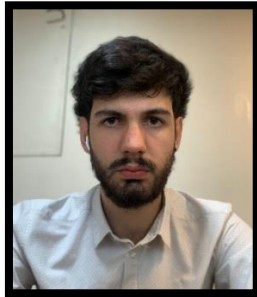


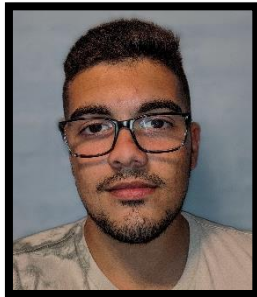
Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

PROGRAMACIÓN 3

Gianfranco Bonanni – 274029



Gabriel Moreno – 222706



Docente: Adriana Cabella

Analista en Tecnologías de la Información

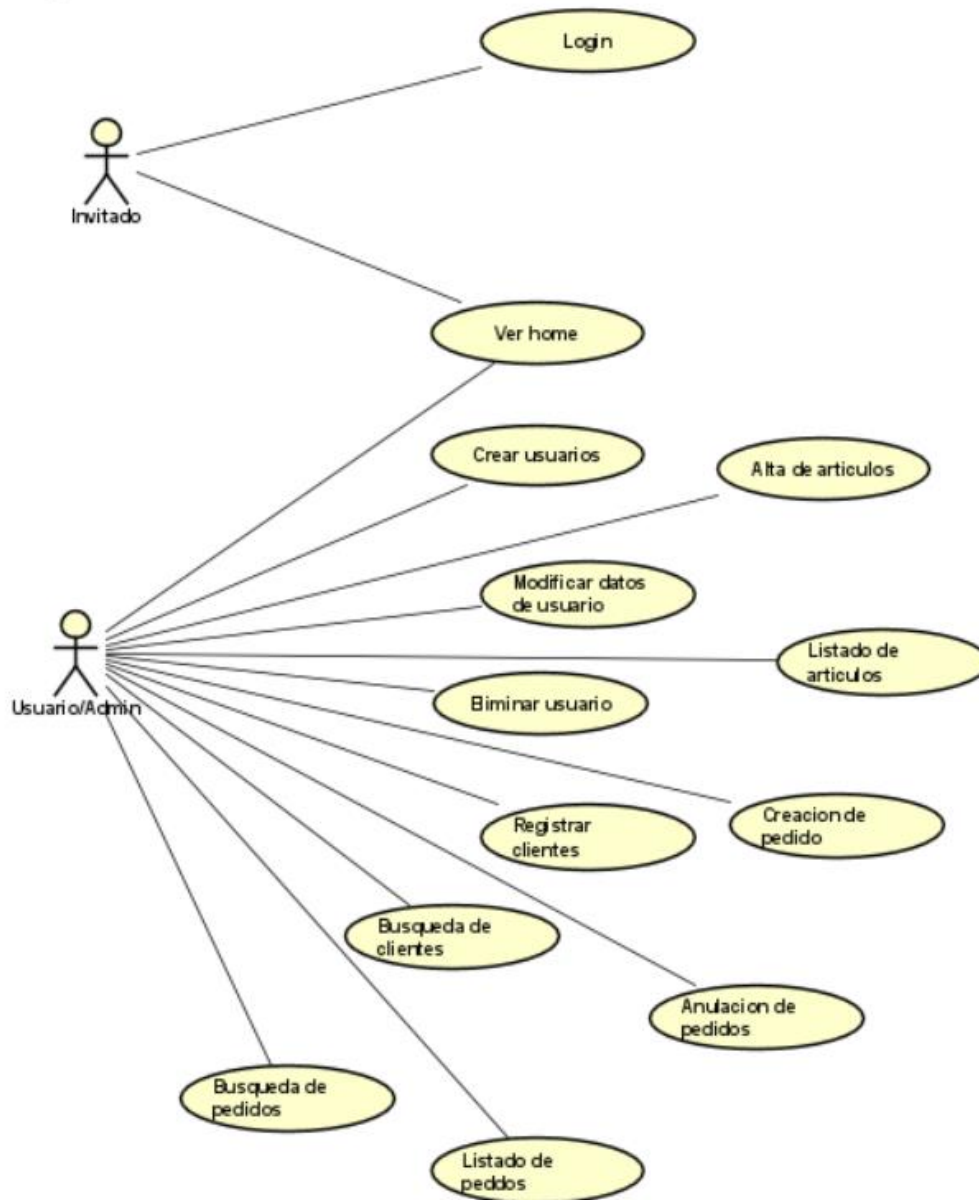
16-05-2024

Contenido

<i>DIAGRAMA CASOS DE USO</i>	3
DESCRIPCION NARRATIVA DE LOS SIGUIENTES CASOS DE USO:	4
DIAGRAMA DE CLASES	5
CODIGO FUENTE DE LA APLICACIÓN.	7

DIAGRAMA CASOS DE USO

uc CasoDeUso



DESCRIPCION NARRATIVA DE LOS SIGUIENTES CASOS DE USO:

Alta pedio.

ITEM	VALUE
UseCase	Creacion de pedido
Summary	Permite registrar un nuevo pedido en el sistema, dejandolo disponible para su visionado.
Actor	Usuario/Admin
Precondition	Hay clientes registrados en el sistema, hay articulos cargados en la base y tiene al menos una linea de pedido.
Postcondition	El pedido quedo realizado en el sistema con todos los datos cargados y al menos una linea agregada al pedido, se le asigno un ID y queda disponible para ser utilizado.
Base Sequence	1. El actor ingresa a registrar un nuevo pedido. 2. El sistema solicita cliente, fecha de entrega, selecciono el articulo, la cantidad. 3. El sistema verifica la fecha del pedido, el stock de los articulos y que tenga la cantidad de articulos suficientes. 6. El sistema comprueba la validez de los datos, le asigna un código autonumérico, la marca como disponible para ser visualizada y la incluye en la lista de pedidos del cliente.
Branch Sequence	3a. El actor cancela el ingreso. 3a - 1. El pedido no es guardado. 4a. No se ingresó Cliente, fecha de entrega, o no se cargo una linea al pedido. 4a - 1. El sistema no registra los datos. 6b - El actor ingresa el stock del articulo cuando no hay suficiente. 6b- 1. El sistema avisa que hay un error, marcando los campos erróneos con una descripción del error.
Exception Sequence	La comunicación con el servidor se corta durante el registro. El pedido es descartado en su totalidad.

Anulacion pedido.

ITEM	VALUE
UseCase	Anulacion de pedidos
Summary	Permite anular un pedido que aun no ha sido entregado.
Actor	Usuario/Admin
Precondition	El pedido no ha sido entregado aun, el pedido esta listado.
Postcondition	El pedido ha sido anulado.
Base Sequence	1. El actor ingresara la fecha de emision del pedido. 2. Se listaran todos los pedidos no entregados de esa fecha mostrando, fecha de entrega prometida, cliente y total del pedido. 3. El cliente seleccionara anular sobre el pedido que desee anular. 4. Le pedira confirmacion sobre esta accion. 5. El pedido quedara anulado del sistema.
Branch Sequence	3a. El actor no encuentra pedido para anular. 3a - 1. La fecha ingresada no corresponde a ningun pedido. 4a. No se confirmo la anulacion del pedido. 4a - 1. No se elimina el pedido por falta de confirmacion de anulacion. 4b. No se lista pedido. 4b - 1. No hay pedido para listar ya que han sido entregados todos para esa fecha. 4b - 2. No se ingresa fecha.
Exception Sequence	La comunicación con el servidor se corta durante la anulacion. El pedido no es anulado.

DIAGRAMA DE CLASES

Diagrama clases

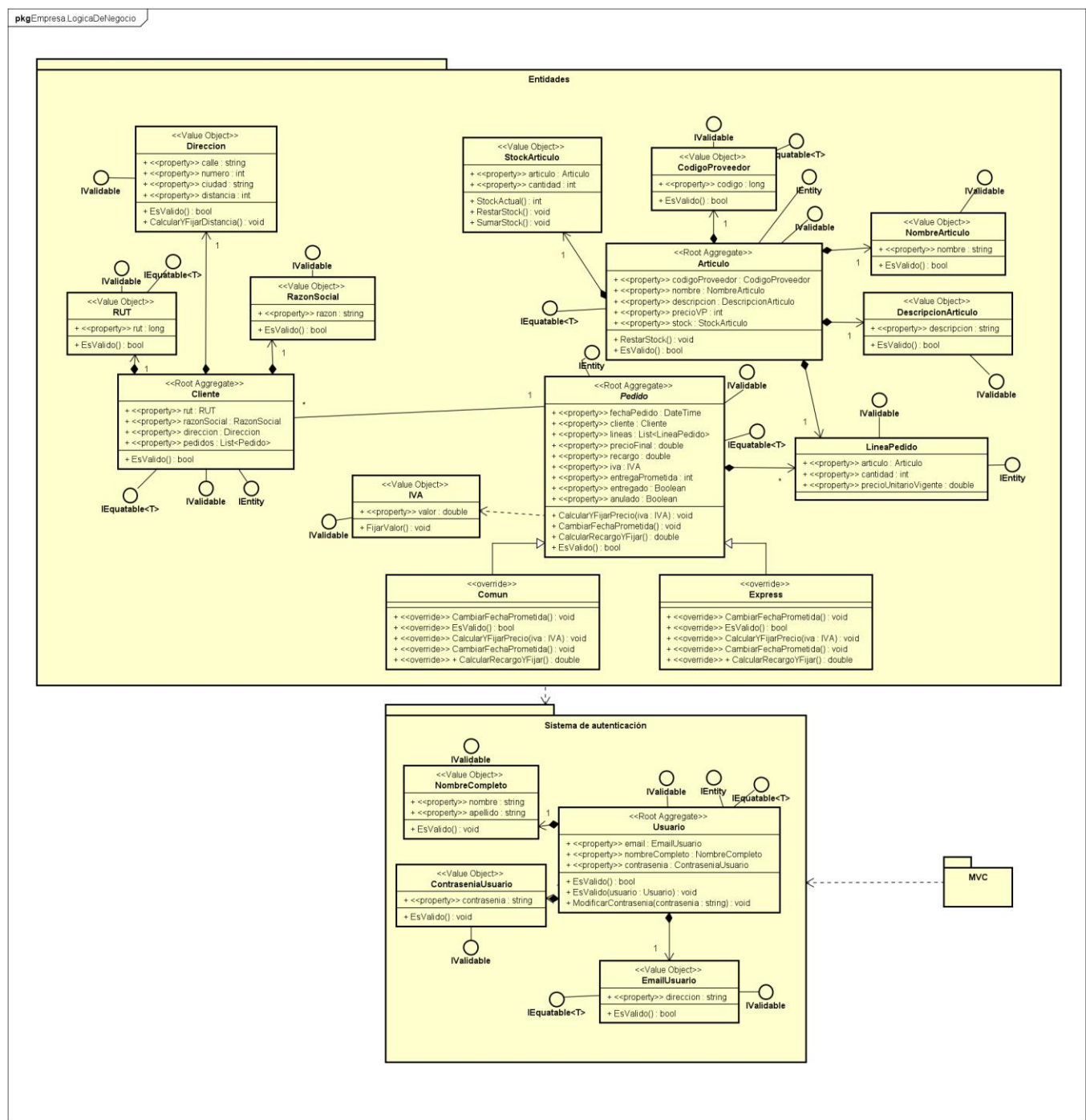
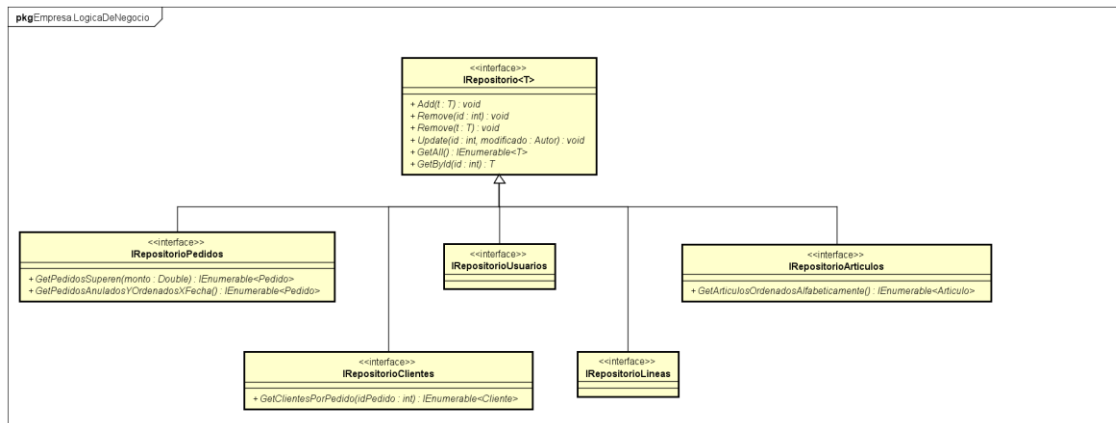
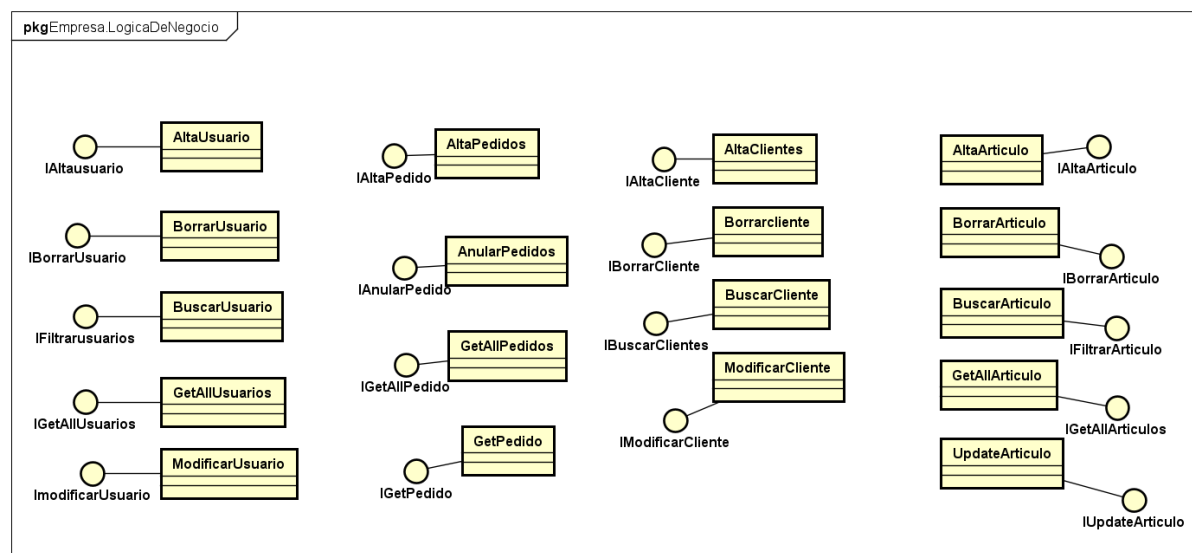


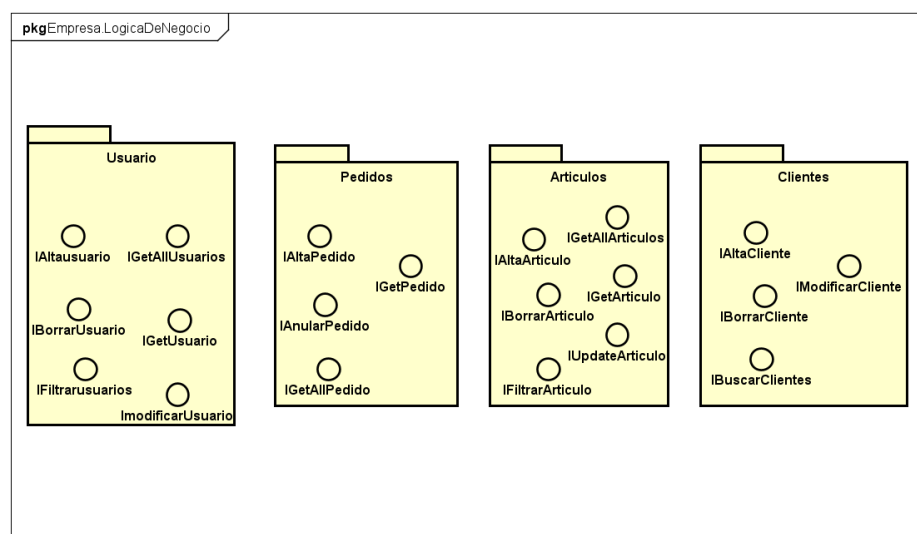
Diagrama Interfaces



ImplementacionCasoUso



Interfaces de Caso de uso



CODIGO FUENTE DE LA APLICACIÓN.

Archivo: UtilidadesString.cs

Localizacion: Papeleria.Comun\UtilidadesString.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.Comun
```

```
{
```

```
    public static class UtilidadesString
```

```
    {
```

```
        public static string FormatearInicialesMayuscula(string texto)
```

```
        {
```

```
            string[] palabras = texto.Split(' ');
```

```
            for (int i = 0; i < palabras.Length; i++)
```

```
            {
```

```
                if (palabras[i].Length > 0)
```

```
                {
```

```
                    palabras[i] = char.ToUpper(palabras[i][0]) + palabras[i].Substring(1);
```

```
        }  
    }  
  
    return string.Join(" ", palabras);  
  
}  
  
}  
  
}
```

Archivo: ClienteConfiguraciones.cs

Localizacion: Papeleria.AccesoDatos\Configuraciones\ClienteConfiguraciones.cs

```
using Empresa.LogicaDeNegocio.Entidades;  
  
using Microsoft.EntityFrameworkCore;  
  
using Microsoft.EntityFrameworkCore.Metadata.Builders;  
  
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;  
  
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.AccesoDatos.Configuraciones
```



```

{

public class ClienteConfiguraciones : IEntityTypeConfiguration<Cliente>

{

    public void Configure(EntityTypeBuilder<Cliente> builder)

    {

        var razonSocialConverter = new ValueConverter<RazonSocial, string>

        (

            rs => rs.RazonSoc,

            rs => new RazonSocial(rs)

        );

        builder.Property(e => e.razonSocial).HasConversion(razonSocialConverter);

        builder.HasIndex(e => e.razonSocial);

    }

}

}

```

Archivo: PapeleriaContext.cs

Localizacion: Papeleria.AccesoDatos\EF\PapeleriaContext.cs

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

```
using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;

using Papeleria.LogicaNegocio.Entidades;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.AccesoDatos.Configuraciones;
```

```
namespace Papeleria.AccesoDatos.EF
```

```
{

    public class PapeleriaContext:DbContext

    {

        public DbSet<Usuario> Usuarios{ get; set; }

        public DbSet<Articulo> Articulos { get; set; }

        public DbSet<Cliente> Clientes { get; set; }

        public DbSet<Pedido> Pedidos { get; set; }

        public DbSet<Express> Expresses { get; set; }

        public DbSet<Comunes> Comuns { get; set; }

        public DbSet<LineaPedido> LineasPedidos { get; set; }
```

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

```
{
```

```
optionsBuilder.UseSqlServer(@"SERVER=(localDB)\Mssqllocaldb;DATABASE=PapeleriaOBL;INTEGRATED SECURITY=True; encrypt=false");
```

```
}
```

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

```
{
```

```
new ClienteConfiguraciones().Configure(modelBuilder.Entity<Cliente>());
```

```
modelBuilder.ApplyConfiguration(new ClienteConfiguraciones());
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: RepositorioArticuloEF.cs

Localizacion: Papeleria.AccesoDatos\EF\RepositorioArticuloEF.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo;
```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.EF
```

```
{
```

```
    public class RepositorioArticuloEF : IRepositoryArticulo
```

```
    {
```

```
        private PapeleriaContext _db;
```

```
        public RepositorioArticuloEF() { _db = new PapeleriaContext(); }
```

```
        public void Add(Articulo obj)
```

```
        {
```

```
            try
```

```
            {
```

```
                _db.Articulos.Add(obj);
```

```
                _db.SaveChanges();
```

```
            }
```

```
            catch (Exception ex)
```

```
            {
```

```
        throw new ArticuloNoValidoException(ex.Message);
```

```
    }
```

```
}
```

```
public IEnumerable<Articulo> GetAll()
```

```
{
```

```
    return _db.Articulos.ToList();
```

```
}
```

```
public Articulo GetArticuloByCodigo(CodigoProveedorArticulos codigo)
```

```
{
```

```
    Articulo? articulo = _db.Articulos.FirstOrDefault(art => art.CodigoProveedor.codigo ==  
codigo.codigo);
```

```
    return articulo;
```

```
}
```

```
public IEnumerable<Articulo> GetArticulosOrdenadosAlfabeticamente()
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public Articulo GetById(int id)
```

```
{
```

```
    Articulo? articulo = _db.Articulos.FirstOrDefault(art => art.Id == id);
```

```
        return articulo;
    }

    public IEnumerable<Articulo> GetObjectsByID(List<int> ids)
    {
        throw new NotImplementedException();
    }

    public void Remove(int id)
    {
        var articulo = _db.Articulos.FirstOrDefault(u => u.Id == id);

        if (articulo != null)
        {
            _db.Articulos.Remove(articulo);

            _db.SaveChanges();
        }
    }

    public void Remove(Articulo obj)
    {
        _db.Articulos.Remove(obj);

        _db.SaveChanges();
    }
}
```

```
public void Update(int id, Artículo obj)

{

    var articulo = _db.Articulos.FirstOrDefault(u => u.Id == id);


    if (articulo != null)

    {

        try

        {

            articulo.ModificarDatos(obj);

            _db.SaveChanges();

        }

        catch (Exception ex)

        {

            throw new ArtículoNoValidoException(ex.Message);

        }

    }

    else

    {

        throw new ArtículoNuloException("El articulo no existe");

    }

}
```

```

public bool ExisteArticuloConNombre(string nombre)

{

    Articulo? articulo = _db.Articulos.FirstOrDefault(art => art.NombreArticulo.Nombre == nombre);

    if (articulo != null) {

        return true;

    }

    return false;

}

}

}

```

Archivo: RepositorioClienteEF.cs

Localizacion: Papeleria.AccesoDatos\EF\RepositorioClienteEF.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Microsoft.EntityFrameworkCore;

using Papeleria.AccesoDatos.Excepciones;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

```



```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.EF
```

```
{
```

```
    public class RepositorioClienteEF : IRepositorioCliente
```

```
    {
```

```
        private PapeleriaContext _db;
```

```
        private IRepositorioPedido _repositorioPedido { get; set; }
```

```
        public RepositorioClienteEF(IRepositorioPedido repositorioPedido)
```

```
        {
```

```
            _db = new PapeleriaContext();
```

```
            _repositorioPedido = repositorioPedido;
```

```
        }
```

```
        public void Add(Cliente obj)
```

```
        {
```

```
            try
```

```
            {
```

```
                _db.Clientes.Add(obj);
```

```
        _db.SaveChanges();

    }

    catch (Exception ex)

    {

        throw new ClienteNoValidoException(ex.Message);

    }

}

public IEnumerable<Cliente> GetAll()

{

    return _db.Clientes.ToList();

}

public Cliente GetById(int id)

{

    Cliente? cliente = _db.Clientes.FirstOrDefault(cli => cli.Id == id);

    return cliente;

}

public Cliente GetCliente(int idCliente)

{

    return _db.Clientes.FirstOrDefault(cli => cli.Id == idCliente);

}
```

```
}
```

```
public Cliente GetClientePorDireccion(DireccionCliente direccionCliente)
```

```
{
```

```
    return _db.Clientes.FirstOrDefault(cli => cli.direccion.Equals(direccionCliente));
```

```
}
```

```
public Cliente GetClientePorRazon(string rsocial)
```

```
{
```

```
    try
```

```
    {
```

```
        //No tocar, funciona...
```

```
        var clientes = _db.Clientes.ToList();
```

```
        var cliente = clientes.FirstOrDefault(c => c.razonSocial.RazonSoc.Equals(rsocial, StringComparison.OrdinalIgnoreCase)); //OrdinalIgnoreCase para que ignore mayusculas
```

```
        return cliente;
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        throw new ClienteNoValidoException("Error al buscar el cliente.");
```

```
    }
```

```
}
```

```
public Cliente GetClientePorRUT(long rut)
```

```
{
```

```
    return _db.Clientes.FirstOrDefault(cli => cli.rut.Rut.Equals(rut));
```

```
}
```

```
public IEnumerable<Cliente> GetClientes()
```

```
{
```

```
    return _db.Clientes.ToList();
```

```
}
```

```
public IEnumerable<Cliente> GetClientesPorPedido(int idPedido)
```

```
{
```

```
    return _db.Clientes.Where(cli => cli.pedidos.Any(pedido => pedido.Id == idPedido)).ToList();
```

```
}
```

```
public IEnumerable<Cliente> GetClientesPedidoSupereMonto(double monto)
```

```
{
```

```
    var pedidosSuperaronMonto = _repositorioPedido.GetPedidosQueSuperenMonto(monto);
```

```
    var clientesDePedidosQueSuperaronMonto = pedidosSuperaronMonto.Select(pedido =>  
pedido.cliente).Distinct();
```

```
    return clientesDePedidosQueSuperaronMonto;
```

```
}
```

```
public IEnumerable<Cliente> GetObjectsByID(List<int> ids)

{

    throw new NotImplementedException();

}
```

```
public void Remove(int id)

{

    var cliente = GetById(id);

    if (cliente != null)

    {

        _db.Clientes.Remove(cliente);

        _db.SaveChanges();

    }

}
```

```
public void Remove(Cliente obj)

{

    if (obj != null)

    {

        _db.Clientes.Remove(obj);

        _db.SaveChanges();

    }

}
```

```
}
```

```
public void Update(int id, Cliente obj)
```

```
{
```

```
    if (id == null || obj == null)
```

```
    {
```

```
        throw new ArgumentNullException("No se recibió ningun usuario.");
```

```
    }
```

```
    Cliente cliente = GetById(id);
```

```
    try
```

```
    {
```

```
        cliente.Update(obj);
```

```
        _db.Clientes.Update(cliente);
```

```
        _db.SaveChanges();
```

```
    }
```

```
    catch (Exception e)
```

```
    {
```

```
        throw new InfraException("Hubo un problema, reintente.");
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: RepositorioLineaPedidoEF.cs

Localizacion: Papeleria.AccesoDatos\EF\RepositorioLineaPedidoEF.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Empresa.LogicaDeNegocio.Sistema;
```

```
using Papeleria.LogicaNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.EF
```

```
{
```

```
    public class RepositorioLineaPedidoEF : IRepositoryLineaPedido
```

```
    {
```

```
        private PapeleriaContext _db;
```

```
        public RepositorioLineaPedidoEF()
```

```
{  
  
    _db = new PapeleriaContext();  
  
}
```

```
public void Add(LineaPedido obj)
```

```
{  
  
    try  
  
    {  
  
        _db.LineasPedidos.Add(obj);  
  
        _db.SaveChanges();  
  
    }  
  
    catch (Exception ex)  
  
    {  
  
        throw new ClienteNoValidoException(ex.Message);  
  
    }  
  
}
```

```
public LineaPedido Get(int id)
```

```
{  
  
    throw new NotImplementedException();  
  
}
```



```
public IEnumerable<LineaPedido> GetAll()

{

    return _db.LineasPedidos.ToList();

}
```

```
public IEnumerable<LineaPedido> GetByArticulo(Articulo articulo)

{

    throw new NotImplementedException();

}
```

```
public LineaPedido GetById(int id)

{

    LineaPedido? lineaPedido= _db.LineasPedidos.FirstOrDefault(lnPd => lnPd.Id == id);

    return lineaPedido;

}
```

```
public IEnumerable<LineaPedido> GetObjectsByID(List<int> ids)

{

    throw new NotImplementedException();

}
```

```
public void Remove(int id)

{
```

```
        throw new NotImplementedException();  
    }  
  
    public void Remove(LineaPedido obj)  
  
    {  
  
        throw new NotImplementedException();  
  
    }  
  
    public void Update(int id, LineaPedido obj)  
  
    {  
  
        throw new NotImplementedException();  
  
    }  
  
    }  
  
}
```

Archivo: RepositorioPedidoEF.cs

Localizacion: Papeleria.AccesoDatos\EF\RepositorioPedidoEF.cs

using Empresa.LogicaDeNegocio.Entidades;

using Microsoft.EntityFrameworkCore;

using Papeleria.LogicaNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

```
using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.EF
```

```
{

    public class RepositorioPedidoEF : IRepositoryPedido

    {

        private PapeleriaContext _db;

        public RepositorioPedidoEF()

        {

            _db = new PapeleriaContext();

        }

        public void Add(Pedido obj)

        {

            try

            {
```

```

        _db.ChangeTracker.Clear(); // NO TOCAR :)

        _db.Pedidos.Attach(obj);

        _db.SaveChanges();

    }

    catch (Exception ex)

    {

        throw new PedidoNoValidoException(ex.Message);

    }

}

public IEnumerable<Pedido> GetAll()

{

    List<Cliente> clientes = _db.Clientes.ToList(); // NO TOCAR :)

    List<Articulo> articulos = _db.Articulos.ToList();

    List<Pedido> pedidos = _db.Pedidos.ToList();

    var lineas = _db.LineasPedidos.ToList();

    foreach (Pedido pedido in pedidos)

    {

        var filtrados = lineas;

    }

    return pedidos;

```

```
}
```

```
public Pedido GetById(int id)
```

```
{
```

```
    Pedido? pedido= _db.Pedidos.FirstOrDefault(pedido => pedido.Id == id);
```

```
    return pedido;
```

```
}
```

```
public IEnumerable<Pedido> GetObjectsByID(List<int> ids)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public Pedido GetPedido(Pedido pedido)
```

```
{
```

```
    Pedido pedidoEncontrado = _db.Pedidos.FirstOrDefault(p => p.Id == pedido.Id);
```

```
    if (pedidoEncontrado != null)
```

```
    {
```

```
        return pedidoEncontrado;
```

```
    }
```

```
    return null;
```

```
}
```

```
public IEnumerable<Pedido> GetPedidosPorCliente(Cliente cliente)
```

```
{
```

```
    List<Pedido> pedidosCliente = _db.Pedidos.Where(p => p.cliente.Id == cliente.Id).ToList();
```

```
    return pedidosCliente;
```

```
}
```

```
public IEnumerable<Pedido> GetPedidosPorDireccion(DireccionCliente direccionPedido)
```

```
{
```

```
    var pedidosPorDireccion = _db.Pedidos.Where(p => p.cliente.direccion ==  
direccionPedido).ToList();
```

```
    return pedidosPorDireccion;
```

```
}
```

```
public IEnumerable<Pedido> GetPedidosPorRazon(RazonSocial rsocial)
```

```
{
```

```
    return _db.Pedidos.Where(p => p.cliente.razonSocial.Equals(rsocial)).ToList();
```

```
}
```

```
public IEnumerable<Pedido> GetPedidosPorRUT(RUT rut)
```

```
{
```

```
    return _db.Pedidos.Where(p => p.cliente.rut.Rut == rut.Rut).ToList();
```

```
}
```

```
public IEnumerable<Pedido> GetPedidosQueSuperenMonto(double monto)
```

```
{  
  
    List<Cliente> clientes = _db.Clientes.ToList(); // NO TOCAR :)  
  
    var pedidos = _db.Pedidos.ToList();  
  
    var pedidosSuperaronMonto = pedidos.Where(pedido => pedido.precioFinal >= monto);  
  
    var cantidad = pedidosSuperaronMonto.Count();  
  
    return pedidosSuperaronMonto;  
  
}
```

```
public IEnumerable<Pedido> GetPedidosPorFecha(DateTime date)
```

```
{  
  
    IEnumerable<Pedido> pedidos = _db.Pedidos.Where(pedido => pedido.fechaPedido.Date ==  
date);  
  
    return pedidos;  
  
}
```

```
public void Remove(int id)
```

```
{  
  
    var pedido = _db.Pedidos.FirstOrDefault(p => p.Id == id);  
  
    if (pedido != null)  
    {  
  
        _db.Pedidos.Remove(pedido);  
  
        _db.SaveChanges();  
  
    }  
  
}
```

```
public void Remove(Pedido obj)
```

```
{
```

```
    _db.Pedidos.Remove(obj);
```

```
    _db.SaveChanges();
```

```
}
```

```
public void Update(int id, Pedido obj)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Anular(int id)
```

```
{
```

```
    var pedido = _db.Pedidos.FirstOrDefault(p => p.Id == id);
```

```
    if (pedido != null)
```

```
    {
```

```
        try
```

```
        {
```

```
            pedido.AnularPedido();
```

```
            _db.SaveChanges();
```

```
        }
```

```
        catch (Exception ex)
```



```

        {

            throw new PedidoNoValidoException(ex.Message);

        }

    }

    else

    {

        throw new UsuarioNuloExcepcion("El pedido no existe");

    }

}

}

}

```

Archivo: RepositorioUsuarioEF.cs

Localizacion: Papeleria.AccesoDatos\EF\RepositorioUsuarioEF.cs

using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Microsoft.EntityFrameworkCore;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

```
using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Constrasenia;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.EF
```

```
{
```

```
    public class RepositorioUsuarioEF : IRepositoryUsuario
```

```
    {
```

```
        private PapeleriaContext _db;
```

```
        public RepositorioUsuarioEF()
```

```
        {
```

```
            _db = new PapeleriaContext();
```

```
        }
```

```
        public Usuario GetUsuarioPorEmail(string email)
```

```
        {
```

```
            return _db.Usuarios.FirstOrDefault(u => u.Email.Direccion == email);
```

```
        }
```

```
public void ModificarContrasenia(int id, ContrasseniaUsuario contrasseniaNueva)

{

    var usuario = _db.Usuarios.FirstOrDefault(u => u.Id == id);


    if (usuario != null)

    {

        try

        {

            usuario.Contrasenia = new ContrasseniaUsuario(contrasseniaNueva.Valor);

            _db.SaveChanges();

        }

        catch (Exception ex)

        {

            throw new ContrasseniaNoValidoException(ex.Message);

        }

    }

    else

    {

        throw new UsuarioNuloExcepcion("El usuario no existe");

    }

}


public Usuario GetById(int id)
```

```
{  
  
    Usuario? usuario = _db.Usuarios.FirstOrDefault(usr => usr.Id == id);  
  
    return usuario;  
  
}
```

```
public void Add(Usuario obj)
```

```
{  
  
    if (obj == null) throw new UsuarioNuloExcepcion("El usuario es nulo");  
  
    try  
  
    {  
  
        _db.Usuarios.Add(obj);  
  
        _db.SaveChanges();  
  
    }  
  
    catch (Exception ex)  
  
    {  
  
        throw new UsuarioNoValidoExcepcion(ex.Message);  
  
    }  
  
}
```

```
public void Update(int id, Usuario obj)
```

```
{  
  
    var usuario = _db.Usuarios.FirstOrDefault(u => u.Id == id);
```

```
if (usuario != null)

{

    try

    {

        usuario.ModificarDatos(obj);

        _db.SaveChanges();

    }

    catch (Exception ex)

    {

        throw new UsuarioNoValidoExcepcion(ex.Message);

    }

}

else

{

    throw new UsuarioNuloExcepcion("El usuario no existe");

}

}
```

```
public void Remove(int id)

{

    var usuario = _db.Usuarios.FirstOrDefault(u => u.Id == id);

    if (usuario != null)
```

```
{  
  
    _db.Usuarios.Remove(usuario);  
  
    _db.SaveChanges();  
  
}  
  
}
```

```
public void Remove(Usuario obj)
```

```
{  
  
    _db.Usuarios.Remove(obj);  
  
    _db.SaveChanges();  
  
}
```

```
public IEnumerable<Usuario> GetAll()
```

```
{  
  
    return _db.Usuarios.ToList();  
  
}
```

```
public IEnumerable<Usuario> GetObjectsByID(List<int> ids)
```

```
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Usuario Login(string email, string contrasenia)
```

```
{  
  
    try  
  
    {  
  
        Usuario usu = GetUsuarioPorEmail(email);  
  
        if (usu != null)  
  
        {  
  
            if (usu.Email.Direccion == email && usu.Contrasenia.Valor == contrasenia)  
  
            {  
  
                return usu;  
  
            }  
  
        }  
  
        return null;  
  
  
    }  
  
    catch (UsuarioNoValidoExcepcion ex)  
  
    {  
  
        throw new UsuarioNoValidoExcepcion("");  
  
    }  
  
}
```

```
public bool ExisteUsuarioConEmail(string email)
```

```
{  
  
    Usuario usuario = GetUsuarioPorEmail(email);
```

```
        return usuario != null;

    }

}

}
```

Archivo: ArgumentNullRepositorioException.cs

Localizacion: Papeleria.AccesoDatos\Excepciones\ArgumentNullRepositorioException.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.AccesoDatos.Excepciones

{

    public class ArgumentNullRepositorioException : RepositorioException

    {

        public ArgumentNullRepositorioException() { }

        public ArgumentNullRepositorioException(string message) : base(message) { }

    }

}
```

Archivo: InfraException.cs

Localizacion: Papeleria.AccesoDatos\Excepciones\InfraException.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.AccesoDatos.Excepciones

{

public class InfraException : RepositorioException

{

public InfraException() { }

public InfraException(string message) : base(message) { }

}

}

Archivo: RepositorioException.cs

Localizacion: Papeleria.AccesoDatos\Excepciones\RepositoryException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.Excepciones
```

```
{
```

```
    public class RepositoryException : Exception
```

```
    {
```

```
        public RepositoryException() { }
```

```
        public RepositoryException(string message) : base(message) { }
```

```
    }
```

```
}
```

Archivo: RepositoryArticuloMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositoryArticuloMemoria.cs

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.Memoria
```

```
{
```

```
    public class RepositorioArticuloMemoria : IRepositoryArticulo
```

```
    {
```

```
        public void Add(Articulo obj)
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
        public IEnumerable<Articulo> GetAll()
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
        public IEnumerable<Articulo> GetAllArticulos()
```

```
        {
```

```
        throw new NotImplementedException();  
    }  
  

```

```
public IEnumerable<Articulo> GetArticulosOrdenadosAlfabeticamente()  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Articulo GetById(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public IEnumerable<Articulo> GetObjectsByID(List<int> ids)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public void Remove(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public void Remove(Articulo obj)

{

    throw new NotImplementedException();

}
```

```
public void Update(int id, Articulo obj)

{

    throw new NotImplementedException();

}
```

```
public Articulo GetArticuloByCodigo(CodigoProveedorArticulos codigo)

{

    throw new NotImplementedException();

}
```

```
public bool ExisteArticuloConNombre(string nombre)

{

    throw new NotImplementedException();

}

}
```

Archivo: RepositorioClienteMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositorioClienteMemoria.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.Memoria
```

```
{

    public class RepositorioClienteMemoria : IRepositoryCliente

    {

        public void Add(Cliente obj)

        {

            throw new NotImplementedException();

        }

        public IEnumerable<Cliente> GetAll()

        {
```

```
        throw new NotImplementedException();  
    }  
  

```

```
public Cliente GetById(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Cliente GetCliente(int idCliente)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Cliente GetClientePorDireccion(DireccionCliente direccionCliente)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Cliente GetClientePorRazon(string rsocial)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Cliente GetClientePorRUT(long rut)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Cliente> GetClientes()

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Cliente> GetClientesPedidoSupereMonto(double monto)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Cliente> GetClientesPorPedido(int idPedido)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Cliente> GetObjectsByID(List<int> ids)

{

    throw new NotImplementedException();

}
```



```
}
```

```
public void Remove(int id)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Remove(Cliente obj)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Update(int id, Cliente obj)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: RepositorioComunesMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositorioComunesMemoria.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.AccesoDatos.Memoria

{

    public class RepositorioComunesMemoria : IRepositoryComun

    {

        public void Add(Comunes obj)

        {

            throw new NotImplementedException();

        }


        public IEnumerable<Comunes> GetAll()

        {

            throw new NotImplementedException();

        }


        public Comunes GetById(int id)
```

```
{  
  
    throw new NotImplementedException();  
  
}
```

```
public IEnumerable<Comunes> GetObjectsByID(List<int> ids)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public void Remove(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public void Remove(Comunes obj)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public void Update(int id, Comunes obj)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
}
```

```
}
```

```
*****
```

Archivo: RepositorioExpressMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositorioExpressMemoria.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.Memoria
```

```
{
```

```
    public class RepositorioExpressMemoria : IRepositoryExpress
```

```
    {
```

```
        public void Add(Express obj)
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
public IEnumerable<Express> GetAll()

{

    throw new NotImplementedException();

}
```

```
public Express GetById(int id)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Express> GetObjectsByID(List<int> ids)

{

    throw new NotImplementedException();

}
```

```
public void Remove(int id)

{

    throw new NotImplementedException();

}
```

```
public void Remove(Express obj)

{
```

```
        throw new NotImplementedException();

    }

    public void Update(int id, Express obj)

    {

        throw new NotImplementedException();

    }

}

}
```

Archivo: RepositorioLineaPedidoMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositorioLineaPedidoMemoria.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.Memoria
```

```
{
```

```
    public class RepositorioLineaPedidoMemoria : IRepositoryLineaPedido
```

```
    {
```

```
        public void Add(LineaPedido obj)
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
        public LineaPedido Get(int id)
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
        public IEnumerable<LineaPedido> GetAll()
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
        public IEnumerable<LineaPedido> GetByArticulo(Articulo articulo)
```

```
        {
```

```
            throw new NotImplementedException();
```

```
        }
```

```
public LineaPedido GetById(int id)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<LineaPedido> GetObjectsByID(List<int> ids)

{

    throw new NotImplementedException();

}
```

```
public void Remove(int id)

{

    throw new NotImplementedException();

}
```

```
public void Remove(LineaPedido obj)

{

    throw new NotImplementedException();

}
```

```
public void Update(int id, LineaPedido obj)

{
```



```
        throw new NotImplementedException();

    }

}

}
```

Archivo: RepositorioPedidoMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositorioPedidoMemoria.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.AccesoDatos.Memoria

{

    public class RepositorioPedidoMemoria : IRepositoryPedido

    {

        public void Add(Pedido obj)
```

```
{  
  
    throw new NotImplementedException();  
  
}
```

```
public void Anular(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public IEnumerable<Pedido> GetAll()  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Pedido GetById(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Cliente GetCliente(int idCliente)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Cliente GetClientePorDireccion(DireccionCliente direccionCliente)

{

    throw new NotImplementedException();

}
```

```
public Cliente GetClientePorRazon(RazonSocial rsocial)

{

    throw new NotImplementedException();

}
```

```
public Cliente GetClientePorRUT(RUT rut)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Cliente> GetClientes()

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Cliente> GetClientesPorPedido(int idPedido)

{
```

```
        throw new NotImplementedException();
    }

    public IEnumerable<Pedido> GetObjectsByID(List<int> ids)
    {
        throw new NotImplementedException();
    }

    public Pedido GetPedido(Pedido pedido)
    {
        throw new NotImplementedException();
    }

    public Pedido GetPedidoById(int id)
    {
        throw new NotImplementedException();
    }

    public IEnumerable<Pedido> GetPedidos()
    {
        throw new NotImplementedException();
    }
```

```
public IEnumerable<Pedido> GetPedidosPorCliente(Cliente cliente)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Pedido> GetPedidosPorDireccion(DireccionCliente direccionPedido)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Pedido> GetPedidosPorFecha(DateTime date)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Pedido> GetPedidosPorRazon(RazonSocial rsocial)

{

    throw new NotImplementedException();

}
```

```
public IEnumerable<Pedido> GetPedidosPorRUT(RUT rut)

{

    throw new NotImplementedException();

}
```

```
}
```

```
public IEnumerable<Pedido> GetPedidosQueSuperenMonto(double monto)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Remove(int id)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Remove(Pedido obj)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Update(int id, Pedido obj)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
}
```

```
}
```

Archivo: RepositorioUsuarioMemoria.cs

Localizacion: Papeleria.AccesoDatos\Memoria\RepositorioUsuarioMemoria.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.AccesoDatos.Memoria
```

```
{

    public class RepositorioUsuarioMemoria : IRepositoryUsuario

    {

        private List<Usuario> _usuarios=new List<Usuario>();

        public void Add(Usuario obj)
```

```
{  
  
    if (obj == null) {  
  
        throw new UsuarioNuloExcepcion("El usuario no puede ser nulo.");  
  
    }  
  
    _usuarios.Add(obj);  
  
}
```

```
public bool ExisteUsuarioConEmail(string email)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public IEnumerable<Usuario> GetAll()  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Usuario GetById(int id)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public IEnumerable<Usuario> GetObjectsByID(List<int> ids)
```



```
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Usuario GetUsuario(int idUsuario)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Usuario GetUsuarioPorEmail(EmailUsuario email)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Usuario GetUsuarioPorEmail(string email)  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public IEnumerable<Usuario> GetUsuarios()  
  
{  
  
    throw new NotImplementedException();  
  
}
```

```
public Usuario Login(string email, string contrasenia)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void ModificarContrasenia(int id, ContraseñaUsuario contraseniaNueva)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Remove(int id)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Remove(Usuario obj)
```

```
{
```

```
    throw new NotImplementedException();
```

```
}
```

```
public void Update(int id, Usuario obj)
```

```
{
```

```
        throw new NotImplementedException();

    }

}

}
```

Archivo: 20240516144751_initialization.cs

Localizacion: Papeleria.AccesoDatos\Migrations\20240516144751_initialization.cs

```
using System;
```

```
using Microsoft.EntityFrameworkCore.Migrations;
```

```
#nullable disable
```

```
namespace Papeleria.AccesoDatos.Migrations
```

```
{
```

```
    /// <inheritdoc />
```

```
    public partial class initialization : Migration
```

```
    {
```

```
        /// <inheritdoc />
```

```
        protected override void Up(MigrationBuilder migrationBuilder)
```

```
        {
```

```
            migrationBuilder.CreateTable(
```

name: "Articulos",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

PrecioVP = table.Column<double>(type: "float", nullable: false),

CodigoProveedor_codigo = table.Column<long>(type: "bigint", nullable: false),

Descripcion_Descripcion = table.Column<string>(type: "nvarchar(max)", nullable: false),

NombreArticulo_Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),

Stock_cantidad = table.Column<int>(type: "int", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK_Articulos", x => x.Id);

});

migrationBuilder.CreateTable(

name: "Clientes",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

razonSocial = table.Column<string>(type: "nvarchar(450)", nullable: false),

```

    direccion_Calle = table.Column<string>(type: "nvarchar(max)", nullable: false),

    direccion_Ciudad = table.Column<string>(type: "nvarchar(max)", nullable: false),

    direccion_Distancia = table.Column<int>(type: "int", nullable: false),

    direccion_Numero = table.Column<int>(type: "int", nullable: false),

    rut_Rut = table.Column<long>(type: "bigint", nullable: false)

},

constraints: table =>

{

    table.PrimaryKey("PK_Clientes", x => x.Id);

});

```

```

migrationBuilder.CreateTable(

    name: "Usuarios",

    columns: table => new

    {

        Id = table.Column<int>(type: "int", nullable: false)

            .Annotation("SqlServer:Identity", "1, 1"),

        Contraseña_Valor = table.Column<string>(type: "nvarchar(max)", nullable: false),

        Email_Direccion = table.Column<string>(type: "nvarchar(max)", nullable: false),

        NombreCompleto_Apellido = table.Column<string>(type: "nvarchar(max)", nullable: false),

        NombreCompleto_Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false)

    },

    constraints: table =>

```

```
{  
  
    table.PrimaryKey("PK_Usuarios", x => x.Id);  
  
});
```

```
migrationBuilder.CreateTable(  
  
    name: "Pedidos",  
  
    columns: table => new  
  
    {  
  
        Id = table.Column<int>(type: "int", nullable: false)  
  
            .Annotation("SqlServer:Identity", "1, 1"),  
  
        fechaPedido = table.Column<DateTime>(type: "datetime2", nullable: false),  
  
        clienteId = table.Column<int>(type: "int", nullable: false),  
  
        recargo = table.Column<double>(type: "float", nullable: false),  
  
        entregaPrometida = table.Column<int>(type: "int", nullable: false),  
  
        precioFinal = table.Column<double>(type: "float", nullable: false),  
  
        entregado = table.Column<bool>(type: "bit", nullable: false),  
  
        anulado = table.Column<bool>(type: "bit", nullable: false),  
  
        Discriminator = table.Column<string>(type: "nvarchar(8)", maxLength: 8, nullable: false),  
  
        iva_valor = table.Column<double>(type: "float", nullable: false)  
  
    },  
  
    constraints: table =>  
  
    {  
  
        table.PrimaryKey("PK_Pedidos", x => x.Id);
```

```

table.ForeignKey(

    name: "FK_Pedidos_Clientes_clienteId",

    column: x => x.clienteId,

    principalTable: "Clientes",

    principalColumn: "Id",

    onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateTable(

    name: "LineasPedidos",

    columns: table => new

    {

        Id = table.Column<int>(type: "int", nullable: false)

            .Annotation("SqlServer:Identity", "1, 1"),

        ArtículoId = table.Column<int>(type: "int", nullable: false),

        Cantidad = table.Column<int>(type: "int", nullable: false),

        PrecioUnitarioVigente = table.Column<double>(type: "float", nullable: false),

        pedidoId = table.Column<int>(type: "int", nullable: false)

    },

    constraints: table =>

    {

        table.PrimaryKey("PK_LineasPedidos", x => x.Id);

        table.ForeignKey(

```

```
name: "FK_LineasPedidos_Articulos_ArticuloId",

column: x => x.ArticuloId,

principalTable: "Articulos",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

table.ForeignKey(

name: "FK_LineasPedidos_Pedidos_pedidoId",

column: x => x_pedidoId,

principalTable: "Pedidos",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});
```

```
migrationBuilder.CreateIndex(

name: "IX_Clientes_razonSocial",

table: "Clientes",

column: "razonSocial");
```

```
migrationBuilder.CreateIndex(

name: "IX_LineasPedidos_ArticuloId",

table: "LineasPedidos",

column: "ArticuloId");
```



```
migrationBuilder.CreateIndex(  
    name: "IX_LineasPedidos_pedidoId",  
    table: "LineasPedidos",  
    column: "pedidoId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Pedidos_clienteId",  
    table: "Pedidos",  
    column: "clienteId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Pedidos_precioFinal",  
    table: "Pedidos",  
    column: "precioFinal");
```

```
}
```

```
/// <inheritdoc />
```

```
protected override void Down(MigrationBuilder migrationBuilder)
```

```
{
```

```
    migrationBuilder.DropTable(  
        name: "LineasPedidos");
```

```
    migrationBuilder.DropTable(  
        name: "Pedidos");
```

```
name: "Usuarios");
```

```
migrationBuilder.DropTable(
```

```
name: "Articulos");
```

```
migrationBuilder.DropTable(
```

```
name: "Pedidos");
```

```
migrationBuilder.DropTable(
```

```
name: "Clientes");
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: 20240516144751_initialization.Designer.cs

Localizacion: Papeleria.AccesoDatos\Migrations\20240516144751_initialization.Designer.cs

```
*****
```

```
// <auto-generated />
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using Microsoft.EntityFrameworkCore;
```

```
using Microsoft.EntityFrameworkCore.Infrastructure;
```

```

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Migrations;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

using Papeleria.AccesoDatos.EF;

#nullable disable

namespace Papeleria.AccesoDatos.Migrations
{
    [DbContext(typeof(PapeleriaContext))]

    [Migration("20240516144751_initialization")]

    partial class initialization
    {
        /// <inheritdoc />

        protected override void BuildTargetModel(ModelBuilder modelBuilder)
        {

#pragma warning disable 612, 618

            modelBuilder

                .HasAnnotation("ProductVersion", "8.0.4")

                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

```

```

modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Articulo", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

    b.Property<double>("PrecioVP")
        .HasColumnType("float");

    b.ComplexProperty<Dictionary<string, object>>("CodigoProveedor",
"Empresa.LogicaDeNegocio.Entidades.Articulo.CodigoProveedor#CodigoProveedorArticulos", b1 =>
    {
        b1.IsRequired();

        b1.Property<long>("codigo")
            .HasColumnType("bigint");
    });

    b.ComplexProperty<Dictionary<string, object>>("Descripcion",
"Empresa.LogicaDeNegocio.Entidades.Articulo.Descripcion#DescripcionArticulo", b1 =>
    {
        b1.IsRequired();
    });
}

```

```

        b1.Property<string>("Descripcion")

            .IsRequired()

            .HasColumnType("nvarchar(max)");

    });

```

```

        b.ComplexProperty<Dictionary<string, object>>("NombreArticulo",
"Empresa.LogicaDeNegocio.Entidades.Articulo.NombreArticulo#NombreArticulo", b1 =>

        {

            b1.IsRequired();

            b1.Property<string>("Nombre")

                .IsRequired()

                .HasColumnType("nvarchar(max)");

        });

```

```

        b.ComplexProperty<Dictionary<string, object>>("Stock",
"Empresa.LogicaDeNegocio.Entidades.Articulo.Stock#StockArticulo", b1 =>

        {

            b1.IsRequired();

            b1.Property<int>("cantidad")

                .HasColumnType("int");

        });

```

```
b.HasKey("Id");
```

```
b.ToTable("Articulos");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Cliente", b =>
```

```
{
```

```
b.Property<int>("Id")
```

```
.ValueGeneratedOnAdd()
```

```
.HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
b.Property<string>("razonSocial")
```

```
.IsRequired()
```

```
.HasColumnType("nvarchar(450)");
```

```
b.ComplexProperty<Dictionary<string, object>>("direccion",  
"Empresa.LogicaDeNegocio.Entidades.Cliente.direccion#DireccionCliente", b1 =>
```

```
{
```

```
b1.IsRequired();
```

```
b1.Property<string>("Calle")
```

```
.IsRequired()
```

```
.HasColumnType("nvarchar(max)");
```

```
b1.Property<string>("Ciudad")
```

```
.IsRequired()
```

```
.HasColumnType("nvarchar(max)");
```

```
b1.Property<int>("Distancia")
```

```
.HasColumnType("int");
```

```
b1.Property<int>("Numero")
```

```
.HasColumnType("int");
```

```
});
```

```
b.ComplexProperty<Dictionary<string, object>>("rut",  
"Empresa.LogicaDeNegocio.Entidades.Cliente.rut#RUT", b1 =>
```

```
{
```

```
b1.IsRequired();
```

```
b1.Property<long>("Rut")
```

```
.HasColumnType("bigint");
```

```
});
```

```
b.HasKey("Id");
```

```
b.HasIndex("razonSocial");
```

```
b.ToTable("Clientes");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Pedido", b =>
```

```
{
```

```
    b.Property<int>("Id")
```

```
        .ValueGeneratedOnAdd()
```

```
        .HasColumnType("int");
```

```
    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
    b.Property<string>("Discriminator")
```

```
        .IsRequired()
```

```
        .HasMaxLength(8)
```

```
        .HasColumnType("nvarchar(8)");
```

```
    b.Property<bool>("anulado")
```

```
        .HasColumnType("bit");
```

```
    b.Property<int>("clienteId")
```



```
.HasColumnType("int");
```

```
b.Property<int>("entregaPrometida")
```

```
.HasColumnType("int");
```

```
b.Property<bool>("entregado")
```

```
.HasColumnType("bit");
```

```
b.Property<DateTime>("fechaPedido")
```

```
.HasColumnType("datetime2");
```

```
b.Property<double>("precioFinal")
```

```
.HasColumnType("float");
```

```
b.Property<double>("recargo")
```

```
.HasColumnType("float");
```

```
b.ComplexProperty<Dictionary<string, object>>("iva",  
"Empresa.LogicaDeNegocio.Entidades.Pedido.iva#IVA", b1 =>
```

```
{
```

```
    b1.IsRequired();
```

```
    b1.Property<double>("valor")
```

```
        .HasColumnType("float");
```

```
});
```

```
b.HasKey("Id");
```

```
b.HasIndex("clienteId");
```

```
b.HasIndex("precioFinal");
```

```
b.ToTable("Pedidos");
```

```
b.HasDiscriminator<string>("Discriminator").HasValue("Pedido");
```

```
b.UseTphMappingStrategy();
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Sistema.Usuario", b =>
```

```
{
```

```
b.Property<int>("Id")
```

```
.ValueGeneratedOnAdd()
```

```
.HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```

        b.ComplexProperty<Dictionary<string, object>>("Contrasenia",
"Empresa.LogicaDeNegocio.Sistema.Usuario.Contrasenia#ContraseniaUsuario", b1 =>

        {

            b1.IsRequired();

            b1.Property<string>("Valor")

                .IsRequired()

                .HasColumnType("nvarchar(max)");

        });

```

```

        b.ComplexProperty<Dictionary<string, object>>("Email",
"Empresa.LogicaDeNegocio.Sistema.Usuario.Email#EmailUsuario", b1 =>

        {

            b1.IsRequired();

            b1.Property<string>("Direccion")

                .IsRequired()

                .HasColumnType("nvarchar(max)");

        });

```

```

        b.ComplexProperty<Dictionary<string, object>>("NombreCompleto",
"Empresa.LogicaDeNegocio.Sistema.Usuario.NombreCompleto#NombreCompleto", b1 =>

        {

            b1.IsRequired();

```

```
        b1.Property<string>("Apellido")

            .IsRequired()

            .HasColumnType("nvarchar(max)");

        b1.Property<string>("Nombre")

            .IsRequired()

            .HasColumnType("nvarchar(max)");

    });
```

```
b.HasKey("Id");
```

```
b.ToTable("Usuarios");

});
```

```
modelBuilder.Entity("Papeleria.LogicaNegocio.Entidades.LineaPedido", b =>

{

    b.Property<int>("Id")

        .ValueGeneratedOnAdd()

        .HasColumnType("int");

    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

});
```

```
b.Property<int>("ArticuloId")
```

```
.HasColumnType("int");
```

```
b.Property<int>("Cantidad")
```

```
.HasColumnType("int");
```

```
b.Property<double>("PrecioUnitarioVigente")
```

```
.HasColumnType("float");
```

```
b.Property<int>("pedidoId")
```

```
.HasColumnType("int");
```

```
b.HasKey("Id");
```

```
b.HasIndex("ArticuloId");
```

```
b.HasIndex("pedidoId");
```

```
b.ToTable("LineasPedidos");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Comunes", b =>
```

```
{
```

```
b.HasBaseType("Empresa.LogicaDeNegocio.Entidades.Pedido");
```

```
b.HasDiscriminator().HasValue("Comunes");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Express", b =>
```

```
{
```

```
b.HasBaseType("Empresa.LogicaDeNegocio.Entidades.Pedido");
```

```
b.HasDiscriminator().HasValue("Express");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Pedido", b =>
```

```
{
```

```
b.HasOne("Empresa.LogicaDeNegocio.Entidades.Cliente", "cliente")
```

```
.WithMany("pedidos")
```

```
.HasForeignKey("clienteId")
```

```
.OnDelete(DeleteBehavior.Cascade)
```

```
.IsRequired();
```

```
b.Navigation("cliente");
```

```
});
```

```
modelBuilder.Entity("Papeleria.LogicaNegocio.Entidades.LineaPedido", b =>
```

```
{
```

```
    b.HasOne("Empresa.LogicaDeNegocio.Entidades.Articulo", "Articulo")
```

```
        .WithMany()
```

```
        .HasForeignKey("ArticuloId")
```

```
        .OnDelete(DeleteBehavior.Cascade)
```

```
        .IsRequired();
```

```
    b.HasOne("Empresa.LogicaDeNegocio.Entidades.Pedido", "pedido")
```

```
        .WithMany("lineas")
```

```
        .HasForeignKey("pedidoId")
```

```
        .OnDelete(DeleteBehavior.Cascade)
```

```
        .IsRequired();
```

```
    b.Navigation("Articulo");
```

```
    b.Navigation("pedido");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Cliente", b =>
```

```
{
```

```
    b.Navigation("pedidos");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Pedido", b =>
```

```
{
```

```
    b.Navigation("lineas");
```

```
});
```

```
#pragma warning restore 612, 618
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: PapeleriaContextModelSnapshot.cs

Localizacion: Papeleria.AccesoDatos\Migrations\PapeleriaContextModelSnapshot.cs

```
*****
```

```
// <auto-generated />
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using Microsoft.EntityFrameworkCore;
```

```
using Microsoft.EntityFrameworkCore.Infrastructure;
```

```
using Microsoft.EntityFrameworkCore.Metadata;
```

```
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
```

```
using Papeleria.AccesoDatos.EF;
```



```
#nullable disable
```

```
namespace Papeleria.AccesoDatos.Migrations
```

```
{
```

```
[DbContext(typeof(PapeleriaContext))]
```

```
partial class PapeleriaContextModelSnapshot : ModelSnapshot
```

```
{
```

```
protected override void BuildModel(ModelBuilder modelBuilder)
```

```
{
```

```
#pragma warning disable 612, 618
```

```
modelBuilder
```

```
.HasAnnotation("ProductVersion", "8.0.4")
```

```
.HasAnnotation("Relational:MaxIdentifierLength", 128);
```

```
SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Articulo", b =>
```

```
{
```

```
b.Property<int>("Id")
```

```
.ValueGeneratedOnAdd()
```

```
.HasColumnType("int");
```

```
SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
b.Property<double>("PrecioVP")
```

```
.HasColumnType("float");
```

```
b.ComplexProperty<Dictionary<string, object>>("CodigoProveedor",  
"Empresa.LogicaDeNegocio.Entidades.Articulo.CodigoProveedor#CodigoProveedorArticulos", b1 =>
```

```
{
```

```
b1.IsRequired();
```

```
b1.Property<long>("codigo")
```

```
.HasColumnType("bigint");
```

```
});
```

```
b.ComplexProperty<Dictionary<string, object>>("Descripcion",  
"Empresa.LogicaDeNegocio.Entidades.Articulo.Descripcion#DescripcionArticulo", b1 =>
```

```
{
```

```
b1.IsRequired();
```

```
b1.Property<string>("Descripcion")
```

```
.IsRequired()
```

```
.HasColumnType("nvarchar(max)");
```

```
});
```

```

        b.ComplexProperty<Dictionary<string, object>>("NombreArticulo",
"Empresa.LogicaDeNegocio.Entidades.Articulo.NombreArticulo#NombreArticulo", b1 =>

        {

            b1.IsRequired();

            b1.Property<string>("Nombre")

                .IsRequired()

                .HasColumnType("nvarchar(max)");

        });

```

```

        b.ComplexProperty<Dictionary<string, object>>("Stock",
"Empresa.LogicaDeNegocio.Entidades.Articulo.Stock#StockArticulo", b1 =>

        {

            b1.IsRequired();

            b1.Property<int>("cantidad")

                .HasColumnType("int");

        });

```

```

        b.HasKey("Id");

        b.ToTable("Articulos");

    });

```

```

modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Cliente", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

    b.Property<string>("razonSocial")
        .IsRequired()
        .HasColumnType("nvarchar(450)");

    b.ComplexProperty<Dictionary<string, object>>("direccion",
"Empresa.LogicaDeNegocio.Entidades.Cliente.direccion#DireccionCliente", b1 =>
    {
        b1.IsRequired();

        b1.Property<string>("Calle")
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b1.Property<string>("Ciudad")
            .IsRequired()
            .HasColumnType("nvarchar(max)");
    }
    );
}

```

```
b1.Property<int>("Distancia")
```

```
.HasColumnType("int");
```

```
b1.Property<int>("Numero")
```

```
.HasColumnType("int");
```

```
});
```

```
b.ComplexProperty<Dictionary<string, object>>("rut",  
"Empresa.LogicaDeNegocio.Entidades.Cliente.rut#RUT", b1 =>
```

```
{
```

```
b1.IsRequired();
```

```
b1.Property<long>("Rut")
```

```
.HasColumnType("bigint");
```

```
});
```

```
b.HasKey("Id");
```

```
b.HasIndex("razonSocial");
```

```
b.ToTable("Clientes");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Pedido", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

    b.Property<string>("Discriminator")
        .IsRequired()
        .HasMaxLength(8)
        .HasColumnType("nvarchar(8)");

    b.Property<bool>("anulado")
        .HasColumnType("bit");

    b.Property<int>("clienteId")
        .HasColumnType("int");

    b.Property<int>("entregaPrometida")
        .HasColumnType("int");

    b.Property<bool>("entregado")
```

```
.HasColumnType("bit");
```

```
b.Property<DateTime>("fechaPedido")
```

```
.HasColumnType("datetime2");
```

```
b.Property<double>("precioFinal")
```

```
.HasColumnType("float");
```

```
b.Property<double>("recargo")
```

```
.HasColumnType("float");
```

```
b.ComplexProperty<Dictionary<string, object>>("iva",  
"Empresa.LogicaDeNegocio.Entidades.Pedido.iva#IVA", b1 =>
```

```
{
```

```
    b1.IsRequired();
```

```
    b1.Property<double>("valor")
```

```
        .HasColumnType("float");
```

```
});
```

```
b.HasKey("Id");
```

```
b.HasIndex("clienteId");
```

```
b.HasIndex("precioFinal");
```

```
b.ToTable("Pedidos");
```

```
b.HasDiscriminator<string>("Discriminator").HasValue("Pedido");
```

```
b.UseTphMappingStrategy();
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Sistema.Usuario", b =>
```

```
{
```

```
    b.Property<int>("Id")
```

```
        .ValueGeneratedOnAdd()
```

```
        .HasColumnType("int");
```

```
    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));
```

```
    b.ComplexProperty<Dictionary<string, object>>("Contrasenia",  
"Empresa.LogicaDeNegocio.Sistema.Usuario.Contrasenia#ContraseniaUsuario", b1 =>
```

```
{
```

```
    b1.IsRequired();
```

```
    b1.Property<string>("Valor")
```

```
        .IsRequired()
```



```
.HasColumnType("nvarchar(max)");

});
```

```
b.ComplexProperty<Dictionary<string, object>>("Email",
"Empresa.LogicaDeNegocio.Sistema.Usuario.Email#EmailUsuario", b1 =>

{

    b1.IsRequired();

    b1.Property<string>("Direccion")

        .IsRequired()

        .HasColumnType("nvarchar(max)");

});
```

```
b.ComplexProperty<Dictionary<string, object>>("NombreCompleto",
"Empresa.LogicaDeNegocio.Sistema.Usuario.NombreCompleto#NombreCompleto", b1 =>

{

    b1.IsRequired();

    b1.Property<string>("Apellido")

        .IsRequired()

        .HasColumnType("nvarchar(max)");

    b1.Property<string>("Nombre")

        .IsRequired()
```

```
        .HasColumnType("nvarchar(max)");

    });
```

```
b.HasKey("Id");
```

```
b.ToTable("Usuarios");

});
```

```
modelBuilder.Entity("Papeleria.LogicaNegocio.Entidades.LineaPedido", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

    b.Property<int>("ArticuloId")
        .HasColumnType("int");

    b.Property<int>("Cantidad")
        .HasColumnType("int");

    b.Property<double>("PrecioUnitarioVigente")
```

```
.HasColumnType("float");
```

```
b.Property<int>("pedidoId")
```

```
.HasColumnType("int");
```

```
b.HasKey("Id");
```

```
b.HasIndex("ArticuloId");
```

```
b.HasIndex("pedidoId");
```

```
b.ToTable("LineasPedidos");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Comunes", b =>
```

```
{
```

```
b.HasBaseType("Empresa.LogicaDeNegocio.Entidades.Pedido");
```

```
b.HasDiscriminator().HasValue("Comunes");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Express", b =>
```

```
{
```

```
b.HasBaseType("Empresa.LogicaDeNegocio.Entidades.Pedido");
```

```
b.HasDiscriminator().HasValue("Express");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Pedido", b =>
```

```
{
```

```
b.HasOne("Empresa.LogicaDeNegocio.Entidades.Cliente", "cliente")
```

```
.WithMany("pedidos")
```

```
.HasForeignKey("clienteId")
```

```
.OnDelete(DeleteBehavior.Cascade)
```

```
.IsRequired();
```

```
b.Navigation("cliente");
```

```
});
```

```
modelBuilder.Entity("Papeleria.LogicaNegocio.Entidades.LineaPedido", b =>
```

```
{
```

```
b.HasOne("Empresa.LogicaDeNegocio.Entidades.Articulo", "Articulo")
```

```
.WithMany()
```

```
.HasForeignKey("ArticuloId")
```

```
.OnDelete(DeleteBehavior.Cascade)
```

```
.IsRequired();
```

```
b.HasOne("Empresa.LogicaDeNegocio.Entidades.Pedido", "pedido")
```

```
.WithMany("lineas")
```

```
.HasForeignKey("pedidoId")
```

```
.OnDelete(DeleteBehavior.Cascade)
```

```
.IsRequired();
```

```
b.Navigation("Articulo");
```

```
b.Navigation("pedido");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Cliente", b =>
```

```
{
```

```
    b.Navigation("pedidos");
```

```
});
```

```
modelBuilder.Entity("Empresa.LogicaDeNegocio.Entidades.Pedido", b =>
```

```
{
```

```
    b.Navigation("lineas");
```

```
});
```

```
#pragma warning restore 612, 618
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: IAlta.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasoDeUsoGeneral\IAlta.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.Interaces
```

```
{
```

```
    public interface IAlta<T>
```

```
    {
```

```
        public void Crear(T obj);
```

```
    }
```

```
}
```

```
*****
```

Archivo: IListar.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasoDeUsoGeneral\IListar.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.Interaces
```

```
{
```

```
    public interface IListar<T>
```

```
    {
```

```
        public IEnumerable<T> ListarTodo();
```

```
        public T ListarUno(int id);
```

```
        public IEnumerable<T> ListarPorNombre(string name);
```

```
        public T ListarUnoPorNombre(string nombre);
```

```
        public List<T> ListarSeleccionPorId(List<int> ids);
```

```
    }
```

```
}
```

Archivo: IRemove.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasoDeUsoGeneral\IRemove.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.Interaces
{

    public interface IRemove<T>

    {

        public void Remove(T obj);

    }

}
```

Archivo: IUpdate.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasoDeUsoGeneral\IUpdate.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```



```
namespace Papeleria.LogicaAplicacion.Interaces
```

```
{  
  
    public interface IUpdate<T>  
  
    {  
  
        public void Update(int id, T obj);  
  
    }  
  
}
```

```
*****
```

Archivo: Articulo.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\Articulo.cs

```
*****
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;  
  
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;  
  
using Papeleria.LogicaNegocio.Excepciones.Articulo;  
  
using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.CodigoProveedor;  
  
using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.DescripcionArticulo;  
  
using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepcions.Nombre;  
  
using Papeleria.LogicaNegocio.InterfacesEntidades;  
  
using System.Collections.Generic;  
  
using System.Timers;
```

```
namespace Empresa.LogicaDeNegocio.Entidades
```

```
{
```

```
public class Articulo : IValidable<Articulo>, IEquatable<Articulo>, IEntity
```

```
{
```

```
public int Id { get; set; }
```

```
public NombreArticulo NombreArticulo{ get; set; }
```

```
public CodigoProveedorArticulos CodigoProveedor{ get; set; }
```

```
public DescripcionArticulo Descripcion{ get; set; }
```

```
public double PrecioVP{ get; set; }
```

```
public StockArticulo Stock{ get; set; }
```

```
public Articulo(long codigoProveedor, string nombre, string descripcion, double precioVP, int stock)
```

```
{
```

```
    this.CodigoProveedor = new CodigoProveedorArticulos(codigoProveedor);
```

```
    this.NombreArticulo = new NombreArticulo(nombre);
```

```
    this.Descripcion = new DescripcionArticulo(descripcion);
```

```
    this.PrecioVP = precioVP;
```

```
    this.Stock = new StockArticulo(stock);
```

```
    esValido();
```

```
}
```

```
public Articulo()
```

```
{
```

```
}
```

```
public bool Equals(Articulo? other)
```

```
{
```

```
    if (other == null) return false;
```

```
    return this.NombreArticulo == other.NombreArticulo;
```

```
}
```

```
public void esValido()
```

```
{
```

```
    if (CodigoProveedor==null) {
```

```
        throw new CodigoProveedorNuloException("El codigo de proveedor no puede ser nulo.");
```

```
    }
```

```
    if (NombreArticulo==null) {
```

```
        throw new NombreNuloException("El nombre del articulo no puede ser nulo.");
```

```
    }
```

```
    if (Descripcion==null) {
```

```
        throw new DescripcionArticuloNuloException("La descripcion no puede ser nula.");
```

```
    }
```

```
}
```

```
public void CambiarPrecioVP(int nuevoPrecio){
```

```
    PrecioVP = nuevoPrecio;
```

```
}
```

```
public void ModificarDatos(Articulo obj)
```

```
{
```

```
    this.CodigoProveedor = obj.CodigoProveedor;
```

```
    this.NombreArticulo = obj.NombreArticulo;
```

```
    this.Descripcion = obj.Descripcion;
```

```
    this.PrecioVP = obj.PrecioVP;
```

```
    this.Stock = obj.Stock;
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: Cliente.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\Cliente.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Sistema;
```

```
using Microsoft.EntityFrameworkCore;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.Direccion;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RazonSocial;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RUT;

using Papeleria.LogicaNegocio.InterfacesEntidades;

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

namespace Empresa.LogicaDeNegocio.Entidades

{

    public class Cliente: IValidable<Cliente>, IEquatable<Cliente>, IEntity

    {

        public int Id { get; set; }

        [Key]

        public RUT rut{ get; set; }

        public RazonSocial razonSocial{ get; set; }

        public DireccionCliente direccion{ get; set; }

        public List<Pedido> pedidos{ get; set; }

        public Cliente(long rut, string razonSocial, string calle, int numero, string ciudad,int distancia)

        {
```

```
this.rut = new RUT(rut);

this.razonSocial = new RazonSocial(razonSocial);

this.direccion = new DireccionCliente(calle, numero, ciudad, distancia);

this.pedidos = new List<Pedido>();

esValido();

}

public Cliente()

{

}

public void esValido()

{

    if (rut == null) {

        throw new RutNuloException("El RUT no puede ser nulo.");

    }

    if (razonSocial == null) {

        throw new RazonSocialNuloException("La razon social no puede ser nula.");

    }

    if (direccion == null)

    {

        throw new DireccionNuloException("La direccion no puede ser nula.");

    }

}
```

```
}
```

```
}
```

```
public void Update(Cliente obj)
```

```
{
```

```
    esValido();
```

```
    rut = obj.rut;
```

```
    razonSocial = obj.razonSocial;
```

```
    direccion = obj.direccion;
```

```
}
```

```
public bool Equals(Cliente? other)
```

```
{
```

```
    if (other == null) return false;
```

```
    return this.rut == other.rut;
```

```
}
```

```
public override bool Equals(object? obj)
```

```
{
```

```
    if (obj == null) return false;
```

```
    return this.Equals(obj as Cliente);
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: Comunes.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\Comunes.cs

```
*****
```

```
using Papeleria.LogicaNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Pedido;
```

```
using System;
```

```
namespace Empresa.LogicaDeNegocio.Entidades
```

```
{
```

```
    public class Comunes : Pedido
```

```
    {
```

```
        public Comunes(): base()
```

```
        {
```

```
            this.fechaPedido = DateTime.Now;
```

```
            this.entregado = false;
```

```
            this.anulado = false;
```



```
esValido();
```

```
}
```

```
public override void AgregarLineaPedido(Articulo articulo, int cantidad)
```

```
{
```

```
    base.AgregarLineaPedido(articulo, cantidad);
```

```
}
```

```
public override double CalcularRecargoYFijar()
```

```
{
```

```
    // Recargo del 5% si la distancia a la direcci❖n de entrega supera los 100 km
```

```
    if (cliente.direccion.Distance > 100)
```

```
    {
```

```
        this.recargo = 0.05;
```

```
        return 0.05;
```

```
    }
```

```
    this.recargo = 0;
```

```
    return 0;
```

```
}
```

```
public override double CalcularYFijarPrecio(IVA iva)
```

```
{
```

```
    this.iva = iva;
```

```
double subtotal = 0;

foreach (LineaPedido line in lineas)

{

    subtotal += line.PrecioUnitarioVigente * line.Cantidad;

}

this.precioFinal = (subtotal * (1 + recargo)) * (1 + iva.valor);

return precioFinal;

}
```

```
public override void CambiarEntregaPrometida(int dias)

{

    entregaPrometida = dias;

}
```

```
public override bool Equals(object? obj)

{

    return base.Equals(obj);

}
```

```
public override bool Equals(Pedido? other)

{

    return base.Equals(other);

}
```

```
public override void SetearEntregado()
```

```
{
```

```
    base.SetearEntregado();
```

```
}
```

```
public override void AnularPedido()
```

```
{
```

```
    base.AnularPedido();
```

```
}
```

```
public override void esValido()
```

```
{
```

```
    base.esValido();
```

```
}
```

```
public override int FijarEntregaPrometida(int dias)
```

```
{
```

```
    this.entregaPrometida = dias;
```

```
    return dias;
```

```
}
```

```
public override string? ToString()
```

```
{
```

```
    return base.ToString();
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: Express.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\Express.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Pedido;
```

```
using System;
```

```
namespace Empresa.LogicaDeNegocio.Entidades
```

```
{
```

```
    public class Express : Pedido
```

```
    {
```

```
        public Express():base()
```

```
    {
```

```
}
```

```
public Express(Cliente cliente, int dias, IVA iva, List<LineaPedido> lista):base(cliente,dias,iva,lista)
```

```
{
```

```
    this.fechaPedido = DateTime.Now;
```

```
    this.cliente = cliente;
```

```
    this.lineas = lista;
```

```
    this.recargo = CalcularRecargoYFijar();
```

```
    this.iva = iva;
```

```
    this.entregaPrometida = FijarEntregaPrometida(dias);
```

```
    this.precioFinal = CalcularYFijarPrecio(iva);
```

```
    this.entregado = false;
```

```
    this.anulado = false;
```

```
    esValido();
```

```
}
```

```
public override double CalcularRecargoYFijar()
```

```
{
```

```
    this.recargo = 0.10;
```

```
    if (entregaPrometida == 0) {
```

```
        this.recargo = 0.15;
```

```
    }
```

```
    return recargo;
```

```
}
```

```
public override double CalcularYFijarPrecio(IVA iva)
```

```
{
```

```
    this.iva = iva;
```

```
    double subtotal = 0;
```

```
    foreach(LineaPedido line in lineas)
```

```
    {
```

```
        subtotal += line.PrecioUnitarioVigente * line.Cantidad;
```

```
    }
```

```
    this.precioFinal = (subtotal * (1 + recargo)) * (1 + iva.valor);
```

```
    return precioFinal;
```

```
}
```

```
public override void AgregarLineaPedido(Articulo articulo, int cantidad)
```

```
{
```

```
    base.AgregarLineaPedido(articulo, cantidad);
```

```
}
```

```
public override void SetearEntregado()
```

```
{
```

```
    base.SetearEntregado();
```

```
}
```

```
public override void AnularPedido()
```

```
{  
  
    base.AnularPedido();  
  
}  
  
public override void AgregarLineaPedido(LineaPedido linea)  
  
{  
  
    base.AgregarLineaPedido(linea);  
  
}  
  
  
  
public override bool Equals(object? obj)  
  
{  
  
    return base.Equals(obj);  
  
}  
  
  
  
public override void esValido()  
  
{  
  
    base.esValido();  
  
}  
  
  
  
public override bool Equals(Pedido? other)  
  
{  
  
    return base.Equals(other);  
  
}
```

```
public override int FijarEntregaPrometida(int dias)
```

```
{
```

```
    this.entregaPrometida = dias;
```

```
    return dias;
```

```
}
```

```
public override void CambiarEntregaPrometida(int dias)
```

```
{
```

```
    entregaPrometida = dias;
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: LineaPedido.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\LineaPedido.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.Linea;
```

```
using Papeleria.LogicaNegocio.InterfacesEntidades;
```



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

```
namespace Papeleria.LogicaNegocio.Entidades
```

```
{
```

```
    public class LineaPedido : IValidable<LineaPedido>, IEntity
```

```
    {
```

```
        public int Id { get; set; }
```

```
        public Articulo Articulo { get; set; }
```

```
        public int Cantidad { get; set; }
```

```
        public double PrecioUnitarioVigente { get; set; }
```

```
        public Pedido pedido { get; set; }
```

```
        public LineaPedido(Articulo articulo, int cantidad)
```

```
        {
```

```
            Articulo = articulo;
```

```
            Cantidad = cantidad;
```

```
            PrecioUnitarioVigente = articulo.PrecioVP;
```

```
            esValido();
```

```
            this.pedido = null;
```

```
        }
```

```
        public LineaPedido()
```

```
        {
```

```
}
```

```
public void AsignarPedido(Pedido pedido)
```

```
{
```

```
    this.pedido = pedido;
```

```
}
```

```
public void esValido()
```

```
{
```

```
    if (Articulo == null)
```

```
    {
```

```
        throw new ArticuloNuloException("El articulo no puede ser nulo en la linea del pedido.");
```

```
    }
```

```
    if (Cantidad == null || Cantidad < 1)
```

```
    {
```

```
        throw new LineaNuloException("La cantidad no puede ser nulo o menor a 1");
```

```
    }
```

```
}
```

```
}
```

```
}
```

Archivo: Pedido.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\Pedido.cs

```
using System;

using System.Collections.Generic;

using Microsoft.EntityFrameworkCore;

using Papeleria.LogicaNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.InterfacesEntidades;

namespace Empresa.LogicaDeNegocio.Entidades

{

    [Index(nameof(precioFinal))]

    public abstract class Pedido : IValidable<Pedido>, IEquatable<Pedido>, IEntity

    {

        public int Id { get; set; }

        public DateTime fechaPedido { get; set; }
```

```
public Cliente cliente { get; set; }
```

```
public List<LineaPedido> lineas { get; set; }
```

```
public double recargo { get; set; }
```

```
public IVA iva { get; set; }
```

```
public int entregaPrometida { get; set; }
```

```
public double precioFinal { get; set; }
```

```
public bool entregado { get; set; }
```

```
public bool anulado { get; set; }
```

```
public Pedido()
```

```
{
```

```
    this.fechaPedido = DateTime.Now;
```

```
    this.lineas = new List<LineaPedido>();
```

```
    this.entregado = false;
```

```
    this.anulado = false;
```

```
}
```

```
public Pedido(Cliente cliente, int dias, IVA iva, List<LineaPedido> lista)
```

```
{
```

```
    this.fechaPedido = DateTime.Now;
```

```
    this.cliente = cliente;
```

```
this.lineas = lista;

this.recargo = CalcularRecargoYFijar();

this.iva = iva;

this.entregaPrometida = FijarEntregaPrometida(dias);

this.precioFinal = CalcularYFijarPrecio(iva);

esValido();

}
```

```
public abstract double CalcularYFijarPrecio(IVA iva);

public abstract int FijarEntregaPrometida(int dias);

public abstract void CambiarEntregaPrometida(int dias);
```

```
public abstract double CalcularRecargoYFijar();
```

```
public virtual void SetearEntregado()

{

    if (this.anulado)

    {

        throw new Exception("No se puede entregar un pedido que haya sido anulado"); ;

    }

    else

    {

        this.entregado = true;
```

```
    }  
}
```

```
public virtual void AnularPedido()  
{  
    if (this.entregado)  
    {  
        throw new Exception("No se puede anular un pedido ya entregado");  
    }  
    else  
    {  
        this.anulado = true;  
    }  
}
```

```
public virtual void AgregarLineaPedido(Articulo articulo, int cantidad)  
{  
    try  
    {  
        LineaPedido pedido = new LineaPedido(articulo, cantidad);  
        lineas.Add(pedido);  
    }  
}
```

```
        catch (Exception ex)

        {

            throw new PedidoNoValidoException(ex.Message);

        }

    }
```

```
public virtual void AgregarLineaPedido(LineaPedido linea)

{

    try

    {

        lineas.Add(linea);

    }

    catch (Exception ex)

    {

        throw new PedidoNoValidoException(ex.Message);

    }

}
```

```
public virtual void esValido()

{

    /*if (cliente == null)

    {
```

```

        throw new ClienteNuloException("El cliente no puede ser nulo.");

    }*/

    /*if(fechaPedido < DateTime.Now) { throw new PedidoNuloException("La fecha de pedido no
puede ser inferior a la actual"); }*/

    if(entregaPrometida < 0) { throw new PedidoNuloException("La fecha de entrega no puede ser
anterior a la fecha del pedido"); }

    }

public virtual bool Equals(Pedido? other)

{

    if (other == null)

        throw new PedidoNuloException("Debe incluir el pedido a comparar");

    return this.Id == other.Id;

}

}

}

```

Archivo: Usuario.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\Usuario.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Contrasenia;

using Papeleria.LogicaNegocio.InterfacesEntidades;


namespace Empresa.LogicaDeNegocio.Sistema

{

    public class Usuario : IValidable<Usuario>, IEquatable<Usuario>, IEntity

        {

            public int Id { get; set; }

            public EmailUsuario Email{ get; set; }


            public NombreCompleto NombreCompleto{ get; set; }

            public ContraseniaUsuario Contrasenia{ get; set; }


            public Usuario(string email, string nombre, string apellido, string contrasenia)

            {

                this.Email = new EmailUsuario(email);

                this.NombreCompleto = new NombreCompleto(nombre, apellido);

                this.Contrasenia = new ContraseniaUsuario(contrasenia);

```

```
        esValido();

    }

    public Usuario()

    {

    }

    public bool Equals(Usuario? other)

    {

        if (other == null)

            throw new ArgumentNullException("Debe incluir el autor a comparar");

        return this.Id == other.Id || this.Email == other.Email;

    }

    public void esValido()

    {

        esValido(this);

    }

    public void esValido(Usuario usuario)

    {

        if(Email == null) {

            throw new UsuarioNoValidoExcepcion("El email no puede ser nulo para crear el usuario.");
```

```

    }

    if(NombreCompleto == null) {

        throw new UsuarioNoValidoExcepcion("El nombre y apellido no puede ser nulo para crear el
usuario.");

    }

    if(Contrasenia == null) {

        throw new UsuarioNoValidoExcepcion("La contrase❖a no puede ser nula para crear el
usuario.");

    }

}

public void ModificarContrase❖a(string contrasenia)

{

    if (contrasenia == null)

        throw new ContraseniaNuloException("La contrase❖a no puede ser nula.");

    this.Contrasenia = new ContraseniaUsuario(contrasenia);

}

public void ModificarDatos(Usuario usu) {

    if (usu.Contrasenia == null)

        throw new ContraseniaNuloException("La contrase❖a no puede ser nula.");

    if (usu.NombreCompleto == null)

        throw new ContraseniaNuloException("La contrase❖a no puede ser nula.");

    this.NombreCompleto = usu.NombreCompleto;

    this.Contrasenia = usu.Contrasenia;

```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: IEntity.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesEntidades\IEntity.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Sistema;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects;
```

```
namespace Empresa.LogicaDeNegocio.Entidades
```

```
{
```

```
    public interface IEntity
```

```
    {
```

```
        public int Id { get; set; }
```

```
    }
```

```
}
```

Archivo: IValidable.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesEntidades\IValidable.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario;
```

```
namespace Papeleria.LogicaNegocio.InterfacesEntidades
```

```
{

    public interface IValidable<T> where T : class

    {

        void esValido();

    }

}
```

Archivo: IRepository.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositorio.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.InterfacesRepositorio

{

public interface IRepositoryo<T>

{

public T GetById(int id);

public void Add(T obj);

public void Update(int id, T obj);

public void Remove(int id);

public void Remove(T obj);

public IEnumerable<T> GetAll();

public IEnumerable<T> GetObjectsByID(List<int> ids);

}

}

Archivo: IRepositoryoArticulo.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositorioArticulo.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.InterfacesRepositorio
```

```
{

    public interface IRepositorioArticulo:IRepositorio<Articulo>

    {

        public IEnumerable<Articulo> GetArticulosOrdenadosAlfabeticamente();

        public Articulo GetArticuloByCodigo(CodigoProveedorArticulos codigo);

        public bool ExisteArticuloConNombre(string nombre);

    }

}
```

Archivo: IRepositorioCliente.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositorioCliente.cs

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.InterfacesRepositorio

{

public interface IRepositoryoCliente:IRepositoryo<Cliente>

{

IEnumerable<Cliente> GetClientesPorPedido(int idPedido);

public IEnumerable<Cliente> GetClientes();

public Cliente GetById(int id);

public Cliente GetClientePorRUT(long rut);

public Cliente GetClientePorRazon(string rsocial);

public Cliente GetClientePorDireccion(DireccionCliente direccionCliente);

public IEnumerable<Cliente> GetClientesPedidoSupereMonto(double monto);

public Cliente GetCliente(int idCliente);

}

}

Archivo: IRepositoryComun.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositoryComun.cs

using Empresa.LogicaDeNegocio.Entidades;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.InterfacesRepositorio

{

public interface IRepositoryComun:IRepository<Comunes>

{

}

}

Archivo: IRepositoryExpress.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositoryExpress.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaNegocio.InterfacesRepositorio

{

    public interface IRepositoryExpress:IRepository<Express>

    {

    }

}
```

Archivo: IRepositoryLineaPedido.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositoryLineaPedido.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades;

using System;

using System.Collections.Generic;

using System.Linq;
```

```

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaNegocio.InterfacesRepositorio
{

    public interface IRepositoryLineaPedido:IRepository<LineaPedido>

    {

        public IEnumerable<LineaPedido> GetAll();

        public LineaPedido Get(int id);

        public IEnumerable<LineaPedido> GetByArticulo(Articulo articulo);

    }

}

```

Archivo: IRepositoryPedido.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositoryPedido.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using System;

using System.Collections.Generic;

using System.Linq;

```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.InterfacesRepositorio
```

```
{  
  
    public interface IRepositoryPedido:IRepository<Pedido>  
  
    {  
  
        public IEnumerable<Pedido> GetPedidosPorCliente(Cliente cliente);  
  
        public IEnumerable<Pedido> GetPedidosPorRUT(RUT rut);  
  
        public IEnumerable<Pedido> GetPedidosPorRazon(RazonSocial rsocial);  
  
        public IEnumerable<Pedido> GetPedidosPorDireccion(DireccionCliente direccionPedido);  
  
        public Pedido GetPedido(Pedido pedido);  
  
        public IEnumerable<Pedido> GetPedidosQueSuperenMonto(double monto);  
  
        public IEnumerable<Pedido> GetPedidosPorFecha(DateTime date);  
  
        public void Anular(int id);  
  
    }  
  
}
```

```
*****
```

Archivo: IRepositoryUsuario.cs

Localizacion: Papeleria.LogicaNegocio\InterfacesRepositorio\IRepositoryUsuario.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```

using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaNegocio.InterfacesRepositorio
{

    public interface IRepositoryUsuario:IRepository<Usuario>

    {

        public Usuario GetUsuarioPorEmail(string email);

        public void ModificarContrasenia(int id, ContraseniaUsuario contraseniaNueva);

        public Usuario Login(string email, string contrasenia);

        bool ExisteUsuarioConEmail(string email);

    }

}

```

Archivo: ArticulosController.cs

Localizacion: Papeleria.MVC\Controllers\ArticulosController.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using System.Net.Http.Headers;

using Azure;

using static System.Runtime.InteropServices.JavaScript.JSType;

using System.Runtime.Intrinsics.Arm;

using System.Text.Json.Serialization;

using System.Text.Json;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
namespace Papeleria.MVC.Controllers

{

    public class ArticulosController : Controller

    {

        private static IRepositoryArticulo _repoArticulos = new RepositorioArticuloEF();

        private static IAltaArticulo _altaArticulo;

        private static IGetAllArticulos _getAllArticulos;

        private static IUpdateArticulo _modificarArticulo;

        private static IGetArticulo _getArticulo;

        private static IBorrarArticulo _borrarArticulo;


        public ArticulosController()

        {

            _altaArticulo = new AltaArticulo(_repoArticulos);

            _getAllArticulos = new GetAllArticulos(_repoArticulos);

            _modificarArticulo = new UpdateArticulo(_repoArticulos);

            _getArticulo = new BuscarArticulo(_repoArticulos);

            _borrarArticulo = new BorrarArticulo(_repoArticulos);

        }

    }

}
```

```
public ActionResult Index()

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        var articulo = _getAllArticulos.Ejecutar();

        if (articulo == null || articulo.Count() == 0)

        {

            ViewBag.Mensaje = "No existen articulo";

        }

        ViewBag.Mensaje = $"Articulos en total: {articulo.Count()}.";

        return View(articulo);

    }

    return RedirectToAction("Index", "Home");

}

public ActionResult Create()

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        return View();

    }

    return RedirectToAction("Index", "Home");

}
```


[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(ArticuloDTO articulo)

```
{  
  
    try  
  
    {  
  
        _altaArticulo.Ejecutar(articulo);  
  
        TempData["MensajeOK"] = "Articulo creado";  
  
        return RedirectToAction(nameof(Index));  
  
    }  
  
    catch (Exception ex)  
  
    {  
  
        ViewBag.Error = ex.Message;  
  
        return View();  
  
    }  
  
}
```

public ActionResult Edit(int? id)

```
{  
  
    if (HttpContext.Session.GetInt32("LogueadoID") != null)  
  
    {  
  
        ArticuloDTO dto = _getArticulo.GetByIdDTO(id.GetValueOrDefault());  
  
        ArticuloDTO mod = new ArticuloDTO()  
  
        {
```

```

        Id = dto.Id,

        CodigoProveedor = dto.CodigoProveedor,

        Descripcion = dto.Descripcion,

        NombreArticulo = dto.NombreArticulo,

        PrecioVP = dto.PrecioVP,

        Stock = dto.Stock

    };

    if (dto == null)

        return View();

    return View(mod);

}

return RedirectToAction("Index", "Home");

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, ArticuloDTO articulo)

{

    try

    {

        _modificarArticulo.Ejecutar(id, articulo);

        return RedirectToAction(nameof(Index));

    }

    catch

```

```

    {

        return View();

    }

}

// GET: UsuariosController/Delete/5

public ActionResult Delete(int? id)

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        ArticuloDTO dto = _getArticulo.GetByIdDTO(id.GetValueOrDefault());

        ArticuloDTO borrar = new ArticuloDTO()

        {

            Id = dto.Id,

           CodigoProveedor = dto.CodigoProveedor,

            NombreArticulo = dto.NombreArticulo,

            Descripcion = dto.Descripcion,

            PrecioVP = dto.PrecioVP,

            Stock = dto.Stock,

        };

        if (dto == null)

            { return RedirectToAction("Index", "Articulos"); }

        return View(borrar);

    }

}

```

```
        return RedirectToAction("Index", "Home");
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Delete(int id, ArtículoDTO borrar, bool IsChecked)
    {
        try
        {
            if (IsChecked)
            {
                _borrarArticulo.Ejecutar(id);
            }
            else
            {
                ViewBag.msg = "Debe seleccionar el checkbox";
            }

            return RedirectToAction("Index", "Articulos");
        }
        catch
        {
            return View();
        }
    }
}
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: ClientesController.cs

Localizacion: Papeleria.MVC\Controllers\ClientesController.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Microsoft.AspNetCore.Http;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
using Papeleria.AccesoDatos.EF;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes;
```

```
using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
namespace Papeleria.MVC.Controllers
```

```
{
```

```
    public class ClientesController : Controller
```

```

{

private static IRepositoryPedido _pedidos = new RepositorioPedidoEF();

private static IRepositoryCliente _clientesRepo = new RepositorioClienteEF(_pedidos);

private static IBuscarClientes _buscarClientes;

private static IAltaCliente _altaCliente;

private static IModificarCliente _modificarCliente;

private static IBorrarCliente _borrarCliente;

public ClientesController()

{

    _buscarClientes = new BuscarClientes(_clientesRepo);

    _altaCliente = new AltaClientes(_clientesRepo);

    _modificarCliente = new ModificarCliente(_clientesRepo);

    _borrarCliente = new BorrarCliente(_clientesRepo);

}

// GET: ClientesController

public ActionResult Index()

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        var clientes = _buscarClientes.GetAll();

        if (clientes == null || clientes.Count() == 0)

        {

            ViewBag.Mensaje = "No existen clientes";

        }

    }

}

}

```

```

    }

    ViewBag.Mensaje = $"Clientes en total: {clientes.Count()}.";

    return View(clientes);

}

return RedirectToAction("Index", "Home");

}

```

[HttpPost]

```

public IActionResult Index(string razonSocial, double? monto, string submitButton)

{

    TempData["ResultadoBuscarCliente"] = "";

    try

    {

        if (!string.IsNullOrEmpty(submitButton))

        {

            if (submitButton == "Filtrar X Razon")

            {

                if (!string.IsNullOrEmpty(razonSocial))

                {

                    var clienteRazonSocial = _buscarClientes.GetXRazonSocial(razonSocial);

                    if (clienteRazonSocial == null)

                    {

                        TempData["ResultadoBuscarClientes"] = "No se ha encontrado ninguna coincidencia
para esa razon social.";

```

```
        return RedirectToAction("Index", "Clientes");

    }

    return View("Details", clienteRazonSocial);

}

else

{

    TempData["ResultadoBuscarClientes"] = "No se encontraron coincidencias de
búsqueda.";

    return RedirectToAction("Index", "Clientes");

}

}

else if (submitButton == "Filtrar X Monto")

{

    if (monto != null)

    {

        ViewBag.Monto = monto;

        var clientesSuperaronMonto = _buscarClientes.GetXMontoSuperado(monto.Value);

        if (!clientesSuperaronMonto.Any())

        {

            TempData["ResultadoBuscarClientes"] = "No se ha encontrado ninguna coincidencia
para ese monto.";

            return View();

        }

        return View("ListaSuperaronMonto", clientesSuperaronMonto);

    }

}
```



```

    }

    else

    {

        TempData["ResultadoBuscarClientes"] = "Ocurrió un error en la búsqueda";

        return RedirectToAction("Index", "Clientes");

    }

}

TempData["ResultadoBuscarClientes"] = "No se ha encontrado ninguna coincidencia.";

return RedirectToAction("Index", "Clientes");

}

catch (Exception e)

{

    TempData["ResultadoBuscarClientes"] = e.Message;

    return RedirectToAction("Index", "Clientes");

}

}

// GET: ClientesController/Create

public ActionResult Create()

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

```

```
        return View();

    }

    return RedirectToAction("Index", "Home");

}


// POST: ClientesController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(ClienteDTO cliente)

{

    try

    {

        _altaCliente.Ejecutar(cliente);

        TempData["MensajeOK"] = "Cliente creado";

        return RedirectToAction(nameof(Index));

    }

    catch (Exception ex)

    {

        ViewBag.Error = ex.Message;

        return View();

    }

}
```

```
// GET: ClientesController/Edit/5
```

```
public ActionResult Edit(int id)
```

```
{
```

```
    if (HttpContext.Session.GetInt32("LogueadoID") != null)
```

```
    {
```

```
        ClienteDTO dto = _buscarClientes.GetByIdDTO(id);
```

```
        ClienteDTO mod = new ClienteDTO()
```

```
        {
```

```
            Id = dto.Id,
```

```
            rut = dto.rut,
```

```
            razonSocial = dto.razonSocial,
```

```
            Calle = dto.Calle,
```

```
            Numero = dto.Numero,
```

```
            Ciudad = dto.Ciudad,
```

```
            Distancia = dto.Distancia
```

```
        };
```

```
        if (dto == null)
```

```
            return View();
```

```
        return View(mod);
```

```
    }
```

```
    return RedirectToAction("Index", "Home");
```

```
}
```

// POST: ClientesController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, ClienteDTO mod)

{

try

{

 _modificarCliente.Ejecutar(id, mod);

 return RedirectToAction(nameof(Index));

}

catch (Exception ex)

{

 ViewBag.Error = ex.Message;

 return View();

}

}

// GET: ClientesController/Delete/5

public ActionResult Delete(int id)

{

 if (HttpContext.Session.GetInt32("LogueadoID") != null)

 {

 ClienteDTO dto = _buscarClientes.GetByIdDTO(id);

```

    ClienteDTO borrar = new ClienteDTO()

    {

        Id = dto.Id,

        rut = dto.rut,

        razonSocial = dto.razonSocial,

        Calle = dto.Calle,

        Numero = dto.Numero,

        Ciudad = dto.Ciudad,

        Distancia = dto.Distancia

    };

    if (dto == null)

        return RedirectToAction("Index", "Cliente");

    return View(borrar);

}

return RedirectToAction("Index", "Home");

}

// POST: ClientesController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, ClienteDTO borrar, bool IsChecked)

{

    try

```

```

{

    if (IsChecked)

    {

        _borrarCliente.Ejecutar(id, borrar);

    }

    else

    {

        TempData["Error"] = "Debe seleccionar el checkbox";

    }

    return RedirectToAction("Index", "Clientes");

}

catch (Exception ex)

{

    ViewBag.Error = ex.Message;

    return View();

}

}

}

```

Archivo: HomeController.cs

Localizacion: Papeleria.MVC\Controllers\HomeController.cs

```
using Empresa.LogicaDeNegocio.Sistema;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
using Microsoft.VisualStudio.Web.CodeGenerators.Mvc.Templates.BlazorIdentity.Pages.Manage;
```

```
using Papeleria.AccesoDatos.EF;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using Papeleria.MVC.Models;
```

```
using System.Diagnostics;
```

```
namespace Papeleria.MVC.Controllers
```

```
{
```

```
    public class HomeController : Controller
```

```
    {
```

```
        private static IRepositoryUsuario _repoUsuarios = new RepositorioUsuarioEF();
```

```
        private static IRepositoryArticulo _repoArticulo = new RepositorioArticuloEF();
```

```
        private readonly ILogger<HomeController> _logger;
```

```
        public HomeController(ILogger<HomeController> logger)
```

```
        {
```

```
            _logger = logger;
```

```
        }
```

```
public IActionResult Index()

{

    //Comentar si no quieres entrar directo

    /* Usuario usuLogin = _repoUsuarios.Login("prueba@prueba.com", "Prueba123!");

    HttpContext.Session.SetInt32("LogueadoID", usuLogin.Id);*/

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        return RedirectToAction("Index", "Usuarios");

    }

    else

    {

        return View();

    }

}
```

```
public IActionResult Login()

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        return RedirectToAction("Index", "Publicacion");

    }

}
```



```

    }

    else

    {

        return View();

    }

}

[HttpPost]

public IActionResult Login(string Email, string Contraseña)

{

    Usuario usuLogin = _repoUsuarios.Login(Email, Contraseña);

    if (usuLogin != null)

    {

        HttpContext.Session.SetInt32("LogueadoID", usuLogin.Id);

        HttpContext.Session.SetString("LogueadoEmail", usuLogin.Email.Direccion);

        return RedirectToAction("Index", "Usuarios");

    }

    else

    {

        TempData["Error"] = "Error en los datos ingresados";

        return RedirectToAction("Index", "Home");

    }

}

public IActionResult Logout()

```

```
{  
  
    if (HttpContext.Session.GetInt32("LogueadoID") != null)  
  
    {  
  
        HttpContext.Session.Clear();  
  
        return RedirectToAction("Index", "Home");  
  
    }  
  
    else  
  
    {  
  
        return RedirectToAction("Index", "Home");  
  
    }  
  
}  
  
}
```

Archivo: PedidosController.cs

Localizacion: Papeleria.MVC\Controllers\PedidosController.cs

using Empresa.LogicaDeNegocio.Entidades;

using Humanizer;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using Papeleria.AccesoDatos.EF;

```

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Pedidos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos;

using Papeleria.LogicaNegocio.Entidades;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using Papeleria.MVC.Models.PedidosModels;


namespace Papeleria.MVC.Controllers
{
    public class PedidosController : Controller
    {
        private static IRepositoryPedido _pedidos = new RepositorioPedidoEF();

        private static IRepositoryCliente _clientesRepo = new RepositorioClienteEF(_pedidos);

        private static IRepositoryLineaPedido _lineaPedido = new RepositorioLineaPedidoEF();

        private static IRepositoryArticulo _articulos = new RepositorioArticuloEF();

        private static IAltaPedido _altaPedido;
    }
}

```

```
private static IBuscarClientes _buscarClientes;

private static IGetArticulo _getArticulo = new BuscarArticulo(_articulos);

private static IGetAllArticulos _getAllArticulos;

private static IGetAllPedidos _getAllPedidos;

private static IGetPedido _getPedidos;

private static IAnularPedido _anularPedido;

private static PedidoDTO tempPedido;


public PedidosController()

{

    _buscarClientes = new BuscarClientes(_clientesRepo);

    _getAllArticulos = new GetAllArticulos(_articulos);

    _getAllPedidos = new GetAllPedidos(_pedidos);

    _getPedidos = new GetPedidos(_pedidos);

    _altaPedido = new AltaPedidos(_pedidos);

    _anularPedido = new AnularPedido(_pedidos);

    ViewBag.Clientes = _buscarClientes.GetAll();

    ViewBag.Articulos = _getAllArticulos.Ejecutar();

}

public IActionResult Index()

{

    ViewBag.Clientes = _buscarClientes.GetAll();

    ViewBag.Articulos = _getAllArticulos.Ejecutar();
```

```

tempPedido = null;

if (HttpContext.Session.GetInt32("LogueadoID") != null)

{

    var pedidos = _getAllPedidos.Ejecutar();

    if (pedidos == null || pedidos.Count() == 0)

    {

        ViewBag.Mensaje = "No hay pedidos registrados";

    }

    return View(pedidos);

}

return RedirectToAction("Index", "Home");

}

[HttpPost]

public IActionResult Index(DateTime date)

{

    _pedidos = new RepositorioPedidoEF();

    ViewBag.Clientes = _buscarClientes.GetAll();

    ViewBag.Articulos = _getAllArticulos.Ejecutar();

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        if (date != null)

        {

            var pedidos = _getPedidos.GetPedidosPorFecha(date);

```

```

        if (pedidos == null || pedidos.Count() == 0)
        {
            ViewBag.Mensaje = "No existen pedido desde la fecha indicada";
        }

        return View(pedidos);
    }

    else
    {
        RedirectToAction("Index", "Pedidos");
    }
}

return RedirectToAction("Index", "Home");
}

public IActionResult Details(int id)
{
    ViewBag.Clientes = _buscarClientes.GetAll();

    ViewBag.Articulos = _getAllArticulos.Ejecutar();

    if (HttpContext.Session.GetInt32("LogueadoID") != null)
    {
        PedidoDTO pedido = PedidosMappers.ToDto(_getPedidos.GetById(id));

        if (pedido == null)
        {
            return RedirectToAction("Index", "Pedidos");
        }

        return View(pedido);
    }
}

```

```

    }

    return RedirectToAction("Index", "Home");

}

[HttpGet]

public IActionResult Anular(int id)

{

    ViewBag.Clientes = _buscarClientes.GetAll();

    ViewBag.Articulos = _getAllArticulos.Ejecutar();

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        //PedidoDTO pedido = PedidosMappers.ToDto(_getPedidos.GetById(id));

        PedidoDTO pedido = _getPedidos.GetByIdDTO(id);

        if (pedido == null)

        { return RedirectToAction("Index", "Pedidos"); }

        return View(pedido);

    }

    return RedirectToAction("Index", "Home");

}

[HttpPost]

public IActionResult Anular(int id, bool isChecked)

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

```

```
ViewBag.Clientes = _buscarClientes.GetAll();

ViewBag.Articulos = _getAllArticulos.Ejecutar();

Pedido pedido = _getPedidos.GetById(id);

if (pedido == null)

{ return RedirectToAction("Index", "Pedidos"); }

try

{

    if (isChecked)

    {

        _anularPedido.Ejecutar(id);

    }

    else

    {

        ViewBag.msg = "Debe aceptar las responsabilidades (checkbox) antes de anular el pedido";

    }

    return RedirectToAction("Index", "Pedidos");

}

catch (Exception e)

{

    throw new Exception(e.Message);

}

return View(pedido);

}
```



```
        return RedirectToAction("Index", "Home");  
    }  
}
```

[HttpGet]

```
public IActionResult Crear()
```

```
{
```

```
    if (HttpContext.Session.GetInt32("LogueadoID") != null)
```

```
    {
```

```
        ViewBag.Clientes = _buscarClientes.GetAll();
```

```
        ViewBag.Articulos = _getAllArticulos.Ejecutar();
```

```
        if (tempPedido != null)
```

```
        {
```

```
            ViewBag.LineasPedido = tempPedido.LineasPedido;
```

```
        }
```

```
        var viewModel = new PedidoAltaModel
```

```
        {
```

```
            LineasPedido = new List<LineaPedidoModel> { new LineaPedidoModel() } // Al menos una  
línea de pedido
```

```
        };
```

```
        return View(viewModel);
```

```
    }
```

```
    return RedirectToAction("Index", "Home");
```

```
}
```

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Crear(PedidoDTO pedidoAlta)

{

 ViewBag.Clientes = _buscarClientes.GetAll();

 ViewBag.Articulos = _getAllArticulos.Ejecutar();

 try

 {

 if (pedidoAlta == null)

 {

 ViewBag.Error = "El pedido es invalido";

 return View();

 }

 if (pedidoAlta.FechaEntrega < DateTime.Now)

 {

 ViewBag.Error = "Fecha de entrega invalida";

 return View();

 }

 pedidoAlta.FechaPedido = DateTime.Now;

 if (tempPedido != null)

 {

 ViewBag.LineasPedido = tempPedido.LineasPedido;

 }

```
if (ViewBag.LineasPedido.Count() == 0)

{

    ViewBag.Error = "Debe agregar al menos un articulo al pedido";

    return View();

}

pedidoAlta.LineasPedido = tempPedido.LineasPedido;

if ((pedidoAlta.FechaEntrega.Subtract(pedidoAlta.FechaPedido)).Days < 5)

{

    _altaPedido.EjecutarExpress(pedidoAlta);

}

else

{

    _altaPedido.EjecutarComunes(pedidoAlta);

}

tempPedido = null;

return RedirectToAction("Index", "Pedidos");

}

catch (Exception ex)

{

    ViewBag.Error = ex.Message;

    return RedirectToAction("Index", "Pedidos");

}

}
```

[HttpPost]

```
public ActionResult AddArticulo(PedidoDTO pedido, int ArticuloId, int Cantidad)
```

```
{
```

```
    try
```

```
    {
```

```
        ArticuloDTO articulo = _getArticulo.GetByIdDTO(ArticuloId);
```

```
        LineaPedidoDTO altaLinea = new LineaPedidoDTO { idArticulo = articulo.Id, CodigoProveedor  
= articulo.CodigoProveedor, NombreArticulo = articulo.NombreArticulo, Descripcion =  
articulo.Descripcion, PrecioVP = articulo.PrecioVP, Stock = articulo.Stock, PrecioUnitario =  
articulo.PrecioVP, Cantidad = Cantidad, Subtotal = Cantidad * articulo.PrecioVP };
```

```
        if (tempPedido == null)
```

```
        {
```

```
            tempPedido = new PedidoDTO { LineasPedido = new List<LineaPedidoDTO>() };
```

```
        }
```

```
        tempPedido.LineasPedido.Add(altaLinea);
```

```
        return RedirectToAction(nameof(Crear));
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        ViewBag.Error = ex.Message;
```

```
        return View();
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: UsuariosController.cs

Localizacion: Papeleria.MVC\Controllers\UsuariosController.cs

```
*****
```

```
using Microsoft.AspNetCore.Http;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
using Papeleria.AccesoDatos.EF;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
namespace Papeleria.MVC.Controllers
```

```
{
```

```
    public class UsuariosController : Controller
```

```
    {
```

```
        private static IRepositoryUsuario _repoUsuarios = new RepositorioUsuarioEF();
```

```
        private static IAltaUsuario _altaUsuario;
```

```
private static IGetAllUsuarios _getAllUsuarios;

private static IModificarUsuario _modificarUsuario;

private static IGetUsuario _getUsuario;

private static IBorrarUsuario _borrarUsuario;

public UsuariosController()

{

    _altaUsuario = new AltaUsuarios(_repoUsuarios);

    _getAllUsuarios = new GetAllUsuarios(_repoUsuarios);

    _modificarUsuario = new ModificarUsuario(_repoUsuarios);

    _getUsuario = new BuscarUsuario(_repoUsuarios);

    _borrarUsuario = new BorrarUsuario(_repoUsuarios);

}

// GET: UsuariosController

public ActionResult Index()

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        var usuarios = _getAllUsuarios.Ejecutar();

        if (usuarios == null || usuarios.Count() == 0)

        {

            ViewBag.Mensaje = "No existen usuarios";

        }

    }

}
```

```
        ViewBag.Mensaje = $"Usuarios en total: {usuarios.Count()}.";

        return View(usuarios);

    }

    return RedirectToAction("Index", "Home");

}
```

// GET: UsuariosController/Create

```
public ActionResult Create()
```

```
{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        return View();

    }

    return RedirectToAction("Index", "Home");

}
```

// POST: UsuariosController/Create

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create(UsuarioDTO autorDto)
```

```
{

    try

    {
```

```

        _altaUsuario.Ejecutar(autorDto);

        TempData["MensajeOK"] = "Usuario creado";

        return RedirectToAction(nameof(Index));
    }

    catch (Exception ex)

    {

        ViewBag.Error = ex.Message;

        return View();
    }
}

// GET: UsuariosController/Edit/5

public ActionResult Edit(int? id)

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        UsuarioDTO dto = _getUsuario.GetByIdDTO(id.GetValueOrDefault());

        UsuarioDTO mod = new UsuarioDTO()

        {

            Id = dto.Id,

            Nombre = dto.Nombre,

            Apellido = dto.Apellido,

            Email = dto.Email,

```



```

        Contraseña = dto.Contraseña

    };

    if (dto == null)

        return View();

    return View(mod);

}

return RedirectToAction("Index", "Home");

}

// POST: UsuariosController/Edit/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, UsuarioDTO usu)

{

    try

    {

        _modificarUsuario.Ejecutar(id, usu);

        return RedirectToAction(nameof(Index));

    }

    catch (Exception ex)

    {

        ViewBag.Error = ex.Message;

        return View();

```

```

    }

}

// GET: UsuariosController/Delete/5

public ActionResult Delete(int? id)

{

    if (HttpContext.Session.GetInt32("LogueadoID") != null)

    {

        UsuarioDTO dto = _getUsuario.GetByIdDTO(id.GetValueOrDefault());

        UsuarioDTO borrar = new UsuarioDTO()

        {

            Id = dto.Id,

            Nombre = dto.Nombre,

            Apellido = dto.Apellido,

            Email = dto.Email,

            Contraseña = dto.Contraseña

        };

        if (dto == null)

            return RedirectToAction("Index", "Usuarios");

        if (dto.Id == HttpContext.Session.GetInt32("LogueadoID")) {

            TempData["Error"] = "No se puede borrar el usuario logueado";

            return RedirectToAction(nameof(Index));

        }
    }
}

```

```
        return View(borrar);

    }

    return RedirectToAction("Index", "Home");

}


// POST: UsuariosController/Delete/5

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Delete(int id, UsuarioDTO borrar, bool IsChecked)

{

    try

    {

        if (IsChecked)

        {

            _borrarUsuario.Ejecutar(id, borrar);

        }

        else

        {

            TempData["Error"] = "Debe seleccionar el checkbox";

        }

        return RedirectToAction("Index", "Usuarios");

    }

    catch (Exception ex)
```

```
{  
  
    ViewBag.Error = ex.Message;  
  
    return View();  
  
}  
  
}
```

//TODO 2:22:26 - <https://vimeopro.com/universidadortfi/fi-5212-programacion-3-cabella-69235-p3-m3a-remoto/video/929607409>

```
}  
  
}
```

Archivo: ArticuloModel.cs

Localizacion: Papeleria.MVC\Models\ArticuloModel.cs

namespace Papeleria.MVC.Models

```
{  
  
    public class ArticuloModel  
  
    {  
  
  
  
  
  
  
  
  
  
        public int Id { get; set; }  
  
        public longCodigoProveedor { get; set; }  
  
        public string NombreArticulo { get; set; }  
  
        public string Descripcion { get; set; }  
  
        public double PrecioVP { get; set; }  
  
    }  
  
}
```

```
        public int Stock { get; set; }

    }

}
```

Archivo: ErrorViewModel.cs

Localizacion: Papeleria.MVC\Models\ErrorViewModel.cs

```
namespace Papeleria.MVC.Models

{

    public class ErrorViewModel

    {

        public string? RequestId { get; set; }


        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

    }

}
```

Archivo: ArticulosController.cs

Localizacion: Papeleria.WebApi\Controllers\ArticulosController.cs

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Microsoft.AspNetCore.Mvc;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
// For more information on enabling Web API for empty projects, visit
https://go.microsoft.com/fwlink/?LinkID=397860
```

```
namespace Papeleria.WebApi.Controllers

{

    [Route("api/[controller]")]
    [ApiController]

    public class ArticulosController : ControllerBase
    {

        private IRepositoryArticulo _repoArticulos;

        private IGetAllArticulos _cuGetArticulos;

        private IGetArticulo _cuGetArticulo;

        private IAltaArticulo _cuAltaArticulo;

        private IBorrarArticulo _cuBorrarArticulo;

        private IUpdateArticulo _cuModificarArticulo;
```

```

public ArticulosController()

{

    _repoArticulos = new RepositorioArticuloEF();

    _cuGetArticulos = new GetAllArticulos(_repoArticulos);

    _cuGetArticulo = new BuscarArticulo(_repoArticulos);

    _cuAltaArticulo = new AltaArticulo(_repoArticulos);

    _cuBorrarArticulo = new BorrarArticulo(_repoArticulos);

    _cuModificarArticulo = new UpdateArticulo(_repoArticulos);

}

// GET: api/<ArticulosController>

/// <summary>

/// Listar todos los articulos

/// </summary>

/// <returns>Lista de articulos ordenados alfabeticamente.</returns>

[HttpGet]

public ActionResult<IEnumerable<ArticuloDTO>> Get()

{

    try

    {

        var articulosDto = _cuGetArticulos.Ejecutar();

        var ordenada = articulosDto.OrderBy(articulo => articulo.NombreArticulo);

        return Ok(ordenada);

    }

}

```

```
catch (ArticuloNoValidoException ex)
```

```
{
```

```
    return BadRequest(ex.Message);
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    return StatusCode(500, ex.Message);
```

```
}
```

```
}
```

```
// GET api/<ArticulosController>/5
```

```
/// <summary>
```

```
/// Listar articulo particuloar
```

```
/// </summary>
```

```
/// <param name="id">Número entero con el valor Id del articulo a buscar</param>
```

```
/// <returns>Articulo correspondiente al ID - Code 200 | Error 400 (Bad Request) si parametro/articulo  
es invalido | 500 - Error con la DB / Excepcion particular</returns>
```

```
[HttpGet("{id}", Name = "GetAutorByID")]
```

```
public ActionResult<ArticuloDTO> Get(int id)
```

```
{
```

```
    try
```

```
{
```

```
        var articuloDto = _cuGetArticulo.GetById(id);
```



```

        return Ok(articuloDto);
    }

    catch (ArticuloNoValidoException ex)

    {

        return BadRequest(ex.Message);

    }

    catch (Exception ex)

    {

        return StatusCode(500, ex.Message);

    }

}

```

// POST api/<ArticulosController>

/// <summary>

/// Agregar articulo

/// </summary>

/// <param name="articulo">Parametro que toma el articulo armado con sus respectivos atributos y lo
pasa a la aplicacion para registrarlo</param>

/// <returns>201 - Si el Articulo fue creado satisfactoriamente | 400 - Si el Articulo suministrado no es
valido | 500 - Error con la DB / Excepcion particular</returns>

[HttpPost]

public ActionResult<ArticuloDTO> Post(ArticuloDTO articulo)

```

{

    try

    {

```

```
        _cuAltaArticulo.Ejecutar(articulo);

        return CreatedAtRoute("GetAutorByID", new { id = articulo.Id }, articulo);

    }
}
```

```
    catch (ArticuloNoValidoException ex)
```

```
    {
```

```
        return BadRequest(ex.Message);
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        return StatusCode(500, ex.Message);
```

```
    }
```

```
}
```

```
// PUT api/<ArticulosController>/5
```

```
/// <summary>
```

```
/// Modificar articulo
```

```
/// </summary>
```

```
/// <param name="id">Proporciona el ID del objeto a modificar</param>
```

```
/// <param name="articulo">Proporciona el cuerpo del articulo que va a reemplazar al  
existente</param>
```

```
/// <returns>200 - Articulo modificado correctamente | 400 - ID/Articulo nuevo invalido | 500 - Error  
en la DB / Excepcion particular</returns>
```

```
[HttpPut("{id}")]
```

```
public ActionResult<ArticuloDTO> Put(int id, ArticuloDTO articulo)
```

```
{
```

```
    try
```

```
    {
```

```
        _cuModificarArticulo.Ejecutar(id, articulo);
```

```
        return Ok(articulo);
```

```
    }
```

```
    catch (ArticuloNoValidoException ex)
```

```
    {
```

```
        return BadRequest(ex.Message);
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        return StatusCode(500, ex.Message);
```

```
    }
```

```
}
```

```
// DELETE api/<ArticulosController>/5
```

```
/// <summary>
```

```
/// Borrar articulo
```

```
/// </summary>
```

```
/// <param name="id">Proporciona el ID del articulo a borrar</param>
```

```
/// <returns>200 - Artículo borrado correctamente | 400 - ID Invalido o Artículo no valido | 500 - Error  
de la DB / Excepcion particular</returns>
```

```
[HttpDelete("{id}")]
```

```
public ActionResult<ArticuloDTO> Delete(int id)
```

```
{
```

```
    try
```

```
    {
```

```
        _cuBorrarArticulo.Ejecutar(id);
```

```
        return Ok();
```

```
    }
```

```
    catch (ArticuloNoValidoException ex)
```

```
    {
```

```
        return BadRequest(ex.Message);
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        return StatusCode(500, ex.Message);
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: PedidosController.cs

Localizacion: Papeleria.WebApi\Controllers\PedidosController.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Microsoft.AspNetCore.Mvc;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Pedidos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos;

using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.InterfacesRepositorio;


namespace Papeleria.WebApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class PedidosController : ControllerBase
    {
        private static IRepositoryPedido _repoPedidos = new RepositorioPedidoEF();

        private static IRepositoryCliente _clientesRepo = new RepositorioClienteEF(_repoPedidos);

        private static IRepositoryLineaPedido _lineaPedido = new RepositorioLineaPedidoEF();
```

```

private static IRepositoryArticulo _articulos = new RepositorioArticuloEF();

private static IBuscarClientes _buscarClientes;

private static IGetAllArticulos _getAllArticulos;

private static IGetAllPedidos _getAllPedidos;

private static IGetPedido _getPedidos;

private static IAnularPedido _anularPedido;

public PedidosController()

{

    _buscarClientes = new BuscarClientes(_clientesRepo);

    _getAllArticulos = new GetAllArticulos(_articulos);

    _getAllPedidos = new GetAllPedidos(_repoPedidos);

    _getPedidos = new GetPedidos(_repoPedidos);

    _anularPedido = new AnularPedido(_repoPedidos);

}

// GET: api/<ArticulosController>

/// <summary>

/// Listar pedidos anulados

/// </summary>

/// <returns>200 - Retorna la lista con los pedidos anulados en orden descendente por Fecha | Status
correspondiente al error</returns>

[HttpGet]

public ActionResult<IEnumerable<PedidoDTO>> Get()

{

    try

```

```
{

    _repoPedidos = new RepositorioPedidoEF();

    var clientes = _buscarClientes.GetAll();

    var pedidosDto = _getAllPedidos.Ejecutar();

    var ordenada = pedidosDto.OrderByDescending(pedido => pedido.FechaPedido);

    var pedidosFinal = new List<PedidoDTO>();

    foreach (var pedido in ordenada)

    {

        if(pedido.anulado)

        {

            pedidosFinal.Add(pedido);

        }

    }

    return Ok(pedidosFinal);

}

catch (PedidoNoValidoException ex)

{

    return BadRequest(ex.Message);

}

catch (Exception ex)

{

    return StatusCode(500, ex.Message);

}
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: ArticulosController.cs

Localizacion: Papeleria.WebMvc\Controllers\ArticulosController.cs

```
*****
```

```
using Microsoft.AspNetCore.Http;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace Papeleria.WebMvc.Controllers
```

```
{
```

```
    public class ArticulosController : Controller
```

```
    {
```

```
        // GET: ArticulosController
```

```
        public ActionResult Index()
```

```
        {
```

```
            return View();
```

```
        }
```



```
// GET: ArticulosController/Details/5
```

```
public ActionResult Details(int id)
```

```
{  
  
    return View();  
}
```

```
// GET: ArticulosController/Create
```

```
public ActionResult Create()
```

```
{  
  
    return View();  
}
```

```
// POST: ArticulosController/Create
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create(IFormCollection collection)
```

```
{  
  
    try  
  
    {  
  
        return RedirectToAction(nameof(Index));  
  
    }  
  
    catch  
  
    {  
  

```

```
        return View();

    }

}
```

// GET: ArticulosController/Edit/5

```
public ActionResult Edit(int id)
```

```
{

    return View();

}
```

// POST: ArticulosController/Edit/5

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Edit(int id, IFormCollection collection)
```

```
{

    try

    {

        return RedirectToAction(nameof(Index));

    }

    catch

    {

        return View();

    }

}
```

```
}
```

```
// GET: ArticulosController/Delete/5
```

```
public ActionResult Delete(int id)
```

```
{
```

```
    return View();
```

```
}
```

```
// POST: ArticulosController/Delete/5
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Delete(int id, IFormCollection collection)
```

```
{
```

```
    try
```

```
    {
```

```
        return RedirectToAction(nameof(Index));
```

```
    }
```

```
    catch
```

```
    {
```

```
        return View();
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: ClientesController.cs

Localizacion: Papeleria.WebMvc\Controllers\ClientesController.cs

```
*****
```

```
using Microsoft.AspNetCore.Http;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace Papeleria.WebMvc.Controllers
```

```
{
```

```
    public class ClientesController : Controller
```

```
    {
```

```
        // GET: ClientesController
```

```
        public ActionResult Index()
```

```
        {
```

```
            return View();
```

```
        }
```

```
        // GET: ClientesController/Details/5
```

```
        public ActionResult Details(int id)
```

```
        {
```

```
            return View();
```

```
}
```

```
// GET: ClientesController/Create
```

```
public ActionResult Create()
```

```
{
```

```
    return View();
```

```
}
```

```
// POST: ClientesController/Create
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create(IFormCollection collection)
```

```
{
```

```
    try
```

```
    {
```

```
        return RedirectToAction(nameof(Index));
```

```
    }
```

```
    catch
```

```
    {
```

```
        return View();
```

```
    }
```

```
}
```

```
// GET: ClientesController/Edit/5
```

```
public ActionResult Edit(int id)
```

```
{  
  
    return View();  
}
```

```
// POST: ClientesController/Edit/5
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Edit(int id, IFormCollection collection)
```

```
{  
  
    try  
  
    {  
  
        return RedirectToAction(nameof(Index));  
  
    }  
  
    catch  
  
    {  
  
        return View();  
  
    }  
}
```

```
// GET: ClientesController/Delete/5
```

```
public ActionResult Delete(int id)
```

```
{  
  
    return View();  
  
}
```

```
// POST: ClientesController/Delete/5
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Delete(int id, IFormCollection collection)
```

```
{  
  
    try  
  
    {  
  
        return RedirectToAction(nameof(Index));  
  
    }  
  
    catch  
  
    {  
  
        return View();  
  
    }  
  
}
```

```
*****
```

Archivo: HomeController.cs

Localizacion: Papeleria.WebMvc\Controllers\HomeController.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
using Papeleria.WebMvc.Models;
```

```
using System.Diagnostics;
```

```
namespace Papeleria.WebMvc.Controllers
```

```
{
```

```
    public class HomeController : Controller
```

```
    {
```

```
        private readonly ILogger<HomeController> _logger;
```

```
        public HomeController(ILogger<HomeController> logger)
```

```
        {
```

```
            _logger = logger;
```

```
        }
```

```
        public IActionResult Index()
```

```
        {
```

```
            return View();
```

```
        }
```

```
        public IActionResult Privacy()
```



```
{  
  
    return View();  
  
}
```

```
[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]  
  
public IActionResult Error()  
  
{  
  
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??  
HttpContext.TraceIdentifier });  
  
}  
  
}
```

Archivo: PedidosController.cs

Localizacion: Papeleria.WebMvc\Controllers\PedidosController.cs

```
using Microsoft.AspNetCore.Http;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace Papeleria.WebMvc.Controllers
```

```
{  
  
    public class PedidosController : Controller  
  
    {
```

```
// GET: PedidosController
```

```
public ActionResult Index()
```

```
{  
  
    return View();  
}
```

```
// GET: PedidosController/Details/5
```

```
public ActionResult Details(int id)
```

```
{  
  
    return View();  
}
```

```
// GET: PedidosController/Create
```

```
public ActionResult Create()
```

```
{  
  
    return View();  
}
```

```
// POST: PedidosController/Create
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create(ICollection collection)
```

```
{
```

```
try

{

    return RedirectToAction(nameof(Index));

}

catch

{

    return View();

}

}
```

// GET: PedidosController/Edit/5

```
public ActionResult Edit(int id)
```

```
{

    return View();

}
```

// POST: PedidosController/Edit/5

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Edit(int id, IFormCollection collection)
```

```
{

    try

    {
```

```
        return RedirectToAction(nameof(Index));
    }

    catch

    {

        return View();
    }
}
```

// GET: PedidosController/Delete/5

```
public ActionResult Delete(int id)
```

```
{

    return View();
}
```

// POST: PedidosController/Delete/5

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Delete(int id, IFormCollection collection)
```

```
{

    try

    {

        return RedirectToAction(nameof(Index));
    }
}
```

```
        catch

        {

            return View();

        }

    }

}
```

Archivo: UsuariosController.cs

Localizacion: Papeleria.WebMvc\Controllers\UsuariosController.cs

```
using Microsoft.AspNetCore.Http;
```

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace Papeleria.WebMvc.Controllers
```

```
{
```

```
    public class UsuariosController : Controller
```

```
    {
```

```
        // GET: UsuariosController
```

```
        public ActionResult Index()
```

```
        {
```

```
            return View();
```

```
}
```

```
// GET: UsuariosController/Details/5
```

```
public ActionResult Details(int id)
```

```
{
```

```
    return View();
```

```
}
```

```
// GET: UsuariosController/Create
```

```
public ActionResult Create()
```

```
{
```

```
    return View();
```

```
}
```

```
// POST: UsuariosController/Create
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create(IFormCollection collection)
```

```
{
```

```
    try
```

```
    {
```

```
        return RedirectToAction(nameof(Index));
```

```
    }
```

```
        catch

        {

            return View();

        }

    }
```

// GET: UsuariosController/Edit/5

```
public ActionResult Edit(int id)

{

    return View();

}
```

// POST: UsuariosController/Edit/5

```
[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(int id, IFormCollection collection)

{

    try

    {

        return RedirectToAction(nameof(Index));

    }

    catch

    {

    }
```

```
        return View();  
    }  
}
```

// GET: UsuariosController/Delete/5

```
public ActionResult Delete(int id)  
  
{  
  
    return View();  
}
```

// POST: UsuariosController/Delete/5

```
[HttpPost]  
  
[ValidateAntiForgeryToken]  
  
public ActionResult Delete(int id, IFormCollection collection)  
  
{  
  
    try  
  
    {  
  
        return RedirectToAction(nameof(Index));  
  
    }  
  
    catch  
  
    {  
  
        return View();  
  
    }  
}
```



```
    }  
  
    }  
  
}
```

Archivo: ErrorViewModel.cs

Localizacion: Papeleria.WebMvc\Models\ErrorViewModel.cs

namespace Papeleria.WebMvc.Models

```
{  
  
    public class ErrorViewModel  
  
    {  
  
        public string? RequestId { get; set; }  
  
  
  
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);  
  
    }  
  
}
```

Archivo: ArticulosMappers.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\MapeosDatos\ArticulosMappers.cs

using Empresa.LogicaDeNegocio.Sistema;

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;


namespace Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos
{

    public class ArticulosMappers

    {

        public static Articulo FromDto(ArticuloDTO dto)

        {

            if (dto == null) throw new ArticuloNuloException(nameof(dto));

            return new Articulo(dto.CodigoProveedor, dto.NombreArticulo, dto.Descripcion, dto.PrecioVP,
dto.Stock);

        }

        public static Articulo FromDtoUpdate(ArticuloDTO dto)

        {

            if (dto == null) throw new ArticuloNuloException(nameof(dto));
```

```

        var articulo = new Articulo(dto.CodigoProveedor, dto.NombreArticulo, dto.Descripcion,
dto.PrecioVP, dto.Stock);

        articulo.Id = dto.Id;

        return articulo;

    }

    public static ArticuloDTO ToDto(Articulo articulo)

    {

        if (articulo == null) throw new ArticuloNuloException();

        return new ArticuloDTO()

        {

            Id = articulo.Id,

            CodigoProveedor = articulo.CodigoProveedor.codigo,

            NombreArticulo = articulo.NombreArticulo.Nombre,

            Descripcion = articulo.Descripcion.Descripcion,

            PrecioVP = articulo.PrecioVP,

            Stock = articulo.Stock.StockActual()

        };

    }

    public static IEnumerable<ArticuloDTO> FromLista(IEnumerable<Articulo> articulos)

    {

        if (articulos == null)

        {

            throw new ArticuloNuloException("La lista de articulos no puede ser nula");

```

```
    }

    return articulos.Select(articulo => ToDto(articulo));

}

}

}
```

Archivo: ClientesMappers.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\MapeosDatos\ClientesMappers.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Microsoft.EntityFrameworkCore.Metadata.Internal;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using static System.Runtime.InteropServices.JavaScript.JSType;
```

```
namespace Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos
```

```
{
```

```
    public class ClientesMappers
```

```
    {
```

```
        public static Cliente FromDto(ClienteDTO dto)
```

```
        {
```

```
            if (dto == null) throw new UsuarioNuloExcepcion(nameof(dto));
```

```
            return new Cliente(dto.rut, dto.razonSocial, dto.Calle, dto.Numero, dto.Ciudad, dto.Distancia);
```

```
        }
```

```
        public static ClienteDTO ToDto(Cliente cliente)
```

```
        {
```

```
            if (cliente == null)
```

```
                throw new ClienteNuloException("El cliente no puede ser nulo.");
```

```
            return new ClienteDTO()
```

```
            {
```

```
                Id = cliente.Id,
```

```
                rut = cliente.rut.Rut,
```

```
razonSocial = cliente.razonSocial.RazonSoc,
```

```
Calle = cliente.direccion.Calle,
```

```
Numero = cliente.direccion.Numero,
```

```
Ciudad = cliente.direccion.Ciudad,
```

```
Distancia = cliente.direccion.Distancia
```

```
};
```

```
}
```

```
public static IEnumerable<ClienteDTO> FromLista(IEnumerable<Cliente> clientes)
```

```
{
```

```
    if (clientes == null)
```

```
    {
```

```
        throw new UsuarioNuloExcepcion("La lista de clientes no puede ser nula");
```

```
    }
```

```
    return clientes.Select(cliente => ToDto(cliente));
```

```
}
```

```
internal static Cliente FromDtoUpdate(ClienteDTO dto)
```

```
{
```

```
    if (dto == null)
```

```
        throw new ClienteNuloException(nameof(dto));
```

```
    var cliente = new Cliente(dto.rut, dto.razonSocial, dto.Calle, dto.Numero, dto.Ciudad,  
dto.Distancia);
```

```
        cliente.Id = dto.Id;

        return cliente;

    }

}

}
```

Archivo: LineaPedidoMappers.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\MapeosDatos\LineaPedidoMappers.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaNegocio.Entidades;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos
```

```
{
```

```
    internal class LineaPedidoMappers
```

```
    {
```

```
        private static IRepositoryArticulo _repoArticulos = new RepositorioArticuloEF();
```

```
        private static IGetArticulo _getArticulo;
```

```
        public static LineaPedido FromDTO(LineaPedidoDTO dto)
```

```
        {
```

```
            _getArticulo = new BuscarArticulo(_repoArticulos);
```

```
            if (dto == null)
```

```
            {
```

```
                throw new PedidoNuloException("LineaPedido nula");
```

```
            }
```

```
            Articulo articulo = _getArticulo.GetById(dto.idArticulo);
```

```
            return new LineaPedido(articulo, dto.Cantidad);
```

```
        }
```

```
        public static LineaPedido FromDTOUpdate(LineaPedidoDTO dto)
```

```
        {
```

```
            _getArticulo = new BuscarArticulo(_repoArticulos);
```

```
            if (dto == null)
```



```

{

    throw new PedidoNuloException("LinePedido nula");

}

Articulo articulo = _getArticulo.GetById(dto.idArticulo);

LineaPedido linea = new LineaPedido(articulo, dto.Cantidad);

linea.Id = dto.id;

return linea;

}

public static LineaPedidoDTO ToDto(LineaPedido linea)

{

    if (linea == null) throw new PedidoNuloException();

    ArticuloDTO articulo = ArticulosMappers.ToDto(linea.Articulo);

    return new LineaPedidoDTO()

    {

        id = linea.Id,

        PedidoID = linea.pedido.Id,

        idArticulo = articulo.Id,

       CodigoProveedor = articulo.CodigoProveedor,

        NombreArticulo = articulo.NombreArticulo,

        Descripcion = articulo.Descripcion,

        PrecioVP = articulo.PrecioVP,

        Stock = articulo.Stock,

        Cantidad = linea.Cantidad,

```

PrecioUnitario = articulo.PrecioVP,

Subtotal = articulo.PrecioVP*linea.Cantidad

};

}

public static IEnumerable<LineaPedidoDTO> FromLista(IEnumerable<LineaPedido> lineas)

{

if (lineas == null)

{

throw new PedidoNuloException("La lista de articulos no puede ser nula");

}

return lineas.Select(linea => ToDto(linea));

}

}

}

Archivo: PedidosMappers.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\MapeosDatos\PedidosMappers.cs

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;
```

```
using Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;
```

```
using Papeleria.LogicaNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Pedido;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos
```

```
{
```

```
    public class PedidosMappers
```

```
    {
```

```
        private static IRepositoryPedido _pedidos = new RepositorioPedidoEF();
```

```
        private static IRepositoryCliente _clientesRepo = new RepositorioClienteEF(_pedidos);
```

```
        private static IBuscarClientes _buscarClientes = new BuscarClientes(_clientesRepo);
```

```
        public static Pedido FromExpress(PedidoDTO pedido)
```

```
        {
```

```
            if (pedido == null) throw new PedidoNuloException(nameof(pedido));
```

```
List<LineaPedido> lineasFinal = new List<LineaPedido>();

foreach (LineaPedidoDTO linea in pedido.LineasPedido)

{

    LineaPedido l = LineaPedidoMappers.FromDTO(linea);

    lineasFinal.Add(l);

}

var pedidoFinal = new Express();

pedidoFinal.Id = pedido.Id;

Cliente cliente = _buscarClientes.GetById(pedido.ClienteID);

pedidoFinal.cliente = cliente;

pedidoFinal.lineas = lineasFinal;

pedidoFinal.CalcularRecargoYFijar();

pedidoFinal.FijarEntregaPrometida((pedido.FechaEntrega - pedido.FechaPedido).Days);

pedidoFinal.CalcularYFijarPrecio(new IVA(pedido.iva));

foreach (LineaPedido linea in lineasFinal)

{

    linea.AsignarPedido(pedidoFinal);

}

try

{

    pedidoFinal.esValido();

}

catch (Exception ex)
```

```

{

    throw new PedidoNuloException("Pedido invalido");

}

return pedidoFinal;

}

public static Pedido FromComunes(PedidoDTO pedido)

{

    if (pedido == null) throw new PedidoNuloException(nameof(pedido));

    List<LineaPedido> lineasFinal = new List<LineaPedido>();

    foreach(LineaPedidoDTO linea in pedido.LineasPedido)

    {

        LineaPedido l = LineaPedidoMappers.FromDTO(linea);

        lineasFinal.Add(l);

    }

    var pedidoFinal = new Comunes();

    pedidoFinal.Id = pedido.Id;

    Cliente cliente = _buscarClientes.GetById(pedido.ClienteID);

    pedidoFinal.cliente = cliente;

    pedidoFinal.lineas = lineasFinal;

    pedidoFinal.CalcularRecargoYFijar();

    pedidoFinal.FijarEntregaPrometida((pedido.FechaEntrega - pedido.FechaPedido).Days);

    pedidoFinal.CalcularYFijarPrecio(new IVA(pedido.iva));

    foreach(LineaPedido linea in lineasFinal)

```

```
{

    linea.AsignarPedido(pedidoFinal);

}

try

{

    pedidoFinal.esValido();

}

catch (Exception ex)

{

    throw new PedidoNuloException("Pedido invalido");

}

return pedidoFinal;

}

public static PedidoDTO ToExpressDto(Express pedido)

{

    if (pedido == null) throw new PedidoNuloException();

    return new PedidoDTO()

    {

        Id = pedido.Id,

        FechaPedido = pedido.fechaPedido,

        FechaEntrega = pedido.fechaPedido.AddDays(pedido.entregaPrometida),

        ClienteID = pedido.cliente.Id,

        LineasPedido = LineaPedidoMappers.FromLista(pedido.lineas).ToList(),
```

```

        recargo = pedido.recargo,

        iva = pedido.iva.valor,

        precioFinal = pedido.precioFinal,

        entregado = pedido.entregado,

        anulado = pedido.anulado

    };

}

public static PedidoDTO ToComunesDto(Comunes pedido)
{
    if (pedido == null) throw new PedidoNuloException();

    return new PedidoDTO()
    {
        Id = pedido.Id,

        FechaPedido = pedido.fechaPedido,

        FechaEntrega = pedido.fechaPedido.AddDays(pedido.entregaPrometida),

        ClienteID = pedido.cliente.Id,

        LineasPedido = LineaPedidoMappers.FromLista(pedido.lineas).ToList(),

        recargo = pedido.recargo,

        iva = pedido.iva.valor,

        precioFinal = pedido.precioFinal,

        entregado = pedido.entregado,

        anulado = pedido.anulado

    };
}

```

```
}
```

```
public static PedidoDTO ToDto(Pedido pedido)
```

```
{
```

```
    if (pedido == null) throw new PedidoNuloException();
```

```
    return new PedidoDTO()
```

```
    {
```

```
        Id = pedido.Id,
```

```
        FechaPedido = pedido.fechaPedido,
```

```
        FechaEntrega = pedido.fechaPedido.AddDays(pedido.entregaPrometida),
```

```
        ClienteID = pedido.cliente.Id,
```

```
        LineasPedido = LineaPedidoMappers.FromLista(pedido.lineas).ToList(),
```

```
        recargo = pedido.recargo,
```

```
        iva = pedido.iva.valor,
```

```
        precioFinal = pedido.precioFinal,
```

```
        entregado = pedido.entregado,
```

```
        anulado = pedido.anulado
```

```
    };
```

```
}
```

```
public static IEnumerable<PedidoDTO> FromLista(IEnumerable<Pedido> pedidos)
```

```
{
```

```
    if (pedidos == null)
```

```
    {
```



```
        throw new PedidoNuloException("La lista de pedidos no puede ser nula");
    }

    return pedidos.Select(pedido => ToDto(pedido));
}
```

```
public static IEnumerable<PedidoDTO> FromListaExpress(IEnumerable<Express> pedidos)
{
    if (pedidos == null)
    {
        throw new PedidoNuloException("La lista de pedidos no puede ser nula");
    }

    return pedidos.Select(pedido => ToExpressDto(pedido));
}
```

```
public static IEnumerable<PedidoDTO> FromListaComunes(IEnumerable<Comunes> pedidos)
{
    if (pedidos == null)
    {
        throw new PedidoNuloException("La lista de pedidos no puede ser nula");
    }

    return pedidos.Select(pedido => ToComunesDto(pedido));
}
}
```

}

Archivo: UsuariosMappers.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\MapeosDatos\UsuariosMappers.cs

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Papeleria.LogicaNegocio.Entidades;

using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

namespace Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos

{

public class UsuariosMappers

{

public static Usuario FromDto(UsuarioDTO dto) {

```

        if (dto == null) throw new UsuarioNuloExcepcion(nameof(dto));

        return new Usuario(dto.Email,dto.Nombre,dto.Apellido,dto.Contrasenia);

    }

    public static Usuario FromDtoUpdate(UsuarioDTO dto)

    {

        if (dto == null) throw new UsuarioNuloExcepcion(nameof(dto));

        var usuario = new Usuario(dto.Email, dto.Nombre, dto.Apellido, dto.Contrasenia);

        usuario.Id = dto.Id;

        return usuario;

    }

    public static UsuarioDTO ToDto(Usuario usuario) {

        if (usuario == null) throw new UsuarioNuloExcepcion();

        return new UsuarioDTO()

        {

            Id = usuario.Id,

            Email = usuario.Email.Direccion,

            Nombre = usuario.NombreCompleto.Nombre,

            Apellido = usuario.NombreCompleto.Apellido,

            Contrasenia = usuario.Contrasenia.Valor

        };

    }

    public static IEnumerable<UsuarioDTO> FromLista(IEnumerable<Usuario> usuarios) {

```

```
if (usuarios == null) {
```

```
    throw new UsuarioNuloExcepcion("La lista de usuarios no puede ser nula");
```

```
}
```

```
return usuarios.Select(usuario => ToDto(usuario));
```

```
}
```

[//https://vimeopro.com/universidadortfi/fi-5212-programacion-3-cabella-69235-p3-m3a-remoto/video/929607409](https://vimeopro.com/universidadortfi/fi-5212-programacion-3-cabella-69235-p3-m3a-remoto/video/929607409)

//1:36:07

```
}
```

```
}
```

Archivo: AltaArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Articulos\AltaArticulo.cs

```
using Empresa.LogicaDeNegocio.Sistema;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo;
```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using System;
```

```
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Email;


namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos

{

    public class AltaArticulo : IAltaArticulo

    {

        private IRepositoryArticulo _repoArticulos;


        public AltaArticulo(IRepositoryArticulo repo)

        {

            _repoArticulos = repo;

        }


        public void Ejecutar(ArticuloDTO dto)

        {

            if (dto == null)

                throw new ArticuloNuloException("Nulo");

        }

    }

}
```

```

bool nombreExiste = _repoArticulos.ExisteArticuloConNombre(dto.NombreArticulo);

if (nombreExiste)

{

    throw new ArticuloNoValidoException("El nombre del articulo ya está en uso.");

}

else

{

    Articulo articulo = ArticulosMappers.FromDto(dto);

    _repoArticulos.Add(articulo);

}

}

}

```

Archivo: BorrarArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Articulos\BorrarArticulo.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos

{

    public class BorrarArticulo : IBorrarArticulo

    {

        private IRepositoryArticulo _repoArticulos;


        public BorrarArticulo(IRepositoryArticulo repo)

        {

            _repoArticulos = repo;

        }

        public void Ejecutar(int id)

        {

            var articulo = _repoArticulos.GetById(id);

            if(articulo == null)
```

```
{

    throw new ArticuloNuloException("Articulo no puede ser nulo");

}

try

{

    _repoArticulos.Remove(articulo);

} catch (Exception ex)

{

    throw new ArticuloNoValidoException(ex.Message);

}

}

public void Ejecutar(Articulo articulo)

{

    if (articulo == null)

    {

        throw new ArticuloNuloException("Articulo no puede ser nulo");

    }

    try

    {

        _repoArticulos.Remove(articulo);

    }

    catch (Exception ex)

    {
```



```
        throw new ArticuloNoValidoException(ex.Message);

    }

}

}

}
```

Archivo: BuscarArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Articulos\BuscarArticulo.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos
```

```
{  
  
    public class BuscarArticulo : IGetArticulo  
  
    {  
  
        private IRepositoryArticulo _repoArticulos;  
  
        public BuscarArticulo(IRepositoryArticulo repo)  
  
        {  
  
            _repoArticulos = repo;  
  
        }  
  
        public ArticuloDTO GetByIdDTO(int id)  
  
        {  
  
            var articulo = _repoArticulos.GetById(id);  
  
            if (articulo == null)  
  
            {  
  
                throw new ArticuloNuloException("Articulo no encontrado con el ID especificado");  
  
            }  
  
            var articuloReturn = ArticulosMappers.ToDto(articulo);  
  
            return articuloReturn;  
  
        }  
  
        public Articulo GetById(int id)
```

```
{  
  
    return _repoArticulos.GetById(id);  
  
}
```

```
public ArticuloDTO GetArticuloPorCodigo(CodigoProveedorArticulos codigoProveedor)  
  
{  
  
    var articulo = _repoArticulos.GetArticuloByCodigo(codigoProveedor);  
  
    if(articulo == null)  
  
    {  
  
        throw new ArticuloNuloException("Articulo no encontrado con el codigo especificado");  
  
    }  
  
    var articuloReturn = ArticulosMappers.ToDto(articulo);  
  
    return articuloReturn;  
  
}
```

```
public IEnumerable<ArticuloDTO> GetArticulosPorNombre(string nombre)  
  
{  
  
    throw new NotImplementedException();  
  
}  
  
}  
  
}
```

Archivo: GetAllArticulos.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Articulos\GetAllArticulos.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos
```

```
{
```

```
    public class GetAllArticulos : IGetAllArticulos
```

```
    {
```

```
        private IRepositoryArticulo _repoArticulos;
```

```
        public GetAllArticulos(IRepositoryArticulo repo)
```

```
        {
```

```
            _repoArticulos = repo;
```

```
}
```

```
public IEnumerable<ArticuloDTO> Ejecutar()
```

```
{
```

```
    var articulos = _repoArticulos.GetAll();
```

```
    if (articulos == null || articulos.Count() == 0)
```

```
    {
```

```
        throw new ArticuloNuloException("No se encontraron articulos en el sistema");
```

```
    }
```

```
    return ArticulosMappers.FromLista(articulos);
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: UpdateArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Articulos\UpdateArticulo.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Articulos
```

```
{
```

```
    public class UpdateArticulo : IUpdateArticulo
```

```
    {
```

```
        private IRepositoryArticulo _repoArticulos;
```

```
        public UpdateArticulo(IRepositoryArticulo repo)
```

```
        {
```

```
            _repoArticulos = repo;
```

```
        }
```

```
        public void Ejecutar(int id, ArticuloDTO articuloMod)
```

```
        {
```

```
            if (articuloMod == null) { throw new ArticuloNuloException("Articulo modificado no puede ser nulo"); }
```

```
            try
```

```
            {
```

```

        var articulo = ArticulosMappers.FromDtoUpdate(articuloMod);

        _repoArticulos.Update(id, articulo);

    }

    catch (Exception ex) { throw new ArticuloNoValidoException(ex.Message); }

}

}

}

```

Archivo: AltaClientes.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Clientes\AltaClientes.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes
```

```
{
```

```
    public class AltaClientes : IAltaCliente
```

```
    {
```

```
        private IRepositoryCliente _repoClientes;
```

```
        public AltaClientes(IRepositoryCliente repo)
```

```
        {
```

```
            _repoClientes = repo;
```

```
        }
```

```
        public void Ejecutar(ClienteDTO dto)
```

```
        {
```

```
            if (dto == null)
```

```
            {
                throw new ClienteNuloException("Nulo");
            }

```

```
            Cliente cliente= ClientesMappers.FromDto(dto);
```

```
            _repoClientes.Add(cliente);
```



```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: BorrarCliente.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Clientes\BorrarCliente.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente;
```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes
```

```
{
```

```
    public class BorrarCliente : IBorrarCliente
```

```
{

private IRepositoryCliente _repoCliente;


public BorrarCliente(IRepositoryCliente repo)

{

    _repoCliente = repo;

}


public void Ejecutar(int id, ClienteDTO clienteBorrar)

{

    if (clienteBorrar == null)

        throw new ClienteNuloException("Usuario no puede ser nulo.");

    try

    {

        var usuario = _repoCliente.GetById(clienteBorrar.Id);

        _repoCliente.Remove(usuario);

    }

    catch (Exception ex)

    {

        throw new UsuarioNoValidoExcepcion(ex.Message);

    }

}

}
```

}

Archivo: BuscarClientes.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Clientes\BuscarClientes.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using Papeleria.AccesoDatos.EF;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Empresa.LogicaDeNegocio.Entidades;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes
```

```
{
```

```
    public class BuscarClientes : IBuscarClientes
```

```
    {
```

```
        private IRepositoryioCliente _repoClientes;
```

```
        public BuscarClientes(IRepositoryioCliente repo)
```

```
        {
```

```
            _repoClientes = repo;
```

```
        }
```

```
        public IEnumerable<ClienteDTO> GetAll()
```

```
        {
```

```
            var clientesOrigen = _repoClientes.GetAll();
```

```
            if (clientesOrigen == null || clientesOrigen.Count() == 0)
```

```
            {
```

```
                throw new ClienteNuloException("No hay clientes registrados");
```

```
            }
```

```
            return ClientesMappers.FromLista(clientesOrigen);
```

```
        }
```

```
        public ClienteDTO GetByIdDTO(int id)
```

```
        {
```

```
            var cliente = _repoClientes.GetById(id);
```

```
        if (cliente == null)

        {

            throw new ClienteNuloException("Articulo no encontrado con el ID especificado");

        }

        var clienteReturn = ClientesMappers.ToDto(cliente);

        return clienteReturn;

    }

    public Cliente GetById(int id)

    {

        return _repoClientes.GetById(id);

    }


    public IEnumerable<ClienteDTO> GetXMontoSuperado(double monto)

    {

        var clientesOrigen = _repoClientes.GetClientesPedidoSupereMonto(monto);

        if (clientesOrigen == null || clientesOrigen.Count() == 0)

        {

            throw new ClienteNuloException("No hay clientes registrados que hayan superado ese monto.");

        }

        return ClientesMappers.FromLista(clientesOrigen);

    }


    public ClienteDTO GetXRazonSocial(string razon)
```

```

{

    var clientesOrigen = _repoClientes.GetClientePorRazon(razon);

    if (clientesOrigen == null)

    {

        throw new ClienteNuloException("No hay clientes registrados con esa razon social.");

    }

    return ClientesMappers.ToDto(clientesOrigen);

}

}

}

```

Archivo: ModificarCliente.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Clientes\ModificarCliente.cs

```

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Clientes
```

```
{
```

```
    public class ModificarCliente : IModificarCliente
```

```
    {
```

```
        private IRepositoryCliente _repoClientes;
```

```
        public ModificarCliente(IRepositoryCliente repo)
```

```
        {
```

```
            _repoClientes = repo;
```

```
        }
```

```
        public void Ejecutar(int id, ClienteDTO clienteModificado)
```

```
        {
```

```
            if (clienteModificado == null)
```

```
            {
                throw new ClienteNoValidoException("Cliente no puede ser nulo.");
            }

```

```
            try
```

```
            {
```

```
                var cliente = ClientesMappers.FromDtoUpdate(clienteModificado);
```

```
        _repoClientes.Update(id, cliente);

    }

    catch (Exception ex)

    {

        throw new UsuarioNoValidoExcepcion(ex.Message);

    }

}

}

}
```

Archivo: AltaPedidos.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Pedidos\AltaPedidos.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos;

using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;
```



```
using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Pedidos

{

    public class AltaPedidos : IAltaPedido

    {

        private IRepositoryPedido _repoPedidos;


        public AltaPedidos(IRepositoryPedido _repositorioPedidos)

        {

            _repoPedidos = _repositorioPedidos;

        }


        public void EjecutarExpress(PedidoDTO dto)

        {

            if (dto == null) {

                throw new PedidoNoValidoException("");

            }

            Pedido pedido = PedidosMappers.FromExpress(dto);

            _repoPedidos.Add(pedido);

        }

    }

}
```

```
}
```

```
public void EjecutarComunes(PedidoDTO dto)
```

```
{
```

```
    if (dto == null)
```

```
    {
```

```
        throw new PedidoNoValidoException("");
```

```
    }
```

```
    Pedido pedido = PedidosMappers.FromComunes(dto);
```

```
    _repoPedidos.Add(pedido);
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: AnularPedido.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Pedidos\AnularPedido.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Pedidos
{

    public class AnularPedido : IAnularPedido

    {

        private IRepositoryPedido _repoPedidos;


        public AnularPedido(IRepositoryPedido _repositorioPedidos)

        {

            _repoPedidos = _repositorioPedidos;

        }

        public void Ejecutar(int id)

        {

            _repoPedidos.Anular(id);

        }

    }

}
```

Archivo: GetAllPedidos.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Pedidos\GetAllPedidos.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos;

using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Pedidos
{
    public class GetAllPedidos : IGetAllPedidos
    {
        private IRepositoryPedido _repositorioPedido;

        public GetAllPedidos(IRepositoryPedido repo)
        {
            _repositorioPedido = repo;
        }
    }
}
```

```
public IEnumerable<PedidoDTO> Ejecutar()

{

    var pedidosOrigen = _repositorioPedido.GetAll();

    return PedidosMappers.FromLista(pedidosOrigen);

}

}

}
```

Archivo: GetPedidos.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Pedidos\GetPedidos.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos;

using Papeleria.LogicaNegocio.Excepciones.Cliente;

using Papeleria.LogicaNegocio.Excepciones.Pedido;

using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Pedidos
```

```
{  
  
    public class GetPedidos : IGetPedido  
  
    {  
  
        private static IRepositoryPedido _repoPedidos = new RepositorioPedidoEF();  
  
        private static IRepositoryCliente _clientesRepo = new RepositorioClienteEF(_repoPedidos);  
  
        public GetPedidos(IRepositoryPedido repo)  
  
        {  
  
            _repoPedidos = repo;  
  
        }  
  
        public Pedido GetById(int id)  
  
        {  
  
            return _repoPedidos.GetById(id);  
  
        }  
  
        public PedidoDTO GetByIdDTO(int id)  
  
        {
```

```

var pedido = _repoPedidos.GetById(id);

if (pedido == null)

{

    throw new PedidoNuloException("Pedido no encontrado con el ID especificado");

}

var pedidoReturn = PedidosMappers.ToDto(pedido);

return pedidoReturn;

}

```

```

public IEnumerable<PedidoDTO> GetPedidosPorFecha(DateTime date)

{

    var pedidos = _repoPedidos.GetPedidosPorFecha(date);

    if (pedidos == null)

    {

        throw new PedidoNuloException("No se encontraron pedidos desde la fecha especificada");

    }

    var pedidoReturn = PedidosMappers.FromLista(pedidos);

    return pedidoReturn;

}

}

}

```

Archivo: AltaUsuarios.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Usuarios\AltaUsuarios.cs

```
using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.AccesoDatos.EF;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.Interaces;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepcions.Email;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios

{

    public class AltaUsuarios : IAltaUsuario

    {

        private IRepositoryUsuario _repoUsuarios;
```



```
public AltaUsuarios(IRepositorioUsuario repo)
```

```
{
```

```
    _repoUsuarios = repo;
```

```
}
```

```
public void Ejecutar(UsuarioDTO dto)
```

```
{
```

```
    if (dto == null)
```

```
        throw new UsuarioNuloExcepcion("No han llegado datos.");
```

```
    bool emailExistente = _repoUsuarios.ExisteUsuarioConEmail(dto.Email);
```

```
    if (emailExistente)
```

```
    {
```

```
        throw new EmailNoValidoException("El email ya está en uso.");
```

```
    }
```

```
    else
```

```
    {
```

```
        Usuario usuario = UsuariosMappers.FromDto(dto);
```

```
        _repoUsuarios.Add(usuario);
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: BorrarUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Usuarios\BorrarUsuario.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;
```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios
```

```
{
```

```
    public class BorrarUsuario : IBorrarUsuario
```

```
    {
```

```
        private IRepositoryUsuario _repoUsuarios;
```

```
public BorrarUsuario(IRepositorioUsuario repo)

{

    _repoUsuarios = repo;

}


public void Ejecutar(int id, UsuarioDTO usuarioBorrar)

{

    if (usuarioBorrar == null)

        throw new UsuarioNuloExcepcion("Usuario no puede ser nulo.");

    try

    {

        var usuario = _repoUsuarios.GetById(usuarioBorrar.Id);

        _repoUsuarios.Remove(usuario);

    }

    catch (Exception ex)

    {

        throw new UsuarioNoValidoExcepcion(ex.Message);

    }

}

}
```

Archivo: BuscarUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Usuarios\BuscarUsuario.cs

```
using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;

using Papeleria.LogicaAplicacion.Interaces;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios

{

    public class BuscarUsuario : IGetUsuario

    {

        private IRepositoryUsuario _repoUsuarios;

        public BuscarUsuario(IRepositoryUsuario repo)
```

```
{

    _repoUsuarios = repo;

}

public UsuarioDTO GetByIdDTO(int id)

{

    var usu = _repoUsuarios.GetById(id);

    if (usu == null)

    {

        throw new UsuarioNuloExcepcion("No hay usuario con ese id");

    }

    var usuDto = UsuariosMappers.ToDto(usu);

    return usuDto;

}

public Usuario GetById(int id)

{

    return _repoUsuarios.GetById(id);

}

}
```

Archivo: GetAllUsuarios.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Usuarios\GetAllUsuarios.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.MapeosDatos;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;
```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario;
```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios
```

```
{
```

```
    public class GetAllUsuarios : IGetAllUsuarios
```

```
    {
```

```
        private IRepositoryUsuario _repositorioUsuarios;
```

```
        public GetAllUsuarios(IRepositoryUsuario repo)
```

```
        {
```

```
            _repositorioUsuarios = repo;
```

```

    }

    public IEnumerable<UsuarioDTO> Ejecutar()

    {

        var usuariosOrigen = _repositorioUsuarios.GetAll();

        if (usuariosOrigen == null || usuariosOrigen.Count() == 0)

        {

            throw new UsuarioNuloExcepcion("No hay autores registrados");

        }

        return UsuariosMappers.FromLista(usuariosOrigen);

    }

}

}

```

Archivo: ModificarUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\ImplementacionCasosUso\Usuarios\ModificarUsuario.cs

```

using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaAplicacion.DataTransferObjects.MapecosDatos;

using Papeleria.LogicaAplicacion.Interfaces;

using Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios;

using Papeleria.LogicaNegocio.Excepciones.Usuario;

```

```
using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.ImplementacionCasosUso.Usuarios
{

    public class ModificarUsuario:IModificarUsuario

    {

        private IRepositoryUsuario _repoUsuarios;


        public ModificarUsuario(IRepositoryUsuario repo)

        {

            _repoUsuarios = repo;

        }


        public void Ejecutar(int id, UsuarioDTO usuarioModificado)

        {

            if (usuarioModificado == null)

                throw new UsuarioNuloExcepcion("Usuario no puede ser nulo.");

            try
```



```

    {

        var usuario = UsuariosMappers.FromDtoUpdate(usuarioModificado);

        _repoUsuarios.Update(id, usuario);

    }

    catch (Exception ex)

    {

        throw new UsuarioNoValidoExcepcion(ex.Message);

    }

}

}

}

```

Archivo: IAltaArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Articulos\IAltaArticulo.cs

```

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos
```

```
{  
  
    public interface IAltaArticulo  
  
    {  
  
        void Ejecutar(ArticuloDTO dto);  
  
    }  
  
}
```

```
*****
```

Archivo: IBorrarArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Articulos\IBorrarArticulo.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;  
  
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;  
  
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos
```

```
{  
  
    public interface IBorrarArticulo  
  
    {  
  
        void Ejecutar(int id);  
  
        void Ejecutar(Articulo articulo);  
  
    }  
  
}
```

```
*****
```

Archivo: IFiltrarArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Articulos\IFiltrarArticulo.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;  
  
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos
```

```
{
```

```
public interface IFiltrarArticulo

{

    IEnumerable<ArticuloDTO> GetArticuloPorCodigo(long codigoProveedor);

    IEnumerable<ArticuloDTO> GetArticulosPorNombre(string nombre);

}

}
```

Archivo: IGetAllArticulos.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Articulos\IGetAllArticulos.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos

```
{

    public interface IGetAllArticulos
```

```

{

    public IEnumerable<ArticuloDTO> Ejecutar();

}

}

```

Archivo: IGetArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Articulos\IGetArticulo.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos

```

{

    public interface IGetArticulo

    {

```

```

        ArticuloDTO GetByIdDTO(int id);

        Articulo GetById(int id);

        ArticuloDTO GetArticuloPorCodigo(CodigoProveedorArticulos codigoProveedor);

        IEnumerable<ArticuloDTO> GetArticulosPorNombre(string nombre);

    }

}

```

Archivo: IUpdateArticulo.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Articulos\IUpdateArticulo.cs

```

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Articulos

{

    public interface IUpdateArticulo

    {

```

```
        void Ejecutar(int id, ArticuloDTO articuloModificado);

    }

}
```

Archivo: IAltaCliente.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Clientes\IAltaCliente.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes
```

```
{

    public interface IAltaCliente

    {

        void Ejecutar(ClienteDTO dto);

    }

}
```

Archivo: IBorrarCliente.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Clientes\IBorrarCliente.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes
```

```
{

    public interface IBorrarCliente

    {

        void Ejecutar(int id, ClienteDTO cliente);

    }

}
```

Archivo: IBuscarClientes.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Clientes\IBuscarClientes.cs

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes

{

public interface IBuscarClientes

{

public IEnumerable<ClienteDTO> GetAll();

public ClienteDTO GetByIdDTO(int id);

public Cliente GetById(int id);

public ClienteDTO GetXRazonSocial(string razon);

public IEnumerable<ClienteDTO> GetXMontoSuperado(double monto);

}

}

Archivo: IModificarCliente.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Clientes\IModificarCliente.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Clientes
```

```
{

    public interface IModificarCliente

    {

        void Ejecutar(int id, ClienteDTO clienteModificado);

    }

}
```

Archivo: IAltaPedido.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Pedidos\IAltaPedido.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos
```

```
{
```

```
    public interface IAltaPedido
```

```
    {
```

```
        void EjecutarExpress(PedidoDTO dto);
```

```
        void EjecutarComunes(PedidoDTO dto);
```

```
    }
```

```
}
```

Archivo: IAnularPedido.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Pedidos\IAnularPedido.cs

```
using Microsoft.EntityFrameworkCore.Metadata;
```

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos

{

    public interface IAnularPedido

    {

        public void Ejecutar(int id);

    }

}
```

Archivo: IGetAllPedidos.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Pedidos\IGetAllPedidos.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos
```

```
{  
  
    public interface IGetAllPedidos  
  
    {  
  
        public IEnumerable<PedidoDTO> Ejecutar();  
  
    }  
}
```

```
*****
```

Archivo: IGetPedido.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Pedidos\IGetPedido.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;  
  
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;  
  
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos;  
  
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Pedidos

{

    public interface IGetPedido

    {

        PedidoDTO GetByIdDTO(int id);

        Pedido GetById(int id);

        IEnumerable<PedidoDTO> GetPedidosPorFecha(DateTime date);

    }

}
```

Archivo: IAltaUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Usuarios\IAltaUsuario.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios
```

```
{  
  
    public interface IAltaUsuario  
  
    {  
  
        void Ejecutar(UsuarioDTO dto);  
  
    }  
  
}
```

```
*****
```

Archivo: IBorrarUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Usuarios\IBorrarUsuario.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios
```

```
{  
  
    public interface IBorrarUsuario  
  
    {
```

```
        void Ejecutar(int id, UsuarioDTO usu);

    }

}
```

Archivo: IFiltrarUsuarios.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Usuarios\IFiltrarUsuarios.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios
```

```
{

    public interface IFiltrarUsuarios

    {

        IEnumerable<UsuarioDTO> GetUsuarioPorEmail(string email);

    }

}
```

Archivo: IGetAllUsuarios.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Usuarios\IGetAllUsuarios.cs

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios
```

```
{
```

```
    public interface IGetAllUsuarios
```

```
    {
```

```
        public IEnumerable<UsuarioDTO> Ejecutar();
```

```
    }
```

```
}
```

Archivo: IGetUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Usuarios\IGetUsuario.cs

```
using Empresa.LogicaDeNegocio.Sistema;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios
```

```
{

    public interface IGetUsuario

    {

        UsuarioDTO GetByIdDTO(int id);

        Usuario GetById(int id);

    }

}
```

```
*****
```

Archivo: IModificarUsuario.cs

Localizacion: Papeleria.LogicaAplicacion\InterfacesCasosUso\Usuarios\IModificarUsuario.cs

```
*****
```

```
using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios;

using System;
```

```
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.InterfacesCasosUso.Usuarios
{

    public interface IModificarUsuario
    {

        void Ejecutar(int id, UsuarioDTO usuarioModificado);

    }

}
```

Archivo: ArticuloDuplicadoException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Articulo\ArticuloDuplicadoException.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo
{
    internal class ArticuloDuplicadoException
    {
    }
}
```

Archivo: ArticuloNoValidoException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Articulo\ArticuloNoValidoException.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo
{
    public class ArticuloNoValidoException : Exception
    {
        public ArticuloNoValidoException()
```

```
{  
  
}
```

```
public ArticuloNoValidoException(string? message) : base(message)
```

```
{  
  
}
```

```
public ArticuloNoValidoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{  
  
}
```

```
protected ArticuloNoValidoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{  
  
}
```

```
}
```

```
}
```

```
*****
```

Archivo: ArticuloNuloException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Articulo\ArticuloNuloException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo
```

```
{
```

```
    public class ArticuloNuloException : Exception
```

```
    {
```

```
        public ArticuloNuloException()
```

```
        {
```

```
        }
```

```
        public ArticuloNuloException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public ArticuloNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected ArticuloNuloException(SerializationInfo info, StreamingContext context) : base(info,
context)

    {

    }

}

}
```

Archivo: ClienteDuplicadoException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteDuplicadoException.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Cliente

{

    public class ClienteDuplicadoException : Exception

    {

        public ClienteDuplicadoException()

        {


```

```
}
```

```
public ClienteDuplicadoException(string? message) : base(message)
```

```
{
```

```
}
```

```
public ClienteDuplicadoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected ClienteDuplicadoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: ClienteNoValidoException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteNoValidoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```



```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente
```

```
{
```

```
    public class ClienteNoValidoException : Exception
```

```
    {
```

```
        public ClienteNoValidoException()
```

```
        {
```

```
        }
```

```
        public ClienteNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public ClienteNoValidoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected ClienteNoValidoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{  
  
}  
  
}  
  
}
```

Archivo: ClienteNuloException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteNuloException.cs

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Runtime.Serialization;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.LogicaNegocio.Excepciones.Cliente  
{  
  
    public class ClienteNuloException : Exception  
    {  
  
        public ClienteNuloException()  
  
        {  
  
        }  
  
    }  
  
}
```

```
public ClienteNuloException(string? message) : base(message)
```

```
{
```

```
}
```

```
public ClienteNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected ClienteNuloException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: PedidoDuplicadoException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Pedido\PedidoDuplicadoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Pedido
```

```
{
```

```
    public class PedidoDuplicadoException : Exception
```

```
    {
```

```
        public PedidoDuplicadoException()
```

```
        {
```

```
        }
```

```
        public PedidoDuplicadoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public PedidoDuplicadoException(string? message, Exception? innerException) : base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected PedidoDuplicadoException(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
        {
```

```
    }  
  
    }  
  
}
```

Archivo: PedidoNoValidoException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Pedido\PedidoNoValidoException.cs

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Runtime.Serialization;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.LogicaNegocio.Excepciones.Pedido  
{  
  
    public class PedidoNoValidoException : Exception  
    {  
  
        public PedidoNoValidoException()  
  
        {  
  
        }  
  
    }  
  
}
```

```
public PedidoNoValidoException(string? message) : base(message)
```

```
{
```

```
}
```

```
public PedidoNoValidoException(string? message, Exception? innerException) : base(message, innerException)
```

```
{
```

```
}
```

```
protected PedidoNoValidoException(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: PedidoNuloException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Pedido\PedidoNuloException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Pedido
```

```
{
```

```
    public class PedidoNuloException : Exception
```

```
    {
```

```
        public PedidoNuloException()
```

```
        {
```

```
        }
```

```
        public PedidoNuloException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public PedidoNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected PedidoNuloException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
        {
```

```
        }
```

```
}  
  
}
```

```
*****
```

Archivo: UsuarioDuplicadoExcepcion.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioDuplicadoExcepcion.cs

```
*****
```

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Runtime.Serialization;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.LogicaNegocio.Excepciones.Usuario  
{  
  
    public class UsuarioDuplicadoExcepcion : Exception  
  
    {  
  
        public UsuarioDuplicadoExcepcion()  
  
        {  
  
        }  
  
  
  
  
        public UsuarioDuplicadoExcepcion(string? message) : base(message)
```



```
{  
  
}
```

```
public UsuarioDuplicadoExcepcion(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{  
  
}
```

```
protected UsuarioDuplicadoExcepcion(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{  
  
}
```

```
}
```

```
}
```

```
*****
```

Archivo: UsuarioNoValidoExcepcion.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioNoValidoExcepcion.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario
```

```
{
```

```
    public class UsuarioNoValidoExcepcion : Exception
```

```
    {
```

```
        public UsuarioNoValidoExcepcion()
```

```
        {
```

```
        }
```

```
        public UsuarioNoValidoExcepcion(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public UsuarioNoValidoExcepcion(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected UsuarioNoValidoExcepcion(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
*****
```

Archivo: UsuarioNuloExcepcion.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioNuloExcepcion.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario
```

```
{
```

```
    public class UsuarioNuloExcepcion : Exception
```

```
    {
```

```
        public UsuarioNuloExcepcion()
```

```
        {
```

```
        }
```

```
        public UsuarioNuloExcepcion(string? message) : base(message)
```

```
    {
```

```
}
```

```
public UsuarioNuloExcepcion(string? message, Exception? innerException) : base(message, innerException)
```

```
{
```

```
}
```

```
protected UsuarioNuloExcepcion(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: LineaPedidoModel.cs

Localizacion: Papeleria.MVC\Models\PedidosModels\LineaPedidoModel.cs

```
*****
```

```
using System.ComponentModel.DataAnnotations;
```

```
namespace Papeleria.MVC.Models.PedidosModels
```

```
{
```

```
public class LineaPedidoModel
```

```
{
```

```
[Required(ErrorMessage = "El artículo es obligatorio")]
```

```
public int ArtículoId { get; set; }
```

```
[Required(ErrorMessage = "La cantidad es obligatoria")]
```

```
[Range(1, int.MaxValue, ErrorMessage = "La cantidad debe ser mayor que cero")]
```

```
public int Cantidad { get; set; }
```

```
}
```

```
}
```

```
*****
```

Archivo: PedidoAltaModel.cs

Localizacion: Papeleria.MVC\Models\PedidosModels\PedidoAltaModel.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;
```

```
using Papeleria.LogicaNegocio.Entidades;
```

```
using Microsoft.AspNetCore.Mvc.Rendering;
```

```
using System.ComponentModel.DataAnnotations;
```

```
namespace Papeleria.MVC.Models.PedidosModels
```

```
{
```

```
    public class PedidoAltaModel
```

```
{
```

```
[Required(ErrorMessage = "El cliente es obligatorio")]
```

```
public int ClienteId { get; set; }
```

```
public List<LineaPedidoModel> LineasPedido { get; set; }
```

```
[Required(ErrorMessage = "La fecha de entrega es obligatoria")]
```

```
[DataType(DataType.Date)]
```

```
public DateTime FechaEntrega { get; set; }
```

```
public int ArticuloId { get; set; }
```

```
public int Cantidad { get; set; }
```

```
public double iva { get; set; }
```

```
public double PrecioUnitario { get; set; }
```

```
public double Subtotal { get; set; }
```

```
}
```

```
}
```

```
*****
```

Archivo: PedidoListModel.cs

Localizacion: Papeleria.MVC\Models\PedidosModels\PedidoListModel.cs

```
*****
```

```
using System.ComponentModel.DataAnnotations;
```

```
namespace Papeleria.MVC.Models.PedidosModels
```

```

{

public class PedidoListModel

{

    public int Id { get; set; }

    [Required(ErrorMessage = "El cliente es obligatorio")]

    public int ClienteId { get; set; }


    public List<LineaPedidoModel> LineasPedido { get; set; }


    [DataType(DataType.Date)]

    public DateTime FechaPedido { get; set; }


    [Required(ErrorMessage = "La fecha de entrega es obligatoria")]

    [DataType(DataType.Date)]

    public DateTime FechaEntrega { get; set; }

    public double IVA { get; set; }

    public double PrecioFinal { get; set; }


    }

}

```

Archivo: ArticuloDTO.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\Dtos\Articulos\ArticuloDTO.cs

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos

{

public class ArticuloDTO

{

public int Id { get; set; }

public longCodigoProveedor { get; set; }

public string NombreArticulo { get; set; }

public string Descripcion { get; set; }

public double PrecioVP { get; set; }

public int Stock { get; set; }

}

}

Archivo: ClienteDTO.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\Dtos\Cientes\ClienteDTO.cs

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Entidades.ValueObjects.Cientes;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.ComponentModel.DataAnnotations;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Cientes
```

```
{
```

```
    public class ClienteDTO
```

```
    {
```

```
        public int Id { get; set; }
```

```
        public long rut { get; set; }
```

```
        public string razonSocial { get; set; }
```

```
        public string Calle { get; init; }
```

```
public int Numero { get; init; }
```

```
public string Ciudad { get; init; }
```

```
public int Distancia { get; init; }
```

```
}
```

```
}
```

```
*****
```

Archivo: LineaPedidoDTO.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\Dtos\Pedidos\LineaPedidoDTO.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos
```

```
{
```

```
    public class LineaPedidoDTO
```

```
{
```

```

public int id { get; set; }

public int PedidoID { get; set; }

public int idArticulo { get; set; }

public long CodigoProveedor { get; set; }

public string NombreArticulo { get; set; }

public string Descripcion { get; set; }

public double PrecioVP { get; set; }

public int Stock { get; set; }

public int Cantidad { get; set; }

public double PrecioUnitario { get; set; }

public double Subtotal { get; set; }

}

}

```

Archivo: PedidoDTO.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\Dtos\Pedidos\PedidoDTO.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes;

using Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos;

using Papeleria.LogicaNegocio.Entidades;

using System;

```

```
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos
{
    public class PedidoDTO
    {
        public int Id { get; set; }

        public DateTime FechaPedido { get; set; }

        public DateTime FechaEntrega { get; set; }

        public int ClienteID { get; set; }

        public int PedidoID { get; set; }

        public List<LineaPedidoDTO> LineasPedido { get; set; }

        public double recargo { get; set; }

        public double iva { get; set; }

        public double precioFinal { get; set; }

        public bool entregado { get; set; }

        public bool anulado { get; set; }
    }
}
```

Archivo: UsuarioDTO.cs

Localizacion: Papeleria.LogicaAplicacion\DataTransferObjects\Dtos\Usuarios\UsuarioDTO.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios
```

```
{
```

```
    public class UsuarioDTO
```

```
    {
```

```
        public int Id { get; set; }
```

```
        public string Email { get; set; }
```

```
        public string Nombre { get; set; }
```

```
        public string Apellido { get; set; }
```

```
        public string Contrasenia { get; set; }
```

```
    }
```

```
}
```

Archivo:CodigoProveedorArticulos.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Articulos\CodigoProveedorArticulos.cs

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.CodigoProveedor;
```

```
using Papeleria.LogicaNegocio.InterfacesEntidades;
```

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

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos
```

```
{
```

```
    [ComplexType]
```

```
    public record CodigoProveedorArticulos : IValidable<CodigoProveedorArticulos>
```

```
    {
```

```
        public long codigo { get; init; }
```

```
        public CodigoProveedorArticulos()
```

```
        {
```

```
        }
```

```
        public CodigoProveedorArticulos(long codigo)
```

```
        {
```

```
            if (codigo == null) {
```

```
                throw new CodigoProveedorNuloException("El codigo no puede ser nulo");
```

```

    }

    this.codigo = codigo;

    esValido();

}

public void esValido()

{

    if (codigo.ToString().Length != 13) {

        throw newCodigoProveedorNoValidoException("No puede tener un largo distinto de 13
digitos.");

    }

}

}

}

```

Archivo: DescripcionArticulo.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Articulos\DescripcionArticulo.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.DescripcionArticulo;

using Papeleria.LogicaNegocio.InterfacesEntidades;

```

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

```
using System.Drawing;
```

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos
```

```
{
```

```
    [ComplexType]
```

```
    public record DescripcionArticulo : IValidable<DescripcionArticulo>
```

```
    {
```

```
        public string Descripcion { get; init; }
```

```
        public DescripcionArticulo()
```

```
        {
```

```
        }
```

```
        public DescripcionArticulo(string descripcion)
```

```
        {
```

```
            Descripcion = descripcion;
```

```
            esValido();
```

```
        }
```

```
        public void esValido()
```

```
        {
```



```

        if (Descripcion.Length < 5) {

            throw new DescripcionArticuloNoValidoException("La descripcion no puede ser menor a 5
caracteres.");

        }

    }

}

}

```

Archivo: NombreArticulo.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Articulos\NombreArticulo.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Excepciones.Articulo;

using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.NombreArticulo;

using Papeleria.LogicaNegocio.InterfacesEntidades;

using System.ComponentModel.DataAnnotations.Schema;

```

namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos

```

{

    [ComplexType]

    public record NombreArticulo : IValidable<NombreArticulo>, IEquatable<NombreArticulo>

```

```
{

    public string Nombre { get; init; }

}

public NombreArticulo(string nombre)

{

    Nombre = nombre;

    esValido();

}

public NombreArticulo()

{

}

public void esValido()

{

    if (Nombre == null || Nombre.Length<1) {

        throw new NombreArticuloNuloException("El nombre del articulo no puede ser nulo o vacio.");

    }

}

}
```

Archivo: StockArticulo.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Articulos\StockArticulo.cs

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.StockArticulo;

using Papeleria.LogicaNegocio.InterfacesEntidades;

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

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Articulos
```

```
{
```

```
    [ComplexType]
```

```
    public record StockArticulo:IValidable<StockArticulo>
```

```
    {
```

```
        public int cantidad { get; set; }
```

```
        public StockArticulo(int cantidad)
```

```
        {
```

```
            this.cantidad = cantidad;
```

```
            esValido();
```

```
        }
```

```
public StockArticulo()

{

}

public int StockActual()

{

    return cantidad;

}


public void RestarStock(int ctd)

{

    if (ctd > cantidad)

    {

        throw new StockArticuloNoValidoException("La cantidad a restar no puede ser mayor al stock actual.");

    }

    else if (ctd < 1)

    {

        throw new StockArticuloNoValidoException("La cantidad a restar no puede ser menor a 1.");

    }

    else if (ctd == null)

    {

        throw new StockArticuloNuloException("El stock a restar no puede ser nulo");

    }

}
```

```
}

else

{

    cantidad -= ctd;

}

}

public void SumarStock(int ctd)

{

    if (ctd < 1)

    {

        throw new StockArticuloNoValidoException("La cantidad a sumar no puede ser menor a 1.");

    }

    else if (ctd == null)

    {

        throw new StockArticuloNuloException("El stock a sumar no puede ser nulo");

    }

    else {

        cantidad += ctd;

    }

}

public void esValido()
```

```

    {

        if (cantidad < 0) {

            throw new StockArticuloNoValidoException("El stock no puede ser menor a 0");

        }

    }

}

}

}

```

Archivo: DireccionCliente.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Clientes\DireccionCliente.cs

```

using Empresa.LogicaDeNegocio.Entidades;

using Microsoft.EntityFrameworkCore.Metadata.Conventions;

using Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.Direccion;

using Papeleria.LogicaNegocio.InterfacesEntidades;

using System.ComponentModel.DataAnnotations.Schema;

using static System.Runtime.InteropServices.JavaScript.JSType;

namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Clientes

{

```

[ComplexType]

```
public record DireccionCliente : IValidable<DireccionCliente>

{

    public string Calle { get; init; }

    public int Numero { get; init; }

    public string Ciudad { get; init; }

    public int Distancia { get; init; }

    public DireccionCliente(string calle, int numero, string ciudad, int distancia)

    {

        if (calle == null || ciudad == null || numero == null) { throw new DireccionNuloException("Debe
ingresar todos los campos para direccion"); }

        Calle = calle;

        Numero = numero;

        Ciudad = ciudad;

        //Distancia = CalcularYFijarDistancia();

        Distancia = distancia;

        esValido();

    }

    public DireccionCliente()

    {
```

```
}
```

```
//public int CalcularYFijarDistancia()
```

```
//{
```

```
//    return 20;
```

```
//}
```

```
public void esValido()
```

```
{
```

```
    if(Calle == null || Ciudad == null) { throw new DireccionNuloException("Debe ingresar todos los  
campos para direccion"); }
```

```
    if(Numero <= 0) { throw new DireccionNoValidoException("El número debe ser 1 o mayor"); }
```

```
    if(Ciudad.Any(char.IsDigit)) { throw new DireccionNoValidoException("Ciudad no puede  
contener números"); }
```

```
    if (Distancia < 0) { throw new DireccionNoValidoException("La distancia no puede ser menor a  
0km"); }
```

```
}
```

```
}
```

```
}
```

```
*****
```


Archivo: RazonSocial.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Cientes\RazonSocial.cs

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Microsoft.EntityFrameworkCore;
```

```
using Papeleria.Comun;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RazonSocial;
```

```
using Papeleria.LogicaNegocio.InterfacesEntidades;
```

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

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Cientes
```

```
{
```

```
    [ComplexType]
```

```
    public record RazonSocial : IValidable<RazonSocial>
```

```
    {
```

```
        public string RazonSoc { get; init; }
```

```
        public RazonSocial(string razon)
```

```
        {
```

```
            RazonSoc = razon;
```

```
            esValido();
```

```
        }
```

```

public RazonSocial()

{

}

public void esValido()

{

    if (RazonSoc == null) { throw new RazonSocialNuloException("Raz❖n social no puede ser nula");
}

    if (RazonSoc.Length <= 1) { throw new RazonSocialNoValidoException("Raz❖n social
invalida"); }

}

}

}

```

Archivo: RUT.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Cientes\RUT.cs

```
using Empresa.LogicaDeNegocio.Entidades;
```

```
using Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RUT;
```

```
using Papeleria.LogicaNegocio.InterfacesEntidades;
```

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

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Cientes
```

```
{  
  
    [ComplexType]  
  
    public record RUT : IValidable<RUT>, IEquatable<RUT>  
  
    {  
  
        public long Rut { get; init; }  
  
  
  
        public RUT(long rut)  
  
        {  
  
            if(rut == null)  
  
            {  
  
                throw new RutNuloException("RUT no puede estar vacio");  
  
            }  
  
            Rut = rut;  
  
            esValido();  
  
        }  
  
  
  
        public void esValido()  
  
        {  
  
            if (Rut.ToString().Length != 12)  
  
            {  
  
                throw new RutNoValidoException("RUT Invalido, tiene que tener un largo de 12.");  
  
            }  
  
        }  
  
    }  
  
}
```

```
}
```

```
}
```

```
public override int GetHashCode()
```

```
{
```

```
    return base.GetHashCode();
```

```
}
```

```
public override string? ToString()
```

```
{
```

```
    return base.ToString();
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: IVA.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Pedidos\IVA.cs

```
*****
```

```
using Empresa.LogicaDeNegocio.Entidades;

using Papeleria.LogicaNegocio.Excepciones.Pedido.PedidoValueObjects;

using Papeleria.LogicaNegocio.InterfacesEntidades;

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

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Pedidos
```

```
{

    [ComplexType]

    public record IVA : IValidable<IVA>

    {

        public double valor { get; init; }

        public IVA(double valor)

        {

            if (valor >= 1)

            {

                valor = valor / 100;

            }

            this.valor = valor;

            esValido();

        }

        public IVA() { }

        public void esValido()
```

```

{

    if(valor < 0)

    {

        throw new IvaNoValidoException("Valor de IVA no puede ser inferior a 0");

    }

    if(valor == null) { throw new IvaNuloException("IVA no puede estar vacio"); }

}

public void FijarValor()

{

}

}

}

```

Archivo: ContraseniasUsuario.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Usuarios\ContraseniasUsuario.cs

```

using Empresa.LogicaDeNegocio.Entidades;

```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Contrasenia;
```

```
using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Email;
```

```
using Papeleria.LogicaNegocio.InterfacesEntidades;
```

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

```
using System.Security.Cryptography;
```

```
using System.Text;
```

```
using System.Text.RegularExpressions;
```

```
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario
```

```
{
```

```
    [ComplexType]
```

```
    public record ContraseniaUsuario : IValidable<ContraseniaUsuario>
```

```
    {
```

```
        public string Valor { get; set; }
```

```
        public ContraseniaUsuario(string contrasenia)
```

```
        {
```

```
            Valor = contrasenia;
```

```
            esValido();
```

```
        }
```

```
        public ContraseniaUsuario()
```

```
        {
```

```
        }
```

```

//public void esValido(string contrasenia) {

//    if (Valor == null)

//    {

//        throw new ContrasenianuloException("La contrasenia no puede ser nula.");

//    }

//    if (Valor.Length < 6)

//    {

//        throw new ContrasenianoValidoException("La contrasea debe contener un largo minimo de
6 caracteres.");

//    }

//    if (!Regex.IsMatch(Valor, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[.,;!])[A-Za-z\d.,;!]+$"))

//    {

//        throw new ContrasenianoValidoException("La contrasea debe contener al menos una letra
mayuscula, una minuscula, un dígito y un caracter de puntuación: punto, punto y coma, coma, signo
de admiración de cierre.");

//    }

//}

public void esValido()

{

    if (Valor == null)

    {

        throw new ContrasenianuloException("La contrasenia no puede ser nula.");

    }

    if (Valor.Length < 6)

```



```

{

    throw new ContraseñaNoValidoException("La contraseña debe contener un largo minimo de 6
caracteres.");

}

if (!Regex.IsMatch(Valor, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[.,;!])[A-Za-z\d.,;!]+$"))

{

    throw new ContraseñaNoValidoException("La contraseña debe contener al menos una letra
mayúscula, una minúscula, un dígito y un carácter de puntuación: punto, punto y coma, coma, signo
de admiración de cierre.");

}

}

public string Encriptar(string contraseña)

{

    using (SHA256 sha256Hash = SHA256.Create())

    {

        byte[] bytes = sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(contraseña));

        StringBuilder builder = new StringBuilder();

        for (int i = 0; i < bytes.Length; i++)

        {

            builder.Append(bytes[i].ToString("x2"));

        }

        return builder.ToString();

    }

```

```
}  
  
}  
  
}
```

Archivo: EmailUsuario.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Usuarios\EmailUsuario.cs

```
using Empresa.LogicaDeNegocio.Entidades;  
  
using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Email;  
  
using Papeleria.LogicaNegocio.InterfacesEntidades;  
  
using System.ComponentModel.DataAnnotations.Schema;  
  
using System.Drawing;  
  
namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario  
{  
  
    [ComplexType]  
  
    public record EmailUsuario : IValidable<EmailUsuario>, IEquatable<EmailUsuario>  
    {  
  
        public string Direccion { get; init; }  
  
        public EmailUsuario(string direccion) {  
  
            if (direccion == null)
```

```

{

    throw new ArgumentNullException(nameof(direccion), "No puede ser nulo");

}

Direccion = direccion;

esValido();

}

public void esValido(EmailUsuario emailUsuario)

{

    if (emailUsuario == null) {

        throw new EmailNuloException("El email no puede ser nulo.");

    }

    if (emailUsuario.Direccion.Length < 6) {

        throw new EmailNoValidoException("Email no v lido. Largo m nimo 6.");

    }

    if (!emailUsuario.Direccion.Contains("@")) {

        throw new EmailNoValidoException("Email no v lido. Debe incluir un arroba.");

    }

    if (emailUsuario.Direccion.IndexOf("@") == 0 || emailUsuario.Direccion.IndexOf("@") ==
emailUsuario.Direccion.Length-1) {

        throw new EmailNoValidoException("Email no v lido. No puede contener @ en el principio o
en el final.");

    }

}

```

```

public void esValido()

{

    esValido(this);

}

}

}

```

Archivo: NombreCompleto.cs

Localizacion: Papeleria.LogicaNegocio\Entidades\ValueObjects\Usuarios\NombreCompleto.cs

```

using Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepcions.Nombre;

using Papeleria.LogicaNegocio.InterfacesEntidades;

using System.Text.RegularExpressions;

using Papeleria.Comun;

using System.ComponentModel.DataAnnotations.Schema;

```

```

namespace Papeleria.LogicaNegocio.Entidades.ValueObjects.Usuario

```

```

{

    [ComplexType]

    public record NombreCompleto : IValidable<NombreCompleto>

    {

        public string Nombre { get; init; }
    }
}

```

```
public string Apellido { get; init; }
```

```
public NombreCompleto(string nombre, string apellido)
```

```
{
```

```
    Nombre = UtilidadesString.FormatInicialesMayuscula(nombre);
```

```
    Apellido = UtilidadesString.FormatInicialesMayuscula(apellido);
```

```
    esValido();
```

```
}
```

```
//public void esValido(string nombre, string apellido) {
```

```
//    if (nombre == null || apellido == null)
```

```
//    {
```

```
//        throw new NombreNuloException("El nombre o apellido no pueden ser nulos.");
```

```
//    }
```

```
//    if (nombre.Length <= 2 && !nombre.Any(c => char.IsDigit(c)))
```

```
//    {
```

```
//        throw new NombreNoValidoException($"{nombre}: no es un nombre valido.");
```

```
//    }
```

```
//    if (apellido.Length <= 2 && !apellido.Any(c => char.IsDigit(c)))
```

```
//    {
```

```
//        throw new NombreNoValidoException($"{apellido}: no es un apellido valido.");
```

```
//    }
```

```
//    if (!Regex.IsMatch(nombre, @"^[a-zA-Z]+(['-]?[a-zA-Z]+)*$"))
```

```
//    {
```

```

//      throw new NombreNoValidoException($"{nombre}: no es un nombre valido.");

//  }

//  if (!Regex.IsMatch(apellido, @"^[a-zA-Z]+(['-]?[a-zA-Z]+)*$"))

//  {

//      throw new NombreNoValidoException($"{apellido}: no es un apellido valido.");

//  }

//}

public void esValido()

{

    if (Nombre == null || Apellido == null) {

        throw new NombreNuloException("El nombre o apellido no pueden ser nulos.");

    }

    if (Nombre.Length <= 2 && !Nombre.Any(c => char.IsDigit(c))) {

        throw new NombreNoValidoException($"{Nombre}: no es un nombre valido.");

    }

    if (Apellido.Length <= 2 && !Apellido.Any(c => char.IsDigit(c)))

    {

        throw new NombreNoValidoException($"{Apellido}: no es un apellido valido.");

    }

    if (!Regex.IsMatch(Nombre, @"^[a-zA-Z]+(['-]?[a-zA-Z]+)*$")) {

        throw new NombreNoValidoException($"{Nombre}: no es un nombre valido.");

    }

    if (!Regex.IsMatch(Apellido, @"^[a-zA-Z]+(['-]?[a-zA-Z]+)*$")) {

```

```
        throw new NombreNoValidoException($"{Apellido}: no es un apellido valido.");
    }
}

}
```

Archivo: IvaNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Pedido\PedidoValueObjects\IvaNoValidoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Pedido.PedidoValueObjects
```

```
{
```

```
    public class IvaNoValidoException : Exception
```

```
    {
```

```
public IvaNoValidoException()
```

```
{
```

```
}
```

```
public IvaNoValidoException(string? message) : base(message)
```

```
{
```

```
}
```

```
public IvaNoValidoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected IvaNoValidoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: IvaNuloException.cs

Localizacion: Papeleria.LogicaNegocio\Excepciones\Pedido\PedidoValueObjects\IvaNuloException.cs

```
*****
```



```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Pedido.PedidoValueObjects
```

```
{
```

```
    public class IvaNuloException : Exception
```

```
    {
```

```
        public IvaNuloException()
```

```
        {
```

```
        }
```

```
        public IvaNuloException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public IvaNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```

        protected IvaNuloException(SerializationInfo info, StreamingContext context) : base(info, context)
        {

        }

    }

}

```

Archivo: CodigoProveedorDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\CodigoProveedor\CodigoProveedorDuplicadoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.CodigoProveedor
```

```
{
```

```
    public class CodigoProveedorDuplicadoException : Exception
```

```
    {
```

```
        public CodigoProveedorDuplicadoException()
```

```
{  
  
}
```

```
publicCodigoProveedorDuplicadoException(string? message) : base(message)
```

```
{  
  
}
```

```
publicCodigoProveedorDuplicadoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
{  
  
}
```

```
protectedCodigoProveedorDuplicadoException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
{  
  
}
```

```
}
```

```
}
```

```
*****
```

Archivo:CodigoProveedorNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\CodigoProveedor\CodigoProveedorNoValidoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.CodigoProveedor
```

```
{
```

```
    public class CodigoProveedorNoValidoException : Exception
```

```
    {
```

```
        public CodigoProveedorNoValidoException()
```

```
        {
```

```
        }
```

```
        public CodigoProveedorNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public CodigoProveedorNoValidoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```

        protected CodigoProveedorNoValidoException(SerializationInfo info, StreamingContext context) :
base(info, context)

    {

    }

}

}

```

Archivo: CodigoProveedorNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\CodigoProveedor\CodigoProveedorNuloException.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.CodigoProveedor
{

    public class CodigoProveedorNuloException : Exception

    {

```

```
public CodigoProveedorNuloException()
```

```
{
```

```
}
```

```
public CodigoProveedorNuloException(string? message) : base(message)
```

```
{
```

```
}
```

```
public CodigoProveedorNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected CodigoProveedorNuloException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: DescripcionArticuloDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\DescripcionArticulo\Descripcion
ArticuloDuplicadoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.DescripcionArticulo
```

```
{
```

```
    public class DescripcionArticuloDuplicadoException : Exception
```

```
    {
```

```
        public DescripcionArticuloDuplicadoException()
```

```
        {
```

```
        }
```

```
        public DescripcionArticuloDuplicadoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public DescripcionArticuloDuplicadoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected DescripcionArticuloDuplicadoException(SerializationInfo info, StreamingContext context)
        : base(info, context)

        {

        }

    }

}
```

Archivo: DescripcionArticuloNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\DescripcionArticulo\Descripcion
ArticuloNoValidoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.DescripcionArticulo
```

```
{
```

```
    public class DescripcionArticuloNoValidoException : Exception
```



```

{

    public DescripcionArticuloNoValidoException()

    {

    }

    public DescripcionArticuloNoValidoException(string? message) : base(message)

    {

    }

    public DescripcionArticuloNoValidoException(string? message, Exception? innerException) :
base(message, innerException)

    {

    }

    protected DescripcionArticuloNoValidoException(SerializationInfo info, StreamingContext context)
: base(info, context)

    {

    }

}

```

Archivo: DescripcionArticuloNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\DescripcionArticulo\Descripcion
ArticuloNuloException.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.DescripcionArticulo

{

public class DescripcionArticuloNuloException : Exception

{

public DescripcionArticuloNuloException()

{

}

public DescripcionArticuloNuloException(string? message) : base(message)

{

}

```
public DescripcionArticuloNuloException(string? message, Exception? innerException) :  
base(message, innerException)  
  
{  
  
}
```

```
protected DescripcionArticuloNuloException(SerializationInfo info, StreamingContext context) :  
base(info, context)  
  
{  
  
}  
  
}  
  
}
```

Archivo: LineaDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\Linea\LineaDuplicadoException.
cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.Linea

```
{  
  
    internal class LineaDuplicadoException  
  
    {  
  
    }  
  
}
```

Archivo: LineaNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\Linea\LineaNoValidoException.cs

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.Linea  
  
{  
  
    internal class LineaNoValidoException  
  
    {  
  
    }  
  
}
```

Archivo: LineaNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\Linea\LineaNuloException.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.Linea

{

public class LineaNuloException : Exception

{

public LineaNuloException()

{

}

public LineaNuloException(string? message) : base(message)

{

}

```
public LineaNuloException(string? message, Exception? innerException) : base(message,
innerException)
```

```
{
```

```
}
```

```
protected LineaNuloException(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: NombreArticuloDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\NombreArticulo\NombreArticulo
DuplicadoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.NombreArticulo

{

    public class NombreArticuloDuplicadoException : Exception

    {

        public NombreArticuloDuplicadoException()

        {

        }

        public NombreArticuloDuplicadoException(string? message) : base(message)

        {

        }

        public NombreArticuloDuplicadoException(string? message, Exception? innerException) :
base(message, innerException)

        {

        }

        protected NombreArticuloDuplicadoException(SerializationInfo info, StreamingContext context) :
base(info, context)

        {

        }

    }

}
```

Archivo: NombreArticuloNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\NombreArticulo\NombreArticuloNoValidoException.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.NombreArticulo

{

public class NombreArticuloNoValidoException : Exception

{

public NombreArticuloNoValidoException()

{

}

public NombreArticuloNoValidoException(string? message) : base(message)

{


```
}
```

```
public NombreArticuloNoValidoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
{
```

```
}
```

```
protected NombreArticuloNoValidoException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: NombreArticuloNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\NombreArticulo\NombreArticulo
NuloException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.NombreArticulo
```

```
{  
  
    public class NombreArticuloNuloException : Exception  
  
    {  
  
        public NombreArticuloNuloException()  
  
        {  
  
        }  
  
  
        public NombreArticuloNuloException(string? message) : base(message)  
  
        {  
  
        }  
  
  
        public NombreArticuloNuloException(string? message, Exception? innerException) : base(message,  
innerException)  
  
        {  
  
        }  
  
  
        protected NombreArticuloNuloException(SerializationInfo info, StreamingContext context) :  
base(info, context)  
  
        {  
  
        }  
  
    }  
}
```

```
}
```

```
*****
```

Archivo: StockArticuloDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\StockArticulo\StockArticuloDuplicadoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.StockArticulo
```

```
{
```

```
    public class StockArticuloDuplicadoException : Exception
```

```
    {
```

```
        public StockArticuloDuplicadoException()
```

```
        {
```

```
        }
```

```
        public StockArticuloDuplicadoException(string? message) : base(message)
```

```
{  
  
}
```

```
public StockArticuloDuplicadoException(string? message, Exception? innerException) :  
base(message, innerException)  
  
{  
  
}
```

```
protected StockArticuloDuplicadoException(SerializationInfo info, StreamingContext context) :  
base(info, context)  
  
{  
  
}  
  
}
```

Archivo: StockArticuloNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\StockArticulo\StockArticuloNoValidoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.StockArticulo
```

```
{
```

```
    public class StockArticuloNoValidoException : Exception
```

```
    {
```

```
        public StockArticuloNoValidoException()
```

```
        {
```

```
        }
```

```
        public StockArticuloNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public StockArticuloNoValidoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected StockArticuloNoValidoException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
        {
```

```
        }
```

```
}  
  
}
```

Archivo: StockArticuloNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Articulo\ArticulosValueObjects\StockArticulo\StockArticuloNuloException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Articulo.ArticulosValueObjects.StockArticulo
```

```
{  
  
    public class StockArticuloNuloException : Exception  
  
    {  
  
        public StockArticuloNuloException()  
  
        {  
  
        }  
  
    }  
  
}
```

```
public StockArticuloNuloException(string? message) : base(message)
```

```
{
```

```
}
```

```
public StockArticuloNuloException(string? message, Exception? innerException) : base(message,
innerException)
```

```
{
```

```
}
```

```
protected StockArticuloNuloException(SerializationInfo info, StreamingContext context) : base(info,
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: DireccionDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\Direccion\DireccionDuplicadoExcep
tion.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.Direccion
```

```
{
```

```
    public class DireccionDuplicadoException : Exception
```

```
    {
```

```
        public DireccionDuplicadoException() { }
```

```
        public DireccionDuplicadoException(string? message) : base(message) { }
```

```
        public DireccionDuplicadoException(string? message, Exception? innerException) : base(message, innerException) { }
```

```
        protected DireccionDuplicadoException(SerializationInfo info, StreamingContext context) : base(info, context) { }
```

```
    }
```

```
}
```

```
*****
```

Archivo: DireccionNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\Direccion\DireccionNoValidoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.Direccion
```

```
{
```

```
    public class DireccionNoValidoException : Exception
```

```
    {
```

```
        public DireccionNoValidoException() { }
```

```
        public DireccionNoValidoException(string? message) : base(message) { }
```

```
        public DireccionNoValidoException(string? message, Exception? innerException) : base(message, innerException) { }
```

```
        protected DireccionNoValidoException(SerializationInfo info, StreamingContext context) : base(info, context) { }
```

```
    }
```

```
}
```

Archivo: DireccionNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\Direccion\DireccionNuloException.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.Direccion

{

public class DireccionNuloException : Exception

{

public DireccionNuloException() { }

public DireccionNuloException(string? message) : base(message) { }

public DireccionNuloException(string? message, Exception? innerException) : base(message, innerException) { }

```
        protected DireccionNuloException(SerializationInfo info, StreamingContext context) : base(info,
context) { }

    }

}
```

Archivo: RazonSocialDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\RazonSocial\RazonSocialDuplicado
Exception.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;


namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RazonSocial
{

    public class RazonSocialDuplicadoException : Exception

    {

        public RazonSocialDuplicadoException() { }

    }

}
```

```
public RazonSocialDuplicadoException(string? message) : base(message) { }
```

```
public RazonSocialDuplicadoException(string? message, Exception? innerException) :  
base(message, innerException) { }
```

```
protected RazonSocialDuplicadoException(SerializationInfo info, StreamingContext context) :  
base(info, context) { }
```

```
}
```

```
}
```

Archivo: RazonSocialNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\RazonSocial\RazonSocialNoValidoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RazonSocial
```

```

{

    public class RazonSocialNoValidoException : Exception

    {

        public RazonSocialNoValidoException() { }

        public RazonSocialNoValidoException(string? message) : base(message) { }

        public RazonSocialNoValidoException(string? message, Exception? innerException) : base(message,
innerException) { }

        protected RazonSocialNoValidoException(SerializationInfo info, StreamingContext context) :
base(info, context) { }

    }

}

```

Archivo: RazonSocialNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\RazonSocial\RazonSocialNuloExcep
tion.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RazonSocial
```

```
{
```

```
    public class RazonSocialNuloException : Exception
```

```
    {
```

```
        public RazonSocialNuloException()
```

```
        {
```

```
        }
```

```
        public RazonSocialNuloException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public RazonSocialNuloException(string? message, Exception? innerException) : base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected RazonSocialNuloException(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
        {
```

```
        }
```

```
}  
  
}
```

Archivo: RutDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\RUT\RutDuplicadoException.cs

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RUT
```

```
{
```

```
    public class RutDuplicadoException : Exception
```

```
    {
```

```
        public RutDuplicadoException() { }
```

```
        public RutDuplicadoException(string? message) : base(message) { }
```

```
public RutDuplicadoException(string? message, Exception? innerException) : base(message,
innerException) { }
```

```
protected RutDuplicadoException(SerializationInfo info, StreamingContext context) : base(info,
context) { }
```

```
}
```

```
}
```

```
*****
```

Archivo: RutNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\RUT\RutNoValidoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RUT
```

```
{
```

```
public class RutNoValidoException : Exception
```

```
{
```



```
public RutNoValidoException() { }
```

```
public RutNoValidoException(string? message) : base(message) { }
```

```
public RutNoValidoException(string? message, Exception? innerException) : base(message, innerException) { }
```

```
protected RutNoValidoException(SerializationInfo info, StreamingContext context) : base(info, context) { }
```

```
}
```

```
}
```

```
*****
```

Archivo: RutNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Cliente\ClienteValueObjects\RUT\RutNuloException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Cliente.ClienteValueObjects.RUT
```

```
{  
  
    public class RutNuloException : Exception  
  
    {  
  
        public RutNuloException() { }  
  
  
        public RutNuloException(string? message) : base(message) { }  
  
  
  
        public RutNuloException(string? message, Exception? innerException) : base(message,  
innerException) { }  
  
  
  
        protected RutNuloException(SerializationInfo info, StreamingContext context) : base(info, context) {  
    }  
  
    }  
  
}
```

```
*****
```

Archivo: ContraseniaNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepciones\Constrasenia\ContraseniaNoValidoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Constrasenia
```

```
{
```

```
    public class ContrasenianNoValidoException : Exception
```

```
    {
```

```
        public ContrasenianNoValidoException()
```

```
        {
```

```
        }
```

```
        public ContrasenianNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public ContrasenianNoValidoException(string? message, Exception? innerException) : base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected ContrasenianNoValidoException(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
        {
```

```
}  
  
}  
  
}
```

Archivo: ContrasenianuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepciones\ConstrasenianuloException.cs

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Runtime.Serialization;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.ConstrasenianuloException  
{  
  
    public class ConstrasenianuloException : Exception  
  
    {  
  
        public ConstrasenianuloException()  
  
        {  
  
        }  
  
    }  
  
}
```

```
public ContrasenianuloException(string? message) : base(message)
```

```
{
```

```
}
```

```
public ContrasenianuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected ContrasenianuloException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: EmailDuplicadoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepcions\Email\EmailDuplicadoException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Email
```

```
{
```

```
    public class EmailDuplicadoException : Exception
```

```
    {
```

```
        public EmailDuplicadoException()
```

```
        {
```

```
        }
```

```
        public EmailDuplicadoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public EmailDuplicadoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected EmailDuplicadoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{  
  
}  
  
}  
  
}
```

Archivo: EmailNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepcions\Email\EmailNoValidoException.cs

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Runtime.Serialization;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepcions.Email  
{  
  
    public class EmailNoValidoException : Exception  
  
    {  
  
        public EmailNoValidoException()  
  
        {  
  
        }  
  
    }  
  
}
```

```
public EmailNoValidoException(string? message) : base(message)
```

```
{
```

```
}
```

```
public EmailNoValidoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected EmailNoValidoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: EmailNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepcions\Email\EmailNuloException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```



```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Email
```

```
{
```

```
    public class EmailNuloException : Exception
```

```
    {
```

```
        public EmailNuloException()
```

```
        {
```

```
        }
```

```
        public EmailNuloException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public EmailNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected EmailNuloException(SerializationInfo info, StreamingContext context) : base(info, context)
```

```
        {
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: NombreNoValidoException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepcions\Nombre\NombreNoValidoException.
cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepcions.Nombre
```

```
{
```

```
    public class NombreNoValidoException : Exception
```

```
    {
```

```
        public NombreNoValidoException()
```

```
        {
```

```
        }
```

```
public NombreNoValidoException(string? message) : base(message)
```

```
{
```

```
}
```

```
public NombreNoValidoException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
{
```

```
}
```

```
protected NombreNoValidoException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
*****
```

Archivo: NombreNuloException.cs

Localizacion:

Papeleria.LogicaNegocio\Excepciones\Usuario\UsuarioExcepcions\Nombre\NombreNuloException.cs

```
*****
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Excepciones.Usuario.UsuarioExcepciones.Nombre
```

```
{
```

```
    public class NombreNuloException : Exception
```

```
    {
```

```
        public NombreNuloException()
```

```
        {
```

```
        }
```

```
        public NombreNuloException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public NombreNuloException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected NombreNuloException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
{  
  
}  
  
}  
  
}
```

Archivo: _ViewImports.cshtml

Localizacion: Papeleria.MVC\Views_ViewImports.cshtml

@using Papeleria.MVC

@using Papeleria.MVC.Models

@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

Archivo: _ViewStart.cshtml

Localizacion: Papeleria.MVC\Views_ViewStart.cshtml

```
@ {  
  
    Layout = "_Layout";  
  
}
```

Archivo: _ViewImports.cshtml

Localizacion: Papeleria.WebMvc\Views_ViewImports.cshtml

@using Papeleria.WebMvc

@using Papeleria.WebMvc.Models

@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

Archivo: _ViewStart.cshtml

Localizacion: Papeleria.WebMvc\Views_ViewStart.cshtml

@ {

Layout = "_Layout";

}

Archivo: Create.cshtml

Localizacion: Papeleria.MVC\Views\Articulos\Create.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos.ArticuloDTO

@ {

ViewData["Title"] = "Crear articulo";

}

<h1>Crear articulo</h1>

<h4>Articulo</h4>

<hr />

<partial name="_Mensajes" />

<div class="row">

<div class="col-md-4">

<form asp-action="Create">

<div asp-validation-summary="ModelOnly" class="text-danger"></div>

<div class="form-group">

<label asp-for="CodigoProveedor" class="control-label"></label>

<input asp-for="CodigoProveedor" class="form-control" />

<label asp-for="NombreArticulo" class="control-label"></label>

<input asp-for="NombreArticulo" class="form-control" />

<label asp-for="Descripcion" class="control-label"></label>

<input asp-for="Descripcion" class="form-control" />

<label asp-for="PrecioVP" class="control-label"></label>

<input asp-for="PrecioVP" class="form-control" />

<label asp-for="Stock" class="control-label"></label>

```

        <input asp-for="Stock" class="form-control" />

        <span asp-validation-for="Stock" class="text-danger"></span>

    </div>

    <div class="form-group">

        <input type="submit" value="Create" class="btn btn-primary" />

    </div>

</form>

</div>

</div>

<div>

    <a asp-action="Index">Back to List</a>

</div>

```

```

@section Scripts {

    @{

        await Html.RenderPartialAsync("_ValidationScriptsPartial");

    }

}

```

Archivo: Delete.cshtml

Localizacion: Papeleria.MVC\Views\Articulos\Delete.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos.ArticuloDTO

@{

 ViewData["Title"] = "Delete";

}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>

<div>

 <h4>Articulo</h4>

 <hr />

 <dl class="row">

 <dt class="col-sm-2">

 @Html.DisplayNameFor(model => model.CodigoProveedor)

 </dt>

 <dd class="col-sm-10">

 @Html.DisplayFor(model => model.CodigoProveedor)

 </dd>

 <dt class="col-sm-2">

 @Html.DisplayNameFor(model => model.NombreArticulo)

 </dt>

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.NombreArticulo)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.Descripcion)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.Descripcion)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.PrecioVP)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.PrecioVP)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.Stock)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.Stock)
```

```
</dd>
```

```
</dl>
```

```
<form asp-action="Delete">

    <input type="submit" value="Delete" class="btn btn-danger" /> |

    @Html.CheckBox("IsChecked") <span>Entiendo que esta operación es riesgosa</span>

    <a asp-action="Index">Volver a la lista</a>

</form>

</div>
```

Archivo: Edit.cshtml

Localizacion: Papeleria.MVC\Views\Articulos\Edit.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos.ArticuloDTO

```
@ {

    ViewData["Title"] = "Edit";

}
```

```
<h1>Edit</h1>
```

```
<h4>Articulo</h4>
```

```
<hr />
```

```
<div class="row">
```

```
    <div class="col-md-4">
```

```
<form asp-action="Edit">

  <div asp-validation-summary="ModelOnly" class="text-danger"></div>

  <input type="hidden" asp-for="Id" />

  <div class="form-group">

    <label asp-for="CodigoProveedor" class="control-label"></label>

    <input asp-for="CodigoProveedor" class="form-control" />

    <span asp-validation-for="CodigoProveedor" class="text-danger"></span>

    <label asp-for="NombreArticulo" class="control-label"></label>

    <input asp-for="NombreArticulo" class="form-control" />

    <span asp-validation-for="NombreArticulo" class="text-danger"></span>

    <label asp-for="Descripcion" class="control-label"></label>

    <input asp-for="Descripcion" class="form-control" />

    <span asp-validation-for="Descripcion" class="text-danger"></span>

    <label asp-for="PrecioVP" class="control-label"></label>

    <input asp-for="PrecioVP" class="form-control" />

    <span asp-validation-for="PrecioVP" class="text-danger"></span>

    <label asp-for="Stock" class="control-label"></label>

    <input asp-for="Stock" class="form-control" />

    <span asp-validation-for="Stock" class="text-danger"></span>

  </div>

  <div class="form-group">

    <input type="submit" value="Save" class="btn btn-primary" />

  </div>
```

</form>

</div>

</div>

<div>

<a asp-action="Index">Back to List

</div>

@section Scripts {

@ {await Html.RenderPartialAsync("_ValidationScriptsPartial");}

}

Archivo: Index.cshtml

Localizacion: Papeleria.MVC\Views\Articulos\Index.cshtml

@model IEnumerable<Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Articulos.ArticuloDTO>

@ {

ViewData["Title"] = "Index";

}

<h1>ARTICULOS</h1>

<p>

<a asp-action="Create">CREAR ARTICULOS

</p>

<partial name="_Mensajes" />

<table class="table">

<thead>

<tr>

<th>

@Html.DisplayNameFor(model => model.CodigoProveedor)

</th>

<th>

@Html.DisplayNameFor(model => model.NombreArticulo)

</th>

<th>

@Html.DisplayNameFor(model => model.Descripcion)

</th>

<th>

@Html.DisplayNameFor(model => model.PrecioVP)

</th>

<th>

@Html.DisplayNameFor(model => model.Stock)

</th>

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
@foreach (var item in Model) {
```

```
<tr>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.CodigoProveedor)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.NombreArticulo)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.Descripcion)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.PrecioVP)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.Stock)
```

```
</td>
```

```
<td>
```

```
<a asp-action="Edit" asp-route-id="@item.Id">Editar</a> |
```

```
<a asp-action="Delete" asp-route-id="@item.Id">Borrar</a>
```

</td>

</tr>

}

</tbody>

</table>

Archivo: Create.cshtml

Localizacion: Papeleria.MVC\Views\Cientes\Create.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Cientes.ClienteDTO

@ {

 ViewData["Title"] = "Create";

}

<h1>Crear Cliente</h1>

<h4>Ingrese los datos del cliente:</h4>

<hr />

<partial name="_Mensajes" />

<div class="row">

 <div class="col-md-4">


```
<form asp-action="Create">
```

```
<div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
<div class="form-group">
```

```
<label asp-for="rut" class="control-label">RUT</label>
```

```
<input asp-for="rut" class="form-control" />
```

```
<span asp-validation-for="rut" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="razonSocial" class="control-label">Razon Social</label>
```

```
<input asp-for="razonSocial" class="form-control" />
```

```
<span asp-validation-for="razonSocial" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="Calle" class="control-label"></label>
```

```
<input asp-for="Calle" class="form-control" />
```

```
<span asp-validation-for="Calle" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="Numero" class="control-label"></label>
```

```
<input asp-for="Numero" class="form-control" />
```

```
<span asp-validation-for="Numero" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="Ciudad" class="control-label"></label>
```

```
<input asp-for="Ciudad" class="form-control" />
```

```
<span asp-validation-for="Ciudad" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="Distancia" class="control-label"></label>
```

```
<input asp-for="Distancia" class="form-control" />
```

```
<span asp-validation-for="Distancia" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="submit" value="Crear" class="btn btn-primary" />
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<a asp-action="Index">Lista</a>
```

```
</div>
```

```
@section Scripts {
```

```
@{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
```

```
}
```

Archivo: Delete.cshtml

Localizacion: Papeleria.MVC\Views\Clientes\Delete.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes.ClienteDTO

<h3>Estas seguro que deseas eliminar este cliente?</h3>

<div>

<h4>ClienteDTO</h4>

<hr />

<dl class="row">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Id)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.rut)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.rut)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.razonSocial)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.razonSocial)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Calle)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Calle)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Numero)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Numero)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Ciudad)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Ciudad)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Distancia)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Distancia)

</dd>

</dl>

<form asp-action="Delete">

<input type="submit" value="Delete" class="btn btn-danger" /> |

@Html.CheckBox("IsChecked") Entiendo que esta operación es riesgosa

<a asp-action="Index">Volver a la lista

</form>

</div>

Archivo: Details.cshtml

Localizacion: Papeleria.MVC\Views\Cientes\Details.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Cientes.ClienteDTO

```
<div>
```

```
<h4>Detalles del cliente</h4>
```

```
<hr />
```

```
<dl class="row">
```

```
<dt class = "col-sm-2">
```

```
@Html.DisplayNameFor(model => model.Id)
```

```
</dt>
```

```
<dd class = "col-sm-10">
```

```
@Html.DisplayFor(model => model.Id)
```

```
</dd>
```

```
<dt class = "col-sm-2">
```

```
@Html.DisplayNameFor(model => model.rut)
```

```
</dt>
```

```
<dd class = "col-sm-10">
```

```
@Html.DisplayFor(model => model.rut)
```

```
</dd>
```

```
<dt class = "col-sm-2">
```

```
@Html.DisplayNameFor(model => model.razonSocial)
```

```
</dt>
```

```
<dd class = "col-sm-10">
```

```
@Html.DisplayFor(model => model.razonSocial)
```

```
</dd>
```

```
<dt class = "col-sm-2">
```

@Html.DisplayNameFor(model => model.Calle)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Calle)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Numero)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Numero)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Ciudad)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Ciudad)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Distance)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Distance)

</dd>

</dl>

</div>

<div>

<a asp-action="Index">LISTA

</div>

Archivo: Edit.cshtml

Localizacion: Papeleria.MVC\Views\Cientes\Edit.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Cientes.ClienteDTO

<h4>Editar CLIENTE</h4>

<hr />

<partial name="_Mensajes" />

<div class="row">

<div class="col-md-4">

<form asp-action="Edit">

<div asp-validation-summary="ModelOnly" class="text-danger"></div>

<div class="form-group">

<label asp-for="Id" class="control-label">ID</label>

<input asp-for="Id" class="form-control" readonly />

</div>

<div class="form-group">

<label asp-for="rut" class="control-label">RUT</label>

<input asp-for="rut" class="form-control" />

</div>

<div class="form-group">

<label asp-for="razonSocial" class="control-label">Razon Social</label>

<input asp-for="razonSocial" class="form-control" />

</div>

<div class="form-group">

<label asp-for="Calle" class="control-label"></label>

<input asp-for="Calle" class="form-control" />

</div>

<div class="form-group">

<label asp-for="Numero" class="control-label"></label>

<input asp-for="Numero" class="form-control" />

</div>

<div class="form-group">

<label asp-for="Ciudad" class="control-label"></label>

```
<input asp-for="Ciudad" class="form-control" />
```

```
<span asp-validation-for="Ciudad" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="Distancia" class="control-label"></label>
```

```
<input asp-for="Distancia" class="form-control" />
```

```
<span asp-validation-for="Distancia" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="submit" value="Guardar" class="btn btn-primary" />
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<a asp-action="Index">Volver a la lista</a>
```

```
</div>
```

```
*****
```

Archivo: Index.cshtml

Localizacion: Papeleria.MVC\Views\Clientes\Index.cshtml

@model IEnumerable<Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Clientes.ClienteDTO>

<p>

<a asp-action="Create">CREAR CLIENTE

</p>

<h1>BUSCAR CLIENTES</h1>

<hr />

<partial name="_Mensajes" />

@* <form method="post">

<input type="text" placeholder="Ingrese razon social" name="razonSocial" /><input type="submit" value="Filtrar" />

</form>

<form method="post">

<input type="number" placeholder="Ingrese un monto" name="monto" /><input type="submit" value="Filtrar" />

</form> *@

<form method="post">

<input type="text" placeholder="Ingrese razón social" name="razonSocial" />

<input type="submit" value="Filtrar X Razon" name="submitButton" />

</form>


```
<form method="post">
```

```
    <input type="number" placeholder="Ingrese un monto" name="monto" />
```

```
    <input type="submit" value="Filtrar X Monto" name="submitButton" />
```

```
</form>
```

```
<p>
```

```
    @if (@TempData["ResultadoBuscarClientes"] != null)
```

```
    {
```

```
        @TempData["ResultadoBuscarClientes"].ToString()
```

```
    }
```

```
</p>
```

```
<hr />
```

```
<table class="table">
```

```
    <thead>
```

```
        <tr>
```

```
            <th>
```

```
                ID
```

```
            </th>
```

```
            <th>
```

```
                RUT
```

```
            </th>
```

<th>

Razón Social

</th>

<th>

Calle

</th>

<th>

Número

</th>

<th>

Ciudad

</th>

<th>

Distancia en KM

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Id)

</td>

<td>

@Html.DisplayFor(modelItem => item.rut)

</td>

<td>

@Html.DisplayFor(modelItem => item.razonSocial)

</td>

<td>

@Html.DisplayFor(modelItem => item.Calle)

</td>

<td>

@Html.DisplayFor(modelItem => item.Numero)

</td>

<td>

@Html.DisplayFor(modelItem => item.Ciudad)

</td>

<td>

@Html.DisplayFor(modelItem => item.Distancia)

</td>

<td>

@Html.ActionLink("Editar", "Edit", new { id = item.Id }) |

@Html.ActionLink("Eliminar", "Delete", new { id=item.Id })

</td>

</tr>

}

</tbody>

</table>

Archivo: ListaSuperaronMonto.cshtml

Localizacion: Papeleria.MVC\Views\Cientes\ListaSuperaronMonto.cshtml

@model IEnumerable<Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Cientes.ClienteDTO>

@ {

 ViewData["Title"] = "ListaSuperaronMonto";

}

<h1>CLIENTES QUE SUPERARON MONTO DE \$@ViewBag.Monto</h1>

<table class="table">

 <thead>

 <tr>

 <th>

 ID

 </th>

<th>

RUT

</th>

<th>

Razon Social

</th>

<th>

@Html.DisplayNameFor(model => model.Calle)

</th>

<th>

@Html.DisplayNameFor(model => model.Numero)

</th>

<th>

@Html.DisplayNameFor(model => model.Ciudad)

</th>

<th>

@Html.DisplayNameFor(model => model.Distancia)

</th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Id)

</td>

<td>

@Html.DisplayFor(modelItem => item.rut)

</td>

<td>

@Html.DisplayFor(modelItem => item.razonSocial)

</td>

<td>

@Html.DisplayFor(modelItem => item.Calle)

</td>

<td>

@Html.DisplayFor(modelItem => item.Numero)

</td>

<td>

@Html.DisplayFor(modelItem => item.Ciudad)

</td>

<td>

@Html.DisplayFor(modelItem => item.Distance)

</td>

</tr>

}

</tbody>

</table>

<div>

<a asp-action="Index">Lista

</div>

Archivo: Index.cshtml

Localizacion: Papeleria.MVC\Views\Home\Index.cshtml

@model Empresa.LogicaDeNegocio.Sistema.Usuario;

@ {

 ViewData["Title"] = "Login";

}

<h1>INICIO DE SESIÓN</h1>

<hr />

<partial name="_Mensajes" />

<div class="row">

 <div class="col-md-4">

```
<form asp-action="Login">

    <div asp-validation-summary="ModelOnly" class="text-danger"></div>

    <div class="form-group">

        <label asp-for="Email" class="control-label">Email</label>

        <input asp-for="Email" class="form-control" />

        <span asp-validation-for="Email" class="text-danger"></span>

    </div>

    <div class="form-group">

        <label asp-for="Contrasenia" class="control-label">Contraseña</label>

        <input type="password" asp-for="Contrasenia" class="form-control" />

        <span asp-validation-for="Contrasenia" class="text-danger"></span>

    </div>

    <br />

    <div class="form-group">

        <input type="submit" value="Iniciar sesión" class="btn btn-primary" />

    </div>

</form>

</div>

</div>
```

Archivo: Anular.cshtml

Localizacion: Papeleria.MVC\Views\Pedidos\Anular.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos.PedidoDTO

@ {

 ViewData["Title"] = "Detalle del pedido";

}

<p>

 <a asp-action="Index">Volver a la lista

</p>

<div class="div">

 <h4>Pedido</h4>

 <hr />

 <dl class="row">

 <dt class="col-sm-2">

 @Html.DisplayNameFor(model => model.FechaPedido)

 </dt>

 <dd class="col-sm-10">

 @Html.DisplayFor(model => model.FechaPedido)

 </dd>

 <dt class="col-sm-2">

 @Html.DisplayNameFor(model => model.FechaEntrega)

 </dt>

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.FechaEntrega)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.ClienteID)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.ClienteID)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.precioFinal)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.precioFinal)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.entregado)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
    @Html.DisplayFor(model => model.entregado)
```

```
</dd>
```

```
<form asp-action="Anular">
```

```
<input type="submit" value="Anular" class="btn btn-danger" /> |
```

```
@Html.CheckBox("IsChecked") <span>Entiendo que esta operación es riesgosa</span>
```

```
<a asp-action="Index">Volver a la lista</a>
```

```
</form>
```

```
</div>
```

```
*****
```

Archivo: Crear.cshtml

Localizacion: Papeleria.MVC\Views\Pedidos\Crear.cshtml

```
*****
```

```
@using Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos
```

```
@model Papeleria.MVC.Models.PedidosModels.PedidoAltaModel
```

```
<h1>Crear Pedido</h1>
```

```
<p>@ViewBag.Error</p>
```

```
<form asp-action="Crear" method="post">
```

```
<div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
<div class="form-group">
```

```
<label asp-for="ClienteId">Cliente:</label>
```

```
<select asp-for="ClienteId" class="form-control" asp-items="@ (new SelectList(ViewBag.Cientes, "Id", "razonSocial"))">
```

```
<option value="">Seleccionar cliente...</option>
```

```
</select>
```


</div>

<div class="form-group">

<label asp-for="FechaEntrega">Fecha de Entrega:</label>

<input asp-for="FechaEntrega" class="form-control" type="date" />

</div>

<div class="form-group">

<label asp-for="iva">Valor IVA:</label>

<input asp-for="iva" class="form-control" type="number" value="22" />

</div>

@if (ViewBag.LineasPedido != null && ViewBag.LineasPedido.Count > 0)

{

<div class="form-group">

<label>Articulos pedidos:</label>

<table class="table table-bordered" id="lineasPedidoAgregados">

<thead>

<tr>

<th>Articulo</th>

<th>Cantidad</th>

</tr>

</thead>

<tbody>

</tbody>

@foreach (LineaPedidoDTO linea in ViewBag.LineasPedido)

{

<tr>

<td>

@linea.NombreArticulo

</td>

<td>

Cantidad: @linea.Cantidad

</td>

</tr>

}

</table>

</div>

}

<button type="submit" class="btn btn-success">Crear Pedido</button>

</form>


```
<div class="form-group">
```

```
    <form asp-action="AddArticulo">
```

```
        <label>Artículos:</label>
```

```
        <table class="table table-bordered" id="lineasPedido">
```

```
            <thead>
```

```
                <tr>
```

```
                    <th>Artículo</th>
```

```
                    <th>Cantidad</th>
```

```
                </tr>
```

```
            </thead>
```

```
            <tbody>
```

```
                @foreach (var linea in Model.LineasPedido)
```

```
                {
```

```
                    <tr>
```

```
                        <td>
```

```
                            <select      asp-for="ArticuloId"      class="form-control"      asp-items="@ (new  
SelectList(ViewBag.Articulos, "Id", "NombreArticulo"))">
```

```
                                <option value="">Seleccionar artículo...</option>
```

```
                            </select>
```

```
                            <span asp-validation-for="ArticuloId" class="text-danger"></span>
```

```
                        </td>
```

```
                        <td><input asp-for="Cantidad" class="form-control" /></td>
```

```

        <span asp-validation-for="Cantidad" class="text-danger"></span>

    </tr>

}

</tbody>

</table>

<button type="submit" class="btn btn-primary" id="agregarLinea">Agregar Línea de
Pedido</button>

</form>

</div>

```

```

@section Scripts {

    @{

        await Html.RenderPartialAsync("_ValidationScriptsPartial");

    }

}

```

Archivo: Details.cshtml

Localizacion: Papeleria.MVC\Views\Pedidos\Details.cshtml

```
@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos.PedidoDTO
```

```
@{
```

```
ViewData["Title"] = "Detalle del pedido";  
  
}
```

```
<p>
```

```
    <a asp-action="Index">Volver a la lista</a>
```

```
</p>
```

```
<div class="div">
```

```
    <h4>Pedido</h4>
```

```
    <hr />
```

```
    <dl class="row">
```

```
        <dt class="col-sm-2">
```

```
            @Html.DisplayNameFor(model => model.FechaPedido)
```

```
        </dt>
```

```
        <dd class="col-sm-10">
```

```
            @Html.DisplayFor(model => model.FechaPedido)
```

```
        </dd>
```

```
        <dt class="col-sm-2">
```

```
            @Html.DisplayNameFor(model => model.FechaEntrega)
```

```
        </dt>
```

```
        <dd class="col-sm-10">
```

```
            @Html.DisplayFor(model => model.FechaEntrega)
```

```
        </dd>
```

```
        <dt class="col-sm-2">
```

```
@Html.DisplayNameFor(model => model.ClienteID)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
@Html.DisplayFor(model => model.ClienteID)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
@Html.DisplayNameFor(model => model.precioFinal)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
@Html.DisplayFor(model => model.precioFinal)
```

```
</dd>
```

```
<dt class="col-sm-2">
```

```
@Html.DisplayNameFor(model => model.entregado)
```

```
</dt>
```

```
<dd class="col-sm-10">
```

```
@Html.DisplayFor(model => model.entregado)
```

```
</dd>
```

```
</dl>
```

```
</div>
```

```
*****
```

Archivo: Index.cshtml

Localizacion: Papeleria.MVC\Views\Pedidos\Index.cshtml

```
*****
```

```
@model IEnumerable<Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Pedidos.PedidoDTO>
```

```
@ {
```

```
    ViewData["Title"] = "Index";
```

```
}
```

```
<h1>Pedidos registrados</h1>
```

```
<p>
```

```
    <a asp-action="Crear">CREAR PEDIDO</a>
```

```
</p>
```

```
<h4>BUSCAR PEDIDOS DESDE FECHA</h4>
```

```
<hr />
```

```
<form method="post">
```

```
    <input type="date" placeholder="Ingrese fecha para buscar pedidos desde la misma" name="date" />
```

```
    <input type="submit" value="Filtrar" name="submitButton" />
```

```
</form>
```

```
<p>@ViewBag.Mensaje</p>
```

```
<table class="table">
```

```
    <thead>
```

```
        <tr>
```

<th>

Fecha Pedido

</th>

<th>

Razon Social

</th>

<th>

Precio Final \$

</th>

<th>

Entregado

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

if (!item.anulado)

{

<tr>

<td>

```

        @Html.DisplayFor(modelItem => item.FechaPedido)

    </td>

    <td>

        @foreach (var cliente in ViewBag.Clientes)

        {

            if (cliente.Id.Equals(item.ClienteID))

            {

                @cliente.razonSocial

                ;

            }

        }

    </td>

    <td>

        @Html.DisplayFor(modelItem => item.precioFinal)

    </td>

    <td>

        @Html.DisplayFor(modelItem => item.entregado)

    </td>

    <td>

        <a asp-action="Details" asp-route-id="@item.Id">Detalle</a> |

        <a asp-action="Anular" asp-route-id="@item.Id">Anular</a>

    </td>

</tr>

```

```
    }

}

</tbody>

</table>
```

```
*****
```

Archivo: Error.cshtml

Localizacion: Papeleria.MVC\Views\Shared\Error.cshtml

```
*****
```

```
@model ErrorViewModel
```

```
@{
```

```
    ViewData["Title"] = "Error";
```

```
}
```

```
<h1 class="text-danger">Error.</h1>
```

```
<h2 class="text-danger">An error occurred while processing your request.</h2>
```

```
@if (Model.ShowRequestId)
```

```
{
```

```
    <p>
```

```
        <strong>Request ID:</strong> <code>@Model.RequestId</code>
```

```
    </p>
```

```
}
```


<h3>Development Mode</h3>

<p>

Swapping to Development environment will display more detailed information about the error that occurred.

</p>

<p>

The Development environment shouldn't be enabled for deployed applications.

It can result in displaying sensitive information from exceptions to end users.

For local debugging, enable the Development environment by setting the ASPNETCORE_ENVIRONMENT environment variable to Development

and restarting the app.

</p>

Archivo: _Layout.cshtml

Localizacion: Papeleria.MVC\Views\Shared_Layout.cshtml

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

```

<title>@ViewData["Title"] - Papeleria.MVC</title>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Papeleria.MVC.styles.css" asp-append-version="true" />

</head>

<body>

    <header>

        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom
        box-shadow mb-3">

            <div class="container-fluid">

                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Papeleria</a>

                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
                target=".navbar-collapse" aria-controls="navbarSupportedContent"

                    aria-expanded="false" aria-label="Toggle navigation">

                    <span class="navbar-toggler-icon"></span>

                </button>

                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

                    <ul class="navbar-nav flex-grow-1">

                        @if (Context.Session.GetInt32("LogueadoID") == null)

                        {

                            <li class="nav-item">

                                <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-
                                action="Index">Inicio</a>

                                </li>

```

```
}
```

```
@if (Context.Session.GetInt32("LogueadoID") != null)
```

```
{
```

```
    <li class="nav-item">
```

```
        <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-  
action="Index">Usuarios</a>
```

```
    </li>
```

```
    <li class="nav-item">
```

```
        <a class="nav-link text-dark" asp-area="" asp-controller="Clientes" asp-  
action="Index">Clientes</a>
```

```
    </li>
```

```
    <li class="nav-item">
```

```
        <a class="nav-link text-dark" asp-area="" asp-controller="Articulos" asp-  
action="Index">Articulos</a>
```

```
    </li>
```

```
    <li class="nav-item">
```

```
        <a class="nav-link text-dark" asp-area="" asp-controller="Pedidos" asp-  
action="Index">Pedidos</a>
```

```
    </li>
```

```
    <li class="nav-item">
```

```
        <a class="nav-link text-dark btn btn-danger" asp-area="" asp-controller="Home" asp-  
action="Logout">LOGOUT</a>
```

```
    </li>
```

```
}
```


</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

© 2024 - Papeleria.MVC - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

<script src="~/js/site.js" asp-append-version="true"></script>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

Archivo: _Mensajes.cshtml

Localizacion: Papeleria.MVC\Views\Shared_Mensajes.cshtml

@*

For more information on enabling MVC for empty projects, visit
<http://go.microsoft.com/fwlink/?LinkID=397860>

*@

@if (ViewBag.Error != null)

{

<div class="alert alert-danger">@ ViewBag.Error</div>

}

@if (TempData["Error"] != null)

{

<div class="alert alert-danger">@TempData["Error"]</div>

}

@if (ViewBag.Mensaje != null)

{

<div class="alert alert-info">@ ViewBag.Mensaje</div>

}

```
@if (TempData["Mensaje"] != null)

{

    <div class="alert alert-info">@TempData["Mensaje"]</div>

}

@if (TempData["MensajeOK"] != null)

{

    <div class="alert alert-success">@TempData["MensajeOK"]</div>

}
```

Archivo: _ValidationScriptsPartial.cshtml

Localizacion: Papeleria.MVC\Views\Shared_ValidationScriptsPartial.cshtml

```
<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
```

Archivo: Create.cshtml

Localizacion: Papeleria.MVC\Views\Usuarios\Create.cshtml

```
@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios.UsuarioDTO
```

```
@ {

    ViewData["Title"] = "Crear Usuario";
```

}

<h1>Crear Usuario</h1>

<h4>Ingresa datos del usuario</h4>

<hr />

<partial name="_Mensajes"/>

<div class="row">

<div class="col-md-4">

<form asp-action="Create">

<div asp-validation-summary="ModelOnly" class="text-danger"></div>

<div class="form-group">

<label asp-for="Email" class="control-label"></label>

<input asp-for="Email" class="form-control" />

</div>

<div class="form-group">

<label asp-for="Nombre" class="control-label"></label>

<input asp-for="Nombre" class="form-control" />

</div>

<div class="form-group">

<label asp-for="Apellido" class="control-label"></label>

```
<input asp-for="Apellido" class="form-control" />
```

```
<span asp-validation-for="Apellido" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<label asp-for="Contrasenia" class="control-label">Contraseña</label>
```

```
<input asp-for="Contrasenia" class="form-control" />
```

```
<span asp-validation-for="Contrasenia" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="submit" value="Guardar usuario" class="btn btn-primary" />
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<a asp-action="Index">Lista</a>
```

```
</div>
```

```
@section Scripts {
```

```
@{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
```

```
}
```

Archivo: Delete.cshtml

Localizacion: Papeleria.MVC\Views\Usuarios\Delete.cshtml

@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios.UsuarioDTO

<h3>Estas seguro que deseas eliminar el usuario?</h3>

<partial name="_Mensajes" />

<div>

<h4>UsuarioDTO</h4>

<hr />

<dl class="row">

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Id)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Id)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Email)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Email)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Nombre)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Nombre)

</dd>

<dt class = "col-sm-2">

@Html.DisplayNameFor(model => model.Apellido)

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Apellido)

</dd>

<dt class = "col-sm-2">

Contraseña

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Contrasenia)

</dd>

</dl>

<form asp-action="Delete">

<input type="submit" value="Delete" class="btn btn-danger" /> |

```
@Html.CheckBox("IsChecked") <span>Entiendo que esta operación es riesgosa</span>
```

```
<a asp-action="Index">Volver a la lista</a>
```

```
</form>
```

```
</div>
```

```
*****
```

Archivo: Edit.cshtml

Localizacion: Papeleria.MVC\Views\Usuarios\Edit.cshtml

```
*****
```

```
@model Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios.UsuarioDTO
```

```
@ {
```

```
    ViewData["Title"] = "Edit";
```

```
}
```

```
<h1>Editar Usuario</h1>
```

```
<hr />
```

```
<partial name="_Mensajes" />
```

```
<div class="row">
```

```
    <div class="col-md-4">
```

```
        <form asp-action="Edit">
```

```
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
<div class="form-group">
```

```
  <label asp-for="Id" class="control-label"></label>
```

```
  <input asp-for="Id" class="form-control" readonly/>
```

```
  <span asp-validation-for="Id" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label asp-for="Email" class="control-label"></label>
```

```
  <input asp-for="Email" class="form-control" readonly/>
```

```
  <span asp-validation-for="Email" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label asp-for="Nombre" class="control-label"></label>
```

```
  <input asp-for="Nombre" class="form-control" />
```

```
  <span asp-validation-for="Nombre" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label asp-for="Apellido" class="control-label"></label>
```

```
  <input asp-for="Apellido" class="form-control" />
```

```
  <span asp-validation-for="Apellido" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label asp-for="Contrasenia" class="control-label"></label>
```

```
  <input asp-for="Contrasenia" class="form-control" />
```

```
<span asp-validation-for="Contrasenia" class="text-danger"></span>
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="submit" value="Guardar" class="btn btn-primary" />
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<a asp-action="Index">Lista</a>
```

```
</div>
```

```
@section Scripts {
```

```
@{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
```

```
}
```

```
*****
```

Archivo: Index.cshtml

Localizacion: Papeleria.MVC\Views\Usuarios\Index.cshtml

```
*****
```

```
@model IEnumerable<Papeleria.LogicaAplicacion.DataTransferObjects.Dtos.Usuarios.UsuarioDTO>
```

```
@ {  
  
    ViewData["Title"] = "Index";  
  
}
```

```
<h1>Lista de usuarios</h1>
```

```
<p>
```

```
    <a asp-action="Create">CREAR USUARIO</a>
```

```
</p>
```

```
<partial name="_Mensajes" />
```

```
<table class="table">
```

```
    <thead>
```

```
        <tr>
```

```
            <th>
```

```
                @Html.DisplayNameFor(model => model.Id)
```

```
            </th>
```

```
            <th>
```

```
                @Html.DisplayNameFor(model => model.Email)
```

```
            </th>
```

```
            <th>
```

```
                @Html.DisplayNameFor(model => model.Nombre)
```

```
            </th>
```

```
            <th>
```

```
@Html.DisplayNameFor(model => model.Apellido)
```

```
</th>
```

```
<th>
```

```
Contraseña
```

```
</th>
```

```
<th></th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
@foreach (var item in Model)
```

```
{
```

```
<tr>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.Id)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.Email)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.Nombre)
```

```
</td>
```

```
<td>
```

```
@Html.DisplayFor(modelItem => item.Apellido)
```

```

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.Contrasenia)

        </td>

        <td>

            @Html.ActionLink("Editar", "Edit", new { id = item.Id }) |

            @Html.ActionLink("Borrar", "Delete", new { id = item.Id })

        </td>

    </tr>

}

</tbody>

</table>

```

Archivo: Create.cshtml

Localizacion: Papeleria.WebMvc\Views\Articulos\Create.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Delete.cshtml

Localizacion: Papeleria.WebMvc\Views\Articulos\Delete.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Details.cshtml

Localizacion: Papeleria.WebMvc\Views\Articulos\Details.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Edit.cshtml

Localizacion: Papeleria.WebMvc\Views\Articulos\Edit.cshtml

@page

@model Papeleria.WebMvc.Views.Articulos.EditModel

@ {

}

Archivo: Index.cshtml

Localizacion: Papeleria.WebMvc\Views\Articulos\Index.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@ {

}

Archivo: Create.cshtml

Localizacion: Papeleria.WebMvc\Views\Clientes\Create.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Delete.cshtml

Localizacion: Papeleria.WebMvc\Views\Clientes\Delete.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Details.cshtml

Localizacion: Papeleria.WebMvc\Views\Clientes\Details.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Edit.cshtml

Localizacion: Papeleria.WebMvc\Views\Clientes\Edit.cshtml

@page

@model Papeleria.WebMvc.Views.Articulos.EditModel

@{

}

Archivo: Index.cshtml

Localizacion: Papeleria.WebMvc\Views\Clientes\Index.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Index.cshtml

Localizacion: Papeleria.WebMvc\Views\Home\Index.cshtml

@ {

 ViewData["Title"] = "Home Page";

}

<div class="text-center">

 <h1 class="display-4">Welcome</h1>

 <p>Learn about building Web apps with ASP.NET Core.</p>

</div>

Archivo: Privacy.cshtml

Localizacion: Papeleria.WebMvc\Views\Home\Privacy.cshtml

@ {

 ViewData["Title"] = "Privacy Policy";

}

<h1>@ViewData["Title"]</h1>

<p>Use this page to detail your site's privacy policy.</p>

Archivo: Create.cshtml

Localizacion: Papeleria.WebMvc\Views\Pedidos\Create.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@ {

}

Archivo: Delete.cshtml

Localizacion: Papeleria.WebMvc\Views\Pedidos\Delete.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@ {

}

Archivo: Details.cshtml

Localizacion: Papeleria.WebMvc\Views\Pedidos\Details.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@ {

}

Archivo: Edit.cshtml

Localizacion: Papeleria.WebMvc\Views\Pedidos\Edit.cshtml

@page

@model Papeleria.WebMvc.Views.Articulos.EditModel

@ {

}

Archivo: Index.cshtml

Localizacion: Papeleria.WebMvc\Views\Pedidos\Index.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: _Error.cshtml

Localizacion: Papeleria.WebMvc\Views\Shared_Error.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: _Layout.cshtml

Localizacion: Papeleria.WebMvc\Views\Shared_Layout.cshtml

<!DOCTYPE html>

<html lang="en">

<head>


```
<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Papeleria.WebMvc</title>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

<link rel="stylesheet" href="~/Papeleria.WebMvc.styles.css" asp-append-version="true" />

</head>

<body>

    <header>

        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom
        box-shadow mb-3">

            <div class="container-fluid">

                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
                action="Index">Papeleria.WebMvc</a>

                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
                target=".navbar-collapse" aria-controls="navbarSupportedContent"

                    aria-expanded="false" aria-label="Toggle navigation">

                    <span class="navbar-toggler-icon"></span>

                </button>

                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

                    <ul class="navbar-nav flex-grow-1">

                        <li class="nav-item">

                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-
                            action="Index">Home</a>

                        </li>
```

```
<li class="nav-item">

    <a    class="nav-link    text-dark"    asp-area=""    asp-controller="Home"    asp-
action="Privacy">Privacy</a>

</li>

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

    <main role="main" class="pb-3">

        @RenderBody()

    </main>

</div>


<footer class="border-top footer text-muted">

    <div class="container">

        &copy; 2024 - Papeleria.WebMvc - <a asp-area="" asp-controller="Home" asp-
action="Privacy">Privacy</a>

    </div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

<script src="~/js/site.js" asp-append-version="true"></script>
```

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

Archivo: _Mensajes.cshtml

Localizacion: Papeleria.WebMvc\Views\Shared_Mensajes.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: _ValidationScriptsPartial.cshtml

Localizacion: Papeleria.WebMvc\Views\Shared_ValidationScriptsPartial.cshtml

<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>

Archivo: Create.cshtml

Localizacion: Papeleria.WebMvc\Views\Usuarios\Create.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Delete.cshtml

Localizacion: Papeleria.WebMvc\Views\Usuarios\Delete.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

Archivo: Details.cshtml

Localizacion: Papeleria.WebMvc\Views\Usuarios\Details.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@ {

}

Archivo: Edit.cshtml

Localizacion: Papeleria.WebMvc\Views\Usuarios\Edit.cshtml

@page

@model Papeleria.WebMvc.Views.Articulos.EditModel

@ {

}

Archivo: Index.cshtml

Localizacion: Papeleria.WebMvc\Views\Usuarios\Index.cshtml

@*

For more information on enabling MVC for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

*@

@{

}

ENCRIPTACION DE CONTRASEÑA

```
public string Encriptar(string contrasenia)
{
    using (SHA256 sha256Hash = SHA256.Create())
    {
        byte[] bytes = sha256Hash.ComputeHash(Encoding.UTF8.GetBytes(contrasenia));

        StringBuilder builder = new StringBuilder();
        for (int i = 0; i < bytes.Length; i++)
        {
            builder.Append(bytes[i].ToString("x2"));
        }
        return builder.ToString();
    }
}
```

Este método en C# llamado Encriptar toma una contraseña de tipo string (contrasenia) y calcula su hash SHA-256. Esta es una práctica común para almacenar contraseñas de forma segura.

Aquí tienes una explicación detallada de lo que hace:

Crea una instancia de SHA256 usando SHA256.Create().

Calcula el hash de la contraseña de entrada convirtiendo primero la cadena de contraseña en bytes usando la codificación UTF-8 (Encoding.UTF8.GetBytes(contrasenia)), y luego calcula el hash usando sha256Hash.ComputeHash().

Itera sobre los bytes del hash y convierte cada byte a su representación hexadecimal usando el especificador de formato "x2" con ToString(), y lo añade a un StringBuilder.

Finalmente, devuelve la representación de cadena hexadecimal del hash.

Este método básicamente convierte una contraseña en texto plano en una cadena de hash, que es una forma más segura de almacenar y comparar contraseñas. Al almacenar contraseñas en una base de datos, normalmente se almacena la versión hash y se compara la versión hash de la contraseña de entrada durante el inicio de sesión o autenticación en lugar de almacenar o comparar contraseñas en texto plano directamente.

Obtenido de “ChatGPT”.