

Proyecto Catalogo

Emanuel Cardona García

Manuela Estrada Villada

Yesid García García

Andrés Gaviria Albanes

Instituto Tecnológico Metropolitano
Tecnología en Desarrollo de Software
Medellín, Colombia

2024

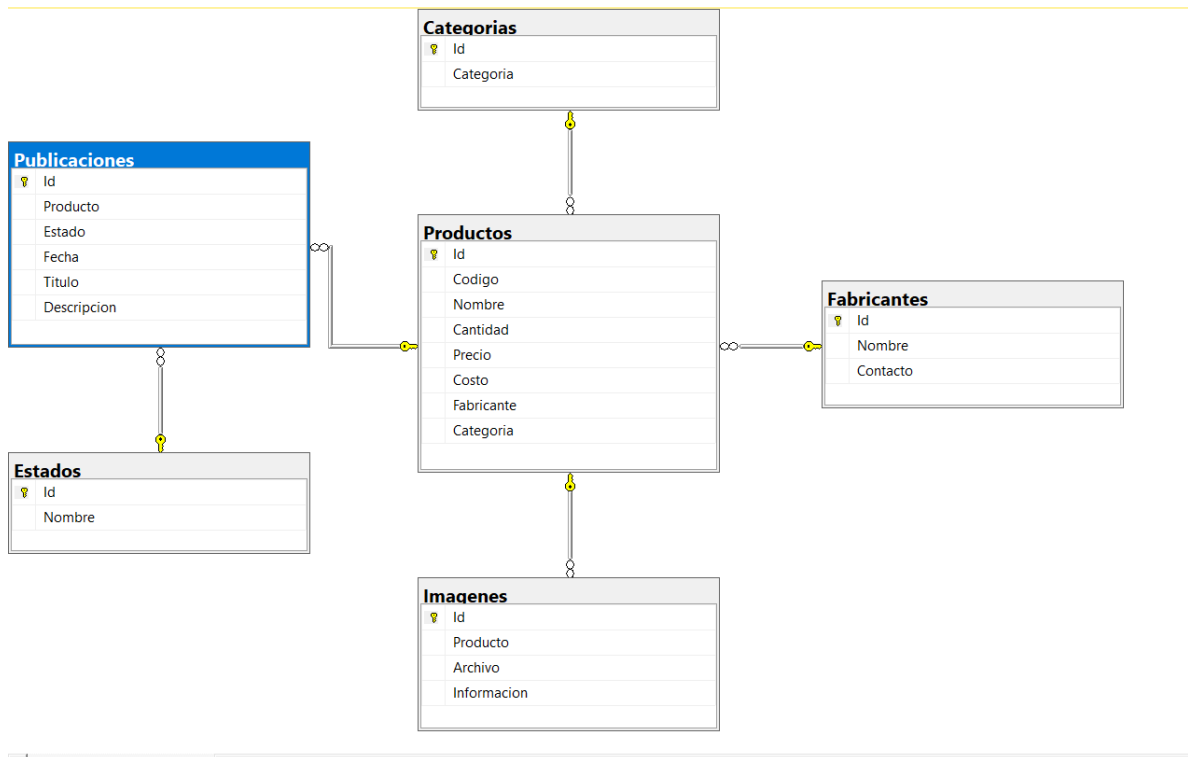
Definición del proyecto:

En este proyecto se busca crear un aplicativo web que permita publicar un producto para su posterior venta. En este aplicativo se almacenarán los datos del producto (código, nombre, cantidad, precio, costo de producción, fabricante, categoría), los datos de la publicación (título, descripción, fecha, estado) y los datos de los fabricantes (nombre, contacto). En la publicación podremos ver la información del producto, la imagen del producto y la información del fabricante además de los datos propios de la publicación como el título, la fecha y la descripción.

Tabla base del proyecto:

ID	Tipo Producto	Cod_Producto	Imagen	Titulo	Descripción	Fecha	Fabricante	Precio	Costo
1	Electronica	E001	####	Apple iPhone 15 Pro Max (256 GB)	Disponible nuevo iPhone 15 Pro Max	2/09/2024	Apple	4850000	1500000
2	Ropa	R001	####	Vestido a cuadros azul	Rebajas colección de verano	26/08/2024	Bershka	170000	90000
3	Mueble	M001	####	Silla ergonomica de oficina	Mejora tus condiciones de trabajo con esta silla	17/06/2024	IKEA	630000	355000
4	Ropa	R002	####	Bolso de mano pequeño	Alto x Ancho x Fondo: 14 x 6 x 8 cm	30/05/2024	Bershka	135000	77000

Diagrama modelo entidad relación:



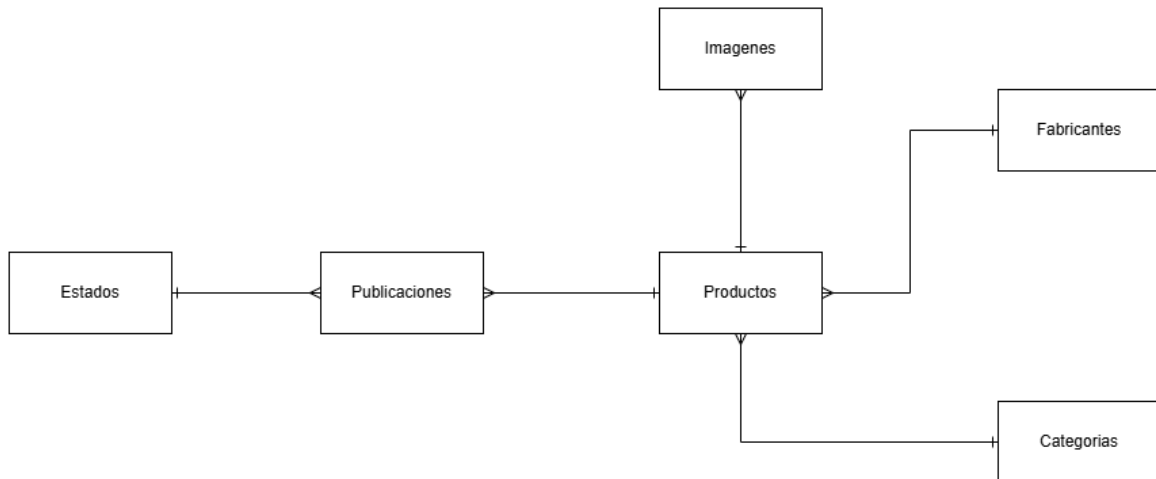


Tabla Estados:

Esta tabla representa los tipos de estados que puede tener una publicación. Contiene los atributos:

- Id [int]: Número identificador único del estado, se genera automáticamente
- Nombre [string]: Nombre del estado

Ejemplo:

ID	Nombre
1	En stock
2	Agotado
3	Inválido

Tabla Categorías:

Esta tabla representa las categorías de los productos. Contiene los atributos:

- Id [int]: Número identificador único de la categoría, se genera automáticamente
- Categoría [string]: Nombre de la categoría

Ejemplo:

ID	Categoría
1	Electronica
2	Ropa
3	Muebles

Tabla Fabricantes:

Esta tabla representa a los fabricantes de los productos. Contiene los atributos:

- Id [int]: Número identificador único del fabricante dentro del sistema, se genera automáticamente
- Nombre [string]: Nombre del fabricante
- Contacto [string]: correo electrónico del fabricante

Ejemplo:

ID	Nombre	Contacto
1	Apple	customer.service@apple.com
2	Bershka	customer.service@bershka.com
3	IKEA	customer.service@ikea.com

Tabla Productos

Esta tabla representa los productos que serán publicados. Contiene los siguientes atributos:

- Id [int]: Número identificador único del producto dentro del sistema, se genera automáticamente
- Código [string]: Referencia del producto proporcionado por el fabricante
- Nombre [string]: Nombre del producto
- Fabricante [int]: Id del fabricante, relaciona la tabla Productos con la tabla Fabricantes
- Categoría [int]: Id de la categoría, relaciona la tabla Productos con la tabla Categorías
- Cantidad [int]: Cantidad en stock del producto
- Precio [float]: Precio del producto
- Costo [float]: Costo de producción del producto

Ejemplo:

ID	Codigo	Nombre	Fabricante	Categoria	Cantidad	Precio	Costo
1	E001	Celular	1	1	100	4850000	1500000
2	R001	Vestido	2	2	0	170000	90000
3	M001	Silla	3	3	20	630000	355000
4	R002	Bolso	2	2	15	135000	77000

Tabla Imágenes:

Esta tabla representa las imágenes del producto que van a ser publicadas. Contiene los atributos:

- Id [int]: Número identificador único de la imagen dentro del sistema, se genera automáticamente
- Producto [int]: Id de producto, relaciona la tabla Imágenes con la tabla Productos
- Archivo [string]: URL de la imagen
- Información [string]: Descripción de la imagen

Ejemplo:

ID	Producto	Archivo	Informacion
1	1	iphone15.png	####
2	2	vestidoss2024.jpeg	####
3	2	vestidoss2024-01.jpeg	####
4	3	silla.png	####
5	4	bolso.png	####

Tabla Publicaciones:

Esta tabla representa las publicaciones dentro del sistema. Contiene los atributos:

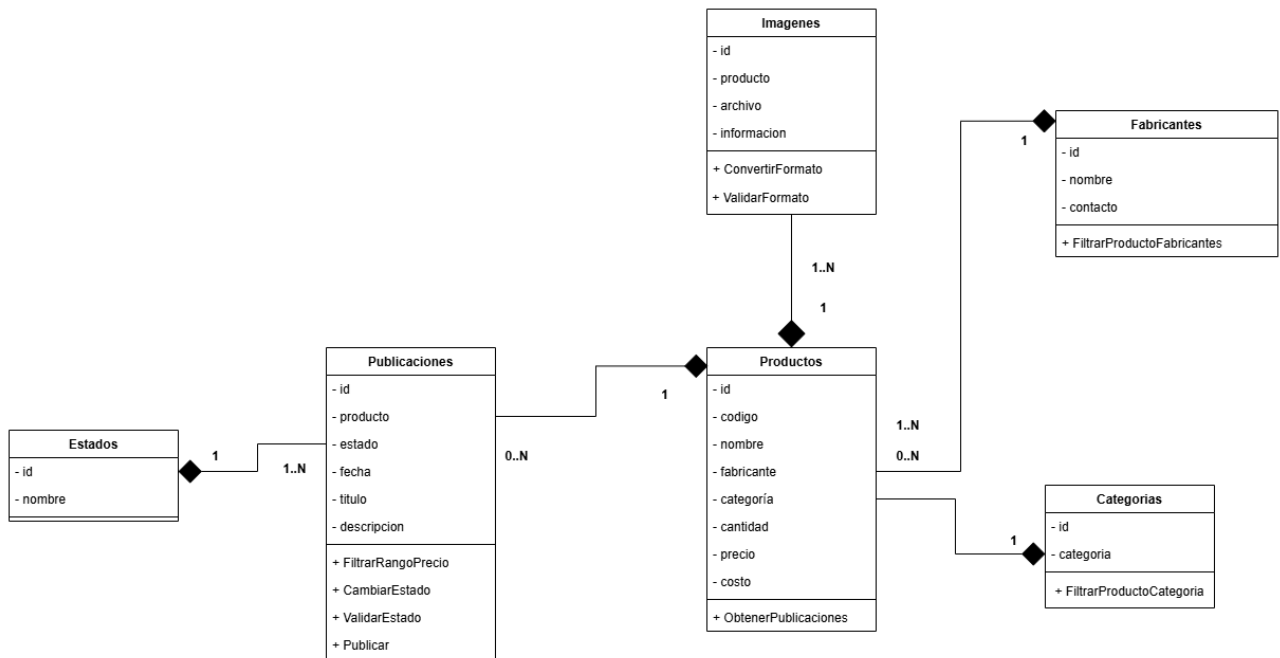
- Id [int]: Número identificador único de la publicación, se genera automáticamente
- Producto [int]: Id de producto, relaciona la tabla Publicaciones con la tabla Productos
- Estado [int]: Id del estado, relaciona la tabla Publicaciones con la tabla Estados
- Fecha [datetime]: Fecha en la que se publicó la publicación
- Título [string]: Titulo de la publicación
- Descripción [string]: Descripción que se hace del producto dentro de la publicación

Ejemplo:

ID	Producto	Estado	Fecha	Titulo	Descripcion
1	1	1	2/09/2024	Apple iPhone 15 Pro Max (256 GB)	Disponible nuevo iPhone 15 Pro Max
2	2	2	26/08/2024	Vestido a cuadros azul	Rebajas colección de verano
3	3	1	17/06/2024	Increíble silla de oficina IKEA	Super cómoda y al mejor precio
4	3	1	15/05/2024	Silla ergonomica de oficina	Mejora tus condiciones de trabajo con esta silla
5	4	3	30/05/2024	Bolso de mano pequeño	Alto x Ancho x Fondo: 14 x 6 x 8 cm

Diagrama de clases:

Dentro del diagrama de clases tenemos una clase para cada tabla del MER, por lo tanto, en este segmento solo se va a abordar los métodos de cada clase, para ver los atributos más detalladamente puede dirigirse al segmento del MER. Cada clase además de los métodos que se muestran contienen su propio CRUD.



Clase Estados:

Esta clase no contiene métodos propios además del CRUD

Clase Categorías:

- FiltrarProductoCategoría(): Lista todos los productos de una categoría en específico

Clase Fabricantes:

- FiltrarProductoFabricantes(): Lista todos los productos de un fabricante en específico

Clase Producto:

- ObtenerPublicaciones(): Obtiene todas las publicaciones de un producto en específico

Clase Imágenes:

- ConvertirFormato(): Cambiar formato de png a jpg o viceversa
- ValidarFormato(): Verificar que el formato de la imagen sea válido

Clase Publicaciones:

- FiltrarRangoPrecio(): Obtiene todos los productos en un rango de precio determinado
- CambiarEstado(): Modificar el estado de una publicación
- ValidarEstado(): Verificar que el estado sea correcto, es decir verificar la cantidad en stock de un producto y que el estado cumpla con este
- Publicar(): Crear la publicación

Código:**Nugets necesarios para el aplicativo:**

Microsoft.EntityFrameworkCore.SqlServer versión 8.08

Script SQL:

```
CREATE DATABASE bd_catalogo
```

```
GO
```

```
USE bd_catalogo
```

```
GO
```

```
CREATE TABLE [Estados] (
```

```
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1,1),
```

```
    [Nombre] NVARCHAR(50) NOT NULL
```

```
);
```

```
GO
```

```
CREATE TABLE [Categorias] (
```

```
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1,1),
```

```
    [Categoria] NVARCHAR(50) NOT NULL
```

```
);
```

GO

```
CREATE TABLE [Fabricantes] (  
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
    [Nombre] NVARCHAR(50) NOT NULL,  
    [Contacto] NVARCHAR(100) NULL  
);
```

GO

```
CREATE TABLE [Productos] (  
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
    [Codigo] NVARCHAR(50) NOT NULL,  
    [Nombre] NVARCHAR(50) NOT NULL,  
    [Cantidad] INT NOT NULL,  
    [Precio] FLOAT NOT NULL,  
    [Costo] FLOAT NOT NULL,  
    [Fabricante] INT NOT NULL,  
    [Categoria] INT NOT NULL,  
    CONSTRAINT [FK_Producto_Fabricante] FOREIGN KEY ([Fabricante])  
REFERENCES [Fabricantes] ([Id]) ON DELETE No Action ON UPDATE No Action,  
    CONSTRAINT [FK_Producto_Categoria] FOREIGN KEY ([Categoria])  
REFERENCES [Categorias] ([Id]) ON DELETE No Action ON UPDATE No Action,  
);
```

GO

```
CREATE TABLE [Imagenes] (  
    [Id] INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
    [Producto] INT NOT NULL,
```



```
[Archivo] NVARCHAR(255) NOT NULL,  
[Informacion] TEXT NULL,  
        CONSTRAINT [FK_Imagen_Producto] FOREIGN KEY ([Producto])  
REFERENCES [Productos] ([Id]) ON DELETE No Action ON UPDATE No Action  
);  
GO
```

```
CREATE TABLE [Publicaciones] (  
    [Id] INT NOT NULL IDENTITY(1,1) PRIMARY KEY,  
    [Producto] INT NOT NULL,  
    [Estado] INT NOT NULL,  
    [Fecha] SMALLDATETIME NOT NULL DEFAULT GETDATE(),  
    [Titulo] NVARCHAR(255) NOT NULL,  
    [Descripcion] TEXT NULL,  
        CONSTRAINT [FK_Publicaciones_Producto] FOREIGN KEY ([Producto])  
REFERENCES [Productos] ([Id]) ON DELETE No Action ON UPDATE No Action,  
        CONSTRAINT [FK_Publicaciones_Estado] FOREIGN KEY ([Estado])  
REFERENCES [Estados] ([Id]) ON DELETE No Action ON UPDATE No Action  
);  
GO
```

-- Insertar datos en la tabla Estados

```
INSERT INTO Estados ([Nombre]) VALUES ('Disponible');  
INSERT INTO Estados ([Nombre]) VALUES ('No Disponible');  
GO
```

-- Insertar datos en la tabla Categorías

```
INSERT INTO Categorías ([Categoría]) VALUES ('Electrónica');
```

```
INSERT INTO Categorias ([Categoria]) VALUES ('Hogar');
```

```
GO
```

-- Insertar datos en la tabla Fabricantes

```
INSERT INTO Fabricantes ([Nombre], [Contacto]) VALUES ('Sony',  
'contacto@sony.com');
```

```
INSERT INTO Fabricantes ([Nombre], [Contacto]) VALUES ('Samsung',  
'contacto@samsung.com');
```

```
GO
```

-- Insertar datos en la tabla Productos

```
INSERT INTO Productos ([Codigo], [Nombre], [Fabricante], [Categoria], [Cantidad],  
[Precio], [Costo])
```

```
VALUES ('P001', 'Televisor LED', 1, 1, 50, 3999960.00,3500000.00);
```

```
INSERT INTO Productos ([Codigo], [Nombre], [Fabricante], [Categoria], [Cantidad],  
[Precio],[Costo])
```

```
VALUES ('P002', 'Refrigerador', 2, 2, 30, 3199960.00,2980000.00);
```

```
GO
```

-- Insertar datos en la tabla Publicaciones

```
INSERT INTO Publicaciones ([Producto], [Estado], [Fecha], [Titulo], [Descripcion])
```

```
VALUES (1, 1, GETDATE(), 'Venta de Televisores', 'Oferta en televisores LED Sony');
```

```
INSERT INTO Publicaciones ([Producto], [Estado], [Fecha], [Titulo], [Descripcion])
```

```
VALUES (2, 2, GETDATE(), 'Refrigeradores en stock', 'Refrigeradores Samsung a  
buen precio');
```

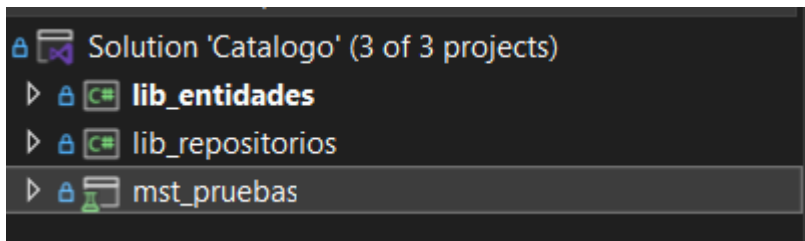
```
GO
```

-- Insertar datos en la tabla Imagenes

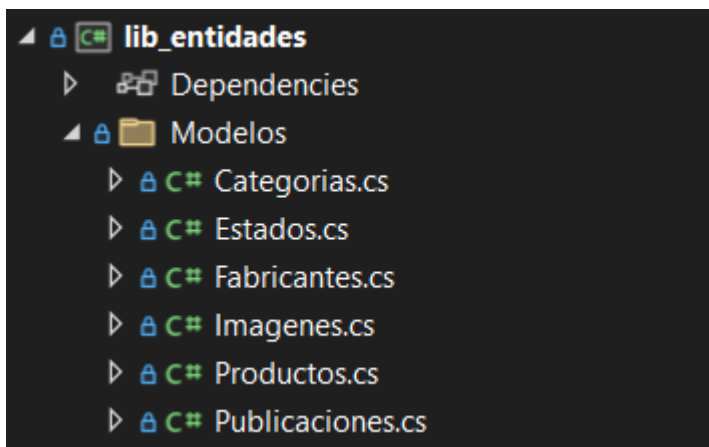
```
INSERT INTO Imagenes ([Producto], [Archivo], [Informacion])
```

```
VALUES (1, 'televisor_sony.jpg', 'Imagen de un televisor LED Sony');  
INSERT INTO Imagenes ([Producto], [Archivo], [Informacion])  
VALUES (2, 'refrigerador_samsung.jpg', 'Imagen de un refrigerador Samsung');  
GO
```

Estructuración del proyecto:



lib_entidades: Contiene la carpeta Modelos la cual guarda las clases que planteamos previamente con sus respectivos atributos y métodos.



Las clases de esta librería utilizan los siguientes namespace:

- using **System.ComponentModel.DataAnnotations;**
- using **System.ComponentModel.DataAnnotations.Schema;**

Categorias.cs:

```

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace lib_entidades.Modelos
{
    23 references
    public class Categorias
    {
        7 references
        [Key] public int Id { get; set; }
        2 references
        public string? Categoria { get; set; }

        0 references
        public void FiltrarProductoCategoria()
        {
            // contenido del método
        }
    }
}

```

Estados.cs:

```

1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace lib_entidades.Modelos
5  {
6      23 references
7      public class Estados
8      {
9          7 references
10         [Key] public int Id { get; set; }
11         2 references
12         public string? Nombre { get; set; }
13     }
14 }

```

Fabricantes.cs:

```
1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace lib_entidades.Modelos
5  {
6      23 references
7      public class Fabricantes
8      {
9          7 references
10         [Key] public int Id { get; set; }
11         2 references
12         public string? Nombre { get; set; }
13         1 reference
14         public string? Contacto { get; set; }
15
16         0 references
17         public void FiltrarProductoFabricantes()
18         {
19             // contenido del método
20         }
21     }
22 }
```

Imagenes.cs:

```
4  namespace lib_entidades.Modelos
5  {
6      22 references
7      public class Imagenes
8      {
9          7 references
10         [Key] public int Id { get; set; }
11         1 reference
12         public int Producto { get; set; }
13         2 references
14         public string? Archivo { get; set; }
15         1 reference
16         public string? Informacion { get; set; }
17
18         0 references
19         [NotMapped] public Productos? _Producto { get; set; }
20
21         0 references
22         public void ConvertirFormato()
23         {
24             // contenido del método
25         }
26
27         0 references
28         public void ValidarFormato()
29         {
30             // contenido del método
31         }
32     }
33 }
```

Productos.cs:

```
4  namespace lib_entidades.Modelos
5  {
6      24 references
7      public class Productos
8      {
9          7 references
10         [Key] public int Id { get; set; }
11         1 reference
12         public string? Codigo { get; set; }
13         1 reference
14         public string? Nombre { get; set; }
15         2 references
16         public int Cantidad { get; set; }
17         1 reference
18         public double Precio { get; set; }
19         1 reference
20         public double Costo { get; set; }
21         1 reference
22         public int Fabricante { get; set; }
23         1 reference
24         public int Categoria { get; set; }
25
26         0 references
27         [NotMapped] public Fabricantes? _Fabricante { get; set; }
28         0 references
29         [NotMapped] public Categorias? _Categoria { get; set; }
30
31         0 references
32         public void ObtenerPublicaciones()
33         {
34             // contenido del método
35         }
36     }
37 }
```

Publicaciones.cs:

```

4  namespace Lib_entidades.Modelos
5  {
6      22 references
7      public class Publicaciones
8      {
9          7 references
10         [Key] public int Id { get; set; }
11         1 reference
12         public int Producto { get; set; }
13         1 reference
14         public int Estado { get; set; }
15         1 reference
16         public DateTime Fecha { get; set; }
17         2 references
18         public string? Titulo { get; set; }
19         1 reference
20         public string? Descripcion { get; set; }
21
22         0 references
23         [NotMapped] public Productos? _Producto { get; set; }
24         0 references
25         [NotMapped] public Estados? _Estado { get; set; }
26     }
27 }

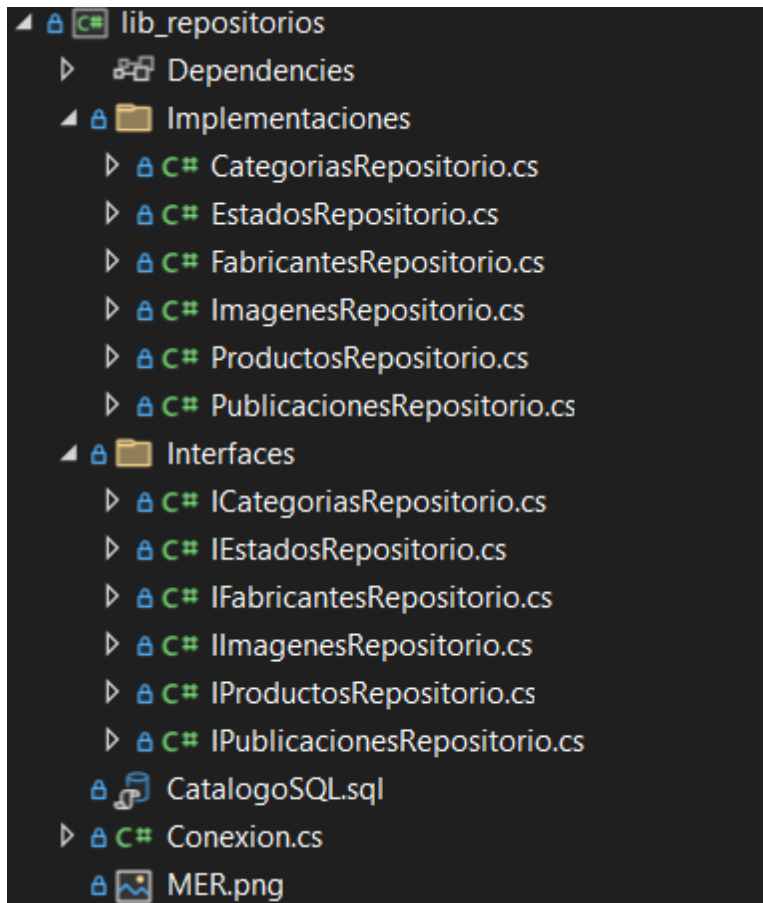
```

```

20  0 references
21  public void Publicar()
22  {
23      // contenido del método
24  }
25  0 references
26  public void ValidarEstado()
27  {
28      // contenido del método
29  }
30  0 references
31  public void CambiarEstado()
32  {
33      // contenido del método
34  }
35  0 references
36  public void FiltrarRangoPrecio()
37  {
38      // contenido del método
39  }
40 }

```

lib_repositorios: Contiene principalmente dos carpetas Implementaciones e Interfaces y una clase Conexión fuera de las carpetas.



Conexión.cs:

Esta clase utiliza los siguientes namespace:

- using **lib_entidades.Modelos;**
- using **Microsoft.EntityFrameworkCore;**
- using **System.Linq.Expressions;**


```

5  namespace lib_repositorios
6  {
7      18 references
8      public class Conexion : DbContext
9      {
10         7 references
11         public string? StringConnection { get; set; }
12
13         0 references
14         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
15         {
16             optionsBuilder.UseSqlServer(this.StringConnection!, p => { });
17             optionsBuilder.UseQueryTrackingBehavior(QueryTrackingBehavior.NoTracking);
18         }
19
20         0 references
21         protected DbSet<Estados>? Estados { get; set; }
22         0 references
23         protected DbSet<Categorias>? Categorias { get; set; }
24         0 references
25         protected DbSet<Fabricantes>? Fabricantes { get; set; }
26         0 references
27         protected DbSet<Productos>? Productos { get; set; }
28         0 references
29         protected DbSet<Imagenes>? Imagenes { get; set; }
30         0 references
31         protected DbSet<Publicaciones>? Publicaciones { get; set; }
32         0 references

```

```

23         0 references
24         public virtual DbSet<T> ObtenerSet<T>() where T : class, new()
25         {
26             return this.Set<T>();
27         }
28
29         6 references
30         public virtual List<T> Listar<T>() where T : class, new()
31         {
32             return this.Set<T>().ToList();
33         }
34
35         6 references
36         public virtual List<T> Buscar<T>(Expression<Func<T, bool>> condiciones) where T : class, new()
37         {
38             return this.Set<T>().Where(condiciones).ToList();
39         }
40
41         0 references
42         public virtual bool Existe<T>(Expression<Func<T, bool>> condiciones) where T : class, new()
43         {
44             return this.Set<T>().Any(condiciones);
45         }
46
47         6 references
48         public virtual void Guardar<T>(T entidad) where T : class, new()
49         {
50             this.Set<T>().Add(entidad);
51         }

```

```

48     public virtual void Modificar<T>(T entidad) where T : class
49     {
50         var entry = this.Entry(entidad);
51         entry.State = EntityState.Modified;
52     }
53
54     6 references
55     public virtual void Borrar<T>(T entidad) where T : class, new()
56     {
57         this.Set<T>().Remove(entidad);
58     }
59
60     0 references
61     public virtual void Separar<T>(T entidad) where T : class, new()
62     {
63         this.Entry(entidad).State = EntityState.Detached;
64     }
65
66     18 references
67     public virtual void GuardarCambios()
68     {
69         this.SaveChanges();
70     }

```

Carpeta Interfaces:

Todas las clases de esta carpeta usan los siguientes namespace:

- using **lib_entidades.Modelos**;
- using **System.Linq.Expressions**;

ICategoriasRepositorio:

```

1  using lib_entidades.Modelos;
2  using System.Linq.Expressions;
3
4  namespace lib_repositorios.Interfaces
5  {
6      2 references
7      public interface ICategoriasRepositorio
8      {
9          2 references
10         List<Categorias> Listar();
11         4 references
12         List<Categorias> Buscar(Expression<Func<Categorias, bool>> condiciones);
13         2 references
14         Categorias Guardar(Categorias entidad);
15         2 references
16         Categorias Modificar(Categorias entidad);
17         2 references
18         Categorias Borrar(Categorias entidad);
19     }

```

IEstadosRepositorio:

```
1  using lib_entidades.Modelos;
2  using System.Linq.Expressions;
3
4  namespace lib_repositorios.Interfaces
5  {
6      2 references
7      public interface IEstadosRepositorio
8      {
9          2 references
10         List<Estados> Listar();
11         4 references
12         List<Estados> Buscar(Expression<Func<Estados, bool>> condiciones);
13         2 references
14         Estados Guardar(Estados entidad);
15         2 references
16         Estados Modificar(Estados entidad);
17         2 references
18         Estados Borrar(Estados entidad);
19     }
20 }
```

IFabricantesRepositorio:

```
1  using lib_entidades.Modelos;
2  using System.Linq.Expressions;
3
4  namespace lib_repositorios.Interfaces
5  {
6      2 references
7      public interface IFabricantesRepositorio
8      {
9          2 references
10         List<Fabricantes> Listar();
11         4 references
12         List<Fabricantes> Buscar(Expression<Func<Fabricantes, bool>>condiciones);
13         2 references
14         Fabricantes Guardar(Fabricantes entidad);
15         2 references
16         Fabricantes Modificar(Fabricantes entidad);
17         2 references
18         Fabricantes Borrar(Fabricantes entidad);
19     }
20 }
```

IImagenesRepositorio:

```
1  using lib_entidades.Modelos;
2  using System.Linq.Expressions;
3
4  namespace lib_repositorios.Interfaces
5  {
6      2 references
7      public interface IImagenesRepositorio
8      {
9          2 references
10         List<Imagenes> Listar();
11         4 references
12         List<Imagenes> Buscar(Expression<Func<Imagenes, bool>> condiciones);
13         2 references
14         Imagenes Guardar(Imageenes entidad);
15         2 references
16         Imagenes Modificar(Imageenes entidad);
17         2 references
18         Imagenes Borrar(Imageenes entidad);
19     }
20 }
```

IProductosRepositorio:

```
1  using lib_entidades.Modelos;
2  using System.Linq.Expressions;
3
4  namespace lib_repositorios.Interfaces
5  {
6      2 references
7      public interface IProductosRepositorio
8      {
9          2 references
10         List<Productos> Listar();
11         4 references
12         List<Productos> Buscar(Expression<Func<Productos, bool>> condiciones);
13         2 references
14         Productos Guardar(Productos entidad);
15         2 references
16         Productos Modificar(Productos entidad);
17         2 references
18         Productos Borrar(Productos entidad);
19     }
20 }
```

IPublicacionesRepositorio:

```
1  using lib_entidades.Modelos;
2  using System.Linq.Expressions;
3
4  namespace lib_repositorios.Interfaces
5  {
6      2 references
7      public interface IPublicacionesRepositorio
8      {
9          2 references
10         List<Publicaciones> Listar();
11         4 references
12         List<Publicaciones> Buscar(Expression<Func<Publicaciones, bool>> condiciones);
13         2 references
14         Publicaciones Guardar(Publicaciones entidad);
15         2 references
16         Publicaciones Modificar(Publicaciones entidad);
17         2 references
18         Publicaciones Borrar(Publicaciones entidad);
19     }
20 }
```

Carpeta Implementaciones:

Todas las clases de esta carpeta usan los siguientes namespaces:

- using lib_entidades.Modelos;
- using lib_repositorios.Interfaces;
- using System.Linq.Expressions;

CategoriasRepositorio:

```
5  namespace lib_repositorios.Implementaciones
6  {
7      2 references
8      public class CategoriasRepositorio : ICategoriasRepositorio
9      {
10         private Conexion? conexion = null;
11
12         1 reference
13         public CategoriasRepositorio(Conexion conexion)
14         {
15             this.conexion = conexion;
16         }
17
18         2 references
19         public List<Categorias> Listar()
20         {
21             return conexion!.Listar<Categorias>();
22         }
23
24         4 references
25         public List<Categorias> Buscar(Expression<Func<Categorias, bool>> condiciones)
26         {
27             return conexion!.Buscar(condiciones);
28         }
29
30         2 references
31         public Categorias Guardar(Categorias entidad)
32         {
33             conexion!.Guardar(entidad);
34             conexion!.GuardarCambios();
35             return entidad;
36         }
37     }
38 }
```

```
31 2 references
32 public Categorias Modificar(Categorias entidad)
33 {
34     conexion!.Modificar(entidad);
35     conexion!.GuardarCambios();
36     return entidad;
37 }
38 2 references
39 public Categorias Borrar(Categorias entidad)
40 {
41     conexion!.Borrar(entidad);
42     conexion!.GuardarCambios();
43     return entidad;
44 }
45 }
46 }
```

EstadosRepositorio:

```
5 namespace lib_repositorios.Implementaciones
6 {
7     2 references
8     public class EstadosRepositorio : IEstadosRepositorio
9     {
10         private Conexion? conexion = null;
11
12         1 reference
13         public EstadosRepositorio(Conexion conexion)
14         {
15             this.conexion = conexion;
16
17         2 references
18         public List<Estados> Listar()
19         {
20             return conexion!.Listar<Estados>();
21         }
22
23         4 references
24         public List<Estados> Buscar(Expression<Func<Estados, bool>> condiciones)
25         {
26             return conexion!.Buscar(condiciones);
27         }
28
29         2 references
30         public Estados Guardar(Estados entidad)
31         {
32             conexion!.Guardar(entidad);
33             conexion!.GuardarCambios();
34             return entidad;
35         }
36     }
37 }
```

```
31 2 references
32 public Estados Modificar(Estados entidad)
33 {
34     conexion!.Modificar(entidad);
35     conexion!.GuardarCambios();
36     return entidad;
37 }
38 2 references
39 public Estados Borrar(Estados entidad)
40 {
41     conexion!.Borrar(entidad);
42     conexion!.GuardarCambios();
43     return entidad;
44 }
45 }
46 }
```

FabricantesRepositorio:

```
5 namespace lib_repositorios.Implementaciones
6 {
7     2 references
8     public class FabricantesRepositorio : IFabricantesRepositorio
9     {
10         private Conexion? conexion = null;
11
12         1 reference
13         public FabricantesRepositorio(Conexion conexion)
14         {
15             this.conexion = conexion;
16         }
17
18         2 references
19         public List<Fabricantes> Listar()
20         {
21             return conexion!.Listar<Fabricantes>();
22         }
23
24         4 references
25         public List<Fabricantes> Buscar(Expression<Func<Fabricantes, bool>> condiciones)
26         {
27             return conexion!.Buscar(condiciones);
28         }
29
30         2 references
31         public Fabricantes Guardar(Fabricantes entidad)
32         {
33             conexion!.Guardar(entidad);
34             conexion!.GuardarCambios();
35             return entidad;
36         }
37     }
38 }
```

```

31 2 references
32 public Fabricantes Modificar(Fabricantes entidad)
33 {
34     conexion!.Modificar(entidad);
35     conexion!.GuardarCambios();
36     return entidad;
37 }
38 2 references
39 public Fabricantes Borrar(Fabricantes entidad)
40 {
41     conexion!.Borrar(entidad);
42     conexion!.GuardarCambios();
43     return entidad;
44 }
45 }
46

```

ImagenesRepositorio:

```

5 namespace Lib_repositorios.Implementaciones
6 {
7     2 references
8     public class ImagenesRepositorio : IImagenesRepositorio
9     {
10         private Conexion? conexion = null;
11
12         1 reference
13         public ImagenesRepositorio(Conexion conexion)
14         {
15             this.conexion = conexion;
16         }
17
18         2 references
19         public List<Imagenes> Listar()
20         {
21             return conexion!.Listar<Imagenes>();
22         }
23
24         4 references
25         public List<Imagenes> Buscar(Expression<Func<Imagenes, bool>> condiciones)
26         {
27             return conexion!.Buscar(condiciones);
28         }
29
30         2 references
31         public Imagenes Guardar(Imagenes entidad)
32         {
33             conexion!.Guardar(entidad);
34             conexion!.GuardarCambios();
35             return entidad;
36         }
37     }
38 }

```



```

31 2 references
32 public Imagenes Modificar(Imagenes entidad)
33 {
34     conexion!.Modificar(entidad);
35     conexion!.GuardarCambios();
36     return entidad;
37 }
38 2 references
39 public Imagenes Borrar(Imagenes entidad)
40 {
41     conexion!.Borrar(entidad);
42     conexion!.GuardarCambios();
43     return entidad;
44 }
45 }
46

```

ProductosRepositorio:

```

5 namespace lib_repositorios.Implementaciones
6 {
7     2 references
8     public class ProductosRepositorio : IProductosRepositorio
9     {
10         private Conexion? conexion = null;
11
12         1 reference
13         public ProductosRepositorio(Conexion conexion)
14         {
15             this.conexion = conexion;
16         }
17
18         2 references
19         public List<Productos> Listar()
20         {
21             return conexion!.Listar<Productos>();
22         }
23
24         4 references
25         public List<Productos> Buscar(Expression<Func<Productos, bool>> condiciones)
26         {
27             return conexion!.Buscar(condiciones);
28         }
29
30         2 references
31         public Productos Guardar(Productos entidad)
32         {
33             conexion!.Guardar(entidad);
34             conexion!.GuardarCambios();
35             return entidad;
36         }
37     }
38 }

```

```

31 2 references
32 public Productos Modificar(Productos entidad)
33 {
34     conexion!.Modificar(entidad);
35     conexion!.GuardarCambios();
36     return entidad;
37 }
38 2 references
39 public Productos Borrar(Productos entidad)
40 {
41     conexion!.Borrar(entidad);
42     conexion!.GuardarCambios();
43     return entidad;
44 }
45 }
46

```

PublicacionRepositorio:

```

5 namespace lib_repositorios.Implementaciones
6 {
7     2 references
8     public class PublicacionesRepositorio : IPublicacionesRepositorio
9     {
10         private Conexion? conexion = null;
11
12         1 reference
13         public PublicacionesRepositorio(Conexion conexion)
14         {
15             this.conexion = conexion;
16         }
17
18         2 references
19         public List<Publicaciones> Listar()
20         {
21             return conexion!.Listar<Publicaciones>();
22         }
23
24         4 references
25         public List<Publicaciones> Buscar(Expression<Func<Publicaciones, bool>> condiciones)
26         {
27             return conexion!.Buscar(condiciones);
28         }
29
30         2 references
31         public Publicaciones Guardar(Publicaciones entidad)
32         {
33             conexion!.Guardar(entidad);
34             conexion!.GuardarCambios();
35             return entidad;
36         }
37     }
38 }

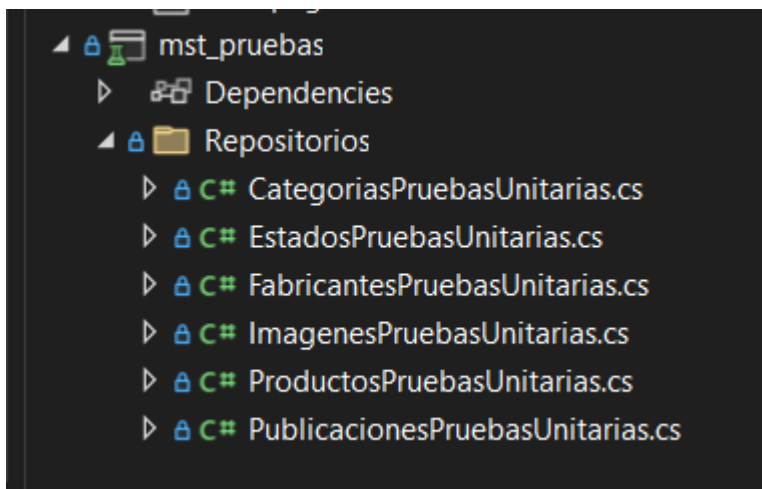
```

```

31 2 references
32 public Publicaciones Modificar(Publicaciones entidad)
33 {
34     conexion!.Modificar(entidad);
35     conexion!.GuardarCambios();
36     return entidad;
37 }
38 2 references
39 public Publicaciones Borrar(Publicaciones entidad)
40 {
41     conexion!.Borrar(entidad);
42     conexion!.GuardarCambios();
43     return entidad;
44 }
45 }
46

```

Mst_pruebas: Contiene la carpeta repositorios con todas las clases para las pruebas unitarias.



Todas las clases de la carpeta Repositorios usan las siguientes librerías:

- using **lib_entidades.Modelos**;
- using **lib_repositorios**;
- using **lib_repositorios.Implementaciones**;
- using **lib_repositorios.Interfaces**;

En las siguientes clases se encuentra la cadena de conexión (línea 16) para conectarse a la base de datos, esta se puede modificar para conectarse al servidor de su máquina, tener en cuenta que la base de datos debe existir en el servidor local.

CategoríasPruebasUnitarias:

```
6 namespace mst_pruebas.Repositorios
7 {
8     [TestClass]
9     public class CategoríasPruebasUnitarias
10     {
11         private ICategoríasRepositorio? iRepositorio = null;
12         private Categorías? entidad = null;
13         public CategoríasPruebasUnitarias()
14         {
15             var conexion = new Conexion();
16             conexion.StringConnection = "server=M4NU_HELPER\\DEV;database=bd_catalogo;uid=sa;pwd=STEMgirls>>>; TrustServerCertificate = true; ";
17             iRepositorio = new CategoríasRepositorio(conexion);
18         }
19
20         [TestMethod]
21         public void Ejecutar()
22         {
23             Guardar();
24             Listar();
25             Buscar();
26             Modificar();
27             Borrar();
28         }
29         private void Guardar()
30         {
31             entidad = new Categorías()
32             {
33                 Categoria = "Prueba",
34             };
35             entidad = iRepositorio!.Guardar(entidad);
36             Assert.IsTrue(entidad.Id != 0);
37         }
```

```
38         private void Listar()
39         {
40             var lista = iRepositorio!.Listar();
41             Assert.IsTrue(lista.Count > 0);
42         }
43         private void Buscar()
44         {
45             var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
46             Assert.IsTrue(lista.Count > 0);
47         }
48         private void Modificar()
49         {
50             entidad!.Categoria = "Prueba Modificada";
51             entidad = iRepositorio!.Modificar(entidad);
52             var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
53             Assert.IsTrue(lista.Count > 0);
54         }
55         private void Borrar()
56         {
57             entidad = iRepositorio!.Borrar(entidad!);
58             var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
59             Assert.IsTrue(lista.Count == 0);
60         }
61     }
62 }
63 }
```

EstadosPruebasUnitarias:

```
6 namespace mst_pruebas.Repositorios
7 {
8     [TestClass]
9     public class EstadosPruebasUnitarias
10     {
11         private IEstadosRepositorio? iRepositorio = null;
12         private Estados? entidad = null;
13         public EstadosPruebasUnitarias()
14         {
15             var conexion = new Conexion();
16             conexion.StringConnection = "server=M4NU_HELPER\\DEV;database=bd_catalogo;uid=sa;pwd=STEMgirls>>>; TrustServerCertificate = true; ";
17             iRepositorio = new EstadosRepositorio(conexion);
18         }
19
20         [TestMethod]
21         public void Ejecutar()
22         {
23             Guardar();
24             Listar();
25             Buscar();
26             Modificar();
27             Borrar();
28         }
29
30         private void Guardar()
31         {
32             entidad = new Estados()
33             {
34                 Nombre = "Prueba",
35             };
36             entidad = iRepositorio!.Guardar(entidad);
37             Assert.IsTrue(entidad.Id != 0);
38         }
39     }
40 }
```

```
39 private void Listar()
40 {
41     var lista = iRepositorio!.Listar();
42     Assert.IsTrue(lista.Count > 0);
43 }
44 private void Buscar()
45 {
46     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
47     Assert.IsTrue(lista.Count > 0);
48 }
49 private void Modificar()
50 {
51     entidad!.Nombre = "Prueba Modificada";
52     entidad = iRepositorio!.Modificar(entidad);
53     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
54     Assert.IsTrue(lista.Count > 0);
55 }
56 private void Borrar()
57 {
58     entidad = iRepositorio!.Borrar(entidad!);
59     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
60     Assert.IsTrue(lista.Count == 0);
61 }
62 }
63 }
64
65 }
```

FabricantesPruebasUnitarias:

```
6 namespace mst_pruebas.Repositorios
7 {
8     [TestClass]
9     public class FabricantesPruebasUnitarias
10     {
11         private IFabricantesRepositorio? iRepositorio = null;
12         private Fabricantes? entidad = null;
13         public FabricantesPruebasUnitarias()
14         {
15             var conexion = new Conexion();
16             conexion.StringConnection = "server=MWNU_HELPER\\DEV;database=bd_catalogo;uid=sa;pwd=STEMgirls>>>; TrustServerCertificate = true; ";
17             iRepositorio = new FabricantesRepositorio(conexion);
18         }
19
20         [TestMethod]
21         public void Ejecutar()
22         {
23             Guardar();
24             Listar();
25             Buscar();
26             Modificar();
27             Borrar();
28         }
29         private void Guardar()
30         {
31             entidad = new Fabricantes()
32             {
33                 Nombre = "Prueba",
34                 Contacto = "contacto@prueba.com"
35             };
36             entidad = iRepositorio!.Guardar(entidad);
37             Assert.IsTrue(entidad.Id != 0);
38         }
39     }
40 }
```

```
39 private void Listar()
40 {
41     var lista = iRepositorio!.Listar();
42     Assert.IsTrue(lista.Count > 0);
43 }
44 private void Buscar()
45 {
46     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
47     Assert.IsTrue(lista.Count > 0);
48 }
49 private void Modificar()
50 {
51     entidad!.Nombre = "Prueba Modificada";
52     entidad = iRepositorio!.Modificar(entidad);
53     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
54     Assert.IsTrue(lista.Count > 0);
55 }
56 private void Borrar()
57 {
58     entidad = iRepositorio!.Borrar(entidad!);
59     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
60     Assert.IsTrue(lista.Count == 0);
61 }
62 }
63 }
64 }
```

ImagenesPruebasUnitarias:

```
6 namespace mst_pruebas.Repositorios
7 {
8     [TestClass]
9     public class ImagenesPruebasUnitarias
10     {
11         private IImagenesRepositorio? iRepositorio = null;
12         private Imagenes? entidad = null;
13         public ImagenesPruebasUnitarias()
14         {
15             var conexion = new Conexion();
16             conexion.StringConnection = "server=MUNU_HELPER\\DEV;database=bd_catologo;uid=sa;pwd=STEMgirls>>>; TrustServerCertificate = true; ";
17             iRepositorio = new ImagenesRepositorio(conexion);
18         }
19         [TestMethod]
20         public void Ejecutar()
21         {
22             Guardar();
23             Listar();
24             Buscar();
25             Modificar();
26             Borrar();
27         }
28         private void Guardar()
29         {
30             entidad = new Imagenes()
31             {
32                 Producto = 1,
33                 Archivo = "prueba.jpg",
34                 Informacion = "Prueba de Imagen",
35             };
36             entidad = iRepositorio!.Guardar(entidad);
37             Assert.IsTrue(entidad.Id != 0);
38         }
39     }
40 }
```

```
39 private void Listar()
40 {
41     var lista = iRepositorio!.Listar();
42     Assert.IsTrue(lista.Count > 0);
43 }
44 private void Buscar()
45 {
46     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
47     Assert.IsTrue(lista.Count > 0);
48 }
49 private void Modificar()
50 {
51     entidad!.Archivo = "Prueba.png";
52     entidad = iRepositorio!.Modificar(entidad);
53     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
54     Assert.IsTrue(lista.Count > 0);
55 }
56 private void Borrar()
57 {
58     entidad = iRepositorio!.Borrar(entidad!);
59     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
60     Assert.IsTrue(lista.Count == 0);
61 }
62 }
63 }
64 }
```

ProductosPruebasUnitarias:

```
6 namespace mst_pruebas.Repositorios
7 {
8     [TestClass]
9     public class ProductosPruebasUnitarias
10     {
11         private IProductosRepositorio? iRepositorio = null;
12         private Productos? entidad = null;
13         public ProductosPruebasUnitarias()
14         {
15             var conexion = new Conexion();
16             conexion.StringConnection = "server=MANU_HELPER\\DEV;database=bd_catalogo;uid=sa;pwd=STEMgirls>>>; TrustServerCertificate = true; ";
17             iRepositorio = new ProductosRepositorio(conexion);
18         }
19         [TestMethod]
20         public void Ejecutar()
21         {
22             Guardar();
23             Listar();
24             Buscar();
25             Modificar();
26             Borrar();
27         }
28     }
29 }
```

```
28 private void Guardar()
29 {
30     entidad = new Productos()
31     {
32         Codigo = "P003",
33         Nombre = "Audifonos",
34         Cantidad = 100,
35         Precio = 259900.00,
36         Costo = 150000.00,
37         Fabricante = 1,
38         Categoria = 1,
39     };
40     entidad = iRepositorio!.Guardar(entidad);
41     Assert.IsTrue(entidad.Id != 0);
42 }
43 private void Listar()
44 {
45     var lista = iRepositorio!.Listar();
46     Assert.IsTrue(lista.Count > 0);
47 }
48 private void Buscar()
49 {
50     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
51     Assert.IsTrue(lista.Count > 0);
52 }
53 private void Modificar()
54 {
55     entidad!.Cantidad = 80;
56     entidad = iRepositorio!.Modificar(entidad);
57     var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
58     Assert.IsTrue(lista.Count > 0);
59 }
```



```

60     private void Borrar()
61     {
62         entidad = iRepositorio!.Borrar(entidad!);
63         var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
64         Assert.IsTrue(lista.Count == 0);
65     }
66 }
67 }
68

```

PublicacionesPruebasUnitarias:

```

6 namespace test_pruebas.Repositorios
7 {
8     [TestClass]
9     public class PublicacionesPruebasUnitarias
10     {
11         private IPublicacionesRepositorio? iRepositorio = null;
12         private Publicaciones? entidad = null;
13         public PublicacionesPruebasUnitarias()
14         {
15             var conexion = new Conexion();
16             conexion.StringConnection = "server=MHNU_HELPER\\DEV;database=bd_catalogo;uid=sa;pwd=STEMgirls>>>; TrustServerCertificate = true; ";
17             iRepositorio = new PublicacionesRepositorio(conexion);
18         }
19         [TestMethod]
20         public void Ejecutar()
21         {
22             Guardar();
23             Listar();
24             Buscar();
25             Modificar();
26             Borrar();
27         }
28         private void Guardar()
29         {
30             entidad = new Publicaciones()
31             {
32                 Producto = 1,
33                 Estado = 1,
34                 Fecha = DateTime.Now,
35                 Titulo = "Prueba",
36                 Descripcion = "Prueba de Publicacion",
37             };
38             entidad = iRepositorio!.Guardar(entidad);
39             Assert.IsTrue(entidad.Id != 0);
40         }
41     }
42 }

```

```

41     private void Listar()
42     {
43         var lista = iRepositorio!.Listar();
44         Assert.IsTrue(lista.Count > 0);
45     }
46     1 reference
47     private void Buscar()
48     {
49         var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
50         Assert.IsTrue(lista.Count > 0);
51     }
52     1 reference
53     private void Modificar()
54     {
55         entidad!.Titulo = "Modificación Prueba";
56         entidad = iRepositorio!.Modificar(entidad);
57         var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
58         Assert.IsTrue(lista.Count > 0);
59     }
60     1 reference
61     private void Borrar()
62     {
63         entidad = iRepositorio!.Borrar(entidad!);
64         var lista = iRepositorio!.Buscar(x => x.Id == entidad!.Id);
65         Assert.IsTrue(lista.Count == 0);
66     }

```

Paso de testeos:

Test Explorer

Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 1,2 sec

0 Warnings 0 Errors

Test	Duration	Traits	E.
✓ mst_pruebas (6)	7 sec		
✓ mst_pruebas.Repositorios (6)	7 sec		
✓ CategoríasPruebasUnitarias (1)	2,3 sec		
✓ EstadosPruebasUnitarias (1)	866 ms		
✓ FabricantesPruebasUnitarias (1)	907 ms		
✓ ImágenesPruebasUnitarias (1)	896 ms		
✓ ProductosPruebasUnitarias (1)	905 ms		
✓ PublicacionesPruebasUnitarias (1)	1,1 sec		

Run | Debug

Group Summary

PublicacionesPruebasUnitarias

Tests in group: 1

Total Duration: 1,1 sec

Outcomes

✓ 1 Passed

