

# O Império dos Dados Contra-Ataca:



**Uma Nova Esperança para Dominar e  
Conquistar o Entity Framework Core**

**Ronaldo Silva**

# Guia Simplificado

---

## Entidades no Entity Framework Core

O Entity Framework Core (EF Core) é uma ferramenta poderosa para lidar com dados em aplicações .NET. Ele permite mapear classes C# para tabelas em um banco de dados, simplificando o desenvolvimento de aplicações baseadas em dados. Vamos explorar os principais tópicos relacionados a entidades no EF Core com exemplos práticos e explicativos.



01

**O QUE SÃO ENTIDADES?**

---

# Conectando o código ao banco de dados

No EF Core, uma entidade é uma classe .NET que mapeia uma tabela ou vista no banco de dados. Por padrão, as propriedades da classe mapeiam para as colunas da tabela correspondente.

## Exemplo básico de uma entidade

```
public class Produto
{
    public int Id { get; set; } // Chave primária
    public string Nome { get; set; }
    public decimal Preco { get; set; }
    public DateTime DataCadastro { get; set; }
}
```

Neste exemplo, a classe Produto representa uma tabela Produtos no banco de dados.



02

**TRABALHANDO COM CHAVES PRIMÁRIAS**

---

# Definindo o identificador único das suas tabelas

Toda entidade precisa de uma chave primária para ser rastreada pelo EF Core. Por padrão, o EF Core assume que uma propriedade chamada Id ou <NomeDaClasse> Id é a chave primária.

## Exemplo com chave primária personalizada

```
public class Cliente
{
    public int ClienteId { get; set; } // Chave primária padrão
    public string Nome { get; set; }
    public string Email { get; set; }
}
```

Se você quiser configurar explicitamente uma chave primária, pode usar o Fluent API:

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Cliente>()
        .HasKey(c => c.ClienteId);
}
```

03

## **TIPOS DE PROPIEDADES**

---

# Mapeando diferentes tipos de dados

O EF Core suporta uma ampla variedade de tipos de dados. Aqui estão alguns exemplos comuns:

## Exemplo com diferentes tipos de dados

```
public class Pedido
{
    public int PedidoId { get; set; }
    public string Descricao { get; set; }
    public DateTime DataPedido { get; set; }
    public bool Entregue { get; set; }
    public decimal ValorTotal { get; set; }
}
```

Essas propriedades serão mapeadas automaticamente para tipos correspondentes no banco de dados, como int, nvarchar, datetime, bit e decimal.





04

## **RELACIONAMENTOS ENTRE ENTIDADES**

---

# Definindo como as tabelas se conectam

No EF Core, os relacionamentos são configurados por propriedades de navegação e chaves estrangeiras.

## Relacionamento um-para-muitos

Um cliente pode ter vários pedidos:

```
public class Cliente
{
    public int ClienteId { get; set; }
    public string Nome { get; set; }

    public List<Pedido> Pedidos { get; set; } = new();
}

public class Pedido
{
    public int PedidoId { get; set; }
    public string Descricao { get; set; }

    public int ClienteId { get; set; } // Chave estrangeira
    public Cliente Cliente { get; set; } // Propriedade de navegação
}
```

Configure o relacionamento no Fluent API:

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Pedido>()
        .HasOne(p => p.Cliente)
        .WithMany(c => c.Pedidos)
        .HasForeignKey(p => p.ClienteId);
}
```

Usando entity como um Jedi

05

**CONTROLE TOTAL SOBRE O MAPEAMENTO**

---

# Controle total sobre o mapeamento

A Fluent API permite configurar detalhes adicionais, como restrições e nomes de colunas.

## Exemplo com configurações avançadas

```
Untitled-1
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Produto>()
        .Property(p => p.Nome)
        .HasMaxLength(100)
        .IsRequired();

    modelBuilder.Entity<Produto>()
        .Property(p => p.Preco)
        .HasColumnType("decimal(18,2)");
}
```

Aqui, configuramos um limite de 100 caracteres para o nome e definimos o tipo da coluna Preco no banco de dados.



# CONCLUSÃO

---

# O Coração do Entity

---

As entidades são o coração do EF Core, conectando o seu código ao banco de dados de maneira poderosa e flexível. Com o que você aprendeu aqui, já pode começar a criar modelos ricos e bem configurados. Continue explorando e praticando para aproveitar tudo o que o EF Core oferece!



# AGRADECIMENTOS

---

# Obrigado pela Leitura

Esse Ebook foi gerado com o auxílio de IA e diagramado por mim.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação criteriosa para avaliar os exemplos gerados, por favor utilize-o com cautela. Projeto disponível em:

<https://github.com/ronaldofas/ebook-bootcamp-ia>

## Autor



Ronaldo Silva



Entity Framework

