Após Clonar a solução mudar o card no taskboard

Iniciar criação Solução.

Criar Solução em branco

Uma imagem contendo captura de tela

Descrição gerada automaticamente

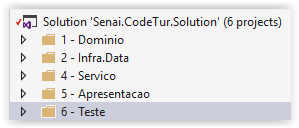
Nome Senai.CodeTur.Solution

Location é o caminho que você fez o clone do seu repositório

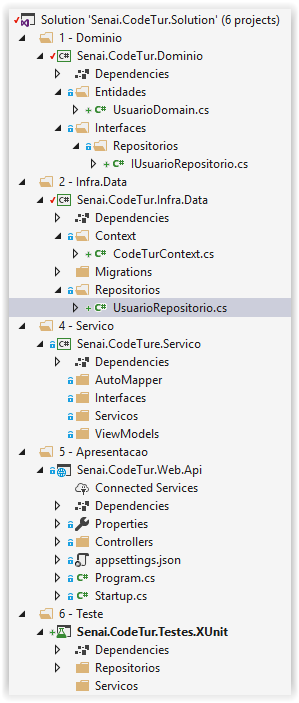
E:\Senai\github\2s-2019-3T\**CodeTur\Backend**

Deixar desmarcado e clicar em ok

Adicionar as pastas abaixo



Criar os projetos em suas respectivas pastas e incluir as pastas dentro dele como figura abaixo



Class Library .Net Core Dominio, Infra.Data e Servicos

Testes Criar Test XUnit

Apresentação criar api

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Deixar desmarcado HTTPS e clicar em OK

Fechar Visual Studio

Ir via cmd até a pasta do projeto e dar

Git add .

Git commit -m “Estrutura projeto”

Git push -u origin master

Verificar no Azure Repos se o mesmo subiu

Iniciar o desenvolvimento

Alterar taskBoard a medida que vai fazendo

Criar classe UsuarioDomain no projeto Dominio, pasta Entidades

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Criar a interface

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Projeto Dominio

Uma imagem contendo captura de tela, monitor

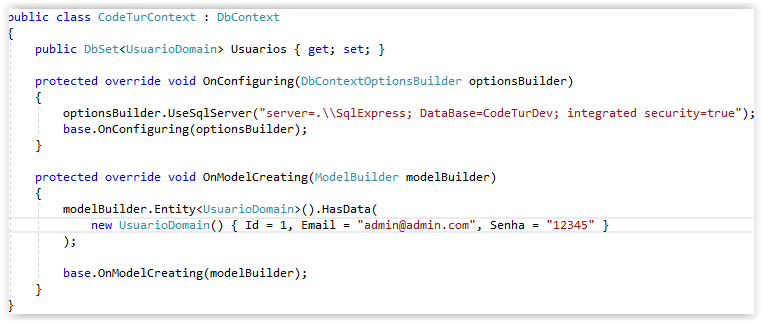
Descrição gerada automaticamente

Projeto Infra.Data Baixar pacotes do nugget

Microsoft.EntityFrameworkCore.SqlServer, Microsoft.EntityFrameworkCore.Design

Microsoft.EntityFrameworkCore.Tools

Criar CodeTurContext na pasta Context



Definir Senai.CodeTur.Infra.Data como projeto inicial Set Startup Project

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

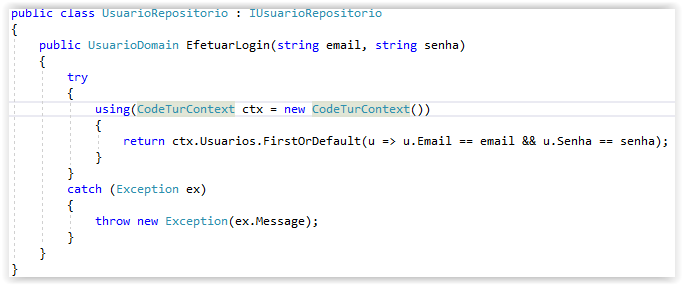
Selecionar Infra.Data como Default Project

Add-Migration Criacao-Banco

Depois Update-Database

Mostrar banco criado

Criar repositório **UsuarioRepositorio herdar IUsuarioRepositorio**



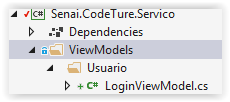
Criar no projeto Testes na pasta Repositórios

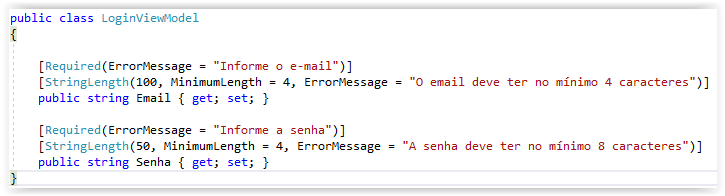


Rodar os tests em Test -> All Tests

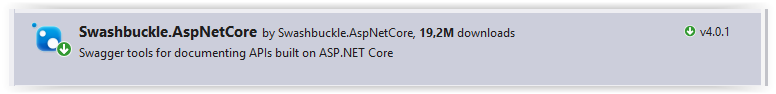
Retirar a senha do efetuar Login para dar erro no teste

Criar LoginViewModel no projeto Serviço





Baixar pacote Swashbuckle.AspNetCore



Configurar Startup.cs

Criar ContaController na Api

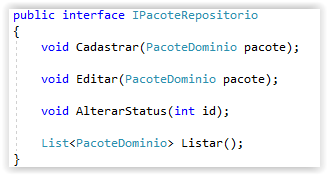
Copiar método Login

Testar postman

Criar PacoteDominio



Criar Interface Pacote



Adicionar no Contexto e executar migration

Criar PacoteRepositorio

Implementar Metodos

Criar Tests para o pacote

Implementar Testes

Criar controller pacote