

# Controle Fácil

Segue abaixo o passo a passo com os comandos que utilizaremos para criar o projeto:

1. Crie uma pasta chamada ControleFacil, e abra o prompt de comando (cmd) dentro da mesma.
2. Crie uma Solution com o seguinte comando:
  - a. **`dotnet new sln --name ControleFacil`**
3. Crie duas pastas na raiz do projeto, **src** e **test**
4. Entre na pasta **src**, execute o comando abaixo para criar o projeto principal da API:
  - a. **`dotnet new webapi --name ControleFacil.Api`**
5. Entre na pasta **test** para criar o projeto de teste com o comando:
  - a. **`dotnet new xunit --name ControleFacil.Test`**
6. Volte a raiz do projeto, pasta onde está a **Solution**
7. Adicione os projetos de Api e Test na Solution:
  - a. **`dotnet sln add "src/ControleFacil.Api/ControleFacil.Api.csproj"`**
  - b. **`dotnet sln add "test/ControleFacil.Test/ControleFacil.Test.csproj"`**
8. Entrar na pasta **test/ControleFacil.Test** para add o projeto como dependência.
  - a. **`dotnet add reference "../src/ConmtroleFacil.Api/ControleFacil.Api.csproj"`**

Vamos precisar nesse projeto de algumas dependências para funcionamento da nossa API.

Para isso instale no seu VsCode a biblioteca de nugget:

- <https://marketplace.visualstudio.com/items?itemName=jmrog.vscode-nuget-packag>  
[e-manager](#).
- <https://marketplace.visualstudio.com/items?itemName=patcx.vscode-nuget-gallery>

As dependências estão abaixo para facilitar, basta copiar este **ItemGroup** e substituir no seu projeto dentro do arquivo ControleFacil.Api.csproj

- Dependências da API:

*<ItemGroup>*

*<PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.10"/>*

*<PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0"/>*

*<PackageReference Include="Npgsql.EntityFrameworkCore.PostgreSQL" Version="7.0.4"/>*

*<PackageReference Include="AutoMapper" Version="12.0.1"/>*

*<PackageReference Include="Microsoft.AspNetCore.Authentication" Version="2.2.0"/>*

*<PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer"*

*Version="7.0.10"/>*

*<PackageReference Include="Microsoft.AspNetCore.Mvc.NewtonsoftJson"*

*Version="7.0.10"/>*

*<PackageReference Include="Microsoft.EntityFrameworkCore" Version="7.0.10"/>*

*<PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="7.0.10"/>*

*<PackageReference Include="FluentValidation.AspNetCore" Version="10.4.0"/>*

*<PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="7.0.10"/>*

*</ItemGroup>*

Crie no seu appsettings.json a ConnectionString e a KeySecret, segue abaixo um pronto de exemplo:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "KeySecret": "df3464c69cfe1d48186c143bcffd6c9d6bdcd717",
  "HorasValidadeToken": 8,
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "PADRAO": "User
ID=postgres;Password=123456;Host=localhost;Port=5432;Database=controle-facil;Pooling=true;"
  }
}
```

Neste projeto vamos trabalhar com Migrations, com isso, para rodar os comandos para criar e executar migrations, vamos precisar do dotnet-ef.

Instale o **dotnet-ef globalmente** na sua máquina, ele será responsável para efetuar as migrations, vamos usar a versão 7.0.10 que é a mais recente até a criação do curso.

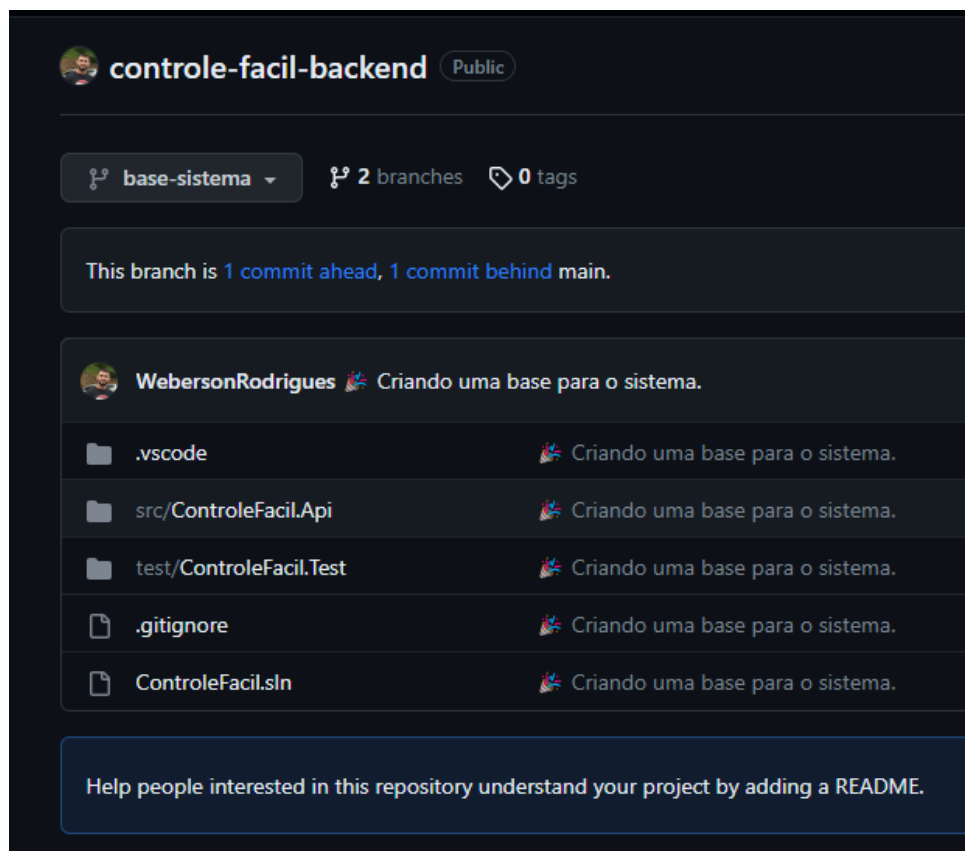
Segue abaixo alguns comandos do dotnet-ef

- Instalar o dotnet-ef
  - **`dotnet tool install --global dotnet-ef --version 7.0.10-*`**
- Criar migrations
  - **`dotnet ef migrations add CriarEntidadeUsuario`**
- Executar migrations
  - **`dotnet ef database updatedotnet ef database update`**

Configure o seu projeto conforme o do modelo base, para isto baixe o projeto base em:

<https://github.com/WebersonRodrigues/controle-facil-backend>

O projeto base esta na branch base-sistema



A branch main está o projeto completo.

Siga o passo a passo dos vídeos para continuar o projeto.

**Bons estudos!**