

# Guia Prático de CRUD com ASP.NET MVC

Este documento explica de forma clara e organizada como funciona um sistema CRUD (Create, Read, Update, Delete) utilizando ASP.NET MVC com Entity Framework e MySQL.

## 1. Model – Pessoa.cs

A classe Pessoa representa o modelo da aplicação. Ela define a estrutura dos dados que serão armazenados no banco de dados e manipulados pelo sistema.

- 1 Id: chave primária com auto incremento
- 2 Nome: nome da pessoa
- 3 Idade: idade da pessoa

O MySQL gerencia automaticamente o auto-incremento da chave primária, não sendo necessário implementar lógica adicional no código.

## 2. Banco de Dados – BancoDados.cs

Esta classe é responsável pela conexão com o banco de dados e pela configuração do Entity Framework.

O Entity Framework atua como um intermediário entre o código C# e o banco de dados, traduzindo comandos C# em SQL automaticamente.

**DbSet<Pessoa> PessoaDB:** representa a tabela Pessoa no banco de dados e permite realizar operações CRUD.

## 3. Controller – HomeController.cs

O Controller é responsável pela lógica da aplicação e pela comunicação entre a View e o Model.

- 1 Adicionar: insere um novo registro no banco
- 2 Excluir: remove um registro existente
- 3 Editar: carrega os dados para edição
- 4 ConfirmarEdição: salva as alterações no banco

Após a execução de cada função, o ASP.NET MVC renderiza automaticamente a View com o mesmo nome da Action.

## 4. Views – Index.cshtml e Editar.cshtml

As Views são responsáveis pela interface com o usuário. Elas capturam os dados do formulário e exibem as informações vindas do Controller.

O atributo **asp-route-id** permite capturar automaticamente o ID do registro, facilitando as operações de edição e exclusão.

Conclusão: Este fluxo garante uma separação clara de responsabilidades, tornando o sistema mais organizado, escalável e fácil de manter.