

Criar classes:

MinhaApp.Dominio

MinhaApp.Repositorio

MinhaApp.Servico

Criar api:

MinhaApp.WebAPI

Referenciar:

Api > serviço

Serviço > repositório

Repositório > dominio

Instalar nuget no repositório e na api:

EntiryFrameworkCore e in memory

Criar classe Aluno em Dominio:

```
namespace MinhaApp.Dominio
{
    public class Aluno
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public decimal Nota { get; set; }

        public Aluno(string nome, decimal nota)
        {
            Nome = nome;
            Nota = nota;
        }

        public bool EstaAprovado() => return Nota > 7;
    }
}
```

Criar interface IAlunoRepositorio;

using MinhaApp.Dominio;

```
namespace MinhaApp.Repositorio
{
    public interface IAlunoRepositorio
    {
        void Salvar(Aluno aluno);
        IEnumerable<Aluno> Listar();
    }
}
```

Criar classe AppDbContext em repositório;

```
using MinhaApp.Dominio;

namespace MinhaApp.Repositorio
{
    public class AppDbContext : DbContext
    {
        public DbSet<Aluno> Alunos { get; set; }

        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) { }
    }
}
```

Criar a classe AlunoRepositorio;

```
namespace MinhaApp.Repositorio
{
    public class AlunoRepositorio : IAlunoRepositorio
    {
        private readonly AppDbContext _context;

        public AlunoRepositorio(AppDbContext context)
        {
            _context = context;
        }

        public void Salvar(Aluno aluno)
        {
            _context.Alunos.Add(aluno);
            _context.SaveChanges();
        }

        public IEnumerable<Aluno> Listar()
        {
            return _context.Alunos.ToList();
        }
    }
}
```

Criar classe AlunoDto em servico;

```
namespace MinhaApp.Servico
{
    public class AlunoDto
    {
        public string Nome { get; set; }
        public decimal Nota { get; set; }
    }
}
```

Criar interface IAlunoServico em servico;

```
namespace MinhaApp.Servico
{
    public interface IAlunoServico
    {
        void Adicionar(AlunoDto alunoDto);
        IEnumerable<AlunoDto> Listar();
    }
}
```

Criar classe AlunoServico em servico;

```
namespace MinhaApp.Servico
{
    public class AlunoServico : IAlunoServico
    {
        private readonly IAlunoRepositorio _repositorio;

        public AlunoServico(IAlunoRepositorio repositorio)
        {
            _repositorio = repositorio;
        }

        public void Adicionar(AlunoDto alunoDto)
        {
            var aluno = new Aluno(alunoDto.Nome, alunoDto.Nota);
            _alunoRepositorio.Salvar(aluno);
        }

        public IEnumerable<AlunoDto> Listar()
        {
            return _alunoRepositorio.Listar().Select(aluno =>
                new AlunoDto { Nome = aluno.Nome, Nota = aluno.Nota });
        }
    }
}
```

**Botão direito no controller da api e adicionar controller;
Escolher api controller empty;
Nomear controlador como AlunoController.**

```
using Microsoft.AspNetCore.Mvc;
using MinhaApp.Servico;

namespace MinhaApp.WebAPI.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class AlunoController : ControllerBase
    {
        private readonly IAlunoServico _servico;

        public AlunoController(IAlunoServico servico)
        {
            _servico = servico;
        }

        [HttpPost]
        public IActionResult Adicionar(AlunoDto dto)
        {
            _servico.Adicionar(dto);
            return Ok("Aluno adicionado com sucesso");
        }

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

No Program.cs Da WebAPI adicionar:

```
using MinhaApp.Repositorio;
using MinhaApp.Servico;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDbContext<AppDbContext>(opt =>
    opt.UseInMemoryDatabase("AlunosDB"));
builder.Services.AddScoped<IAlunoRepositorio, AlunoRepositorio>();
```

```
builder.Services.AddScoped<IAlunoServico,AlunoServico>();
```

```
var app = builder.Build();
```

```
app.UseRouting();;
```

```
app.UseEndpoints(endpoints => endpoints.MapControllers());
```

```
app.Run();
```