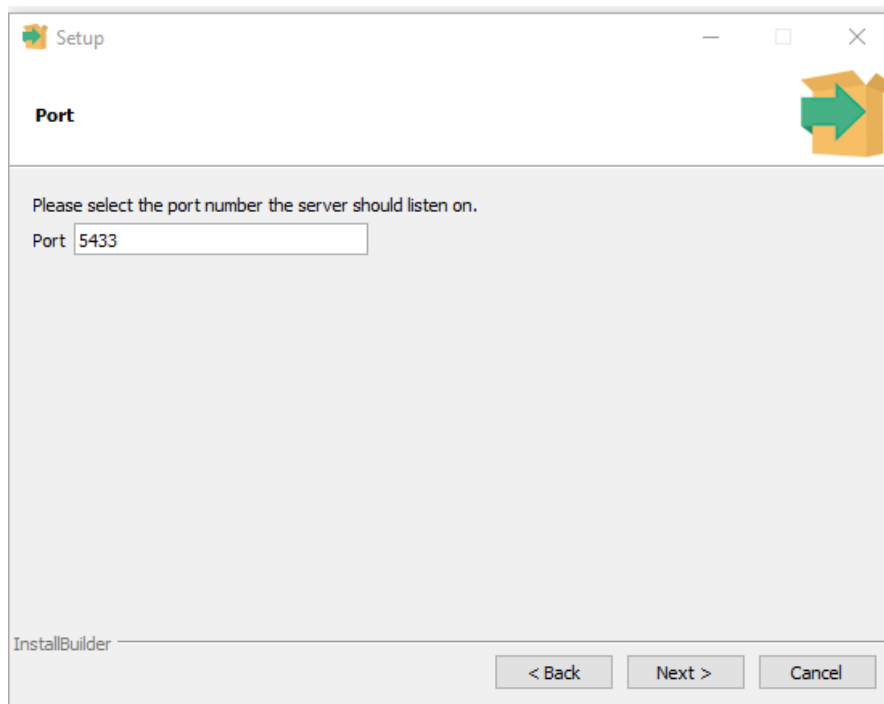
Após baixar o postgres no link: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>



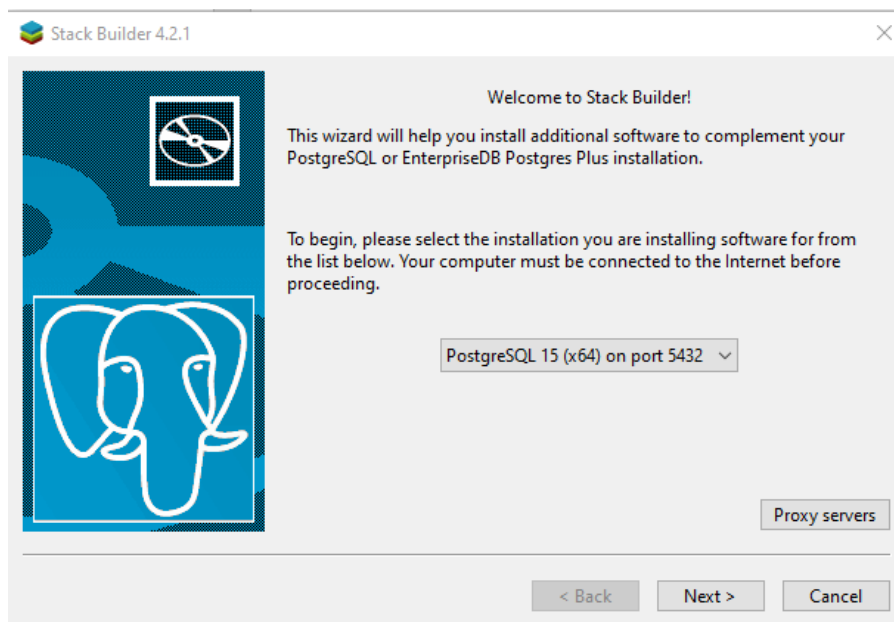
Insira uma senha desejada:



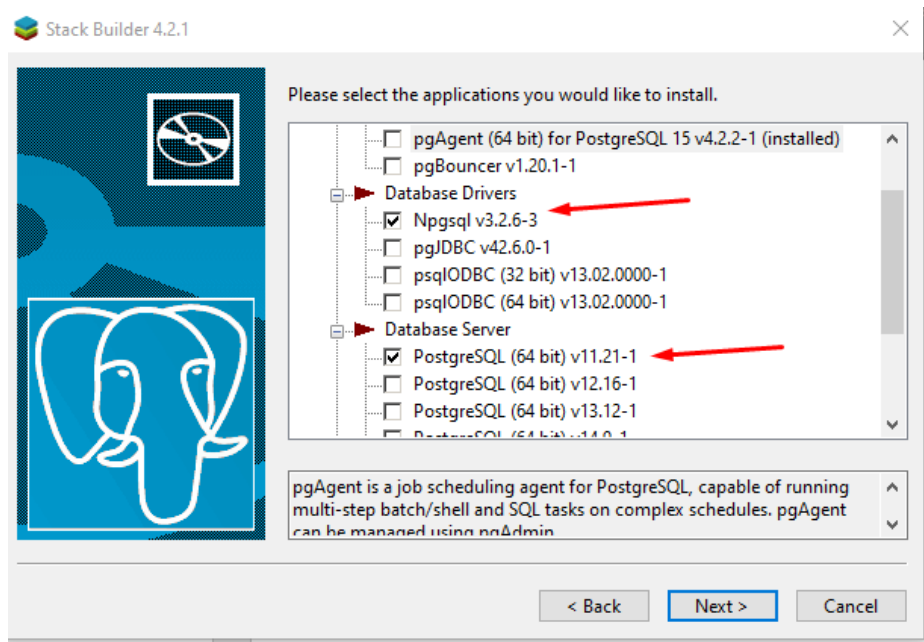
Escolha a porta em que o banco ficará disponível (padrão 5433):



Avance todas as etapas.



Selecione:



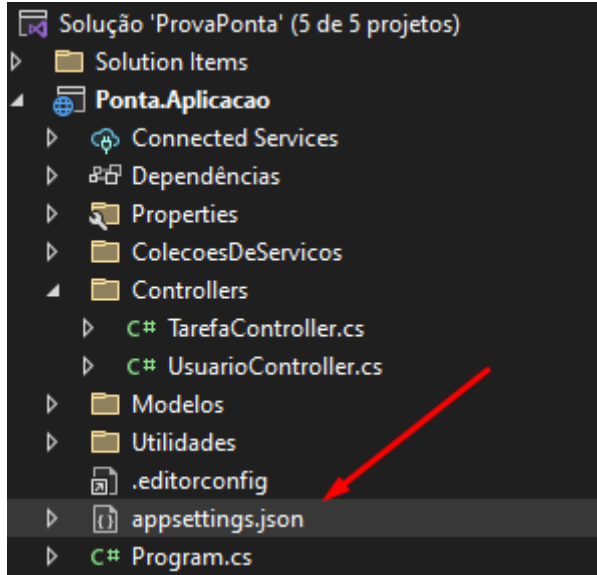
Estas configurações são primordiais para exclusão do projeto.

Observação: **usuário e senha** posteriormente será utilizado para conexão com o banco de dados, **anote os mesmos para prosseguir com as próximas etapas.**

Configuração do banco no visual Studio

Após a configuração do banco insira o **ID, Password, Port e Database** caso queria criar o banco com um nome diferente configurado no momento da instalação:

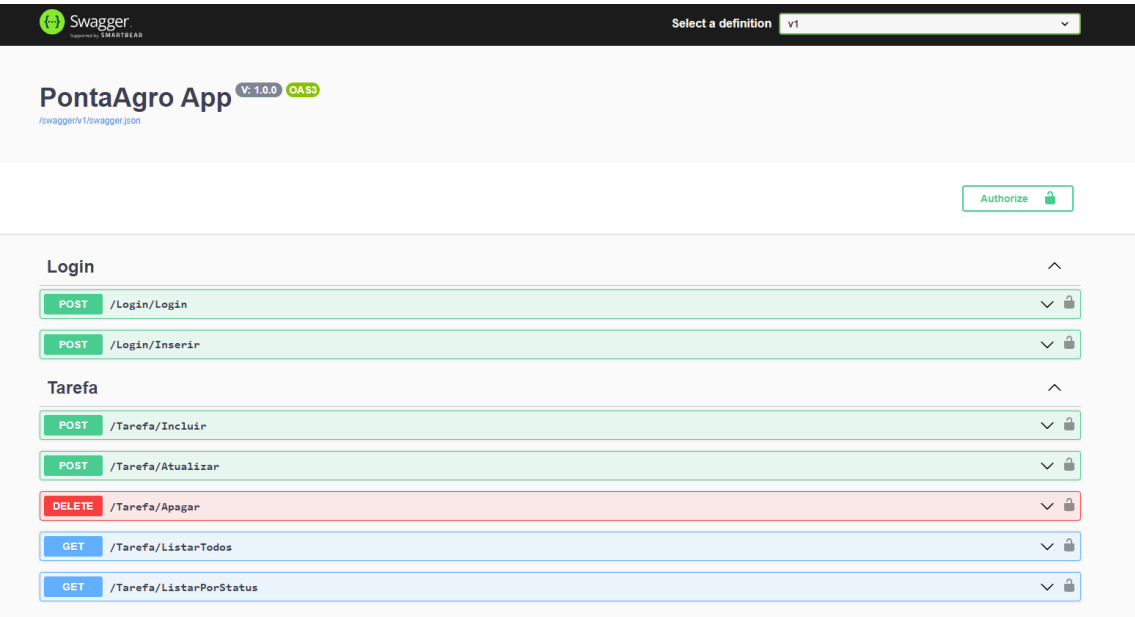
Arquivo: **appsettings.json**



```
{
  "Jwt": {
    "Key": "43e4dbf0-52ed-4203-895d-42b586496bd4"
  },
  "Version": "1.0.0",
  "ConnectionString": "User ID=postgres;Password=root;Host=localhost;Port=5432;Database=PontaAgro;Pooling=true;Connection Lifetime=0;Include Error Detail=true;",
  "AllowedHosts": "*"
}
```

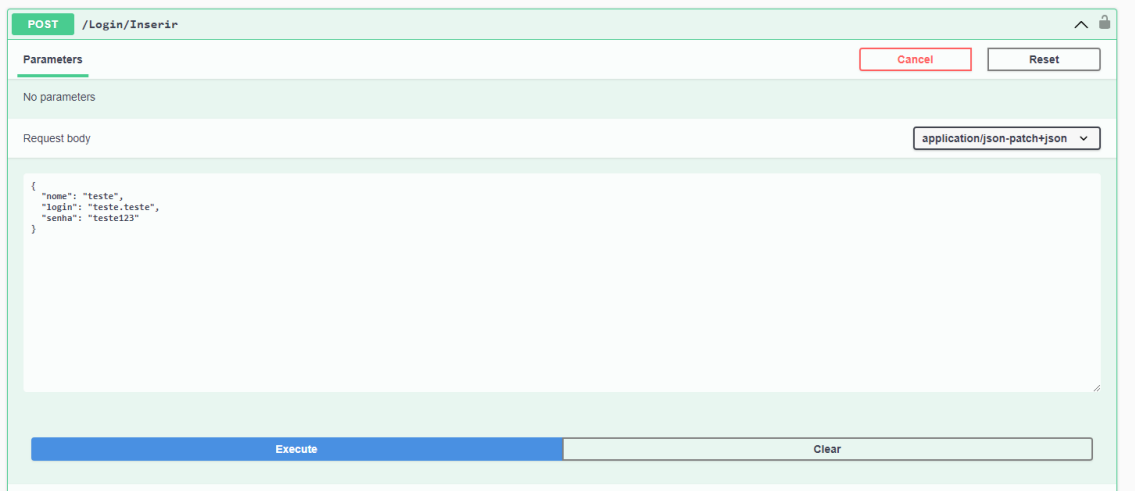
Execute o projeto no visual Studio.

Configuração de execução da primeira vez da aplicação



The image shows the Swagger UI for the PontaAgro App. The top bar includes the Swagger logo and a dropdown menu for selecting a definition, currently set to 'v1'. Below the header, the app name 'PontaAgro App' is displayed with version 'V: 1.0.0' and a 'OAS3' tag. An 'Authorize' button is visible on the right. The main content area lists two API groups: 'Login' and 'Tarefa'. The 'Login' group contains two POST endpoints: '/Login/Login' and '/Login/Inserir'. The 'Tarefa' group contains five endpoints: two POST endpoints ('/Tarefa/Incluir' and '/Tarefa/Atualizar'), one DELETE endpoint ('/Tarefa/Apagar'), and two GET endpoints ('/Tarefa/ListarTodos' and '/Tarefa/ListarPorStatus'). Each endpoint is represented by a colored bar with its method, path, and a lock icon.

Ao executar o projeto as migrations irá adicionar o banco e tabelas envolvidas, com a aplicação executando pela primeira vez, devemos inserir o primeiro usuário na base, **sem autenticação**.

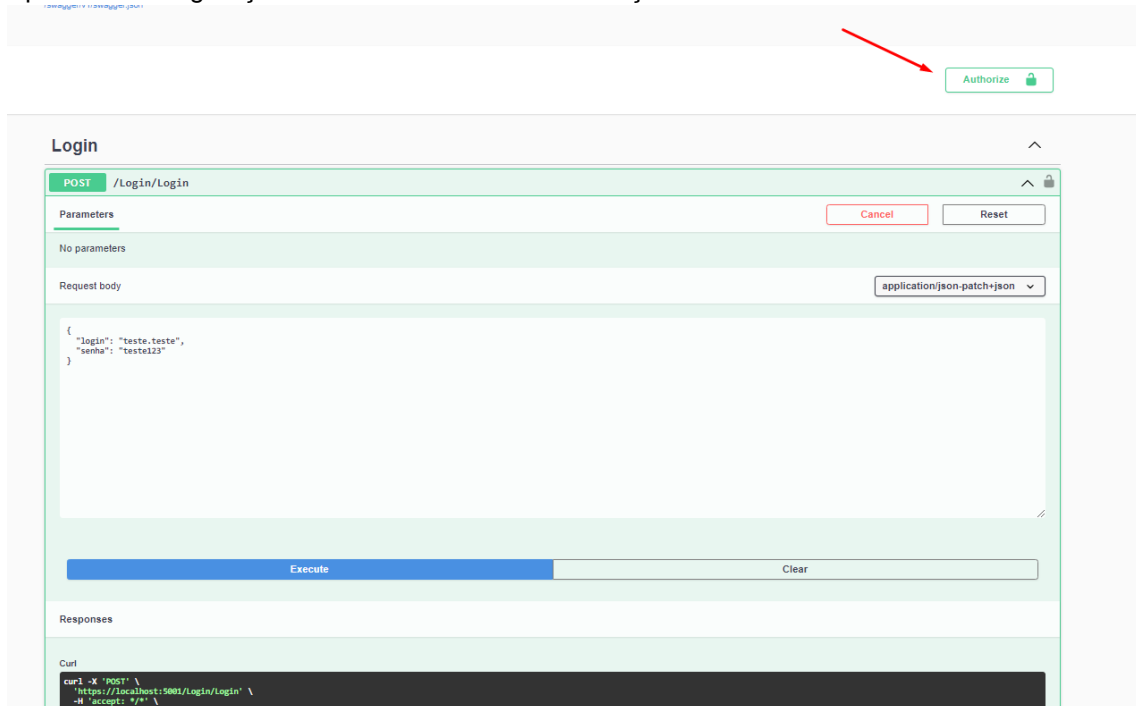


The image shows the Swagger UI interface for the POST endpoint '/Login/Inserir'. The top bar indicates the method 'POST' and the path '/Login/Inserir'. Below the header, there are 'Parameters' and 'Request body' sections. The 'Parameters' section shows 'No parameters'. The 'Request body' section shows a dropdown menu for the content type, currently set to 'application/json-patch+json'. The request body is a JSON object:

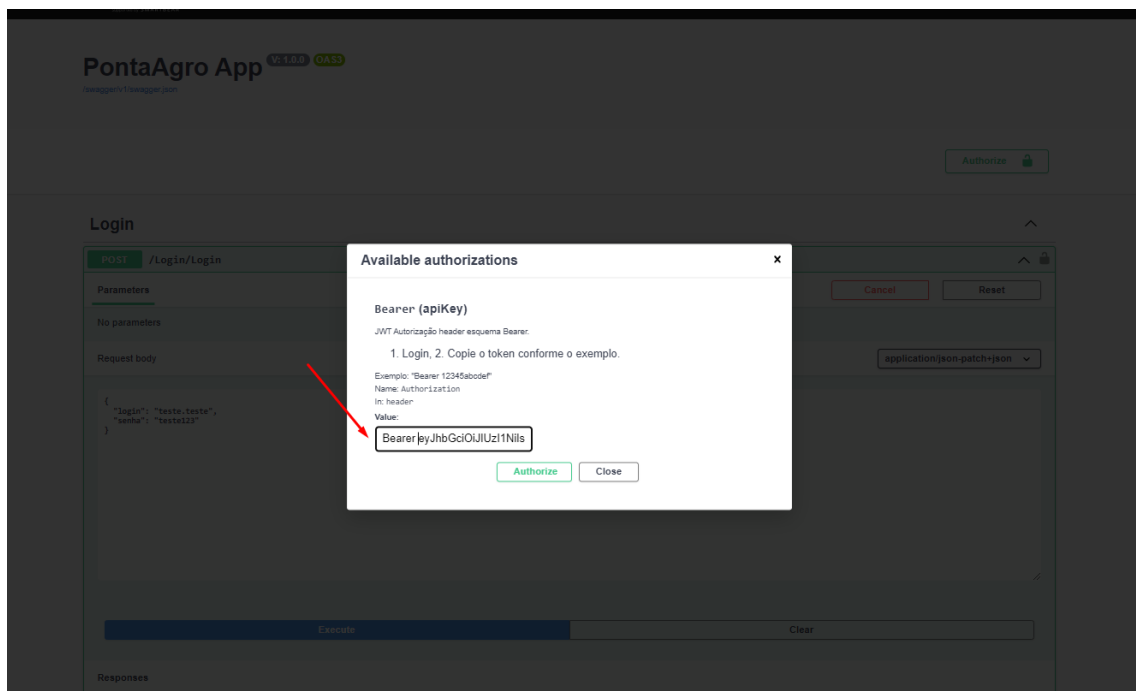
```
{  "nome": "teste",  "login": "teste.teste",  "senha": "testel23"}
```

. At the bottom, there are 'Execute' and 'Clear' buttons.

Após esta configuração iremos realizar uma autorização com nosso do nosso token retornado:



Escreva Bearer e cole o token retornado do login



Após autorizado podemos utilizar a API de acordo com a necessidade do usuário:

GET

/Tarefa/ListarPorStatus

Parameters

Name

Description

status

integer(\$int32)

0

(query)

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:5001/Tarefa/ListarPorStatus?status=0' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0bmVudWQiOiJ1Iiwidm8pcXV1ZXSHAMU0130ZWM2S58ZWM2S1sTaW51Z1GMY2NjuzNjYyOQwIjXZhwIjoxNjE2MjUwOTI4LCPyXQ10JE20TY1hzY1Hj9..zIR3_3Yarv3M8P8KPk' \
  <
```

Request URL

https://localhost:5001/Tarefa/ListarPorStatus?status=0

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "titulo": "Tarefa 1 ",
  "descricao": "Descricao",
  "data": "2023-10-05T07:49:16.37-03:00",
  "status": "Pendente"
}
```

Download

Response headers

```
content-length: 113
content-type: application/json; charset=utf-8
date: Thu, 05 Oct 2023 20:14:13 GMT
server: Kestrel
```

Responses

Code

Description

Links

200

Success

No links

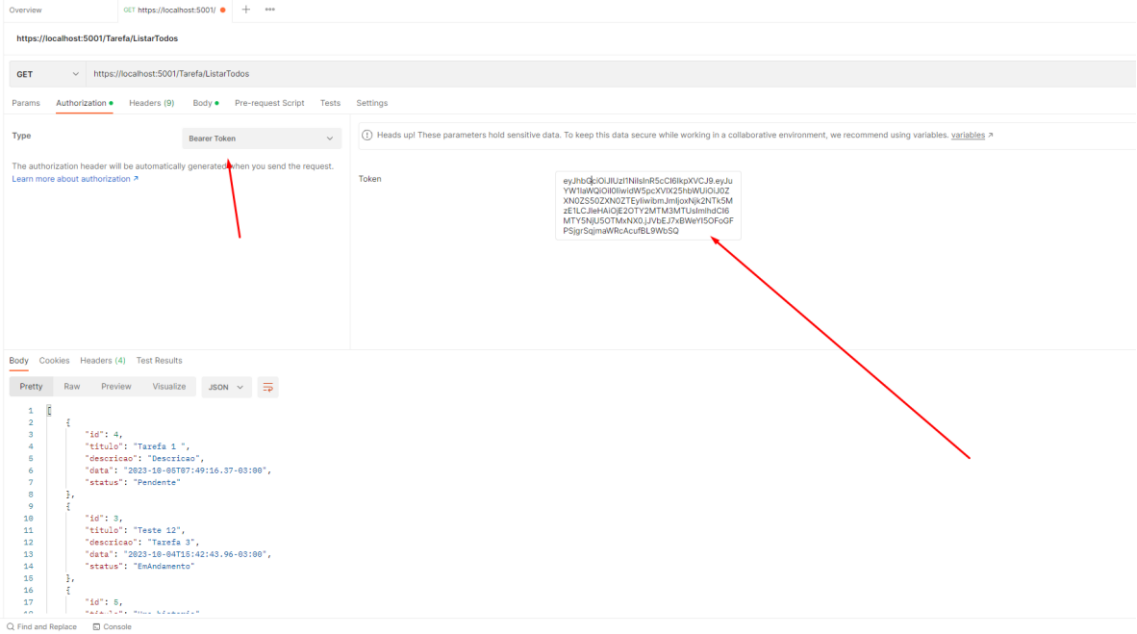
Configuração de execução da primeira vez da aplicação Postman

Ao executar o projeto as migrations irá adicionar o banco e tabelas envolvidas, com a aplicação executando pela primeira vez, devemos inserir o primeiro usuário na base, **sem autenticação**.

Após esta configuração iremos realizar uma autorização com nosso do nosso token retornado:



Selecione a opção **Bearer Token** e cole o token na caixa de ao lado conforme a imagem.



Observação: esta configuração do token deve ser realizada em todos os endpoints da aplicação. **Com exceção do endpoint inclusão do login.**

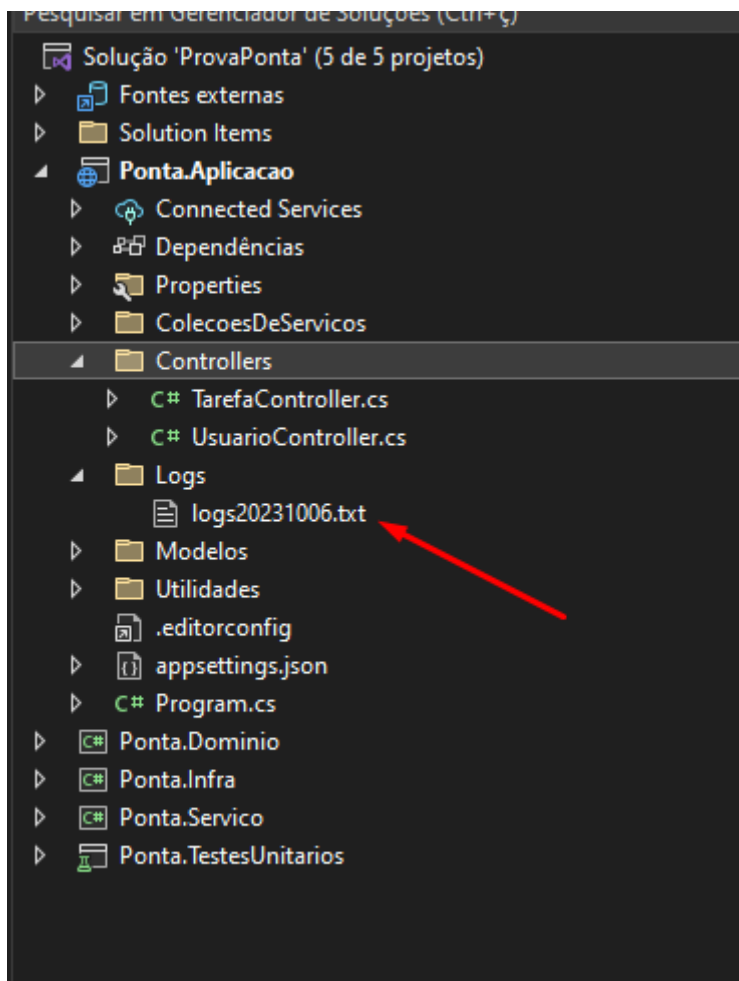
Inserção/Alteração e Seleção por Status

Para realizar a Inserção/Alteração e Seleção por Status será considerado a seguinte regra:

- **0 – Pendente.**
- **1 – Em Andamento.**
- **2 – Concluído.**

Logs

Ao iniciar a aplicação será gerado por dia o monitoramento e rastreamento de toda aplicação em arquivo .txt



Eventos importantes de acordo com a necessidade do usuário estão sendo monitorados como exemplo na **TarefaController**:

```
4 referências
private IActionResult Execute(Func<object> func)
{
    try
    {
        var result = func();
        camposlogs.AddRange(TratamentoDeLogs.CamposLogs(logLabelController + "|" + HttpContext.User.Identity.Name + "|" + JsonConvert.SerializeObject(result) + " | Sucesso"));
        _logger.LogInformation(TratamentoDeLogs.FormatarLabel(logLabel, camposlogs));
    }
    catch (Exception ex)
    {
        camposlogs.AddRange(TratamentoDeLogs.CamposLogs(logLabelController + "|" + HttpContext.User.Identity.Name + "|" + ex.Message + "\r\n\r\n" + ex.InnerException?.Message + " | Erro"));
        _logger.LogCritical(TratamentoDeLogs.FormatarLabel(logLabel, camposlogs));
        return BadRequest(ex.Message + "\r\n\r\n" + ex.InnerException?.Message + "\r\n\r\nErro , entre em contato com o setor de TI.");
    }
}
```