

Atividade

Criar estrutura da API com Context e método GET para projetos e teste no navegador.

Nesta atividade, você implementará códigos no projeto Web API da empresa **ExoApi**. Na primeira etapa, sua tarefa será o desenvolvimento dos códigos no projeto para que a aplicação conecte-se ao banco de dados e forneça a lista dos projetos cadastrados por meio da função listar.

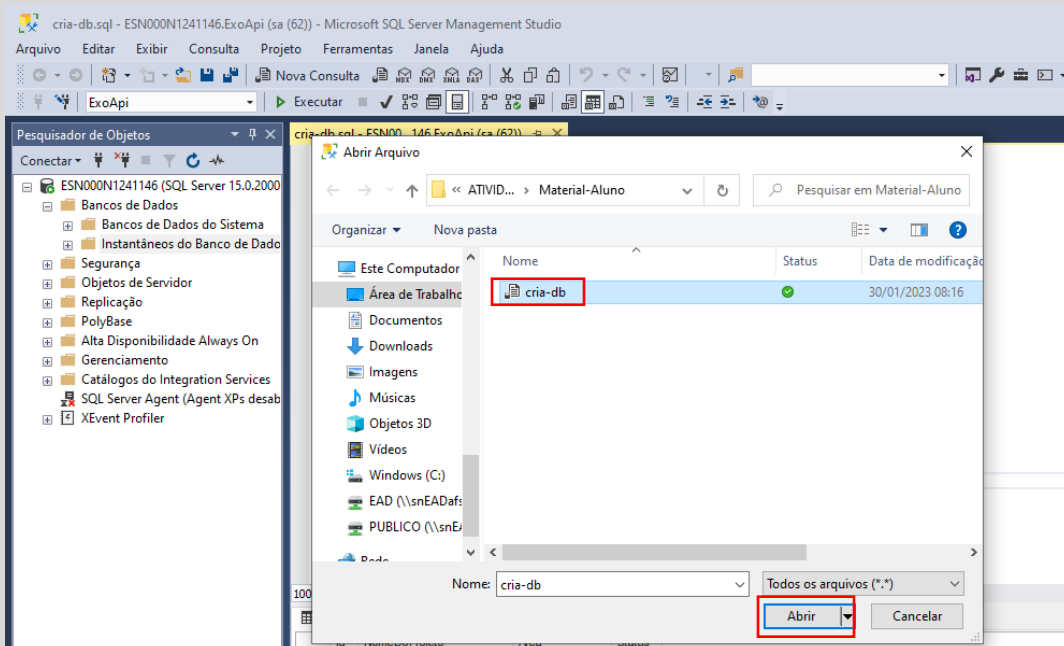
Importante

Para fazer o exercício, é necessário baixar e descompactar o arquivo **ATIVIDADE-03.zip**, que está anexado à atividade.

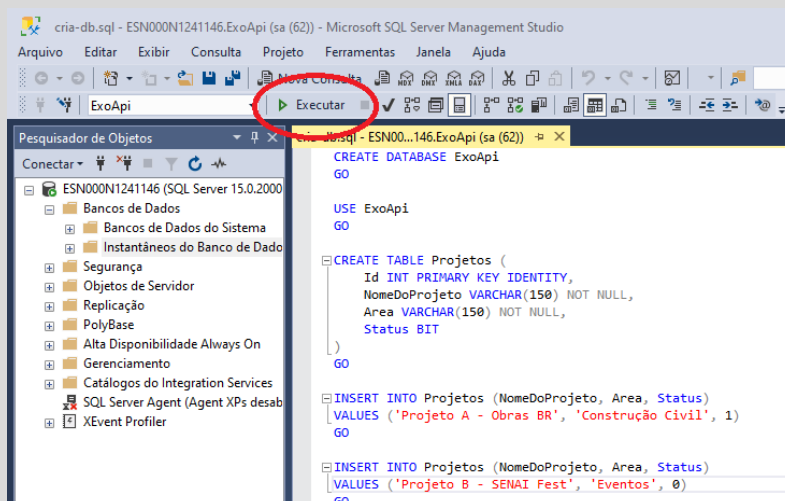


Criando o banco no SSMS

1. Abra o **SQL Server Management Studio (SSMS)**. Clique em **Arquivo > Abrir Arquivo...** e localize a pasta baixada para realizar a atividade. Na pasta **Material-Aluno**, selecione o script **cria-db.sql** e clique em **Abrir**.

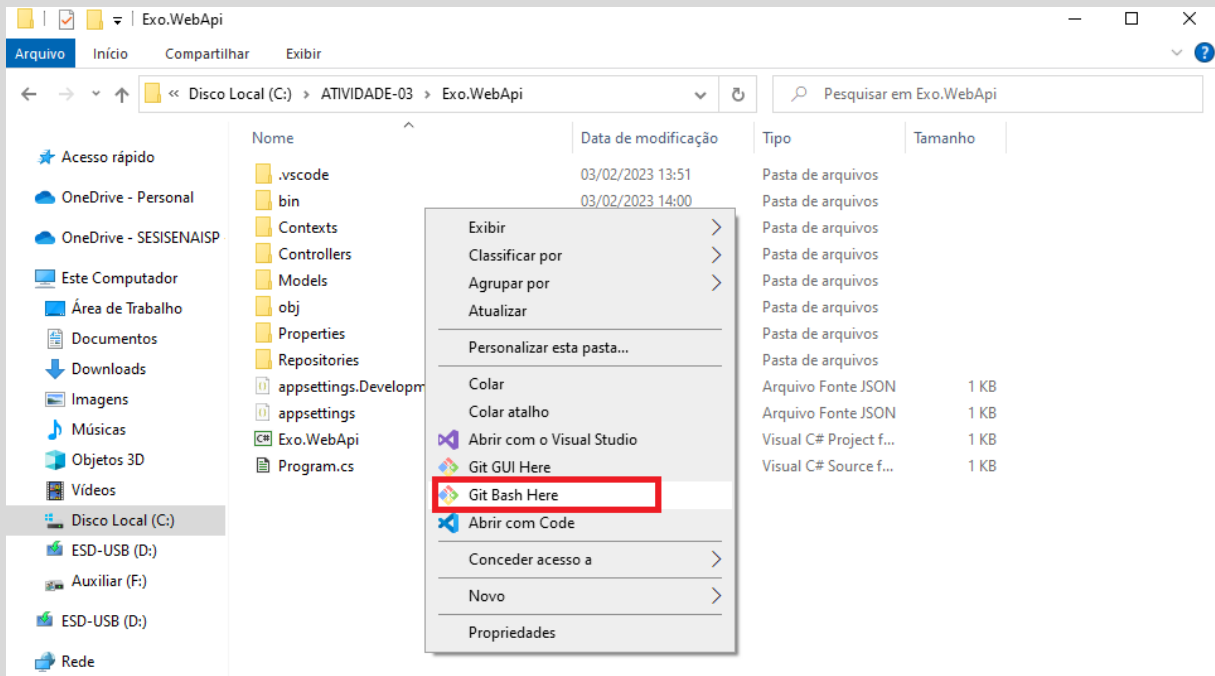


2. Com o arquivo **cria-db.sql** aberto, clique em **Executar** para executar o banco e criar os usuários.



Preparação dos arquivos no VSCode

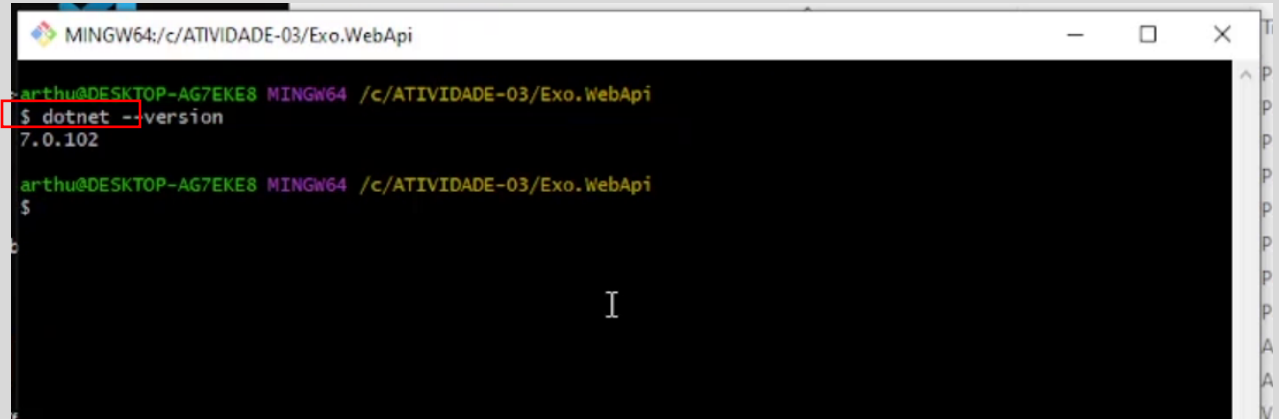
1. Abra a pasta que você baixou a atividade e localize a pasta **Exo.WebApi**. Clique com o botão direito dentro da pasta e escolha **Git Bash Here** para abrir o Terminal.



2. No terminal que será aberto, digite o comando a seguir e dê **Enter** para verificar a versão do dotnet instalada em sua máquina.

```
dotnet --version
```

3. Certifique-se de que sua versão do dotnet seja 6 ou superior. No nosso caso, a versão é 7.0.102, como mostra a figura.



```
MINGW64:/c/ATIVIDADE-03/Exo.WebApi
$ dotnet --version
7.0.102
$
```

4. Agora digite o comando a seguir e dê **Enter** para abrir o **VSCode** com o projeto já aberto.

```
code .
```

5. O VSCode irá abrir com a pasta do projeto e alguns pacotes instalados. Abra o Terminal (menu Terminal > Novo Terminal) e nele digite o comando a seguir para consolidar a instalação dos pacotes.

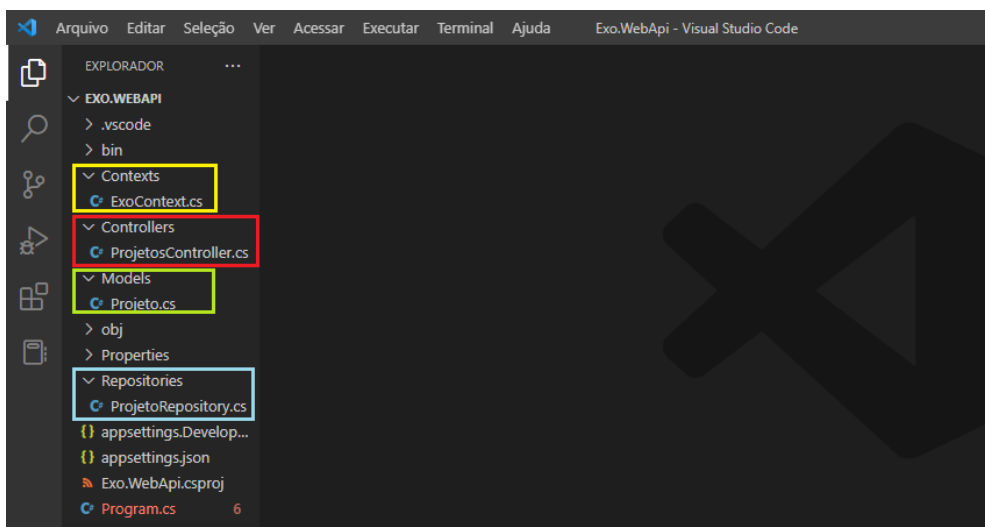
```
dotnet restore
```

Saiba mais

No projeto existem algumas pastas com classes vazias (sem código). Conheça a seguir a função de cada pasta e sua respectiva classe.



Pasta	Classe	Funcionalidade
Context	ExoContext.cs	Cuida das tratativas de conexão com o banco de dados.
Controllers	ProjetosController.cs	Controlador da nossa API. Nesse sistema, ela trabalha com a classe ProjetoRepository.cs e com as operações de manipulação do banco de dados.
Models	Projeto.cs	Modelo do sistema. Nela, encontramos os atributos da entidade Projeto.
Repositories	ProjetoRepository.cs	Faz parte de um padrão de projeto que utilizamos no desenvolvimento da API. Ela é responsável direta pela manipulação com o banco de dados.



Desenvolvimento das classes

1. Abra o arquivo **ExoContext.cs** e copie o código a seguir.

```
using Exo.WebApi.Models;
using Microsoft.EntityFrameworkCore;
using System.Data.SqlClient;
using Microsoft.Data.SqlClient;

namespace Exo.WebApi.Contexts
{
    public class ExoContext : DbContext
    {
        public ExoContext()
        {
        }
        public ExoContext(DbContextOptions<ExoContext> options) :
base(options)
        {
        }
        protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
        {
            if(!optionsBuilder.IsConfigured)
            {
                // Essa string de conexão depende da SUA máquina.
                optionsBuilder.UseSqlServer("Server=localhost\\SQLEXPRESS;"
+ "Database=ExoApi;Trusted_Connection=True;");

                // Exemplo 1 de string de conexão:
                // User
                ID=sa;Password=admin;Server=localhost;Database=ExoApi;-
                // + Trusted_Connection=False;

                // Exemplo 2 de string de conexão:
                //
                Server=localhost\\SQLEXPRESS;Database=ExoApi;Trusted_Connection=True;
            }
        }
        public DbSet<Projeto> Projetos { get; set; }
    }
}
```

Importante

As linhas em verde são comentários. Neles, estão algumas alternativas de string de conexão que poderão ser usadas conforme as configurações de sua máquina.



```
// Essa string de conexão depende da SUA máquina.  
optionsBuilder.UseSqlServer("Server=localhost\\SQLEXPRESS;"  
    + "Database=ExoApi;Trusted_Connection=True;");  
  
// Exemplo 1 de string de conexão:  
// User ID=sa;Password=admin;Server=localhost;Database=ExoApi;-
```

Você sabia

Há alguns tipos de string de conexão:

Um deles é para banco de dados SQL Server, configurado com **usuário sa** na instalação. Se você instalou seu banco de dados dessa forma, use esse exemplo.

Outro tipo mais comum de instalação é a **autenticação do Windows**. Se esse for o seu caso, use esse exemplo de string.

Em ambos os casos, o Server e usuário podem ter nomes diferentes, então o ideal é que você tenha guardado a string de conexão da sua instalação no momento que o SQL Server foi instalado.



2. Abra o arquivo **Projeto.cs** e copie o código a seguir.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Exo.WebApi.Models
{
    public class Projeto
    {
        public int Id { get; set; }
        public string NomeDoProjeto { get; set; }
        public string Area { get; set; }
        public bool Status { get; set; }
    }
}
```


3. Abra o arquivo **ProjetoRepository.cs** e copie o código a seguir.

```
using Exo.WebApi.Contexts;
using Exo.WebApi.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Exo.WebApi.Repositories
{
    public class ProjetoRepository
    {
        private readonly ExoContext _context;
        public ProjetoRepository(ExoContext context)
        {
            _context = context;
        }
        public List<Projeto> Listar()
        {
            return _context.Projetos.ToList();
        }
    }
}
```

4. Abra o arquivo **ProjetosController.cs** e copie o código a seguir.

```
using Exo.WebApi.Models;
using Exo.WebApi.Repositories;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using System;

namespace Exo.WebApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ProjetosController : ControllerBase
    {
        private readonly ProjetoRepository
        _projetoRepository;
        public ProjetosController(ProjetoRepository
projetoRepository)
        {
            _projetoRepository = projetoRepository;
        }

        [HttpGet]
        public IActionResult Listar()
        {
            return Ok(_projetoRepository.Listar());
        }
    }
}
```

5. Salve os arquivos. No Terminal do VSCode, digite o comando a seguir e dê **Enter** para compilar o projeto.

```
dotnet build
```

Testando o projeto no navegador

1. Ainda no Terminal, digite o comando a seguir e dê **Enter** para rodar o projeto.

```
dotnet run
```

Importante

Quando for executado o comando **dotnet run**, o servidor será iniciado e disponibilizará a URL de acesso ao projeto. Será com esse endereço + o sufixo **api/projetos** que você acessará a API desenvolvida por meio do navegador.



```
Arquivo Editar Seleção Ver Acessar Executar ... ProjetosController.cs - Exo.WebApi - Visual Studio Code

EXPLORADOR
EXO.WEBAPI
  .vscode
  bin
  Contexts
  ExoContext.cs
  Controllers
    ProjetosController.cs
  Models
    Projeto.cs
  obj
  Properties
  Repositories
    ProjetoRepository.cs
  appsettings.Develop...
  appsettings.json
  Exo.WebApi.csproj
  Program.cs

Controllers > ProjetoRepository.cs > ...
1 using Exo.WebApi.Models;
2 using Exo.WebApi.Repositories;
3 using Microsoft.AspNetCore.Authorization;
4 using Microsoft.AspNetCore.Mvc;
5 using System;
6
7 namespace Exo.WebApi.Controllers
8 {
9     [Route("api/[controller]")]
10    [ApiController]
11    public class ProjetoRepository : ControllerBase
12    {
13        private readonly ProjetoRepository _projetoRepository;
14        public ProjetoRepository(ProjetoRepository projetoRepository)
```

```
Tempo Decorrido 00:00:04.76
PS C:\ATIVIDADE-03\Exo.WebApi> dotnet run
Compilando...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7154
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5038
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\ATIVIDADE-03\Exo.WebApi\
```

2. Para visualizar os resultados do banco via API, acesse a URL gerada + o sufixo **api/projetos**, como na imagem a seguir.

