

DOCUMENTAÇÃO .NETCORE C# VAGRANT – FICHA P8 (Code First)

DEFICULDADES:

- Houve problemas a utilizar o comando “*add-migrations*”;
- Dificuldade em alterar a leitura do objeto preenchido no controlador para leitura direta na base de dados.

RESOLUÇÃO DAS DIFICULDADES:

- Para resolver o problema do “*add-migrations*”, bastou entrar no projeto de forma correta (clicando no *.sln* invés de abrir através da pasta), ativar o protocolo *TCP* para a base de dados e criar regras de *Firewall* para permitir a conexão do *Host*;
- Para resolver o problema da leitura do objeto foi pedido auxílio ao professor José.

SOLUÇÕES ALTERNATIVAS:

- Podia-se ter usado “*dependency injection*” para ler os dados da base de dados invés do método que foi utilizado.

O QUE FOI FEITO:

SQL:

- Criamos a base de dados *DevOps*;
- Ativamos o protocolo *TCP* para permitir conexão com o *Host*;
- Validamos as tabelas criadas e as relações entre elas.

VAGRANT:

- Confirmamos se o *SQL Server* estava a funcionar corretamente;
- Criamos regras de *Firewall* para permitir a conexão do *Host* passar sem problemas.

PROGRAMA:

- Adicionamos a dependência *Microsoft.EntityFrameworkCore.SqlServer*
- Adicionamos a dependência *Microsoft.EntityFrameworkCore.Tools*
- Criamos a classe *BackEndContext.cs* e fizemo-la herdar a classe *DBContext*;
- Fizemos uma configuração da *Connection String* através de um *Override* do *OnConfiguring*;
- Ativamos o Migration utilizando o comando *add-migration*;
- Utilizamos o comando *update-database* para criar as nossas tabelas;
- Criamos a pasta *Services*;
- Criamos uma classe de serviços para cada objeto que nós tínhamos (*Payment*, *Student*, *Subject*, *Teacher*).