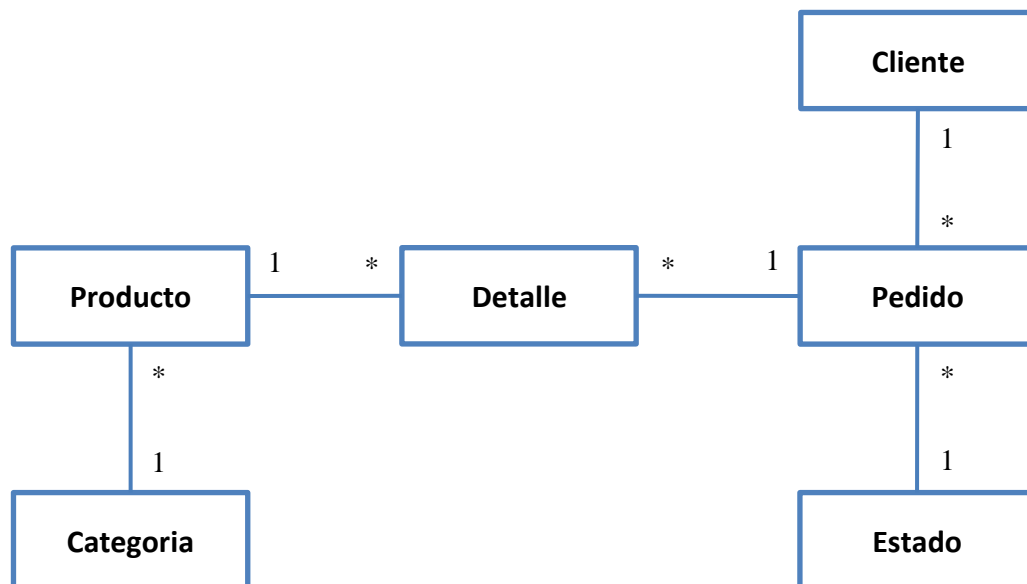


Anexo

Ejemplo de modelo de datos de tienda online

A continuación, se incluye un ejemplo de modelado de datos de una tienda online genérica que puede ayudar a definir de manera adecuada las clases de datos y la clase del contexto de datos en cada caso concreto. Será necesario ajustar la definición de las clases que forman el modelo, así como las especificaciones de validación de las propiedades, considerando las funcionalidades concretas que se requieran y las características del tipo de productos concreto que vaya a ser vendido a través de la aplicación Web de tienda online a desarrollar en cada caso.

- Modelado de datos



- Clases de datos

```
public class Cliente
{
    public int Id { get; set; }
    [Required(ErrorMessage = "El nombre del cliente es un campo requerido.")]
    public string? Nombre { get; set; }
    public string? Email { get; set; }
    public string? Telefono { get; set; }
    public string? Direccion { get; set; }
    public string? Poblacion { get; set; }
    public string?CodigoPostal { get; set; }
    public string? Nif { get; set; }

    public ICollection<Pedido> Pedidos { get; set; }
}
```

Supuesto práctico integrador

```
public class Estado
{
    public int Id { get; set; }
    [Display(Name = "Descripción")]
    [Required(ErrorMessage = "La descripción es un campo requerido.")]
    public string? Description { get; set; }

    public ICollection<Pedido> Pedidos { get; set; }
}
```

```
public class Pedido
{
    [Display(Name = "Núm. Pedido")]
    public int Id { get; set; }
    [Required(ErrorMessage = "La fecha del pedido es un campo requerido.")]
    [DataType(DataType.Date)]
    public DateTime Fecha { get; set; }
    [DataType(DataType.Date)]
    public DateTime? Confirmado { get; set; }
    [DataType(DataType.Date)]
    public DateTime? Preparado { get; set; }
    [DataType(DataType.Date)]
    public DateTime? Enviado { get; set; }
    [DataType(DataType.Date)]
    public DateTime? Cobrado { get; set; }
    [DataType(DataType.Date)]
    public DateTime? Devuelto { get; set; }
    [DataType(DataType.Date)]
    public DateTime? Anulado { get; set; }
    public int ClienteId { get; set; }
    public int EstadoId { get; set; }

    public Cliente? Cliente { get; set; }
    public Estado? Estado { get; set; }
    public ICollection<Detalle> Detalles { get; set; }
}
```

```
public class Detalle
{
    public int Id { get; set; }
    [Display(Name = "Núm. Pedido")]
    public int PedidoId { get; set; }
    [Display(Name = "Id. Producto")]
    public int ProductoId { get; set; }
    public int Cantidad { get; set; }
    [Column(TypeName = "decimal(18, 2)")]
    public decimal Precio { get; set; }
    [Column(TypeName = "decimal(18, 2)")]
    public decimal Descuento { get; set; }

    public virtual Pedido? Pedido { get; set; }
    public virtual Producto? Producto { get; set; }
}
```

```
public class Producto
{
    public int Id { get; set; }
    [Display(Name = "Descripción")]
    [Required(ErrorMessage = "La descripción es un campo requerido.")]
    public string? Descripcion { get; set; }
    public string? Texto { get; set; }
    [DisplayFormat(DataFormatString = "{0:n2}")]
    [Column(TypeName = "decimal(18, 2)")]
    public decimal Precio { get; set; }
    [Display(Name = "Precio")]
    [RegularExpression(@"^-0123456789]+[0-9.]*$",
        ErrorMessage = "El valor introducido debe ser de tipo monetario.")]
    [Required(ErrorMessage = "El precio es un campo requerido")]
    public string PrecioCadena
    {
        get
        {
            return Convert.ToString(Precio).Replace(',', '.');
        }
        set
        {
            Precio = Convert.ToDecimal(value.Replace('.', ','));
        }
    }
    public int? Stock { get; set; }
    public bool? Escaparate { get; set; }
    public string? Imagen { get; set; }
    public int CategoriaId { get; set; }

    public Categoria? Categoria { get; set; }
    public ICollection<Detalle> Detalles { get; set; }
}
```

```
public class Categoria
{
    public int Id { get; set; }
    [Display(Name = "Descripción")]
    [Required(ErrorMessage = "La descripción es un campo requerido.")]
    public string? Descripcion { get; set; }

    public ICollection<Producto> Productos { get; set; }
}
```

Las anotaciones de validación añadidas en algunas propiedades también deberán adaptarse a los requerimientos concretos en cada caso. Las clases de datos anteriores se han definido considerando el comportamiento predeterminado de las convenciones de *Entity Framework Core*: los nombres de propiedad de una clase de datos se usan como nombres de columna, las propiedades de una entidad que se denominan ID o Id se reconocen como clave principal y una propiedad se interpreta como clave ajena si se denomina <nombre de propiedad de navegación><nombre de propiedad clave principal>.

- Clase del contexto de datos

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

    public DbSet<Categoria>? Categorias { get; set; }
    public DbSet<Producto>? Productos { get; set; }
    public DbSet<Cliente>? Clientes { get; set; }
    public DbSet<Estado>? Estados { get; set; }
    public DbSet<Pedido>? Pedidos { get; set; }
    public DbSet<Detalle>? Detalles { get; set; }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        // Deshabilitar la eliminación en cascada en todas las relaciones
        base.OnModelCreating(modelBuilder);
        foreach (var relationship in
            modelBuilder.Model.GetEntityTypes().SelectMany(e => e.GetForeignKeys()))
        {
            relationship.DeleteBehavior = DeleteBehavior.Restrict;
        }
    }
}
```

Considerando el comportamiento predeterminado de las convenciones de *Entity Framework Core*, los nombres de las propiedades *DbSet* se usarán como nombres de las tablas.