

Fundamentos de Asp.Net Core 6

Trabajar con datos reales utilizando Entity Framework Core 6

Resumen del módulo



- **Introducción a Entity Framework Core 6**
- **Añadir EF Core a la aplicación**
- **Uso de migraciones**
- **Añadir datos iniciales**

Introducción a Entity Framework Core 6



Casi todas las aplicaciones web
que construyas necesitarán datos de una
base de datos.



Aunque podemos utilizar ADO.NET de bajo nivel combinado con sentencias SQL, utilizaremos Entity Framework Core.

Introducing Entity Framework Core

ORM

LINQ

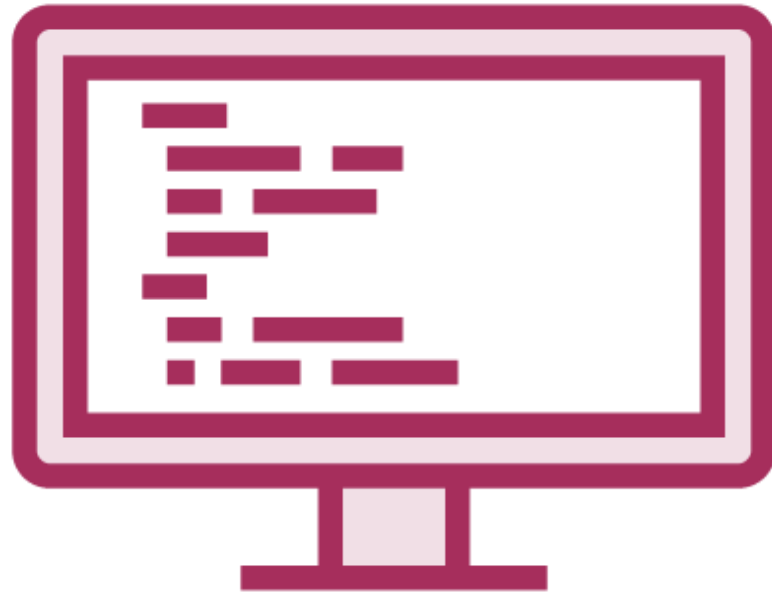
**Ligero y
multiplataforma**

Open-source

**SQL Server y otras
bases de datos no
relacionales**

Code-first

EF Core



Código



Entity Framework



**Base de
datos**

Lo que EF Core hace por ti

? =>Null

Clase

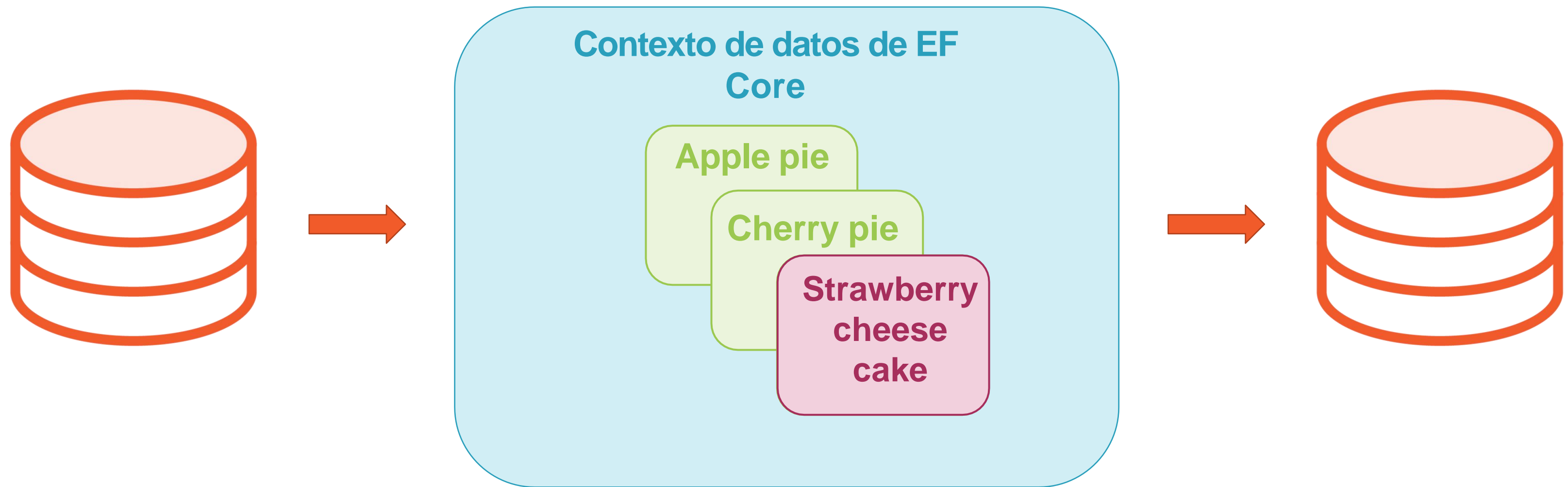
```
public class Pie
{
    public int PieId { get; set; }
    public string? Name { get; set; }
    public string? Description { get; set; }
}
```

Tabla

Pield	Int (PK)
Name	String
Description	string



Seguimiento de cambios en EF Core



Uso de EF Core



Velocidad de desarrollo



Puede trabajar con sentencias SQL



Pero... a veces puede ser menos eficaz que SQL directo.

Añadir EF Core a la aplicación

Añadir EF Core a la aplicación

Paquetes NuGet

```
Microsoft.EntityFrameworkCore.SqlServer
```

```
Microsoft.EntityFrameworkCore.Tools
```

◀ **Paquete NuGet de SQL Server**

◀ **Paquete de ayuda para la consola del gestor de paquetes NuGet**

Añadir EF Core a la aplicación

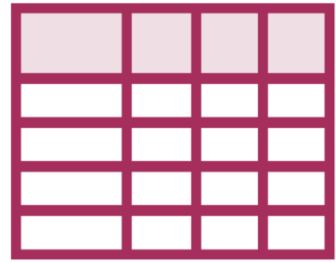
Paquetes NuGet

Clases de dominio

Clases de dominio

```
public class Pie
{
    public int PieId { get; set; }
    public string Name { get; set; }
    public string? ShortDescription { get; set; }
    public decimal Price { get; set; }
    public int CategoryId { get; set; }
    public Category Category { get; set; }
}
```


Creación del mapeado



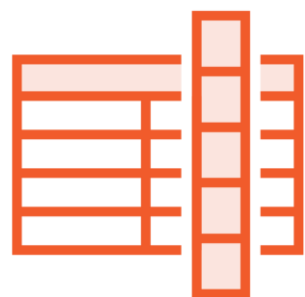
Nombre de tabla y nombre de columna



Pield se convertirá en la clave primaria



CategoryId se convertirá en clave foránea



Tipos de columna utilizados en la base de datos

Añadir EF Core a la aplicación

Paquetes NuGet

Clases de dominio

Contexto de Base de Datos

El contexto de base de datos

```
public class BethanysPieShopDbContext : DbContext
{
    public BethanysPieShopDbContext(DbContextOptions<BethanysPieShopDbContext> options)
        : base(options)
    {
    }

    public DbSet<Pie> Pies { get; set; }
}
```

Añadir EF Core a la aplicación

Paquetes NuGet

Clases de dominio

Contexto de Base de Datos

Configuración de la aplicación

```
{  
  "ConnectionStrings": {  
    "BethanysPieShopDbContextConnection":  
      "Server=(localdb)\\mssqllocaldb;  
        Database=BethanysPieShop;  
        Trusted_Connection=True;  
        MultipleActiveResultSets=true"  
    }  
  }  
}
```

Añadir la cadena de conexión

En appSettings.json

Lectura automática por defecto

```
builder.Services.AddDbContext<BethanysPieShopDbContext>(
    options => {
        options.UseSqlServer(
            builder.Configuration["ConnectionStrings:BethanysPieShopDbContextConnection"]);
    }
);
```

Registro del contexto de base de datos

AddDbContext es un método de extensión

Demo



- **Añadir los paquetes necesarios**
- **Creación del contexto de datos (DbContext)**
- **Cambiar la configuración de la aplicación**

```
_bethanysPieShopDbContext.Pies.Include(c => c.Category).Where(p => p.IsPieOfTheWeek);
```

Consulta de datos mediante LINQ

Añadir nuevos elementos

```
foreach (ShoppingCartItem? shoppingCartItem in shoppingCartItems)
{
    var orderDetail = new OrderDetail
    {
        Amount = shoppingCartItem.Amount,
        PieId = shoppingCartItem.Pie?.PieId,
        Price = shoppingCartItem.Pie?.Price
    };

    order.OrderDetails.Add(orderDetail);
}

_bethanysPieShopDbContext.Orders.Add(order);

_bethanysPieShopDbContext.SaveChanges();
```

Demo

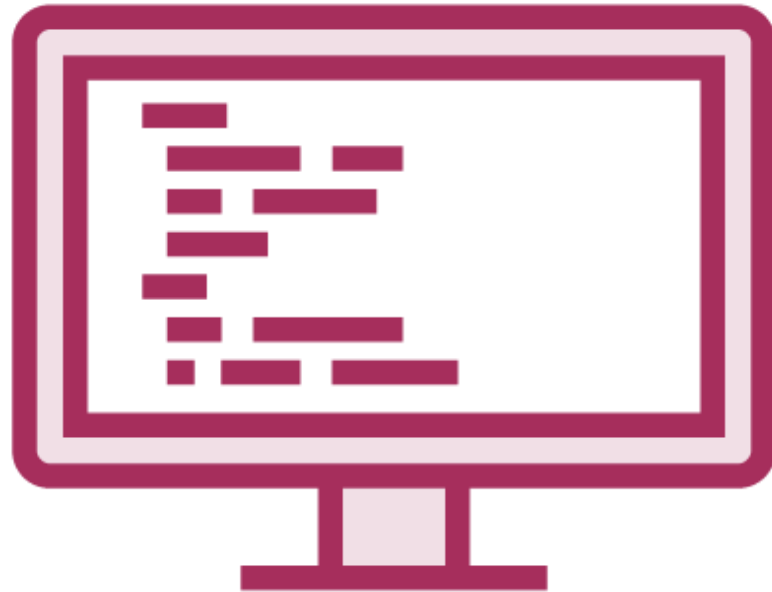


- **Crear el repositorio**

Uso de migraciones

Utilizando EF Core Migrations,
se puede generar código para
sincronizar la base de datos con
el modelo de código.

Crear una migración inicial



Migración de bases de datos

**Consola del administrador
de paquetes**

Comandos

```
>add-migration <MigrationName>
```

Crear una migración inicial



Migración de bases de datos

**Consola del administrador
de paquetes**



Crear base de datos



Base de datos

Comandos

```
>update-database
```

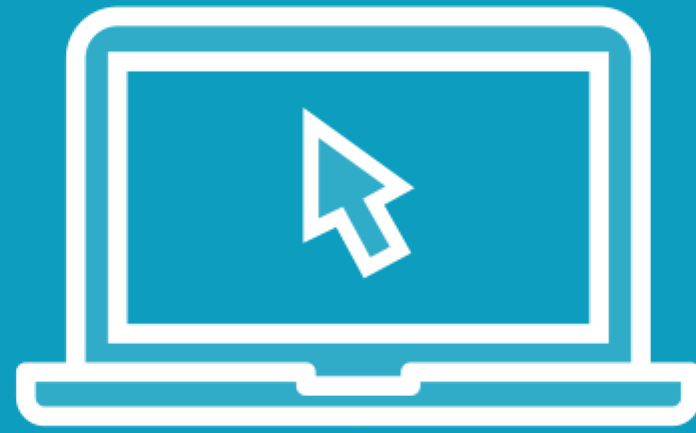
Demo



- **Creación de la migración inicial**
- **Creación de la base de datos**

Añadir datos iniciales

Demo



- **Añadir datos iniciales**

Resumen



- **EF Core es un ORM ligero**
- **Utiliza LINQ para interactuar con la base de datos**
- **Las migraciones se utilizan para sincronizar el modelo y la base de datos**



A continuación:
Navegar por el sitio web