

CRIAR UM PROJETO DE CADASTRO DE PRODUTOS EM ASP.NET CORE MVC COM BANCO DE DADOS

PRÉ-REQUISITOS:

Para desenvolvimento do projeto ASP.NET Core. usamos:

IDE Visual Studio 2022 (ou superior)

Framework .NET 8 (ou superior)

MySQL Workbench : Ferramenta para gerenciar(SGBD) Sistema Gerenciador de Banco de Dados.

NuGet Package Manager: Para instalar os pacotes necessários. (Mysql Data)

METODOLOGIA ÁGIL COM SCRUM

FERRAMENTA : TRELLO

CRIAR OS SEGUINTE LISTAS

BACKLOG: CRIAR OS SEGUINTE CARTÕES:

- CRIAR O PROJETO ASP.NET CORE MVC
- CRIAR O MODELO FISICO DO BANCO DE DADOS
- CONFIGURAR O BANCO DE DADOS NO VISUAL STUDIO
- CRIAR A PASTA REPOSITORIO
- CRIAR AS MODELS
- CRIAR OS CÓDIGOS DO REPOSITÓRIO(Conexao, LoginRepositorio, ProdutoRepositorio)
- CRIAR OS CONTROLLERS(LoginRepositorio, ProdutoController)
- CRIAR AS VIEWS COM BASE NAS CONTROLLERS
- CONFIGURAR A INJEÇÃO DE DEPENDÊNCIAS
- CRIAR A STRING DE CONEXÃO
- TESTAR O PROJETO

EM ANDAMENTO:

REVISÃO:

CONCLUIDO:

OBS: A SPRINT DE CADA CARTÃO NO BACKLOG COM (NOME DO MEMBRO, ETIQUETA(ESCOLHER A COR QUE DESEJAR),PODE COLOCAR UMA DESCRIÇÃO SE NECESSÁRIO, DATA E A TAREFA SENDO FEITA COLOCA EM ANDAMENTO, TERMINOU DEIXA EM REVISÃO E COLOCA O PROXIMO EM ANDAMENTO, SE TUDO ESTIVER OK COLOQUE EM CONCLUIDO

VERSIONAMENTO DO PROJETO

AO CRIAR O PROJETO NA AREA DE TRABALHO CRIAR O ARQUIVO .gitignore (no terminal do Git Bash, criar no modo raiz: touch .gitignore

NO SITE <https://www.toptal.com/developers/gitignore> ESCOLHER AS OPÇÕES Visual Studio, ASP.NET CORE e clique em criar

copiar o arquivo gerado e colar no .gitignore

AGORA SIM VAMOS INICIALIZAR O GIT

1º git init

2º git config --global user.name "Seu Primeiro Nome"

3º git config --global user.email seuEmail@email.com (sem aspas duplas)

4º git status

5º git add .

6º git commit -m "Criando o Projeto Nome do projeto"

7º ir ao GitHub e criar um repositório com o mesmo nome do projeto e deixar público e criar repositório

8º pegar as 3 linhas criadas e colocar no Git Bash (dar enter)

9º verificar se o projeto ja está no GitHub

AGORA VAMOS CRIAR A BRANCH

10º git checkout -b dev

11º git branch (verifica as branches existentes)

A CADA PASSO DO PROJETO ABAIXO UM COMMIT NA DEV

depois que terminar voltar para a branch master

12º git checkout master

TRAZ TUDO DA BRANCH DEV PARA A MASTER

12º git merge dev

continuando na BRANCH master

13º git push (para subir as atualizações para o gitHub)

PASSO 1: CRIAR O PROJETO ASP.NET CORE MVC (MODEL, VIEW, CONTROLLER)

Abra o Visual Studio e crie um projeto.

Selecione "ASP.NET Core Web App (Model-View-Controller)" e clique em "Add".

Defina o nome do projeto (ex: "ExemploProjeto") e escolha o local no desktop para salvar.

Selecione ".NET 8.0 (ou superior)" e clique em "Criar".

PASSO 2: CONFIGURAR O BANCO DE DADOS MYSQL

Crie o Banco de Dados e depois SALVE o script do banco dentro da pasta do projeto

Abra o MySQL Workbench e crie um banco de dados.

Crie as tabelas necessárias:

Usuarios: (Id, Nome, Email, Senha)

Produtos: (Id, Nome, Descricao, Preco, quantidade)

Instale no Pacote NuGet Package Manager > Gerenciar Pacotes NuGet para Solução):

No Visual Studio, abra o "NuGet Package Manager" (Ferramentas(Tools) OU Botão direito em cima do nome do projeto ir em "NuGet Package Manager".

Pesquise por "MySQL.Data" e instale a versão mais recente.

PASSO 3: CRIAR A PASTA REPOSITÓRIO

Botão direito do mouse no Nome do Projeto Add->Folder-> Crie a Pasta "Repositorio":

Botão direito do mouse em cima de "Repositorio" Add->class->Crie a classe chamada "LoginRepositorio.cs".

Botão direito do mouse em cima de "Repositorio" Add->class->Crie a classe chamada "ProdutoRepositorio.cs".

Não esquecer de importar o `using MySql.Data.MySqlClient` e `using System.Data;`

*****DAR O COMMIT COM A BRANCH DEV *****

PASSO 4: CRIAR OS MODELOS (MODELS)

Botão direito do mouse em cima da pasta "Models".

Botão direito do mouse em cima de "Models" Add->class->Crie a classe "Usuario.cs".

Botão direito do mouse em cima de "Models" Add->class->Crie a classe "Produto.cs".

Código das classes Usuario e Produtos que correspondentes às tabelas do banco de dados:

// Usuario.cs

```
{
    public class Usuario
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public string Email { get; set; }
        public string Senha { get; set; }
    }
}
```

// Produto.cs

```
{
    public class Produto
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public string Descricao { get; set; }
        public decimal Preco { get; set; }
        public int quantidade { get; set; }
    }
}
```

*****DAR O COMMIT COM A BRANCH DEV *****

PASSO 5: CRIAR OS CÓDIGOS DO REPOSITÓRIOS

Aqui é colocado os códigos do **MYSQL** como **SELECT, INSERT, UPDATE, DELETE**

USAR COMO BASE OS CÓDIGOS DO PROJETO CRIADO EM AULA

*****DAR O COMMIT COM A BRANCH DEV *******

PASSO 6: CRIAR OS CONTROLADORES (CONTROLLERS)

USAR COMO BASE OS CÓDIGOS DO PROJETO CRIADO EM AULA

*****DAR O COMMIT COM A BRANCH DEV *******

PASSO 7: CRIAR AS VIEWS (VIEWS)

CRIAR AS VIEWS, LOGIN, CADASTRAR PRODUTO, EDITAR PRODUTO O EXCLUIR SERÁ FEITO POR EVENTO ONCLICK NO PRODUTO

CRIAR A ESTILIAZAÇÃO DO PROJETO COM BOOTSTRAP , COLOCAR UM VÍDEO NA PÁGINA HOME

USAR COMO BASE OS CÓDIGOS DO PROJETO CRIADO EM AULA

*****DAR O COMMIT COM A BRANCH DEV *******

PASSO 8: CONFIGURAR A INJEÇÃO DE DEPENDÊNCIA

Registre as Dependências:

No arquivo **Program.cs** localizado na solução do projeto

Injeção de dependência no LoginRepositorio e ProdutoRepositorio.

```
builder.Services.AddScoped<LoginRepositorio>();  
builder.Services.AddScoped<ProdutoRepositorio>();
```

DAR O COMMIT COM A BRANCH DEV **

PASSO 9: STRING DE CONEXÃO

clique em **appsettings.json** localizado na solução do projeto adicione a connection string do MySQL:

```
{  
  "Logging": {  
    "LogLevel": {  
      "Default": "Information",  
      "Microsoft.AspNetCore": "Warning"  
    }  
  },  
  "AllowedHosts": "*",  
  "ConnectionStrings": {  
    "MySQLConnection": "server=localhost;database=EcommerceDB;user=COLOCAR O USUARIO DO  
    MYSQL;password= COLOCAR A SENHA DO MYSQL"  
  }  
}
```

DAR O COMMIT COM A BRANCH DEV **

PASSO 10: EXECUTAR O PROJETO

Execute o projeto e teste as funcionalidades de cadastro de usuario e depois logar com o usuario cadastrado e cadastrar, editar, excluir o produto e verificar no banco de dados se funcionou corretamente.