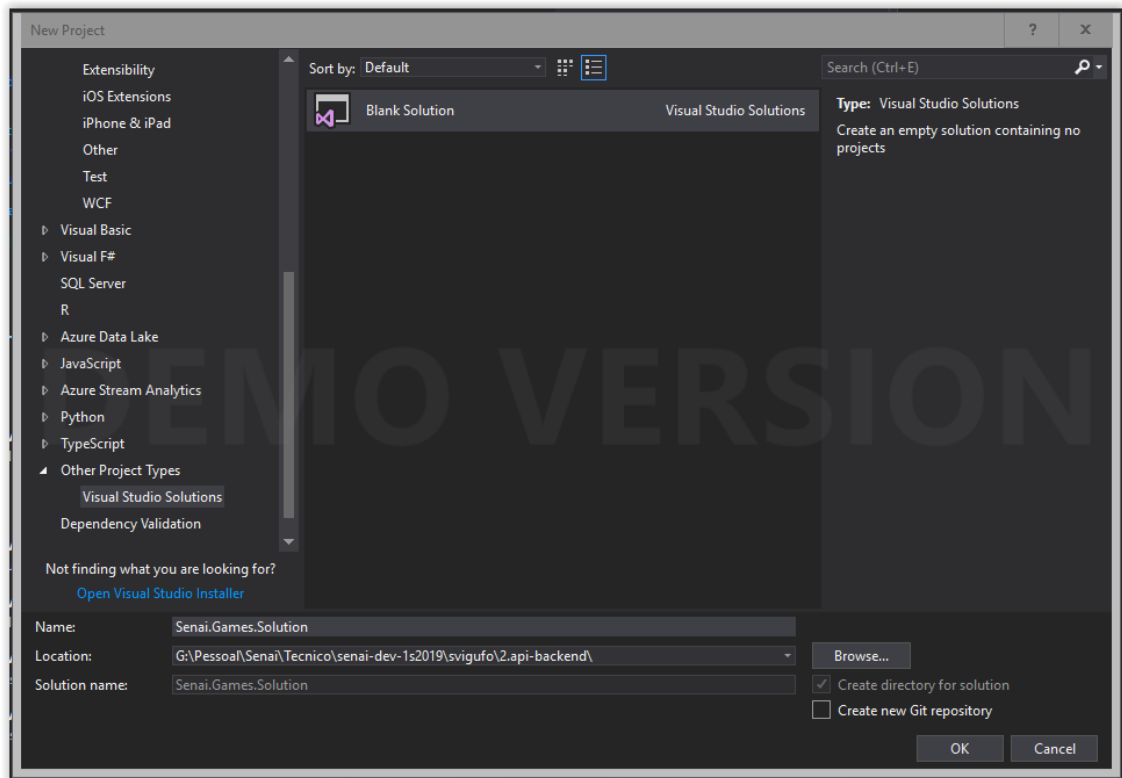
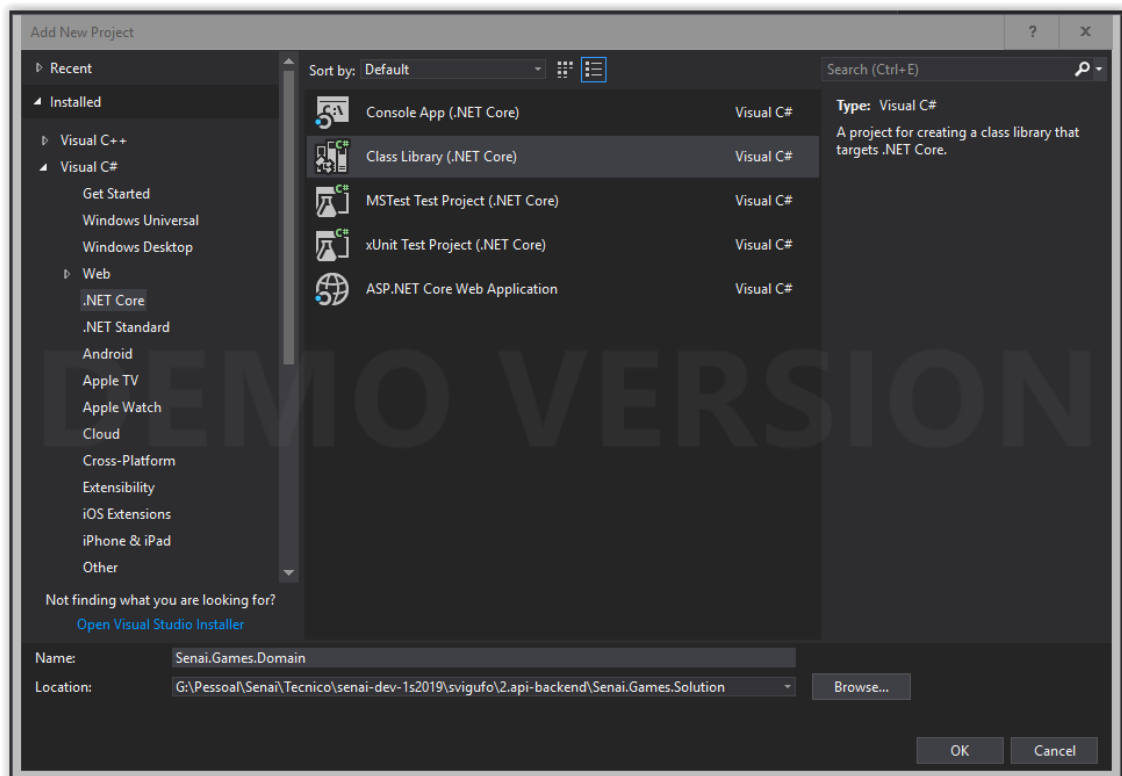


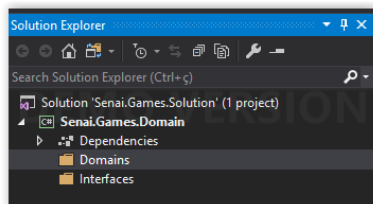
Explicar o que é uma solução e o que são projetos



Criar projeto .Net Core -> Class Library -> Senai.Games.Domain



Apagar a Class1.cs e criar uma pasta Domains e Interfaces

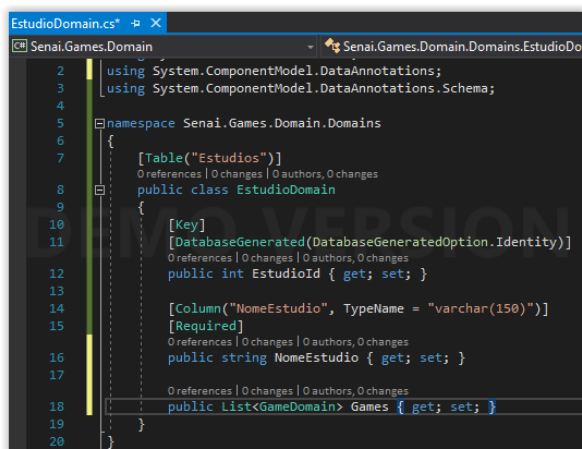


Explicar sobre ORM

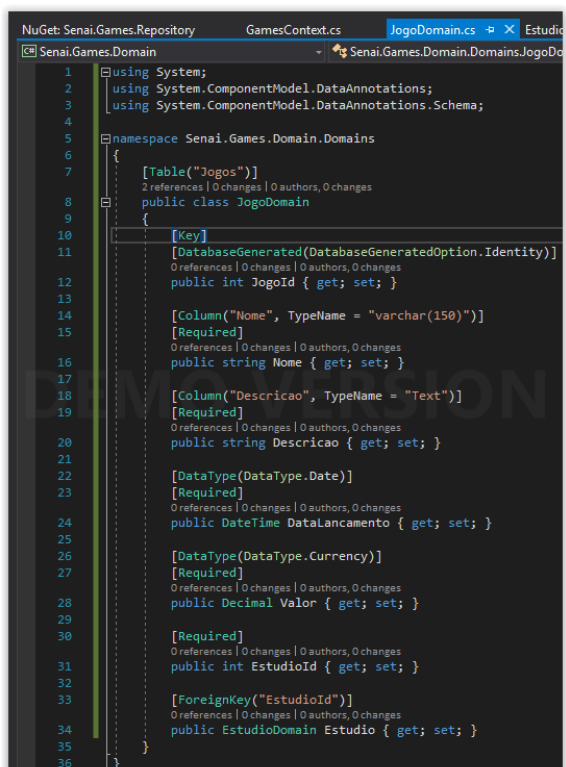
Apresentar o Entity Framework

Criar Classe EstudioDomain e incluir os DataAnnotations

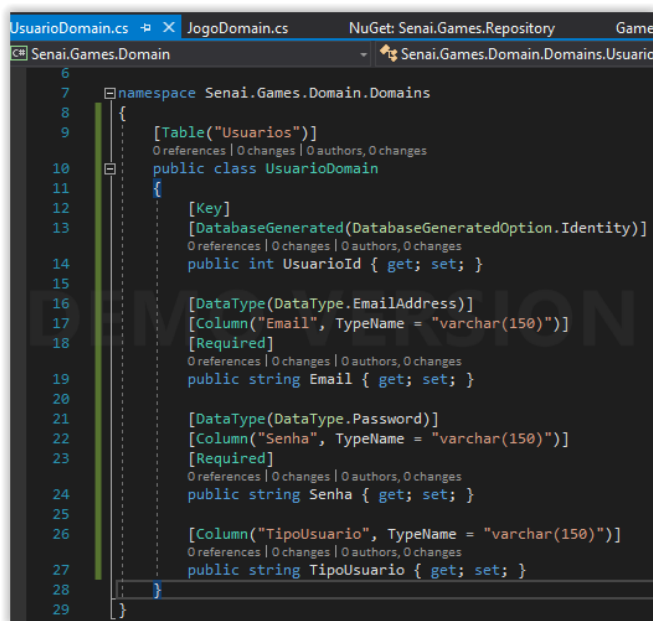
Explicar sobre a propriedade "List<GameDomain> Games"



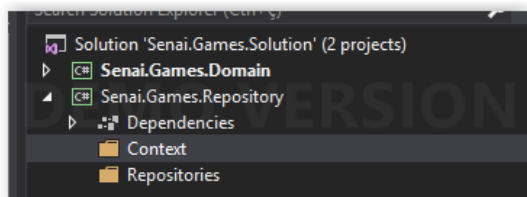
Criar Classe JogoDomain e incluir os DataAnnotations



Criar UsuarioDomain e incluir os DataAnnotations



Criar Projeto Senai.Games.Repository e incluir as pastas



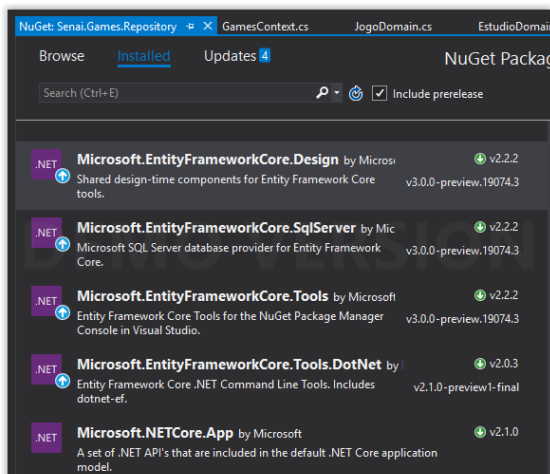
Baixar pacotes do nugget

Microsoft.EntityFrameworkCore.SqlServer 2.1.1

Microsoft.EntityFrameworkCore.Design 2.1.1

Microsoft.EntityFrameworkCore.Tools 2.1.1

Microsoft.EntityFrameworkCore.Tools 2.0.3



Explicar o que é um contexto

Criar class GamesContext dentro da pasta Context

```
public class GamesContext : DbContext
{
    References<IChanges> IAuthors, IChanges
    public DbSet<JogoDomain> Jogos { get; set; }
    References<IChanges> IAuthors, IChanges
    public DbSet<EstudioDomain> Estudios { get; set; }
    References<IChanges> IAuthors, IChanges
    public DbSet<UsuarioDomain> Usuarios { get; set; }

    References<IChanges> IAuthors, IChanges
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer("server=\\SqlExpress;database=InLock_Games;trusted_connection=true;");
        base.OnConfiguring(optionsBuilder);
    }

    References<IChanges> IAuthors, IChanges
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<UsuarioDomain>().HasData(
            new UsuarioDomain
            {
                UsuarioId = 1, Email = "admin@admin.com", Senha = "admin", TipoUsuario = "ADMINISTRADOR"
            },
            new UsuarioDomain
            {
                UsuarioId = 2, Email = "cliente@cliente.com", Senha = "cliente", TipoUsuario = "CLIENTE"
            });

        modelBuilder.Entity<EstudioDomain>().HasData(
            new EstudioDomain { EstudioId = 1, NomeEstudio = "Square Enix" },
            new EstudioDomain { EstudioId = 2, NomeEstudio = "Blizzard" },
            new EstudioDomain { EstudioId = 3, NomeEstudio = "Rockstar Studios" });

        modelBuilder.Entity<JogoDomain>().HasData(
            new JogoDomain
            {
                JogoId = 1, Nome = "Diablo 3", DataLancamento = Convert.ToDateTime("15/05/2012"), Descricao = "É um jogo que contém bastante ação e é viciante, seja você um novato ou um fã",
                Valor = Convert.ToDecimal("99,00"), EstudioId = 2
            },
            new JogoDomain
            {
                JogoId = 2, Nome = "Red Dead Redemption II", DataLancamento = Convert.ToDateTime("26/10/2018"), Descricao = "Jogo eletrônico de ação-aventura western",
                Valor = Convert.ToDecimal("120,00"), EstudioId = 3
            });

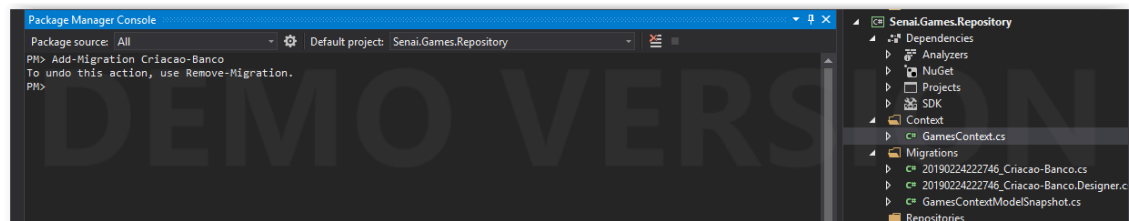
        base.OnModelCreating(modelBuilder);
    }
}
```

Definir Senai.Games.Repository como projeto inicial Set Startup Project

Explicar Migrations

Fazer utilizando o package manager console, tools – nugget package manager – package manager console

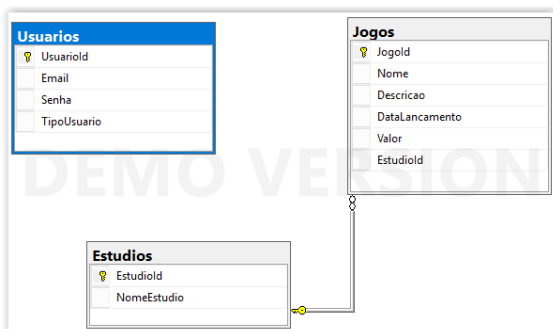
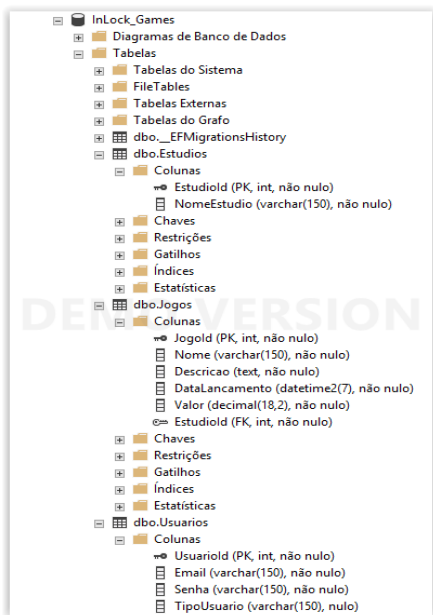
Add-Migration Criacao-Banco



Update-Database

```
Package Manager Console
Package source: All
PM> Update-Database
Applying migration '20190224222746_Criacao-Banco'.
Done.
PM>
```

Abrir Sql Management Studio e mostrar o banco criado e suas tabelas



SQLQuery1.sql - (L...K8\CorujasDev (56))*

```
select * from Estudios
select * from Jogos
select * from Usuarios
```

100 %

Resultados

	Estudioid	NomeEstudio
1	1	Square Enix
2	2	Blizzard
3	3	Rockstar Studios

Mensagens

	Jogoid	Nome	Descricao	DataLancamento	Valor	Estudioid
1	1	Diablo 3	É um jogo que contém bastante ação e é viciante,....	2012-05-15 00:00:00.0000000	99.00	2
2	2	Red Dead Redemption II	Jogo eletrônico de ação-aventura western	2018-10-26 00:00:00.0000000	120.00	3

	Usuarioid	Email	Senha	TipoUsuario
1	1	admin@admin.com	admin	ADMINISTRADOR
2	2	cliente@cliente.com	cliente	CLIENTE

Criar Interface IEstudioRepository na pasta Interfaces do projeto Senai.Games.Domain

```
public interface IEstudioRepository
{
    /// <summary>
    /// Cadastra um novo estúdio
    /// </summary>
    /// <param name="estudio">EstudioDomain</param>
    1 reference | 0 changes | 0 authors, 0 changes
    void Cadastrar(EstudioDomain estudio);

    /// <summary>
    /// Lista todos os Estúdios
    /// </summary>
    /// <returns>List<EstudioDomain></returns>
    1 reference | 0 changes | 0 authors, 0 changes
    List<EstudioDomain> Listar();

    /// <summary>
    /// Busca um estúdio pelo seu Id
    /// </summary>
    /// <param name="id">Id do estúdio</param>
    /// <returns>Retorna um EstudioDomain ou Null</returns>
    1 reference | 0 changes | 0 authors, 0 changes
    EstudioDomain BuscarPorId(int id);
}
```

Crie uma Classe EstudioRepository e implementar a interface na pasta Repositories no projeto Senai.Games.Repository

```
public class EstudioRepository : IEstudioRepository
{
    /// <summary>
    /// Busca um estúdio pelo seu Id
    /// </summary>
    /// <param name="id">Id do estúdio</param>
    /// <returns>Retorna um EstudioDomain ou Null</returns>
    1 reference | 0 changes | 0 authors, 0 changes
    public EstudioDomain BuscarPorId(int id)
    {
        using (GamesContext ctx = new GamesContext())
        {
            return ctx.Estudios.Include("Jogos").FirstOrDefault(g => g.EstudioId == id);
        }
    }

    /// <summary>
    /// Cadastra um novo estúdio
    /// </summary>
    /// <param name="estudio">EstudioDomain</param>
    1 reference | 0 changes | 0 authors, 0 changes
    public void Cadastrar(EstudioDomain estudio)
    {
        using (GamesContext ctx = new GamesContext())
        {
            ctx.Estudios.Add(estudio);
            ctx.SaveChanges();
        }
    }

    /// <summary>
    /// Lista todos os Estúdios
    /// </summary>
    /// <returns>List<EstudioDomain></returns>
    1 reference | 0 changes | 0 authors, 0 changes
    public List<EstudioDomain> Listar()
    {
        using (GamesContext ctx = new GamesContext())
        {
            return ctx.Estudios.Include("Jogos").ToList();
            //Explicar o include
            //return ctx.Estudios.Include("Jogos").ToList();
        }
    }
}
```

//Explicar o include

Criar Interface IUserioRepository na pasta Interfaces do projeto Senai.Games.Domain

```
public interface IUserioRepository
{
    /// <summary>
    /// cadastra um novio usuário
    /// </summary>
    /// <param name="usuario">UsuarioDomain</param>
    0 references | 0 changes | 0 authors, 0 changes
    void Cadastrar(UsuarioDomain usuario);

    /// <summary>
    /// Lista os usuarios
    /// </summary>
    /// <returns>List<UsuarioDomain></returns>
    0 references | 0 changes | 0 authors, 0 changes
    List<UsuarioDomain> Listar();

    /// <summary>
    /// Busca um usuário pelo email e senha
    /// </summary>
    /// <param name="email">Email do usuário</param>
    /// <param name="senha">Senha do usuário</param>
    /// <returns>Retorna o UsuarioDomain ou Null</returns>
    0 references | 0 changes | 0 authors, 0 changes
    UsuarioDomain BuscarEmailSenha(string email, string senha);
}
```

Crie uma Classe UsuarioRepository e implementar a interface na pasta IUserioRepository no projeto Senai.Games.Repository

```
public class UsuarioRepository : IUserioRepository
{
    /// <summary>
    /// Busca um usuário pelo email e senha
    /// </summary>
    /// <param name="email">Email do usuário</param>
    /// <param name="senha">Senha do usuário</param>
    /// <returns>Retorna o UsuarioDomain ou Null</returns>
    1 reference | 0 changes | 0 authors, 0 changes
    public UsuarioDomain BuscarEmailSenha(string email, string senha)
    {
        using (GamesContext ctx = new GamesContext())
        {
            return ctx.Usuarios.FirstOrDefault(u => u.Email == email && u.Senha == senha);
        }
    }

    /// <summary>
    /// cadastra um novio usuário
    /// </summary>
    /// <param name="usuario">UsuarioDomain</param>
    1 reference | 0 changes | 0 authors, 0 changes
    public void Cadastrar(UsuarioDomain usuario)
    {
        using (GamesContext ctx = new GamesContext())
        {
            ctx.Usuarios.Add(usuario);
            ctx.SaveChanges();
        }
    }

    /// <summary>
    /// Lista os usuarios
    /// </summary>
    /// <returns>List<UsuarioDomain></returns>
    1 reference | 0 changes | 0 authors, 0 changes
    public List<UsuarioDomain> Listar()
    {
        using (GamesContext ctx = new GamesContext())
        {
            return ctx.Usuarios.ToList();
        }
    }
}
```

Criar Interface `IJogoRepository` na pasta Interfaces do projeto `Senai.Games.Domain`

```
public interface IJogoRepository
{
    /// <summary>
    /// Cadastra um novo jogo
    /// </summary>
    /// <param name="jogo">JogoDomain</param>
    0 references | 0 changes | 0 authors, 0 changes
    void Cadastrar(JogoDomain jogo);

    /// <summary>
    /// Lista todos os Jogos
    /// </summary>
    /// <returns>List<JogoDomain></returns>
    0 references | 0 changes | 0 authors, 0 changes
    List<JogoDomain> Listar();

    /// <summary>
    /// Busca um jogo pelo seu Id
    /// </summary>
    /// <param name="id">Id do Jogo</param>
    /// <returns>Retorna um JogoDomain ou Null</returns>
    0 references | 0 changes | 0 authors, 0 changes
    JogoDomain BuscarPorId(int id);
}
```

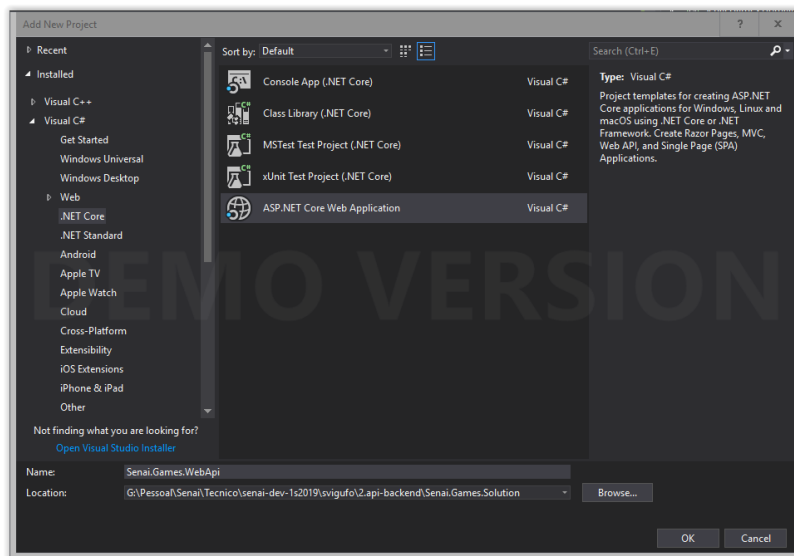
Crie uma Classe `JogoRepository` e implementar a interface na pasta `IJogoRepository` no projeto `Senai.Games.Repository`

```
public class JogoRepository : IJogoRepository
{
    /// <summary>
    /// Busca um jogo pelo seu Id
    /// </summary>
    /// <param name="id">Id do Jogo</param>
    /// <returns>Retorna um JogoDomain ou Null</returns>
    1 reference | 0 changes | 0 authors, 0 changes
    public JogoDomain BuscarPorId(int id)
    {
        using (GamesContext ctx = new GamesContext())
        {
            return ctx.Jogos.Include("Estudio").FirstOrDefault(g => g.JogoId == id);
        }
    }

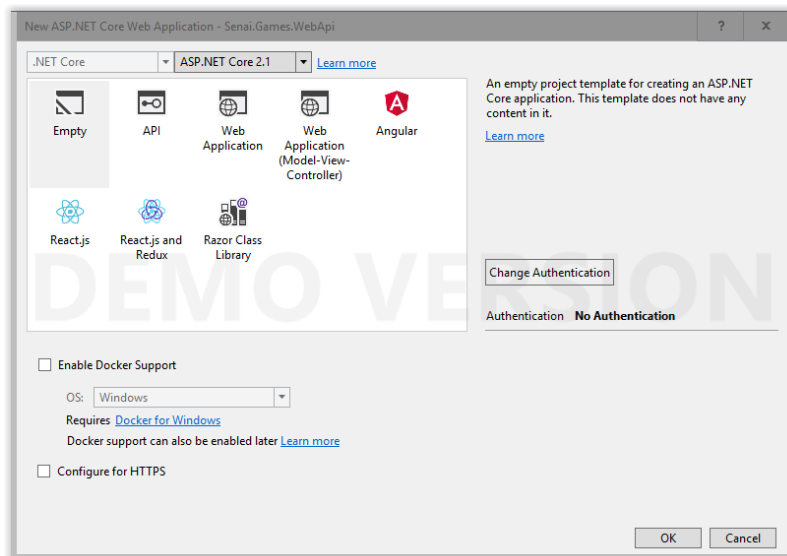
    /// <summary>
    /// Cadastra um novo jogo
    /// </summary>
    /// <param name="jogo">JogoDomain</param>
    1 reference | 0 changes | 0 authors, 0 changes
    public void Cadastrar(JogoDomain jogo)
    {
        using (GamesContext ctx = new GamesContext())
        {
            ctx.Jogos.Add(jogo);
            ctx.SaveChanges();
        }
    }

    /// <summary>
    /// Lista todos os Jogos
    /// </summary>
    /// <returns>List<JogoDomain></returns>
    1 reference | 0 changes | 0 authors, 0 changes
    public List<JogoDomain> Listar()
    {
        using (GamesContext ctx = new GamesContext())
        {
            return ctx.Jogos.Include("Estudio").ToList();
            //Explicar o include
            //return ctx.Estudios.Include("Jogos").ToList();
        }
    }
}
```


Criar Projeto WebApi Senai.Games.WebApi



Tipo de projeto Empty -> Desmarcar Configure for Https



Configurar Startup.cs

Criar pasta Contrllers

Criar Controller LoginController

Criar Controller EstudiosController

```
public class EstudiosController : ControllerBase
{
    4 references | 0 changes | 0 authors, 0 changes
    private IEstudioRepository EstudioRepository { get; set; }

    0 references | 0 changes | 0 authors, 0 changes
    public EstudiosController()
    {
        EstudioRepository = new EstudioRepository();
    }

    [HttpGet("todos")]
    0 references | 0 changes | 0 authors, 0 changes
    public IActionResult Listar()
    {
        try
        {
            return Ok(EstudioRepository.Listar());
        }
        catch (System.Exception ex)
        {
            return BadRequest();
        }
    }

    [HttpGet("{id}")]
    0 references | 0 changes | 0 authors, 0 changes
    public IActionResult BuscarPorId(int id)
    {
        try
        {
            EstudioDomain estudioBuscado = EstudioRepository.BuscarPorId(id);

            if(estudioBuscado == null)
            {
                return NotFound();
            }

            return Ok(estudioBuscado);
        }
        catch (System.Exception ex)
        {
            return BadRequest();
        }
    }

    [HttpPost]
    0 references | 0 changes | 0 authors, 0 changes
    public IActionResult Post(EstudioDomain estudio)
    {
        try
        {
            EstudioRepository.Cadastrar(estudio);
            return Ok();
        }
        catch (System.Exception ex)
        {
            return BadRequest();
        }
    }
}
```

Incluir Autenticação Controller

Explicar Loop entre classes -> Incluir `options.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;`

Criar Controller JogosController

```
4 references | 0 changes | 0 authors, 0 changes
private IRepository<Jogo> _jogoRepository { get; set; }

0 references | 0 changes | 0 authors, 0 changes
public JogosController()
{
    _jogoRepository = new JogoRepository();
}

[HttpGet("todos")]
0 references | 0 changes | 0 authors, 0 changes
public IActionResult Listar()
{
    try
    {
        var Jogos = _jogoRepository.Listar();

        return Ok(Jogos.ToList());
    }
    catch (System.Exception ex)
    {
        return BadRequest();
    }
}

[HttpGet("{id}")]
0 references | 0 changes | 0 authors, 0 changes
public IActionResult BuscarPorId(int id)
{
    try
    {
        JogoDomain JogoBuscado = _jogoRepository.BuscarPorId(id);

        if (JogoBuscado == null)
        {
            return NotFound();
        }

        return Ok(JogoBuscado);
    }
    catch (System.Exception ex)
    {
        return BadRequest();
    }
}

[HttpPost]
0 references | 0 changes | 0 authors, 0 changes
public IActionResult Post(JogoDomain Jogo)
{
    try
    {
        _jogoRepository.Cadastrar(Jogo);
        return Ok();
    }
    catch (System.Exception ex)
    {
        return BadRequest();
    }
}
```

Incluir Autenticação Controller

Criar Controller UsuariosController

```
[Route("api/[controller]")]
[ApiController]
1 reference | 0 changes | 0 authors, 0 changes
public class UsuariosController : ControllerBase
{
    2 references | 0 changes | 0 authors, 0 changes
    private IUserRepository UsuarioRepository { get; set; }

    0 references | 0 changes | 0 authors, 0 changes
    public UsuariosController()
    {
        UsuarioRepository = new UsuarioRepository();
    }

    [Authorize("ADMINISTRADOR")]
    [HttpGet("todos")]
    0 references | 0 changes | 0 authors, 0 changes
    public IActionResult Listar()
    {
        try
        {
            var Usuarios = UsuarioRepository.Listar();

            return Ok(Usuarios.ToList());
        }
        catch (System.Exception ex)
        {
            return BadRequest();
        }
    }
}
```