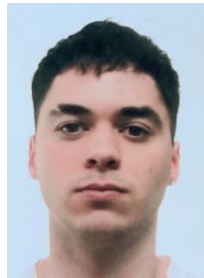


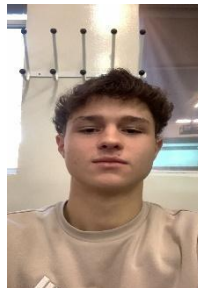
Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

Obligatorio Programación 3

Documento análisis



[Ismael De Hoyos - 333473]



[Joaquín Bellora - 308898]

[M3A]

Docente: [Lilliana Pino]

[Analista en Tecnologías de la Información]

Casos de Uso Narrativos

Identificador: CU-1	Nombre: Alta envío
Autor:	Ismael De Hoyos y Joaquín Bellora
Fecha:	18/5/2025
Descripción: Permite registrar un nuevo envío en el sistema.	
Actor/es: Administrador, funcionario, Cliente	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1-El Actor solicita la creación de una carrera 2-El Actor completa los datos del envío: número de tracking, peso, destino, cliente, etc. 3-El sistema valida los datos ingresados. 4-El sistema guarda el nuevo envío en la base de datos con estado "En Proceso".	
Flujo/s Alternativo/s: <ul style="list-style-type: none">Si el número de tracking ya existe, el sistema muestra un mensaje de error y solicita corregirlo.	
Flujo/s Excepcionales/s: Se interrumpe la comunicación con el servidor. Los datos no son guardados	
Pos condiciones: Se crea el nuevo envío y queda registrado en el sistema listo para su seguimiento.	

Identificador: CU-2	Nombre: Finalizar envío
Autor:	Ismael De Hoyos y Joaquín Bellora
Fecha:	18/5/2025
Descripción: Permite finalizar un envío	
Actor/es: Administrador, funcionario, Cliente	
Precondiciones: <ul style="list-style-type: none"> El actor debe estar logueado en el sistema 	
Flujo Normal: <ol style="list-style-type: none"> 1- El sistema muestra una lista con todos los envíos. 2- El usuario selecciona un envío de la lista para finalizar. 3- El usuario indica que el envío fue entregado y proporciona la fecha de entrega. 4- El sistema valida que el envío esté en estado "En proceso" y que la fecha de entrega sea válida. 5- El sistema actualiza el estado del envío a "Finalizado" y guarda la fecha de entrega. 	
Flujo/s Alternativo/s: <ul style="list-style-type: none"> Si el envío seleccionado ya está finalizado, el sistema muestra un mensaje indicando que no puede finalizarse nuevamente. 	
Flujo/s Excepcionales/s: <p>Se interrumpe la comunicación con el servidor. Los datos no son guardados</p>	
Pos condiciones: <ul style="list-style-type: none"> El envío queda marcado como finalizado con la fecha de entrega registrada. 	

Diagrama Caso de Uso



Código Fuente

Archivo: DemoContext.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaAccesoDatos\DemoContext.cs

```
using LogicaNegocio.EntidadesNegocio;
```

```
using Microsoft.EntityFrameworkCore;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaAccesoDatos
```

```
{
```

```
    public class DemoContext : DbContext
```

```
    {
```

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

```
        public DbSet<Envio> Envios { get; set; }
```

```
        public DbSet<Urgente>Urgente { get; set; }
```

```
        public DbSet<Comun> Comun { get; set; }
```

```
        public DbSet<Agencia> Agencias { get; set; }
```

```
        public DbSet<Seguimiento> Seguimientos { get; set; }
```

```

        public DemoContext(DbContextOptions options) : base(options)
        {

        }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {

            modelBuilder.Entity<Envio>().HasOne(e => e.Funcionario)
                .WithMany()
                .OnDelete(DeleteBehavior.NoAction);

            modelBuilder.Entity<Envio>().HasOne(e => e.Cliente)
                .WithMany()
                .OnDelete(DeleteBehavior.NoAction);

            base.OnModelCreating(modelBuilder);
        }
    }
}

```

Archivo: ImpresionCodigoFuenteSolucion.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImpresionCodigoFuenteSolucion.cs

using System.Text;

using System.IO;

```

namespace ImprimirCodigoFuente
{
    /// <summary>
    /// Para utilizar este código:
    /// 1. Agregar un proyecto aplicación de consola en la solución que tiene el código fuente a
    imprimir
    /// 2. Sustituir el código de la clase Program por este código
    /// 3. Configurar el proyecto de consola como proyecto de inicio
    /// 4. Ejecutar la aplicación
    /// 5. En la misma carpeta de la clase Program quedarán los archivos de texto con el código
    fuente
    /// </summary>
    internal class ImpresionCodigoFuenteSolucion
    {
        /// <summary>
        /// Imprime los archivos de código con extensión .cs y las views .cshtml.
        /// Para agregar otro tipo de archivo simplemente invocar al método Imprimir
        /// indicando *.extensión
        /// </summary>
        /// <param name="args"></param>
        static void Main(string[] args)
        {
            Imprimir("*.cs", "fuentesCs.txt");
            Imprimir("*.cshtml", "views.txt");
        }
        /// <summary>
        /// Imprime los archivos de código fuente de la solución
        /// </summary>
        /// <param name="tipoArchivo">El nombre del archivo a imprimir. Para imprimir todos los
        de un tipo determinado usar "*.extensión" por ejemplo: "*.cs"</param>

```

```
/// <param name="nombreArchivoSalida">El nombre del archivo de texto donde quedará  
el código fuente</param>
```

```
/// <remarks>
```

```
/// Este código funciona siempre que el archivo de la solución (.sln) esté en la raíz de la  
solución,
```

```
/// es decir cuando todos los proyectos están en subcarpetas de la carpeta de la solución
```

```
/// </remarks>
```

```
private static void Imprimir(string tipoArchivo, string nombreArchivoSalida)
```

```
{
```

```
    try
```

```
    {
```

```
        string raizSolucion = ObtenerRutaSolucion();
```

```
        var separador = "*****" + Environment.NewLine;
```

```
        var archivos = System.IO.Directory.GetFiles(raizSolucion, tipoArchivo,  
System.IO.SearchOption.AllDirectories);
```

```
        //se obtienen los archivos .cs excluyendo los que contienen código generado por el  
framework
```

```
        var resultado = archivos.Where(p => !p.Contains("Temporary")
```

```
&& !p.Contains("AssemblyInfo.cs")
```

```
&& !p.Contains("Program.cs")
```

```
        && !p.Contains("AssemblyAttributes")
```

```
&& !p.Contains(".g.cs"))
```

```
        .Select(path => new { Carpeta = path, Nombre = System.IO.Path.GetFileName(path),  
Contenido = System.IO.File.ReadAllText(path) })
```

```
        .Select(info =>
```

```
            separador
```

```
            + "Archivo: " + info.Nombre + Environment.NewLine
```

```
            + "Carpeta: " + info.Carpeta + Environment.NewLine
```

```
            + separador
```

```
            + info.Contenido);
```



```

        var concatenado = string.Join(Environment.NewLine, resultado);

File.WriteAllText(Path.GetDirectoryName(Path.GetDirectoryName(Path.GetDirectoryName(Dire
ctory.GetCurrentDirectory()))
        + @$"\{nombreArchivoSalida}", concatenado, Encoding.UTF8);
    }
    catch (Exception algunError)
    {
        Console.WriteLine(algunError.Message);
    }
}

static string ObtenerRutaSolucion()
{
    string directorioActivo = Directory.GetCurrentDirectory();

    Console.WriteLine("Directorio activo: " + directorioActivo);

    // Navega hacia arriba en la estructura de directorios hasta encontrar la carpeta de la
    solución

    DirectoryInfo directoryInfo = new DirectoryInfo(directorioActivo);

    while (directoryInfo != null && !DirectorioIncluye(directoryInfo, "*.sln"))
    {
        directoryInfo = directoryInfo.Parent;
    }

    if (directoryInfo != null)
    {
        return directoryInfo.FullName;
    }
    else

```

```

        {
            return string.Empty;
        }
    }

    static bool DirectorioIncluye(DirectoryInfo directory, string pattern)
    {
        return directory.GetFiles(pattern).Length > 0;
    }
}

*****

Archivo: WeatherForecast.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\WebAPI\WeatherForecast.cs

*****

namespace WebAPI
{
    public class WeatherForecast
    {
        public DateOnly Date { get; set; }

        public int TemperatureC { get; set; }

        public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);

        public string? Summary { get; set; }
    }
}

```

Archivo: AgenciaMapper.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\Compartido\Mappers\AgenciaMapper.cs

using Compartido.DTOs.Agencia;

using LogicaNegocio.EntidadesNegocio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Compartido.Mappers

{

 public class AgenciaMapper

 {

 public static IEnumerable<ListadoAgenciasDTO>
ListadoAgenciaToListadoAgenciaDTO

 (IEnumerable<Agencia> agencias)

 {

 return agencias.Select(c => new ListadoAgenciasDTO()

 {

 Id = c.Id,

 Nombre = c.Nombre,

 DireccionPostal = c.DireccionPostal,

 Ubicacion = c.Ubicacion

 }).ToList();

 }

 }

```
}
```

```
*****
```

Archivo: EnvioMapper.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\Compartido\Mappers\EnvioMapper.cs

```
*****
```

```
using Compartido.DTOs.Envios;
```

```
using Compartido.DTOs.Seguimiento;
```

```
using LogicaNegocio.EntidadesNegocio;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Compartido.Mappers
```

```
{
```

```
    public class EnvioMapper
```

```
    {
```

```
        public static Comun ComunFromDTO(AltaEnvioComunDTO comunDTO, Usuario  
funcionario, Usuario cliente, Agencia agencia)
```

```
        {
```

```
            return new Comun(comunDTO.NumeroTracking,
```

```
                funcionario,
```

```
                cliente,
```

```
                comunDTO.Peso,
```

```
                Estado.En_Proceso,
```

```
                new List<Seguimiento>(),
```

```
                agencia);
```

```

    }

    public static Urgente UrgenteFromDTO(AltaEnvioUrgenteDTO urgenteDTO,
    Usuario funcionario, Usuario cliente)
    {

        return new Urgente(urgenteDTO.NumeroTracking,
            funcionario,
            cliente,
            urgenteDTO.Peso,
            urgenteDTO.Estado,
            new List<Seguimiento>(),
            urgenteDTO.DireccionPostal,
            urgenteDTO.Entregado,
            urgenteDTO.TiempoEntrega);

    }

    public static ListadoEnviosDTO EnvioToListadoDTO
        (Envio envio)
    {
        return new ListadoEnviosDTO()
        {
            Id = envio.Id,
            NroTracking = envio.NroTracking,
            Peso = envio.Peso,
            FuncionarioNombre = envio.Funcionario?.Nombre ?? "Sin
funcionario",
            ClienteNombre = envio.Cliente?.Nombre ?? "Sin cliente",
            Estado = envio.Estado,
            FechaEntrega = envio.FechaEntrega ?? DateTime.MinValue
        };
    }

```

```

    }

    public static IEnumerable<ListadoEnviosDTO> ListadoEnvioToListadoEnvioDTO
        (IEnumerable<Envio> envios)
    {
        IEnumerable<ListadoEnviosDTO> enviosDTOs = new
List<ListadoEnviosDTO>();

        enviosDTOs = envios.Select(c => new ListadoEnviosDTO()
        {
            Id = c.Id,
            Peso = c.Peso,
            FuncionarioNombre = c.Funcionario.Nombre,
            ClienteNombre = c.Cliente.Nombre,
            Estado = c.Estado,
            FechaEntrega = c.FechaEntrega != null ?
(DateTime)c.FechaEntrega : DateTime.MinValue

        });

        return enviosDTOs;
    }

    public static Seguimiento SeguimientoDTOTOseguimiento(SeguimientoDTO
seguimientoDTO, Usuario usuario)
    {
        if(seguimientoDTO == null)
        {
            throw new ArgumentNullException("Datos Incorrectos");
        }

        return new Seguimiento(seguimientoDTO.Comentario,

seguimientoDTO.FechaHora,seguimientoDTO.Envioid,seguimientoDTO.UsuarioId);
    }

```

```
    }  
  
    }  
}
```

Archivo: UsuarioMapper.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\Compartido\Mappers\UsuarioMapper.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using Compartido.DTOs.Usuario;  
using LogicaNegocio.EntidadesNegocio;  
using static System.Runtime.InteropServices.JavaScript.JSType;  
  
namespace Compartido.Mappers  
{  
    public class UsuarioMapper  
    {  
        public static Usuario UsuarioFromUsuarioDTO(UsuarioDTO usuarioDTO)  
        {  
            if(usuarioDTO == null)  
            {  
                throw new ArgumentNullException("Datos Incorrectos");  
            }  
        }  
    }  
}
```

```

        return new Usuario(usuarioDTO.Ci,
                            usuarioDTO.Nombre,
                            usuarioDTO.Apellido,
                            usuarioDTO.Email,
                            usuarioDTO.Contraseña,

(LogicaNegocio.EntidadesNegocio.Rol)usuarioDTO.Rol)

        ;
    }

    public static IEnumerable<ListadoUsuariosDTO>
ListadoUsuarioToListadoUsuarioDTO

        (IEnumerable<Usuario> usuarios)

    {

        return usuarios.Select(c => new ListadoUsuariosDTO()

        {

            Id = c.Id,
            Ci = c.Ci.Valor,
            Nombre = c.Nombre,
            Apellido = c.Apellido,
            Email = c.Email,
            Contraseña = c.Contraseña,
            rol = (DTOs.Usuario.Rol)c.Rol

        }).ToList();

    }

    public static ListadoUsuariosDTO UsuarioToListadoUsuarioDTO

        (Usuario usuario)

    {

        return new ListadoUsuariosDTO()

        {

```



```

        Id = usuario.Id,
        Ci = usuario.Ci.Valor,
        Nombre = usuario.Nombre,
        Apellido = usuario.Apellido,
        Email = usuario.Email,
        Contraseña = usuario.Contraseña,
        rol = (DTOs.Usuario.Rol)usuario.Rol
    };
}

public static Usuario UsuarioFromListadoUsuarioDTO(ListadoUsuariosDTO
usuarioDTO)
{
    if(usuarioDTO == null)
    {
        throw new ArgumentException("Datos Incorrectos");
    }
    return new Usuario(
        usuarioDTO.Ci,
        usuarioDTO.Nombre,
        usuarioDTO.Apellido,
        usuarioDTO.Email,
        usuarioDTO.Contraseña,
        (LogicaNegocio.EntidadesNegocio.Rol)usuarioDTO.rol);
}

public static UsuarioDTO UsuarioToUsuarioDTO(Usuario usuario)
{
    if (usuario == null)

```

```

        {
            throw new ArgumentNullException("El usuario no puede ser
nulo");
        }

        return new UsuarioDTO
        {
            Id = usuario.Id,
            Ci = usuario.Ci.Valor,
            Nombre = usuario.Nombre,
            Apellido = usuario.Apellido,
            Email = usuario.Email,
            Contraseña = usuario.Contraseña,
            Rol = (DTOs.Usuario.Rol)usuario.Rol
        };
    }
}
}

```

Archivo: 20250516205817_init.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Migrations\20250516205817_init.cs

using System;

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace LogicaAccesoDatos.Migrations

{

/// <inheritdoc />

```

public partial class init : Migration
{
    /// <inheritdoc />
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.CreateTable(
            name: "Ubicacion",
            columns: table => new
            {
                Id = table.Column<int>(type: "int", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                Latitud = table.Column<double>(type: "float", nullable: false),
                Longitud = table.Column<double>(type: "float", nullable: false)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_Ubicacion", x => x.Id);
            });

        migrationBuilder.CreateTable(
            name: "Usuarios",
            columns: table => new
            {
                Id = table.Column<int>(type: "int", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Apellido = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Email = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Contraseña = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Rol = table.Column<int>(type: "int", nullable: false),
                Ci_Valor = table.Column<string>(type: "nvarchar(max)", nullable: false)
            }
        );
    }
}

```

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

```
migrationBuilder.CreateTable(  
    name: "Agencias",  
    columns: table => new  
    {  
        Id = table.Column<int>(type: "int", nullable: false)  
            .Annotation("SqlServer:Identity", "1, 1"),  
        Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),  
        DireccionPostal = table.Column<string>(type: "nvarchar(max)", nullable: false),  
        UbicacionId = table.Column<int>(type: "int", nullable: false)  
    },  
    constraints: table =>  
    {  
        table.PrimaryKey("PK_Agencias", x => x.Id);  
        table.ForeignKey(  
            name: "FK_Agencias_Ubicacion_UbicacionId",  
            column: x => x.UbicacionId,  
            principalTable: "Ubicacion",  
            principalColumn: "Id",  
            onDelete: ReferentialAction.Cascade);  
    });
```

```
migrationBuilder.CreateTable(  
    name: "Envios",  
    columns: table => new  
    {
```

```

Id = table.Column<int>(type: "int", nullable: false)
    .Annotation("SqlServer:Identity", "1, 1"),
NumeroTracking = table.Column<string>(type: "nvarchar(max)", nullable: false),
Funcionariold = table.Column<int>(type: "int", nullable: false),
Clienteld = table.Column<int>(type: "int", nullable: false),
Peso = table.Column<int>(type: "int", nullable: false),
Estado = table.Column<int>(type: "int", nullable: false),
FechaEntrega = table.Column<DateTime>(type: "datetime2", nullable: true),
Discriminator = table.Column<string>(type: "nvarchar(8)", maxLength: 8, nullable:
false),

Agenciald = table.Column<int>(type: "int", nullable: true),
DireccionPostal = table.Column<string>(type: "nvarchar(max)", nullable: true),
Entregado = table.Column<bool>(type: "bit", nullable: true),
TiempoEntrega = table.Column<int>(type: "int", nullable: true)
},
constraints: table =>
{
    table.PrimaryKey("PK_Envios", x => x.Id);
    table.ForeignKey(
        name: "FK_Envios_Agencias_Agenciald",
        column: x => x.Agenciald,
        principalTable: "Agencias",
        principalColumn: "Id",
        onDelete: ReferentialAction.Cascade);
    table.ForeignKey(
        name: "FK_Envios_Usuarios_Clienteld",
        column: x => x.Clienteld,
        principalTable: "Usuarios",
        principalColumn: "Id");
    table.ForeignKey(
        name: "FK_Envios_Usuarios_Funcionariold",

```

```

        column: x => x.Funcionarioid,
        principalTable: "Usuarios",
        principalColumn: "Id");
    });

```

```

migrationBuilder.CreateTable(
    name: "Seguimientos",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Comentario = table.Column<string>(type: "nvarchar(max)", nullable: false),
        FechaHora = table.Column<DateTime>(type: "datetime2", nullable: false),
        Usuarioid = table.Column<int>(type: "int", nullable: false),
        Envioid = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Seguimientos", x => x.Id);
        table.ForeignKey(
            name: "FK_Seguimientos_Envios_Envioid",
            column: x => x.Envioid,
            principalTable: "Envios",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_Seguimientos_Usuarios_Usuarioid",
            column: x => x.Usuarioid,
            principalTable: "Usuarios",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    }
);

```

```
});
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Agencias_UbicacionId",  
    table: "Agencias",  
    column: "UbicacionId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Envios_AgencialId",  
    table: "Envios",  
    column: "AgencialId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Envios_ClienteId",  
    table: "Envios",  
    column: "ClienteId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Envios_FuncionarioId",  
    table: "Envios",  
    column: "FuncionarioId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Seguimientos_EnvioId",  
    table: "Seguimientos",  
    column: "EnvioId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_Seguimientos_UsuarioId",  
    table: "Seguimientos",  
    column: "UsuarioId");
```

```

    }

    /// <inheritdoc />
    protected override void Down(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.DropTable(
            name: "Seguimientos");

        migrationBuilder.DropTable(
            name: "Envios");

        migrationBuilder.DropTable(
            name: "Agencias");

        migrationBuilder.DropTable(
            name: "Usuarios");

        migrationBuilder.DropTable(
            name: "Ubicacion");
    }
}
}
}

```

Archivo: 20250516205817_init.Designer.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--

308898\LogicaAccesoDatos\Migrations\20250516205817_init.Designer.cs

// <auto-generated />

using System;

using System.Collections.Generic;


```

using LogicaAccesoDatos;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Infrastructure;

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Migrations;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;


#nullable disable


namespace LogicaAccesoDatos.Migrations
{
    [DbContext(typeof(DemoContext))]
    [Migration("20250516205817_init")]
    partial class init
    {
        /// <inheritdoc />
        protected override void BuildTargetModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "8.0.10")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

            modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Agencia", b =>
            {
                b.Property<int>("Id")
                    .ValueGeneratedOnAdd()
                    .HasColumnType("int");
            });

```

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

```
b.Property<string>("DireccionPostal")  
    .IsRequired()  
    .HasColumnType("nvarchar(max));
```

```
b.Property<string>("Nombre")  
    .IsRequired()  
    .HasColumnType("nvarchar(max));
```

```
b.Property<int>("UbicacionId")  
    .HasColumnType("int");
```

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

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
```

```
{  
    b.Property<int>("Id")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");
```

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

```
b.Property<int>("ClienteId")  
    .HasColumnType("int");
```

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

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

b.Property<DateTime?>("FechaEntrega")
    .HasColumnType("datetime2");

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

b.Property<string>("NumeroTracking")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

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

b.HasKey("Id");

b.HasIndex("Clienteld");

b.HasIndex("Funcionarioid");

b.ToTable("Envios");

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

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

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Seguimiento", b =>
```

```
{
```

```
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

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

```
    b.Property<string>("Comentario")
        .IsRequired()
        .HasColumnType("nvarchar(max)");
```

```
    b.Property<int>("Envioid")
        .HasColumnType("int");
```

```
    b.Property<DateTime>("FechaHora")
        .HasColumnType("datetime2");
```

```
    b.Property<int>("Usuarioid")
        .HasColumnType("int");
```

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

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

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

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

```
});
```

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

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

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

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

    b.HasKey("Id");

    b.ToTable("Ubicacion");
});
```

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

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

    b.Property<string>("Apellido")
        .IsRequired()
```

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

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

```
.IsRequired()
```

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

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

```
.IsRequired()
```

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

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

```
.IsRequired()
```

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

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

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

```
b.ComplexProperty<Dictionary<string, object>>("Ci",  
"LogicaNegocio.EntidadesNegocio.Usuario.Ci#CedulaUsuario", b1 =>
```

```
{
```

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

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

```
        .IsRequired()
```

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

```
});
```

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comun", b =>
{
    b.HasBaseType("LogicaNegocio.EntidadesNegocio.Envio");

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

    b.HasIndex("AgencialId");

    b.HasDiscriminator().HasValue("Comun");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Urgente", b =>
{
    b.HasBaseType("LogicaNegocio.EntidadesNegocio.Envio");

    b.Property<string>("DireccionPostal")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

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

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

    b.HasDiscriminator().HasValue("Urgente");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Agencia", b =>
```

```

{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Ubicacion", "Ubicacion")
        .WithMany()
        .HasForeignKey("UbicacionId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Ubicacion");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Cliente")
        .WithMany()
        .HasForeignKey("ClientelId")
        .OnDelete(DeleteBehavior.NoAction)
        .IsRequired();

    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Funcionario")
        .WithMany()
        .HasForeignKey("FuncionariolId")
        .OnDelete(DeleteBehavior.NoAction)
        .IsRequired();

    b.Navigation("Cliente");

    b.Navigation("Funcionario");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Seguimiento", b =>
{

```



```
b.HasOne("LogicaNegocio.EntidadesNegocio.Envio", "Envio")
    .WithMany("Seguimientos")
    .HasForeignKey("Envioid")
    .OnDelete(DeleteBehavior.Cascade)
    .IsRequired();
```

```
b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Usuario")
    .WithMany()
    .HasForeignKey("Usuarioid")
    .OnDelete(DeleteBehavior.Cascade)
    .IsRequired();
```

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comun", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Agencia", "Agencia")
        .WithMany()
        .HasForeignKey("Agenciald")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Agencia");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
{
    b.Navigation("Seguimientos");
```

```

        });

#pragma warning restore 612, 618

    }

}

}

*****

Archivo: 20250517181012_init2.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Migrations\20250517181012_init2.cs

*****

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace LogicaAccesoDatos.Migrations
{
    /// <inheritdoc />

    public partial class init2 : Migration
    {
        /// <inheritdoc />

        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.RenameColumn(
                name: "NumeroTracking",
                table: "Envios",
                newName: "NroTracking");
        }

        /// <inheritdoc />

        protected override void Down(MigrationBuilder migrationBuilder)

```

```

    {
        migrationBuilder.RenameColumn(
            name: "NroTracking",
            table: "Envios",
            newName: "NumeroTracking");
    }
}
}

```

Archivo: 20250517181012_init2.Designer.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Migrations\20250517181012_init2.Designer.cs

// <auto-generated />

```

using System;
using System.Collections.Generic;
using LogicaAccesoDatos;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

```

#nullable disable

namespace LogicaAccesoDatos.Migrations

```

{
    [DbContext(typeof(DemoContext))]
    [Migration("20250517181012_init2")]
    partial class init2

```

```

{
    /// <inheritdoc />
    protected override void BuildTargetModel(ModelBuilder modelBuilder)
    {
#pragma warning disable 612, 618
        modelBuilder
            .HasAnnotation("ProductVersion", "8.0.10")
            .HasAnnotation("Relational:MaxIdentifierLength", 128);

        SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

        modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Agencia", b =>
        {
            b.Property<int>("Id")
                .ValueGeneratedOnAdd()
                .HasColumnType("int");

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

            b.Property<string>("DireccionPostal")
                .IsRequired()
                .HasColumnType("nvarchar(max)");

            b.Property<string>("Nombre")
                .IsRequired()
                .HasColumnType("nvarchar(max)");

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

            b.HasKey("Id");

```

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

```
b.ToTable("Agencias");  
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
```

```
{
```

```
    b.Property<int>("Id")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");
```

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

```
    b.Property<int>("ClientId")  
        .HasColumnType("int");
```

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

```
    b.Property<int>("Estado")  
        .HasColumnType("int");
```

```
    b.Property<DateTime?>("FechaEntrega")  
        .HasColumnType("datetime2");
```

```
    b.Property<int>("FuncionarioId")  
        .HasColumnType("int");
```

```
b.Property<string>("NroTracking")
    .IsRequired()
    .HasColumnType("nvarchar(max)");
```

```
b.Property<int>("Peso")
    .HasColumnType("int");
```

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

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

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

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

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Seguimiento", b =>
```

```
{
```

```
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");
```

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

```
    b.Property<string>("Comentario")
        .IsRequired()
        .HasColumnType("nvarchar(max)");
```

```
b.Property<int>("Enviold")  
    .HasColumnType("int");
```

```
b.Property<DateTime>("FechaHora")  
    .HasColumnType("datetime2");
```

```
b.Property<int>("Usuariold")  
    .HasColumnType("int");
```

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

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

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

```
b.ToTable("Seguimientos");  
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Ubicacion", b =>  
{  
    b.Property<int>("Id")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");  
  
    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));  
  
    b.Property<double>("Latitud")  
        .HasColumnType("float");  
  
    b.Property<double>("Longitud")
```

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

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Usuario", b =>
```

```
{
```

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

```
        .ValueGeneratedOnAdd()
```

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

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

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

```
        .IsRequired()
```

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

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

```
        .IsRequired()
```

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

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

```
        .IsRequired()
```

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

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

```
        .IsRequired()
```

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



```
b.Property<int>("Rol")
    .HasColumnType("int");
```

```
    b.ComplexProperty<Dictionary<string, object>>("Ci",
        "LogicaNegocio.EntidadesNegocio.Usuario.Ci#CedulaUsuario", b1 =>
        {
            b1.IsRequired();

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

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

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

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comun", b =>
{
    b.HasBaseType("LogicaNegocio.EntidadesNegocio.Envio");

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

    b.HasIndex("AgencialId");

    b.HasDiscriminator().HasValue("Comun");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Urgente", b =>
```

```

{
    b.HasBaseType("LogicaNegocio.EntidadesNegocio.Envio");

    b.Property<string>("DireccionPostal")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

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

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

    b.HasDiscriminator().HasValue("Urgente");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Agencia", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Ubicacion", "Ubicacion")
        .WithMany()
        .HasForeignKey("UbicacionId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Ubicacion");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Cliente")
        .WithMany()

```

```
.HasForeignKey("ClientId")
.OnDelete(DeleteBehavior.NoAction)
.IsRequired();
```

```
b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Funcionario")
    .WithMany()
    .HasForeignKey("FuncionarioId")
    .OnDelete(DeleteBehavior.NoAction)
    .IsRequired();
```

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

```
b.Navigation("Funcionario");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Seguimiento", b =>
{
```

```
    b.HasOne("LogicaNegocio.EntidadesNegocio.Envio", "Envio")
        .WithMany("Seguimientos")
        .HasForeignKey("EnvioId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();
```

```
    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Usuario")
        .WithMany()
        .HasForeignKey("UsuarioId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();
```

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

```

        b.Navigation("Usuario");
    });

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comun", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Agencia", "Agencia")
        .WithMany()
        .HasForeignKey("AgenciaId")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Agencia");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
{
    b.Navigation("Seguimientos");
});

#pragma warning restore 612, 618
}
}
}

*****

Archivo: DemoContextModelSnapshot.cs
Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Migrations\DemoContextModelSnapshot.cs

*****

// <auto-generated />

using System;

using System.Collections.Generic;

```

```
using LogicaAccesoDatos;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
```

```
#nullable disable
```

```
namespace LogicaAccesoDatos.Migrations
```

```
{
    [DbContext(typeof(DemoContext))]
    partial class DemoContextModelSnapshot : ModelSnapshot
    {
        protected override void BuildModel(ModelBuilder modelBuilder)
        {
```

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

```
        modelBuilder
            .HasAnnotation("ProductVersion", "8.0.10")
            .HasAnnotation("Relational:MaxIdentifierLength", 128);
```

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

```
        modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Agencia", b =>
```

```
        {
            b.Property<int>("Id")
                .ValueGeneratedOnAdd()
                .HasColumnType("int");
```

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

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

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

```
b.Property<string>("Nombre")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.Property<int>("UbicacionId")  
.HasColumnType("int");
```

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

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

```
b.ToTable("Agencias");  
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
```

```
{  
    b.Property<int>("Id")  
        .ValueGeneratedOnAdd()  
        .HasColumnType("int");
```

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

```
b.Property<int>("ClientId")  
.HasColumnType("int");
```

```
b.Property<string>("Discriminator")  
.IsRequired()  
.HasMaxLength(8)
```

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

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

b.Property<DateTime?>("FechaEntrega")
    .HasColumnType("datetime2");

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

b.Property<string>("NroTracking")
    .IsRequired()
    .HasColumnType("nvarchar(max)");

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

b.HasKey("Id");

b.HasIndex("ClientelId");

b.HasIndex("FuncionarioId");

b.ToTable("Envios");

b.HasDiscriminator().HasValue("Envio");

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

```

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Seguimiento", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

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

    b.Property<string>("Comentario")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

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

    b.Property<DateTime>("FechaHora")
        .HasColumnType("datetime2");

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

    b.HasKey("Id");

    b.HasIndex("Envioid");

    b.HasIndex("Usuarioid");

    b.ToTable("Seguimientos");
});

```

```

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Ubicacion", b =>

```



```

{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

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

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

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

    b.HasKey("Id");

    b.ToTable("Ubicacion");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Usuario", b =>
{
    b.Property<int>("Id")
        .ValueGeneratedOnAdd()
        .HasColumnType("int");

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

    b.Property<string>("Apellido")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b.Property<string>("Contraseña")

```

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

```
b.Property<string>("Email")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.Property<string>("Nombre")  
.IsRequired()  
.HasColumnType("nvarchar(max)");
```

```
b.Property<int>("Rol")  
.HasColumnType("int");
```

```
b.ComplexProperty<Dictionary<string, object>>("Ci",  
"LogicaNegocio.EntidadesNegocio.Usuario.Ci#CedulaUsuario", b1 =>
```

```
{  
    b1.IsRequired();  
  
    b1.Property<string>("Valor")  
        .IsRequired()  
        .HasColumnType("nvarchar(max)");  
});
```

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

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

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comun", b =>  
{
```

```
b.HasBaseType("LogicaNegocio.EntidadesNegocio.Envio");
```

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

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

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Urgente", b =>
```

```
{
```

```
b.HasBaseType("LogicaNegocio.EntidadesNegocio.Envio");
```

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

```
.IsRequired()
```

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

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

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

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

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

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

```
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Agencia", b =>
```

```
{
```

```
b.HasOne("LogicaNegocio.EntidadesNegocio.Ubicacion", "Ubicacion")
```

```
.WithMany()
```

```
.HasForeignKey("UbacionId")
.OnDelete(DeleteBehavior.Cascade)
.IsRequired();
```

```
b.Navigation("Ubacion");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Cliente")
        .WithMany()
        .HasForeignKey("Clienteld")
        .OnDelete(DeleteBehavior.NoAction)
        .IsRequired();

    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Funcionario")
        .WithMany()
        .HasForeignKey("Funcionariold")
        .OnDelete(DeleteBehavior.NoAction)
        .IsRequired();

    b.Navigation("Cliente");

    b.Navigation("Funcionario");
});
```

```
modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Seguimiento", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Envio", "Envio")
        .WithMany("Seguimientos")
        .HasForeignKey("Enviold")
```

```

        .onDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("LogicaNegocio.EntidadesNegocio.Usuario", "Usuario")
        .WithMany()
        .HasForeignKey("UsuariId")
        .onDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Envio");

    b.Navigation("Usuario");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comun", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Agencia", "Agencia")
        .WithMany()
        .HasForeignKey("AgencialId")
        .onDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.Navigation("Agencia");
});

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Envio", b =>
{
    b.Navigation("Seguimientos");
});

#pragma warning restore 612, 618
}

```

```
}  
}
```

Archivo: RepositorioAgenciaEF.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Repositorios\RepositorioAgenciaEF.cs

```
using LogicaNegocio.EntidadesNegocio;  
using LogicaNegocio.InterfacesRepositorios;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace LogicaAccesoDatos.Repositorios  
{  
    public class RepositorioAgenciaEF : IRepositoryAgencia  
    {  
        private DemoContext Contexto { get; set; }  
  
        public RepositorioAgenciaEF(DemoContext contexto)  
        {  
            Contexto = contexto;  
        }  
        public void Add(Agencia item)  
        {  
            throw new NotImplementedException();  
        }  
    }  
}
```

```

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

        public IEnumerable<Agencia> FindAll()
        {
            return Contexto.Agencias;
        }

        public Agencia FindById(int id)
        {
            return Contexto.Agencias.Find(id);
        }

        public void Update(Agencia item)
        {
            throw new NotImplementedException();
        }
    }
}

```

Archivo: RepositorioEnvioEF.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Repositorios\RepositorioEnvioEF.cs

```

using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using System;

```

```

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaAccesoDatos.Repositorios
{
    public class RepositorioEnvioEF : IRepositorioEnvio
    {

        private DemoContext Contexto { get; set; }

        public RepositorioEnvioEF(DemoContext contexto)
        {
            Contexto = contexto;
        }

        public void Add(Envio item)
        {
            Envio envioEncontrado = FindByNroTracking(item.NroTracking);
            if (envioEncontrado == null)
            {
                Contexto.Envios.Add(item);
                Contexto.SaveChanges();
            }
            else
            {
                throw new Exception("Ya existe un envio con ese numero de
tracking");
            }
        }
    }
}

```



```

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

public IEnumerable<Envio> FindAll()
{
    return Contexto.Envios.Include(e => e.Funcionario).Include(e =>
e.Cliente);
}

public Envio FindById(int id)
{
    return Contexto.Envios.Where(c => c.Id == id).Include(c =>
c.Seguimientos).SingleOrDefault();
}

public void Update(Envio item)
{
    Envio envioEncontrado = FindByNroTracking(item.NroTracking);

    if (envioEncontrado != null || envioEncontrado == null &&
envioEncontrado.NroTracking == item.NroTracking)
    {
        envioEncontrado.Funcionario = item.Funcionario;
        envioEncontrado.FuncionarioId = item.FuncionarioId;
        envioEncontrado.NroTracking = item.NroTracking;
        envioEncontrado.Cliente = item.Cliente;
        envioEncontrado.ClienteId = item.ClienteId;
        envioEncontrado.Estado = item.Estado;
    }
}

```

```

        envioEncontrado.Id = item.Id;

        envioEncontrado.NroTracking = item.NroTracking;

        envioEncontrado.Peso = item.Peso;

        envioEncontrado.Seguimientos = item.Seguimientos;

        envioEncontrado.FechaEntrega = item.FechaEntrega;

        Contexto.SaveChanges();

    }

}

public Envio FindByNroTracking(string nroTracking)
{
    return Contexto.Envios.Where(e => e.NroTracking ==
nroTracking).Include(e => e.Cliente).Include(e => e.Funcionario).SingleOrDefault();
}

public void FinalizarEnvio(string nro)
{
    Envio envio = FindByNroTracking(nro);
    if (envio != null)
    {
        envio.Estado = Estado.Finalizado;
        envio.FechaEntrega = DateTime.Now;
        Contexto.SaveChanges();
    }
    else
    {
        throw new Exception("No se encontró el envío con el Numero
de Tracking proporcionado.");
    }
}

}

}

```

Archivo: RepositorioUsuarioEF.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAccesoDatos\Repositorios\RepositorioUsuarioEF.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using LogicaNegocio.InterfacesRepositorios;

using LogicaNegocio.EntidadesNegocio;

using LogicaNegocio.ExcepcionesEntidades;

namespace LogicaAccesoDatos.Repositorios

{

 public class RepositorioUsuarioEF : IRepositoryUsuario

 {

 private DemoContext Contexto { get; set; }

 public RepositorioUsuarioEF(DemoContext contexto)

 {

 Contexto = contexto;

 }

 public void Add(Usuario item)

 {

 Usuario usuarioEncontrado = FindByCi(item.Ci.Valor);

 if (usuarioEncontrado == null)

 {

```
        Contexto.Usuarios.Add(item);
        Contexto.SaveChanges();
    }
    else
    {
        throw new UsuarioException("Ya existe un usuario con esa
Cedula");
    }
}
```

```
public void Delete(int id)
{
    Usuario usuarioBuscado = FindById(id);
    if (usuarioBuscado != null)
    {
        Contexto.Usuarios.Remove(usuarioBuscado);
        Contexto.SaveChanges();
    }
    else
    {
        throw new UsuarioException("No existe un Usuario con esa
Cedula");
    }
}
```

```
public IEnumerable<Usuario> FindAll()
{
    return Contexto.Usuarios;
```

```
}
```

```

public void Update(Usuario item)
{

    Usuario usuarioBuscado = FindByCi(item.Ci.Valor);

    if (usuarioBuscado != null || usuarioBuscado == null &&
usuarioBuscado.Ci == item.Ci)
    {

        usuarioBuscado.Nombre = item.Nombre;
        usuarioBuscado.Apellido = item.Apellido;
        usuarioBuscado.Email = item.Email;
        usuarioBuscado.Contraseña = item.Contraseña;
        usuarioBuscado.Rol = item.Rol;

        Contexto.SaveChanges();
    }
}

```

```

}

public Usuario FindById(int id)
{

    return Contexto.Usuarios.Find(id);

}

```

```

private Usuario FindByCi(string cedulaUsuario)
{

    return Contexto.Usuarios.Where(c => c.Ci.Valor ==
cedulaUsuario).SingleOrDefault();

}

```

```

private Usuario FindByEmail(string email)

```

```

        {
            return Contexto.Usuarios.Where(c => c.Email ==
email).SingleOrDefault();
        }

    Usuario IRepositorioUsuario.FindByEmail(string email)
    {
        return FindByEmail(email);
    }

    public IEnumerable<Usuario> FindByRolFuncionario()
    {
        return Contexto.Usuarios.Where(c =>c.Rol == Rol.Funcionario || c.Rol
== Rol.Administrador).ToList();
    }

    public IEnumerable<Usuario> FindByRolCliente()
    {
        return Contexto.Usuarios.Where(c => c.Rol == Rol.Cliente).ToList();
    }
}
}

```

Archivo: Agencia.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Agencia.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```
using LogicaNegocio.ExcepcionesEntidades;
```

```
namespace LogicaNegocio.EntidadesNegocio
```

```
{
```

```
    public class Agencia
```

```
    {
```

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

```
        public string Nombre { get; set; }
```

```
        public string DireccionPostal { get; set; }
```

```
        public Ubicacion Ubicacion { get; set; }
```

```
        public Agencia(string nombre, string direccionPostal, Ubicacion ubicacion)
```

```
        {
```

```
            Nombre = nombre;
```

```
            DireccionPostal = direccionPostal;
```

```
            Ubicacion = ubicacion;
```

```
        }
```

```
        protected Agencia() { }
```

```
    }
```

```
}
```

```
*****
```

Archivo: Comun.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Comun.cs

```
*****
```

```
using LogicaNegocio.ExcepcionesEntidades;
```

```
using System;
```

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

```
using System.Linq;
```

```

using System.Text;

using System.Threading.Tasks;


namespace LogicaNegocio.EntidadesNegocio
{
    public class Comun : Envio
    {
        public Agencia Agencia { get; set; }


        public Comun(string numeroTracking, Usuario funcionario, Usuario cliente, int
peso, Estado estado, List<Seguimiento> seguimientos, Agencia agencia)
            : base( numeroTracking, funcionario, cliente, peso, estado)
        {
            Agencia = agencia;
        }


        protected Comun() { }
    }
}

```

Archivo: Envio.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Envio.cs

```

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Text;

```



```

using System.Threading.Tasks;

using LogicaNegocio.ExcepcionesEntidades;

namespace LogicaNegocio.EntidadesNegocio
{
    public abstract class Envio
    {

        public int Id { get; set; }

        public string NroTracking { get; set; }

        public Usuario Funcionario { get; set; }
        public int Funcionarioid { get; set; }
        public Usuario Cliente { get; set; }
        public int Clienteid { get; set; }

        public int Peso { get; set; }

        public Estado Estado { get; set; }

        public IEnumerable<Seguimiento> Seguimientos { get; set; }

        public DateTime? FechaEntrega { get; set; }
        protected Envio() { }

        public Envio(string nroTracking, Usuario funcionario, Usuario cliente, int peso, Estado
estado)
        {
            NroTracking = nroTracking;
            Funcionario = funcionario;

```

```

        Cliente = cliente;

        Peso = peso;

        Estado = estado;
    }
}

    public enum Estado
    {

        En_Proceso,

        Finalizado

    }
}

```

Archivo: Seguimiento.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Seguimiento.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaNegocio.EntidadesNegocio
{
    public class Seguimiento
    {
        public int Id { get; set; }

        public string Comentario { get; set; }
    }
}

```

```

public DateTime FechaHora { get; set; }

        public Usuario Usuario { get; set; }

        public int Usuariold { get; set; }

        public Envio Envio { get; set; }

        public int Enviold { get; set; }

        public Seguimiento(string comentario, DateTime fechaHora, int enviold, int
usuariold)
        {

            Comentario = comentario;

            FechaHora = fechaHora;

            Enviold = enviold;

            Usuariold = usuariold;

        }

        private Seguimiento() { }

    }
}

```

Archivo: Ubicacion.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Ubicacion.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

namespace LogicaNegocio.EntidadesNegocio
{
    public class Ubicacion
    {
        public int Id { get; set; }
        public double Latitud { get; set; }
        public double Longitud { get; set; }

        public Ubicacion(double latitud, double longitud)
        {
            Latitud = latitud;
            Longitud = longitud;
        }

        protected Ubicacion() { }
    }
}

```

Archivo: Urgente.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Urgente.cs

```

using LogicaNegocio.ExcepcionesEntidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace LogicaNegocio.EntidadesNegocio
{
    public class Urgente : Envio
    {
        public string DireccionPostal { get; set; }

        public Boolean Entregado { get; set; }

        public int TiempoEntrega { get; set; }

        public Urgente(string numeroTracking, Usuario funcionario, Usuario cliente, int
peso, Estado estado, List<Seguimiento> seguimientos, string direccionPostal, bool entregado, int
tiempoEntrega):
            base(numeroTracking, funcionario, cliente, peso, estado)
        {
            DireccionPostal = direccionPostal;

            Entregado = entregado;

            TiempoEntrega = tiempoEntrega;
        }

        protected Urgente() { }
    }
}

```

Archivo: Usuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\EntidadesNegocio\Usuario.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

```

```
using System.Threading.Tasks;

using LogicaNegocio.ExcepcionesEntidades;

using LogicaNegocio.ValueObject;

namespace LogicaNegocio.EntidadesNegocio
{
    public class Usuario : IEquatable<Usuario>
    {

        public int Id { get; set; }

        public CedulaUsuario Ci { get; set; }

        public string Nombre { get; set; }

        public string Apellido { get; set; }

        public string Email { get; set; }

        public string Contraseña { get; set; }

        public Rol Rol { get; set; }


        private Usuario() { }

        public Usuario(string ci, string nombre, string apellido, string email, string
contraseña, Rol rol)
        {

            Ci = new CedulaUsuario(ci);

            Nombre = nombre;

            Apellido = apellido;

            Email = email;

            Contraseña = contraseña;

            Rol = rol;
        }
    }
}
```

```

        }

        public bool Equals(Usuario? other)
        {
            return Ci == other.Ci;
        }
    }

    public enum Rol
    {
        Administrador,
        Funcionario,
        Cliente ,
    }
}

```

Archivo: AgenciaException.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaNegocio\ExcepcionesEntidades\AgenciaException.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.ExcepcionesEntidades
{
    public class AgenciaException : Exception
    {
        public AgenciaException() { }
    }
}

```

```

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

        public AgenciaException(string message, Exception innerException) :
base(message, innerException) { }

    }

}

```

Archivo: EnvioException.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\ExcepcionesEntidades\EnvioException.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaNegocio.ExcepcionesEntidades
{
    public class EnvioException : Exception
    {
        public EnvioException() { }

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

        public EnvioException(string message, Exception innerException) : base(message,
innerException) { }
    }
}

```

Archivo: UsuarioException.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaNegocio\ExcepcionesEntidades\UsuarioException.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaNegocio.ExcepcionesEntidades

{

 public class UsuarioException : Exception

 {

 public UsuarioException() { }

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

 public UsuarioException(string message, Exception innerException) :
base(message, innerException) { }

 }

}

Archivo: IRepository.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaNegocio\InterfacesRepositorios\IRepositorio.cs

using System;

using System.Collections.Generic;

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios
{
    public interface IRepositoryo <T>
    {
        void Add(T item);
        IEnumerable<T> FindAll();
        T FindById(int id);
        void Delete(int id);
        void Update (T item);

    }
}

```

Archivo: IRepositoryoAgencia.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\InterfacesRepositorios\IRepositoryoAgencia.cs

```

using LogicaNegocio.EntidadesNegocio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios

```

```
{  
  
    public interface IRepositoryoAgencia : IRepositoryo<Agencia>  
  
    {  
  
    }  
  
}
```

Archivo: IRepositoryoEnvio.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaNegocio\InterfacesRepositorios\IRepositoryoEnvio.cs

```
using LogicaNegocio.EntidadesNegocio;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;  
  
  
namespace LogicaNegocio.InterfacesRepositorios  
{  
    public interface IRepositoryoEnvio : IRepositoryo<Envio>  
    {  
        Envio FindByNroTracking(string nroTracking);  
  
        void FinalizarEnvio(string nroTracking);  
  
    }  
}
```

Archivo: IRepositoryoUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaNegocio\InterfacesRepositorios\IRepositorioUsuario.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using LogicaNegocio.EntidadesNegocio;
```

```
namespace LogicaNegocio.InterfacesRepositorios
{
    public interface IRepositorioUsuario : IRepositorio<Usuario>
    {
        Usuario FindByEmail(string email);
        IEnumerable<Usuario> FindByRolFuncionario();
        IEnumerable<Usuario> FindByRolCliente();
    }
}
```

Archivo: CedulaUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaNegocio\ValueObject\CedulaUsuario.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
using LogicaNegocio.ExcepcionesEntidades;
```

```
namespace LogicaNegocio.ValueObject
```

```
{
```

```
    [ComplexType]
```

```
    public class CedulaUsuario
```

```
    {
```

```
        public string Valor { get; init; }
```

```
        public CedulaUsuario(string valor)
```

```
        {
```

```
            Valor = valor;
```

```
            Validar();
```

```
        }
```

```
        private void Validar()
```

```
        {
```

```
            if (string.IsNullOrEmpty(Valor))
```

```
            {
```

```
                throw new UsuarioException("La Cedula es obligatoria");
```

```
            }
```

```
            if (Valor.Length < 8)
```

```
            {
```

```
                throw new UsuarioException("La cedula debe ser de 8  
digitos");
```

```
            }
```

```
            if (Valor.Length > 8)
```

```
            {
```

```
                throw new UsuarioException("La cedula debe ser de 8  
digitos");
```

```
            }
```

```
    }  
}  
}
```

Archivo: EnvioController.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Controllers\EnvioController.cs

```
using Compartido.DTOs.Agencia;  
using Compartido.DTOs.Envios;  
using Compartido.DTOs.Usuario;  
using Compartido.DTOs.Seguimiento;  
using LogicaAplicacion.ImplementacionCasosUso.UsuarioCU;  
using LogicaAplicacion.InterfacesCasosUso.AgenciaCU;  
using LogicaAplicacion.InterfacesCasosUso.EnvioCU;  
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;  
using LogicaNegocio.EntidadesNegocio;  
using LogicaNegocio.ExcepcionesEntidades;  
using Microsoft.AspNetCore.Http;  
using Microsoft.AspNetCore.Mvc;  
using MVC.Filtro;  
using MVC.Models.Agencias;  
using MVC.Models.Envios;  
using MVC.Models.Usuarios;  
  
namespace MVC.Controllers  
{  
    [Logueado]  
    public class EnvioController : Controller  
    {
```

```

private IAltaEnvioComun CUALtaEnvioComun { get; set; }
private IAltaEnvioUrgente CUALtaEnvioUrgente { get; set; }

private IListadoEnvios CUListadoEnvios { get; set; }

private IListadoCliente CUListadoCliente { get; set; }

private IListadoAgencias CUListadoAgencia { get; set; }

private IListadoUsuarios CUListaUsuario { get; set; }

private IFinalizarEnvio CUFinalizarEnvio { get; set; }

private IAgregarSeguimiento agregarSeguimiento { get; set; }

public EnvioController(IAltaEnvioComun altaEnvioComun, IAltaEnvioUrgente
altaEnvioUrgente, IListadoEnvios cUListadoEnvios, IFinalizarEnvio
cUFinalizarEnvio, IListadoUsuarios cUListaUsuarios,
IListadoCliente cUListadoClientes, IListadoAgencias listadoAgencias,
IAgregarSeguimiento cUAgregarSeguimiento)
{
    CUALtaEnvioComun = altaEnvioComun;
    CUALtaEnvioUrgente = altaEnvioUrgente;
    CUListadoEnvios = cUListadoEnvios;
    CUFinalizarEnvio = cUFinalizarEnvio;
    CUListaUsuario = cUListaUsuarios;
    CUListadoCliente = cUListadoClientes;
    CUListadoAgencia = listadoAgencias;
    agregarSeguimiento = cUAgregarSeguimiento;
}

// GET: Envio
public ActionResult Index()

```

```

        {
            IEnumerable<ListadoEnviosDTO> enviosDTOs =
CUListadoEnvios.Ejecutar();

            List<ListadoEnviosVM> listadoEnviosVMs =
                new List<ListadoEnviosVM>();

            try
            {

                listadoEnviosVMs = enviosDTOs.Select(c => new
ListadoEnviosVM()

                    {
                        Id = c.Id,
                        Peso = c.Peso,
                        FuncionarioNombre = c.FuncionarioNombre,
                        ClienteNombre = c.ClienteNombre,
                        Estado = c.Estado,
                        FechaEntrega = c.FechaEntrega

                    }).ToList();
            }
            catch (Exception ex)
            {
                ViewBag.Mensaje = "Datos incorrectos";
            }

            return View(listadoEnviosVMs);
        }

// GET: Envio/Details/5
public ActionResult Details(int id)
{
    return View();
}

```



```

    }

    // GET: Envio/Create
    public ActionResult CreateEnvioComun()
    {
        AltaEnvioComunVM comunVM = new AltaEnvioComunVM();

        try
        {
            List<ListadoUsuariosDTO> datosUsuarios =
            CUListaUsuario.Ejecutar().ToList();

            comunVM.Usuarios = datosUsuarios.Select(dtoUsuario => new
            Models.Usuarios.UsuarioViewModel()
            {
                Id = dtoUsuario.Id,
                Nombre = dtoUsuario.Nombre
            }).ToList();

            List<ListadoUsuariosDTO> datosClientes =
            CUListadoCliente.Ejecutar().ToList();

            comunVM.Cliente = datosClientes.Select(dtoUsuario => new
            Models.Usuarios.ClienteViewModel()
            {
                Id = dtoUsuario.Id,
                Nombre = dtoUsuario.Nombre
            }).ToList();

            List < ListadoAgenciasDTO> datosAgencias =
            CUListadoAgencia.Ejecutar().ToList();

            comunVM.Agencia = datosAgencias.Select(dtoAgencia => new
            Models.Agencias.AgenciaViewModel()
            {
                Id = dtoAgencia.Id,
                Nombre = dtoAgencia.Nombre
            })

```

```

        }).ToList();

    }

    catch (EnvioException ex)

    {
        ViewBag.Mensaje = ex.Message;
    }

    catch (Exception ex)

    {
        ViewBag.Mensaje = "Datos incorrectos";
    }

    return View(comunVM);
}

public ActionResult CreateEnvioUrgente()
{
    AltaEnvioUrgenteVM urgenteVM = new AltaEnvioUrgenteVM();

    try
    {

        List<ListadoUsuariosDTO> datosUsuarios =
        CUListaUsuario.Ejecutar().ToList();

        urgenteVM.Usuarios = datosUsuarios.Select(dtoUsuario => new
        Models.Usuarios.UsuarioViewModel()

        {

            Id = dtoUsuario.Id,

            Nombre = dtoUsuario.Nombre

        }).ToList();
    }
}

```

```

        List<ListadoUsuariosDTO> datosClientes =
CUListadoCliente.Ejecutar().ToList();

        urgenteVM.Cliente = datosClientes.Select(dtoUsuario => new
Models.Usuarios.ClienteViewModel()

            {

                Id = dtoUsuario.Id,

                Nombre = dtoUsuario.Nombre

            }).ToList();

```

```

    }

    catch (EnvioException ex)
    {

        ViewBag.Mensaje = ex.Message;

    }

    catch (Exception ex)
    {

        ViewBag.Mensaje = "Datos incorrectos";

    }

    return View(urgenteVM);
}

```

// POST: Envio/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult CreateEnvioComun(AltaEnvioComunVM envioComun)

```

{

    try

    {

        if (ModelState.IsValid)

        {

```

```

        AltaEnvioComunDTO envioDTO = new
AltaEnvioComunDTO()
    {
        NumeroTracking =
envioComun.NumeroTracking,

        Peso = envioComun.Peso,
        Funcionariold = envioComun.Funcionariold,
        Clienteld = envioComun.Clienteld,
        Agenciald = envioComun.Agenciald,
        Estado = Estado.En_Proceso
    };
    CUAltaEnvioComun.Ejecutar(envioDTO);
    return RedirectToAction(nameof(Index));
}

}
catch (UsuarioException ex)
{
    ViewBag.Message = ex.Message;

}
catch (Exception ex)
{
    ViewBag.DatosExcepcion = ex.StackTrace;
    ViewBag.Mensaje = "Error en los datos";
}
envioComun.Usuarios = CUListaUsuario.Ejecutar().Select(dtoUsuario =>
new UsuarioViewModel
{
    Id = dtoUsuario.Id,
    Nombre = dtoUsuario.Nombre
}).ToList();

```

```

        envioComun.Cliente = CUListadoCliente.Ejecutar().Select(dtoUsuario =>
new ClienteViewModel
    {
        Id = dtoUsuario.Id,
        Nombre = dtoUsuario.Nombre
    }).ToList();

```

```

        envioComun.Agencia = CUListadoAgencia.Ejecutar().Select(dtoAgencia
=> new AgenciaViewModel
    {
        Id = dtoAgencia.Id,
        Nombre = dtoAgencia.Nombre
    }).ToList();
        return View(envioComun);
    }

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult CreateEnvioUrgente(AltaEnvioUrgenteVM envioUrgente)
{
    try
    {
        if (ModelState.IsValid)
        {
            AltaEnvioUrgenteDTO envioDTO = new
AltaEnvioUrgenteDTO()
            {
                NumeroTracking =
envioUrgente.NumeroTracking,
                Peso = envioUrgente.Peso,
                FuncionarioId = envioUrgente.FuncionarioId,

```

```

        ClientId = envioUrgente.ClientId,
        DireccionPostal =
envioUrgente.DireccionPostal,

        Entregado = envioUrgente.Entregado,
        TiempoEntrega = envioUrgente.TiempoEntrega,
        Estado = Estado.En_Proceso
    };

    CUALtaEnvioUrgente.Ejecutar(envioDTO);
    return RedirectToAction(nameof(Index));
}

}

catch (UsuarioException ex)
{
    ViewBag.Message = ex.Message;

}

catch (Exception ex)
{
    ViewBag.DatosExcepcion = ex.StackTrace;
    ViewBag.Mensaje = "Error en los datos";
}

envioUrgente.Usuarios = CUListaUsuario.Ejecutar().Select(dtoUsuario
=> new UsuarioViewModel
{
    Id = dtoUsuario.Id,
    Nombre = dtoUsuario.Nombre
}).ToList();

envioUrgente.Cliente = CUListadoCliente.Ejecutar().Select(dtoUsuario
=> new ClienteViewModel
{

```

```

        Id = dtoUsuario.Id,
        Nombre = dtoUsuario.Nombre
    }).ToList();
    return View(envioUrgente);
}

// GET: Envio/Edit/5
public ActionResult Edit(int id)
{
    return View();
}

// POST: Envio/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(int id, IFormCollection collection)
{
    try
    {
        return RedirectToAction(nameof(Index));
    }
    catch
    {
        return View();
    }
}

// GET: Envio/Delete/5
public ActionResult Delete(int id)
{
    return View();
}

```

```
}
```

```
// POST: Envio/Delete/5
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Delete(int id, IFormCollection collection)
```

```
{
```

```
    try
```

```
    {
```

```
        return RedirectToAction(nameof(Index));
```

```
    }
```

```
    catch
```

```
    {
```

```
        return View();
```

```
    }
```

```
}
```

```
[HttpPost]
```

```
public ActionResult CambiarEstadoFinalizado(string nroTracking)
```

```
{
```

```
    try
```

```
    {
```

```
        if (string.IsNullOrEmpty(nroTracking))
```

```
        {
```

```
            ViewBag.Mensaje = "El número de tracking no es  
válido.";
```

```
            return RedirectToAction(nameof(Index));
```

```
        }
```

```
        CUFinalizarEnvio.Ejecutar(nroTracking);
```

```
        return RedirectToAction(nameof(Index));
```

```
    }
```



```

        catch (EnvioException e)
        {

            ViewBag.Mensaje = e.Message;

            return RedirectToAction(nameof(Index));

        }
        catch (Exception ex)
        {

            ViewBag.DatosExcepcion = ex.StackTrace;

            ViewBag.Mensaje = "Error en los datos";

            return RedirectToAction(nameof(Index));

        }

```

```

    }

```

```

    public IActionResult AgregarSeguimiento(int enviold)
    {

        AgregarSeguimientoVM seguimientoVM = new
        AgregarSeguimientoVM();

        try
        {

            seguimientoVM.Enviold = enviold;

            seguimientoVM.UsuarioId =
            (int)HttpContext.Session.GetInt32("usuarioId");

        }
        catch (EnvioException ex)
        {

            ViewBag.Mensaje = ex.Message;

```

```

    }

    catch (Exception ex)
    {
        ViewBag.Mensaje = "Datos incorrectos";
    }

    return View(seguimientoVM);
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult AgregarSeguimiento(AgregarSeguimientoVM
seguimientoVM)
{
    try
    {
        if (ModelState.IsValid)
        {
            SeguimientoDTO seguimientoDTO = new
SeguimientoDTO()
            {
                Envioid = seguimientoVM.Envioid,
                Usuarioid = seguimientoVM.UsuarioId,
                Comentario = seguimientoVM.Comentario,
                FechaHora = DateTime.Now
            };
            agregarSeguimiento.Ejecutar(seguimientoDTO);
            return RedirectToAction(nameof(Index));
        }
    }
    catch (EnvioException ex)
    {
        ViewBag.Message = ex.Message;
    }
}

```

```

        }
        catch (Exception ex)
        {
            ViewBag.DatosExcepcion = ex.StackTrace;
            ViewBag.Mensaje = "Error en los datos";
        }
        return View(seguimientoVM);
    }
}
}

```

Archivo: HomeController.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\MVC\Controllers\HomeController.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
using MVC.Models;
```

```
using System.Diagnostics;
```

```
namespace MVC.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: UsuarioController.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Controllers\UsuarioController.cs

```

using Compartido.DTOs.Usuario;
using LogicaAplicacion.ImplementacionCasosUso.UsuarioCU;
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;
using LogicaNegocio.ExcepcionesEntidades;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

```

```
using MVC.Filtro;
```

```
using MVC.Models.Usuarios;
```

```
namespace MVC.Controllers
```

```
{
```

```
    //[Admin]
```

```
    public class UsuarioController : Controller
```

```
    {
```

```
        private IRegistrarUsuario CURegistroUsuario { get; set; }
```

```
        private IListadoUsuarios CUListadoUsuarios { get; set; }
```

```
        private IEliminarUsuario CUEliminarUsuarios { get; set; }
```

```
        private IBuscarUsuario CUBuscarUsuario { get; set; }
```

```
        private IModificarUsuario CUModificarUsuario { get; set; }
```

```
        private ILoginUsuario CULoginUsuario { get; set; }
```

```
        public UsuarioController(IRegistrarUsuario registroUsuarioCU, IListadoUsuarios  
cUListadoUsuarios,
```

```
        IEliminarUsuario cUEliminarUsuarios,
```

```
        IBuscarUsuario cUBuscarUsuarios,
```

```
        IModificarUsuario cUModificarUsuario, ILoginUsuario cULoginUsuario)
```

```
        {
```

```
            CURegistroUsuario = registroUsuarioCU;
```

```
            CUListadoUsuarios = cUListadoUsuarios;
```

```
            CUEliminarUsuarios = cUEliminarUsuarios;
```

```
            CUBuscarUsuario = cUBuscarUsuarios;
```

```
            CUModificarUsuario = cUModificarUsuario;
```

```
            CULoginUsuario = cULoginUsuario;
```

```
        }
```

```
        // GET: UsuarioController
```

```
        [Logueado]
```

```

public ActionResult Index()
{

    IEnumerable<ListadoUsuariosDTO> usuariosDTO =
    CUListadoUsuarios.Ejecutar();

    List<ListadoUsuariosViewModel> listadoUsuarioViewModel =
        new List<ListadoUsuariosViewModel>();

    try
    {

        listadoUsuarioViewModel = usuariosDTO.Select(static c => new
ListadoUsuariosViewModel()
        {

            Id = c.Id,
            Ci = c.Ci,
            Nombre = c.Nombre,
            Apellido = c.Apellido,
            Email = c.Email,
            Contraseña = c.Contraseña,
            Rol = (Models.Usuarios.Rol)c.rol,

        }).ToList();
    }
    catch (Exception ex)
    {

        ViewBag.Mensaje = "Datos incorrectos";
    }

    return View(listadoUsuarioViewModel);
}

// GET: UsuarioController/Details/5
[Logueado]

```

```
public ActionResult Details(int id)
{
    return View();
}
```

```
// GET: UsuarioController/Create
```

```
[Logueado]
```

```
[Admin]
```

```
public ActionResult Create()
```

```
{
    return View();
}
```

```
// POST: UsuarioController/Create
```

```
[Logueado]
```

```
[Admin]
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create(UsuarioViewModel usuarioVM)
```

```
{
    try
    {
        if (ModelState.IsValid)
        {
            UsuarioDTO usuarioDTO = new UsuarioDTO()
            {
                Nombre = usuarioVM.Nombre,
                Apellido = usuarioVM.Apellido,
                Ci = usuarioVM.Ci,
                Email = usuarioVM.Email,
                Contraseña = usuarioVM.Contraseña,
            }
        }
    }
}
```

```

        Rol =
(Compartido.DTOs.Usuario.Rol)usuarioVM.Rol,

        };

        CURegistroUsuario.Ejecutar(usuarioDTO);
        return RedirectToAction(nameof(Index));
    }

```

```

    }
    catch (UsuarioException ex)
    {
        ViewBag.Message = ex.Message;

    }
    catch (Exception ex)
    {
        ViewBag.DatosExcepcion = ex.StackTrace;
        ViewBag.Mensaje = "Error en los datos";
    }
    return View();
}

```

// GET: UsuarioController/Edit/5

[Logueado]

[Admin]

public ActionResult Edit(int id)

```

{
    ListadoUsuariosViewModel usuarioVM = null;
    try
    {
        if(id <= 0)

```



```

        {
            throw new UsuarioException("El ID es Incorrecta");
        }

        ListadoUsuariosDTO usuariosDTO =
CUBuscarUsuario.Ejecutar(id);

        usuarioVM = new ListadoUsuariosViewModel()
        {
            Id = usuariosDTO.Id,
            Ci = usuariosDTO.Ci,
            Nombre = usuariosDTO.Nombre,
            Apellido = usuariosDTO.Apellido,
            Email = usuariosDTO.Email,
            Contraseña = usuariosDTO.Contraseña,
            Rol = (Models.Usuarios.Rol)usuariosDTO.rol
        };
    }

    catch (UsuarioException ex)
    {
        ViewBag.Mensaje = ex.Message;
    }

    return View(usuarioVM);
}

// POST: UsuarioController/Edit/5
[Logueado]
[Admin]
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(int id, ListadoUsuariosViewModel usuarioVM)
{

```

```

try
{
    if (ModelState.IsValid)
    {
        ListadoUsuariosDTO usuariosDTO = new ListadoUsuariosDTO()
        {
            Id = usuarioVM.Id,
            Ci = usuarioVM.Ci,
            Nombre = usuarioVM.Nombre,
            Apellido = usuarioVM.Apellido,
            Email = usuarioVM.Email,
            Contraseña = usuarioVM.Contraseña,
            rol =
(Compartido.DTOs.Usuario.Rol)usuarioVM.Rol

        };
        CUModificarUsuario.Ejecutar(usuariosDTO);
        return RedirectToAction(nameof(Index));
    }
    else
    {
        ViewBag.Mensaje = "Datos incorrectos";
    }
}
catch (UsuarioException ex)
{
    ViewBag.Mensaje = ex.Message;
}
catch (Exception ex)
{
    ViewBag.Mensaje = "Error en los datos";
}

```

```

    }
    return View();
}

// GET: UsuarioController/Delete/5
[Logueado]
[Admin]
public ActionResult Delete(int id)
{
    ListadoUsuariosViewModel usuarioVM = null;
    try
    {
        if (id <= 0)
        {
            throw new UsuarioException("El id no es correcto");
        }

        ListadoUsuariosDTO usuarioDTO =
            CUBuscarUsuario.Ejecutar(id);
        usuarioVM = new ListadoUsuariosViewModel()
        {
            Id = usuarioDTO.Id,
            Nombre = usuarioDTO.Nombre,
            Apellido = usuarioDTO.Apellido,
            Email = usuarioDTO.Email,
            Contraseña = usuarioDTO.Contraseña,
            Rol = (Models.Usuarios.Rol)usuarioDTO.rol
        };
    }
    catch (UsuarioException ex)
    {
        ViewBag.Mensaje = ex.Message;
    }
}

```

```

    }

    return View(usuarioVM);
}

// POST: UsuarioController/Delete/5
[Logueado]
[Admin]
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Delete(int id, ListadoUsuariosViewModel usuarioVM)
{
    try
    {
        CUEliminarUsuarios.Ejecutar(id);
        return RedirectToAction(nameof(Index));
    }
    catch (UsuarioException ex)
    {
        ViewBag.Mensaje = ex.Message;
    }
    catch (Exception ex)
    {
        ViewBag.Mensaje = "Error en los datos";
    }
    return View();
}

public ActionResult Login()
{
    return View();
}

```

```

// POST: Usuario/Login
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Login(string email, string contraseña)
{
    try
    {
        UsuarioDTO usuarioDTO = CULoginUsuario.Ejecutar(email,
contraseña);

        HttpContext.Session.SetInt32("usuarioId", usuarioDTO.Id);
        HttpContext.Session.SetString("email", usuarioDTO.Email);
        HttpContext.Session.SetString("rol", usuarioDTO.Rol.ToString());
        HttpContext.Session.SetString("nombre",
usuarioDTO.Nombre);

        return RedirectToAction("Index", "Home");
    }
    catch (UsuarioException ex)
    {
        ViewBag.ErrorMessage = ex.Message;
        return View();
    }
}

public IActionResult Logout()
{
    HttpContext.Session.Clear();

    return RedirectToAction("Login");
}

```

```
}  
}
```

Archivo: Admin.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\MVC\Filtro\Admin.cs

using Microsoft.AspNetCore.Mvc.Filters;

using Microsoft.AspNetCore.Mvc;

using Compartido.DTOs.Usuario;

public class Admin : Attribute, IAuthorizationFilter

```
{  
    public void OnAuthorization(AuthorizationFilterContext context)  
    {  
        string rol = context.HttpContext.Session.GetString("rol");  
  
        if (rol == null || rol != Rol.Administrador.ToString())  
        {  
            context.Result = new RedirectResult("/Usuario/Login");  
        }  
    }  
}
```

Archivo: Logueado.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\MVC\Filtro\Logueado.cs

using Microsoft.AspNetCore.Mvc.Filters;

using Microsoft.AspNetCore.Mvc;

```

namespace MVC.Filtro
{
    public class Logueado : Attribute, IAuthorizationFilter
    {
        public void OnAuthorization(AuthorizationFilterContext context)
        {
            if (context.HttpContext.Session.GetString("email") == null)
            {
                context.Result = new RedirectResult("/Home/index");
            }
        }
    }
}

```

Archivo: ErrorViewModel.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\ErrorViewModel.cs

```

namespace MVC.Models
{
    public class ErrorViewModel
    {
        public string? RequestId { get; set; }
    }
}

```

```

        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

    }

}

```

Archivo: EnvioController.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\WebAPI\Controllers\EnvioController.cs

```

using LogicaAplicacion.InterfacesCasosUso.EnvioCU;

using Microsoft.AspNetCore.Mvc;

```

// For more information on enabling Web API for empty projects, visit
<https://go.microsoft.com/fwlink/?LinkID=397860>

```

namespace WebAPI.Controllers

```

```

{
    [Route("api/[controller]")]
    [ApiController]
    public class EnvioController : ControllerBase
    {
        public IListadoEnvios CListadoEnvios { get; set; }

        public IBuscarEnvios CBuscarEnvios { get; set; }

        public EnvioController(IListadoEnvios cListadoEnvios, IBuscarEnvios
cUbuscarEnvios)
        {
            CListadoEnvios = cListadoEnvios;
            CBuscarEnvios = cUbuscarEnvios;
        }
    }
}

```



```
// GET: api/<EnvioController>
[HttpGet]
public IActionResult Get()
{
    //FindAll
    try
    {
        return Ok(CUlistadoEnvios.Ejecutar());
    }
    catch (Exception ex)
    {
        return StatusCode(500);
    }
}
```

```
// GET api/<EnvioController>/5
[HttpGet("{NroTracking}")]
public IActionResult Get(string NroTracking)
{
    //FindById
    try
    {
        if (NroTracking == null)
        {
```

```

        return BadRequest("El NroTracking no puede ser
Nulo");
    }
    return Ok(CUbuscarEnvios.Ejecutar(NroTracking));
}
catch (Exception ex)
{
    return StatusCode(500, "Datos Incorrectos");
}
}

```

```

// POST api/<EnvioController>

```

```

[HttpPost]

```

```

public void Post([FromBody] string value)

```

```

{

```

```

    //Add o Create

```

```

}

```

```

// PUT api/<EnvioController>/5

```

```

[HttpPut("{id}")]

```

```

public void Put(int id, [FromBody] string value)

```

```

{

```

```

    //Update

```

```

}

```

```

// DELETE api/<EnvioController>/5

```

```

[HttpDelete("{id}")]

```

```

public void Delete(int id)

```

```

{

```

```

    //Delete o Remove

```

```
    }  
}  
}
```

Archivo: WeatherForecastController.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\WebAPI\Controllers\WeatherForecastController.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
namespace WebAPI.Controllers
```

```
{  
    [ApiController]  
    [Route("[controller]")]  
    public class WeatherForecastController : ControllerBase  
    {  
        private static readonly string[] Summaries = new[]  
        {  
            "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot",  
            "Sweltering", "Scorching"  
        };  
  
        private readonly ILogger<WeatherForecastController> _logger;  
  
        public WeatherForecastController(ILogger<WeatherForecastController> logger)  
        {  
            _logger = logger;  
        }  
  
        [HttpGet(Name = "GetWeatherForecast")]  
        public IEnumerable<WeatherForecast> Get()
```

```

        {
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
            {
                Date =
DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
                TemperatureC = Random.Shared.Next(-20, 55),
                Summary =
Summaries[Random.Shared.Next(Summaries.Length)]
            })
            .ToArray();
        }
    }
}

```

Archivo: ListadoAgenciasDTO.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\Compartido\DTOs\Agencia>ListadoAgenciasDTO.cs

using LogicaNegocio.EntidadesNegocio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Compartido.DTOs.Agencia

{

public class ListadoAgenciasDTO

{

public int Id { get; set; }

public string Nombre { get; set; }

```
        public string DireccionPostal { get; set; }

        public Ubicacion Ubicacion { get; set; }

    }

}
```

Archivo: AltaEnvioComunDTO.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\Compartido\DTOs\Envios\AltaEnvioComunDTO.cs

```
using LogicaNegocio.EntidadesNegocio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Compartido.DTOs.Envios

{

    public class AltaEnvioComunDTO

    {

        public string NumeroTracking { get; set; }

        public int Peso { get; set; }

        public int FuncionarioId { get; set; }

        public int ClienteId { get; set; }

        public int AgencialId { get; set; }

        public Estado Estado { get; set; }

    }

}
```

Archivo: AltaEnvioUrgenteDTO.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\Compartido\DTOs\Envios\AltaEnvioUrgenteDTO.cs

```
using LogicaNegocio.EntidadesNegocio;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Compartido.DTOs.Envios
```

```
{
```

```
    public class AltaEnvioUrgenteDTO
```

```
    {
```

```
        public string NumeroTracking { get; set; }
```

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

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

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

```
        public string DireccionPostal { get; set; }
```

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

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

```
        public Estado Estado { get; set; }
```

```
    }
```

```
}
```

Archivo: ListadoEnviosDTO.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\Compartido\DTOs\Envios>ListadoEnviosDTO.cs

```
using LogicaNegocio.EntidadesNegocio;
```

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Compartido.DTOs.Envios
{
    public class ListadoEnviosDTO
    {
        public int Id { get; set; }

        public string NroTracking { get; set; }

        public int Peso { get; set; }

        public string FuncionarioNombre { get; set; }

        public string ClienteNombre { get; set; }

        public Estado Estado { get; set; }


        public DateTime FechaEntrega { get; set; }

    }
}

```

Archivo: SeguimientoDTO.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\Compartido\DTOs\Seguimiento\SeguimientoDTO.cs

```

using LogicaNegocio.EntidadesNegocio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

```

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

```
namespace Compartido.DTOs.Seguimiento
```

```
{  
    public class SeguimientoDTO  
    {  
        public string Comentario { get; set; }  
        public DateTime FechaHora { get; set; }  
        public int Enviold { get; set; }  
        public int Usuariold { get; set; }  
    }  
}
```

```
*****
```

```
Archivo: ListadoUsuariosDTO.cs
```

```
Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--  
308898\Compartido\DTOs\Usuario>ListadoUsuariosDTO.cs
```

```
*****
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Compartido.DTOs.Usuario
```

```
{  
    public class ListadoUsuariosDTO  
    {  
        public int Id { get; set; }  
        public string Ci { get; set; }  
        public string Nombre { get; set; }  
    }  
}
```



```

        public string Apellido { get; set; }

        public string Email { get; set; }

        public string Contraseña { get; set; }

        public Rol rol { get; set; }

    }

}

```

Archivo: UsuarioDTO.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\Compartido\DTOs\Usuario\UsuarioDTO.cs

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Compartido.DTOs.Usuario
{
    public class UsuarioDTO
    {
        public int Id { get; set; }

        public string Ci { get; set; }

        public string Nombre { get; set; }

        public string Apellido { get; set; }

        public string Email { get; set; }

        public string Contraseña { get; set; }

        public Rol Rol { get; set; }
    }
}

```

```

    }

    public enum Rol
    {
        Administrador,
        Funcionario,
        Cliente
    }
}

*****

Archivo: ListadoAgencia.cs
Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\AgenciaCU>ListadoAgencia.cs
*****

using Compartido.DTOs.Agencia;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.AgenciaCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.AgenciaCU
{
    public class ListadoAgencia : IListadoAgencias
    {

```

```

        private IRepositoryAgencia RepoAgencia { get; set; }

        public ListadoAgencia(IRepositoryAgencia repoAgencia)
        {
            RepoAgencia = repoAgencia;
        }

        public IEnumerable<ListadoAgenciasDTO> Ejecutar()
        {
            IEnumerable<ListadoAgenciasDTO> agenciasDTO = new
            List<ListadoAgenciasDTO>();

            IEnumerable<Agencia> agencias = RepoAgencia.FindAll().ToList();

            agenciasDTO =
            AgenciaMapper.ListadoAgenciaToListadoAgenciaDTO(agencias);

            return agenciasDTO;
        }
    }
}

```

Archivo: AgregarSeguimiento.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\AgregarSeguimiento.cs

```

using Compartido.DTOs.Seguimiento;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.InterfacesRepositorios;

```

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class AgregarSeguimiento : IAgregarSeguimiento
    {
        public IRepositoryEnvio RepoEnvio { get; set; }

        public IRepositoryUsuario RepoUsuario { get; set; }

        public AgregarSeguimiento(IRepositoryEnvio repoEnvio, IRepositoryUsuario
repoUsuario)
        {
            RepoEnvio = repoEnvio;
            RepoUsuario = repoUsuario;
        }

        public void Ejecutar(SeguimientoDTO seguimientoDTO)
        {
            Envio envio = RepoEnvio.FindById(seguimientoDTO.EnvioId);
            Usuario usuario = RepoUsuario.FindById(seguimientoDTO.UsuarioId);
            Seguimiento seguimientos =
EnvioMapper.SeguimientoDTOTOseguimiento(seguimientoDTO,usuario);
            List<Seguimiento> listaSeguimientos = envio.Seguimientos.ToList();
            listaSeguimientos.Add(seguimientos);
            envio.Seguimientos = listaSeguimientos;
            if (seguimientos != null)
            {

```

```

        RepoEnvio.Update(envio);

    }

    else

    {

        throw new EnvioException("No se encontro un envio con ese
numero de tracking");

    }

}

}

}

```

Archivo: AltaEnvioComun.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\AltaEnvioComun.cs

```

using Compartido.DTOs.Envios;

using Compartido.Mappers;

using LogicaAplicacion.InterfacesCasosUso.EnvioCU;

using LogicaNegocio.EntidadesNegocio;

using LogicaNegocio.InterfacesRepositorios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU

```

```

{

    public class AltaEnvioComun : IAltaEnvioComun
    {

        private IRepositorioEnvio RepoEnvio { get; set; }

        private IRepositorioUsuario RepoUsuario { get; set; }

        private IRepositorioAgencia RepoAgencia { get; set; }

        public AltaEnvioComun(IRepositorioEnvio repositorioEnvio,
IRepositorioUsuario repoUsuario,IRepositorioAgencia repoAgencia)
        {

            RepoEnvio = repositorioEnvio;

            RepoUsuario = repoUsuario;

            RepoAgencia = repoAgencia;

        }

        public void Ejecutar(AltaEnvioComunDTO envioDTO)
        {

            if (envioDTO == null)
            {

                throw new ArgumentNullException("Datos Incorrectos");

            }

            Usuario usuario = RepoUsuario.FindById(envioDTO.FuncionarioId);
            Usuario cliente = RepoUsuario.FindById(envioDTO.ClienteId);
            Agencia agencia = RepoAgencia.FindById(envioDTO.AgencialId);
            if (usuario != null && cliente != null)
            {

                Envio envio = EnvioMapper.ComunFromDTO(envioDTO,usuario,
cliente, agencia);

                RepoEnvio.Add(envio);

            }

        }

    }

}

```

```

        else
        {
            throw new ArgumentException("El funcionario o el cliente no
existen");
        }
    }
}
}
}
}

```

Archivo: AltaEnvioUrgente.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\AltaEnvioUrgente.cs

```

using Compartido.DTOs.Envios;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class AltaEnvioUrgente : IAltaEnvioUrgente
    {
        private IRepositoryioEnvio RepoEnvio{ get; set; }
    }
}

```

```

        private IRepositoryUsuario RepoUsuario { get; set; }

        public AltaEnvioUrgente(IRepositoryEnvio repositorioEnvio,
IRepositoryUsuario repoUsuario)
        {
            RepoEnvio = repositorioEnvio;
            RepoUsuario = repoUsuario;
        }

        public void Ejecutar(AltaEnvioUrgenteDTO envioDTO)
        {
            if (envioDTO == null)
            {
                throw new ArgumentNullException("Datos Incorrectos");
            }
            Usuario usuario = RepoUsuario.FindById(envioDTO.FuncionarioId);
            Usuario cliente = RepoUsuario.FindById(envioDTO.ClienteId);
            if(usuario != null && cliente != null)
            {
                Envio envio =
EnvioMapper.UrgenteFromDTO(envioDTO,usuario, cliente);
                RepoEnvio.Add(envio);
            }
            else
            {
                throw new ArgumentException("El funcionario o el cliente no
existen");
            }
        }
    }
}

```

Archivo: BuscarEnvios.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\BuscarEnvios.cs

```
using Compartido.DTOs.Envios;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.EnvioCU;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class BuscarEnvios : IBuscarEnvios
    {
        private IRepositoryioEnvio RepoEnvio { get; set; }

        public BuscarEnvios(IRepositoryioEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }

        public ListadoEnviosDTO Ejecutar(string NroTracking)
        {
            return EnvioMapper.EnvioToListadoDTO

```

```

        (RepoEnvio.FindByNroTracking(NroTracking));
    }

}

}

```

Archivo: FinalizarEnvio.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\FinalizarEnvio.cs

```

using LogicaAplicacion.InterfacesCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class FinalizarEnvio : IFinalizarEnvio
    {
        private IRepositorioEnvio RepoEnvio { get; set; }

        public FinalizarEnvio(IRepositorioEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }

        public void Ejecutar(string nroTracking)

```

```

        {
            if (string.IsNullOrEmpty(nroTracking))
            {
                throw new ArgumentNullException("Datos incorrectos");
            }
            Envio envio = RepoEnvio.FindByNroTracking(nroTracking);
            if (envio != null)
            {
                RepoEnvio.FinalizarEnvio(envio.NroTracking);
                RepoEnvio.Update(envio);
            }
            else
            {
                throw new ArgumentException("El envio no existe");
            }
        }
    }
}

```

Archivo: ListadoEnvios.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\EnvioCU>ListadoEnvios.cs

```

using Compartido.DTOs.Envios;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfacesRepositorios;
using System;

```

```

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class ListadoEnvios : IListadoEnvios
    {
        private IRepositorioEnvio RepoEnvio { get; set; }


        public ListadoEnvios(IRepositorioEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }


        public IEnumerable<ListadoEnviosDTO> Ejecutar()
        {
            IEnumerable<ListadoEnviosDTO> enviosDTO = new
List<ListadoEnviosDTO>();

            IEnumerable<Envio> envios = RepoEnvio.FindAll();

            enviosDTO = EnvioMapper.ListadoEnvioToListadoEnvioDTO(envios);

            return enviosDTO;
        }
    }
}

```

Archivo: BuscarUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\BuscarUsuario.cs

```
using Compartido.DTOs.Usuario;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU
```

```
{
    public class BuscarUsuario : IBuscarUsuario
    {
        private IRepositoryUsuario RepoUsuario { get; set; }

        public BuscarUsuario(IRepositoryUsuario repoUsuario)
        {
            RepoUsuario = repoUsuario;
        }

        public ListadoUsuariosDTO Ejecutar(int id)
        {
            return UsuarioMapper.UsuarioToListadoUsuarioDTO(RepoUsuario.FindById(id));
        }
    }
}
```

Archivo: EliminarUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\EliminarUsuario.cs

using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;

using LogicaNegocio.InterfacesRepositorios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU

{

public class EliminarUsuario : IEliminarUsuario

{

private IRepositoryUsuario RepoUsuario { get; set; }

public EliminarUsuario

(IRepositoryUsuario repositorioUsuario)

{

RepoUsuario = repositorioUsuario;

}

public void Ejecutar(int id)

{

RepoUsuario.Delete(id);

}

}

}

Archivo: ListadoClientes.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU>ListadoClientes.cs

```
using Compartido.DTOs.Usuario;
using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU
{
    public class ListadoClientes : IListadoCliente
    {

        private IRepositoryUsuario RepoUsuario { get; set; }

        public ListadoClientes(IRepositoryUsuario repousuario)
        {
            RepoUsuario = repousuario;
        }

        public IEnumerable<ListadoUsuariosDTO> Ejecutar()
        {

            IEnumerable<Usuario> usuarios =
RepoUsuario.FindByRolCliente().ToList();
```

```
        IEnumerable<ListadoUsuariosDTO> usuariosDTO =  
        UsuarioMapper.ListadoUsuarioToListadoUsuarioDTO(usuarios);  
  
        return usuariosDTO;
```

```
    }
```

```
}
```

```
}
```

```
*****
```

Archivo: ListadoUsuarios.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU>ListadoUsuarios.cs

```
*****
```

```
using Compartido.Mappers;  
using LogicaNegocio.EntidadesNegocio;  
using LogicaNegocio.InterfacesRepositorios;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;  
using Compartido.DTOs.Usuario;
```

```
namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU
```

```
{
```

```
    public class ListadoUsuarios: IListadoUsuarios
```

```
    {
```

```
        private IRepositoryUsuario RepoUsuario { get; set; }
```

```
        public ListadoUsuarios(IRepositoryUsuario repousuario)
```



```

        {

            RepoUsuario = repousuario;

        }

        public IEnumerable<ListadoUsuariosDTO> Ejecutar()
        {

            IEnumerable<ListadoUsuariosDTO> usuariosDTO = new
List<ListadoUsuariosDTO>();

            IEnumerable<Usuario> usuarios =
RepoUsuario.FindByRolFuncionario().ToList();

            usuariosDTO =
UsuarioMapper.ListadoUsuarioToListadoUsuarioDTO(usuarios);

            return usuariosDTO;

        }

    }
}

```

Archivo: LoginUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\LoginUsuario.cs

```

using Compartido.DTOs.Usuario;

using Compartido.Mappers;

using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;

using LogicaNegocio.EntidadesNegocio;

using LogicaNegocio.ExcepcionesEntidades;

using LogicaNegocio.InterfacesRepositorios;

using System;

using System.Collections.Generic;

```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU
{
    public class LoginUsuario : ILoginUsuario
    {
        private IRepositoryUsuario RepoUsuario { get; set; }

        public LoginUsuario(IRepositoryUsuario repoUsuario)
        {
            RepoUsuario = repoUsuario;
        }

        public UsuarioDTO Ejecutar(string Email, string contraseña)
        {
            if(string.IsNullOrEmpty(Email) || string.IsNullOrEmpty(contraseña))
            {
                throw new ArgumentNullException("Email y Contraseña son
requeridos");
            }

            Usuario usuario = RepoUsuario.FindByEmail(Email);
            if(usuario == null)
            {
                throw new UsuarioException("Usuario no encontrado");
            }

            return UsuarioMapper.UsuarioToUsuarioDTO(usuario);
        }
    }
}
```

```
    }  
}
```

Archivo: ModificarUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\ModificarUsuario.cs

```
using Compartido.DTOs.Usuario;  
using Compartido.Mappers;  
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;  
using LogicaNegocio.EntidadesNegocio;  
using LogicaNegocio.InterfacesRepositorios;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU  
{  
    public class ModificarUsuario : IModificarUsuario  
    {  
        private IRepositoryioUsuario RepoUsuario { get; set; }  
  
        public ModificarUsuario(IRepositoryioUsuario repousuario)  
        {  
            RepoUsuario = repousuario;  
        }  
  
        public void Ejecutar(ListadoUsuariosDTO usuariosDTO)
```

```

    {
        Usuario usuario = UsuarioMapper.UsuarioFromListadoUsuarioDTO(usuarioDTO);
        usuario.Id = usuariosDTO.Id;
        RepoUsuario.Update(usuario);
    }
}
}

```

Archivo: RegistroUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\RegistroUsuario.cs

```

using Compartido.Mappers;
using LogicaAplicacion.InterfacesCasosUso.UsuarioCU;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using LogicaNegocio.EntidadesNegocio;
using Compartido.DTOs.Usuario;

namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU
{
    public class RegistroUsuario : IRegistrarUsuario
    {
        private IRepositoryUsuario RepoUsuario { get; set; }

        public RegistroUsuario(IRepositoryUsuario repoUsuario)
    }
}

```

```

        {
            RepoUsuario = repoUsuario;
        }

        public void Ejecutar(UsuarioDTO usuarioDTO)
        {
            if (usuarioDTO == null)
            {
                throw new ArgumentNullException("Datos Incorrectos");
            }

            Usuario usuario =
            UsuarioMapper.UsuarioFromUsuarioDTO(usuarioDTO);

            RepoUsuario.Add(usuario);
        }
    }
}

```

Archivo: IListadoAgencias.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\AgenciaCU\IListadoAgencias.cs

```

using Compartido.DTOs.Agencia;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso.AgenciaCU
{
    public interface IListadoAgencias

```

```

        {

            IEnumerable<ListadoAgenciasDTO> Ejecutar();

        }

    }

```

Archivo: IAgregarSeguimiento.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\EnvioCU\IAgregarSeguimiento.cs

```

using Compartido.DTOs.Seguimiento;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso.EnvioCU
{
    public interface IAgregarSeguimiento
    {
        void Ejecutar(SeguimientoDTO seguimientoDTO);
    }
}

```

Archivo: IAltaEnvioComun.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\EnvioCU\IAltaEnvioComun.cs

```

using Compartido.DTOs.Envios;

using System;

```

```
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace LogicaAplicacion.InterfacesCasosUso.EnvioCU
{

    public interface IAltaEnvioComun
    {

        void Ejecutar(AltaEnvioComunDTO envioDTO);

    }

}
```

Archivo: IAltaEnvioUrgente.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\EnvioCU\IAltaEnvioUrgente.cs

```
using Compartido.DTOs.Envios;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace LogicaAplicacion.InterfacesCasosUso.EnvioCU
{

    public interface IAltaEnvioUrgente
    {

        void Ejecutar(AltaEnvioUrgenteDTO envioDTO);

    }

}
```

```
}
```

```
*****
```

Archivo: IBuscarEnvios.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\EnvioCU\IBuscarEnvios.cs

```
*****
```

```
using Compartido.DTOs.Envios;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaAplicacion.InterfacesCasosUso.EnvioCU
```

```
{
```

```
    public interface IBuscarEnvios
```

```
    {
```

```
        ListadoEnviosDTO Ejecutar(string NroTracking);
```

```
    }
```

```
}
```

```
*****
```

Archivo: IFinalizarEnvio.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\EnvioCU\IFinalizarEnvio.cs

```
*****
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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



```
namespace LogicaAplicacion.InterfacesCasosUso.EnvioCU
```

```
{  
  
    public interface IFinalizarEnvio  
    {  
  
        void Ejecutar(string nroTracking);  
  
    }  
  
}
```

```
*****
```

```
Archivo: IListadoEnvios.cs
```

```
Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--  
308898\LogicaAplicacion\InterfacesCasosUso\EnvioCU\IListadoEnvios.cs
```

```
*****
```

```
using Compartido.DTOs.Envios;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.InterfacesCasosUso.EnvioCU
```

```
{  
  
    public interface IListadoEnvios  
    {  
  
        IEnumerable<ListadoEnviosDTO> Ejecutar();  
  
    }  
  
}
```

```
*****
```

```
Archivo: IBuscarUsuario.cs
```

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\IBuscarUsuario.cs

```
using Compartido.DTOs.Usuario;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
```

```
{
    public interface IBuscarUsuario
    {
        ListadoUsuariosDTO Ejecutar(int id);
    }
}
```

Archivo: IEliminarUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\IEliminarUsuario.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
```

```
{
    public interface IEliminarUsuario
```

```
{  
    void Ejecutar(int id);  
}  
}
```

Archivo: IListadoCliente.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\IListadoCliente.cs

```
using Compartido.DTOs.Usuario;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
```

```
{  
    public interface IListadoCliente  
    {  
        IEnumerable<ListadoUsuariosDTO> Ejecutar();  
    }  
}
```

Archivo: IListadoUsuarios.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\IListadoUsuarios.cs

```
using Compartido.DTOs.Usuario;  
using System;
```

```
using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
{

    public interface IListadoUsuarios
    {

        IEnumerable<ListadoUsuariosDTO> Ejecutar();

    }

}
```

Archivo: ILoginUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\ILoginUsuario.cs

```
using Compartido.DTOs.Usuario;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
{

    public interface ILoginUsuario
    {

        UsuarioDTO Ejecutar(string Email, string Contraseña);

    }

}
```

```
}
```

```
*****
```

Archivo: IModificarUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\IModificarUsuario.cs

```
*****
```

```
using Compartido.DTOs.Usuario;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
```

```
{
```

```
    public interface IModificarUsuario
```

```
    {
```

```
        void Ejecutar(ListadoUsuariosDTO usuariosDTO);
```

```
    }
```

```
}
```

```
*****
```

Archivo: IRegistrarUsuario.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\LogicaAplicacion\InterfacesCasosUso\UsuarioCU\IRegistrarUsuario.cs

```
*****
```

```
using Compartido.DTOs.Usuario;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaAplicacion.InterfacesCasosUso.UsuarioCU
```

```
{  
  
    public interface IRegistrarUsuario  
  
    {  
  
        void Ejecutar(UsuarioDTO usuarioDTO);  
  
    }  
}
```

```
*****
```

Archivo: AgenciaViewModel.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Agencias\AgenciaViewModel.cs

```
*****
```

```
using LogicaNegocio.EntidadesNegocio;
```

```
namespace MVC.Models.Agencias
```

```
{  
  
    public class AgenciaViewModel  
  
    {  
  
        public int Id { get; set; }  
  
        public string Nombre { get; set; }  
  
        public string DireccionPostal { get; set; }  
  
        public Ubicacion Ubicacion { get; set; }  
  
    }  
}
```

```
*****
```

Archivo: AgregarSeguimientoVM.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Envios\AgregarSeguimientoVM.cs

namespace MVC.Models.Envios

{

public class AgregarSeguimientoVM

{

public string Comentario { get; set; }

public DateTime FechaHora { get; set; }

public int Envioid { get; set; }

public int Usuarioid { get; set; }

}

}

Archivo: AltaEnvioComunVM.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Envios\AltaEnvioComunVM.cs

using LogicaNegocio.EntidadesNegocio;

using MVC.Models.Usuarios;

using MVC.Models.Agencias;

namespace MVC.Models.Envios

{

public class AltaEnvioComunVM

{

public string NumeroTracking { get; set; }

public int Peso { get; set; }

public int Funcionarioid { get; set; }

public int Clienteld { get; set; }

public int Agenciald { get; set; }

public Estado Estado { get; set; }

```

        public IEnumerable<UsuarioViewModel> Usuarios { get; set; }

        = new List<UsuarioViewModel>();

        public IEnumerable<ClienteViewModel> Cliente { get; set; }

        = new List<ClienteViewModel>();

        public IEnumerable<AgenciaViewModel> Agencia { get; set; }

        = new List<AgenciaViewModel>();

    }

}

```

Archivo: AltaEnvioUrgenteVM.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Envios\AltaEnvioUrgenteVM.cs

```

using LogicaNegocio.EntidadesNegocio;

using MVC.Models.Usuarios;

```

```

namespace MVC.Models.Envios

```

```

{

```

```

    public class AltaEnvioUrgenteVM
    {

        public string NumeroTracking { get; set; }

        public int Peso { get; set; }

        public int FuncionarioId { get; set; }

        public int ClienteId { get; set; }

        public string DireccionPostal { get; set; }

        public bool Entregado { get; set; }

        public int TiempoEntrega { get; set; }

        public Estado Estado { get; set; }
    }

```



```

        public IEnumerable<UsuarioViewModel> Usuarios { get; set; }

        = new List<UsuarioViewModel>();

        public IEnumerable<ClienteViewModel> Cliente { get; set; }

        = new List<ClienteViewModel>();

    }

}

```

Archivo: ListadoEnviosVM.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Envios>ListadoEnviosVM.cs

```
using LogicaNegocio.EntidadesNegocio;
```

```
namespace MVC.Models.Envios
```

```

{

    public class ListadoEnviosVM

    {

        public int Id { get; set; }

        public int Peso { get; set; }

        public string FuncionarioNombre { get; set; }

        public string ClienteNombre { get; set; }

        public Estado Estado { get; set; }


        public DateTime FechaEntrega { get; set; }

    }

}

```

Archivo: ClienteViewModel.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\MVC\Models\Usuarios\ClienteViewModel.cs

namespace MVC.Models.Usuarios

```
{  
  
    public class ClienteViewModel  
    {  
  
        public int Id { get; set; }  
  
        public string Ci { get; set; }  
  
        public string Nombre { get; set; }  
  
        public string Apellido { get; set; }  
  
        public string Email { get; set; }  
  
        public string Contraseña { get; set; }  
  
        public Rol Rol { get; set; }  
  
    }  
}
```

Archivo: ListadoUsuariosViewModel.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--308898\MVC\Models\Usuarios>ListadoUsuariosViewModel.cs

namespace MVC.Models.Usuarios

```
{  
  
    public class ListadoUsuariosViewModel  
    {  
  
        public int Id { get; set; }  
  
        public string Ci { get; set; }  
  
        public string Nombre { get; set; }  
  
        public string Apellido { get; set; }  
  
        public string Email { get; set; }  
  
    }  
}
```

```

        public string Contraseña { get; set; }

        public Rol Rol;
    }

    public enum Rol
    {
        Administrador,
        Funcionario,
        Cliente,
    }
}

*****

Archivo: LoginViewModel.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Usuarios\LoginViewModel.cs

*****

namespace MVC.Models.Usuarios
{
    public class LoginViewModel
    {
        public string Email { get; set; }
        public string Contraseña { get; set; }
    }
}

```

```

*****

Archivo: UsuarioViewModel.cs

Carpeta: C:\Users\ismad\source\repos\Prog-P3-2025-1\333473--
308898\MVC\Models\Usuarios\UsuarioViewModel.cs

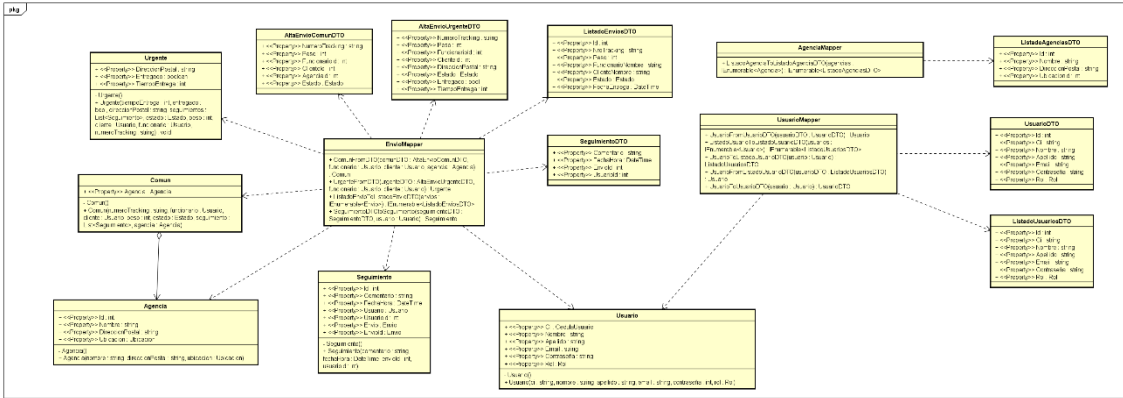
*****

```

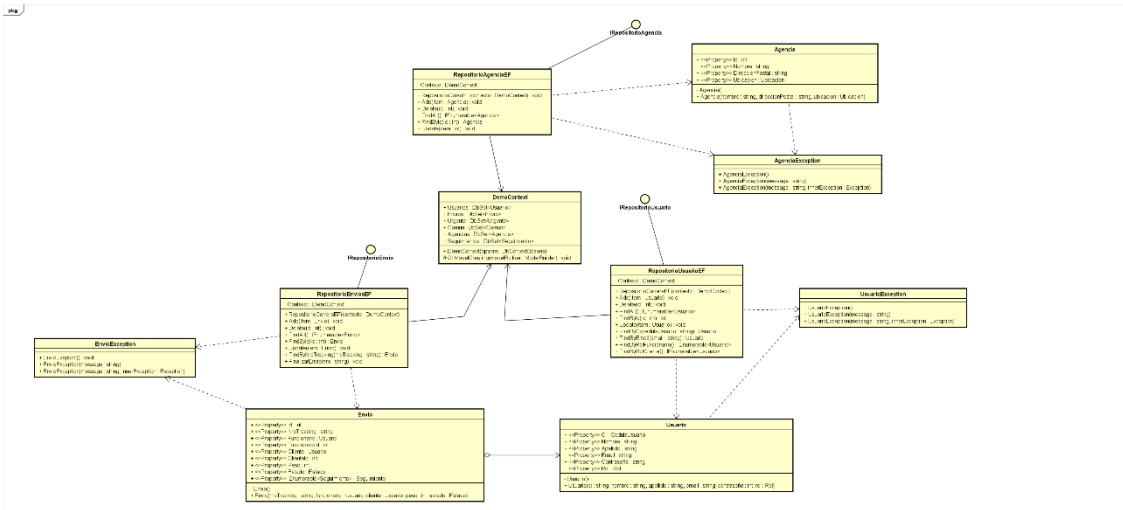
```
namespace MVC.Models.Usuarios
{
    public class UsuarioViewModel
    {
        public int Id { get; set; }
        public string Ci { get; set; }
        public string Nombre { get; set; }
        public string Apellido { get; set; }
        public string Email { get; set; }
        public string Contraseña { get; set; }
        public Rol Rol { get; set; }
    }
}
```

Diagrama de clases

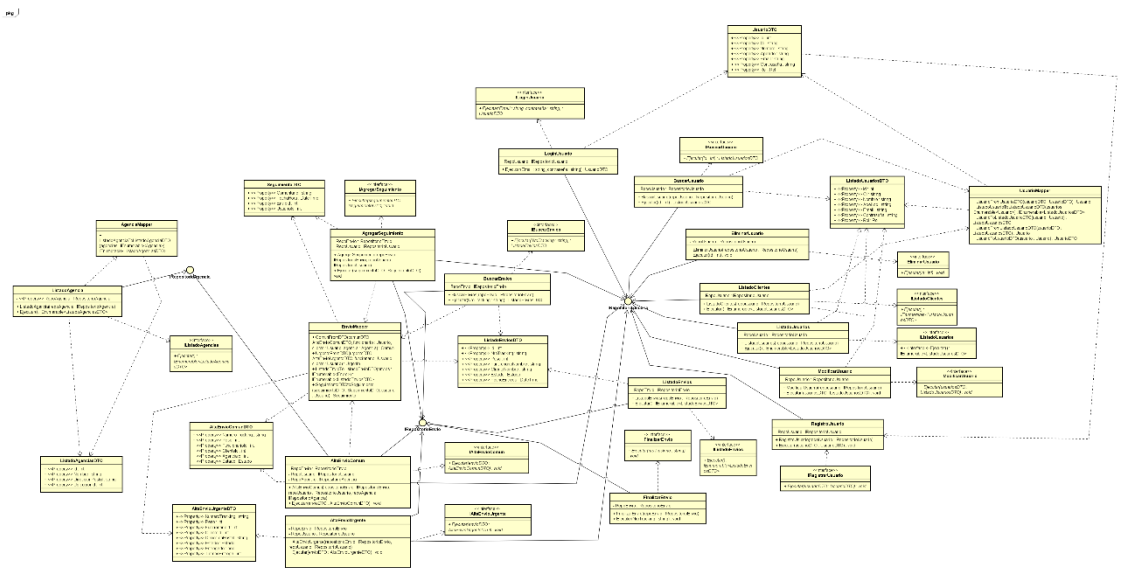
Compartido



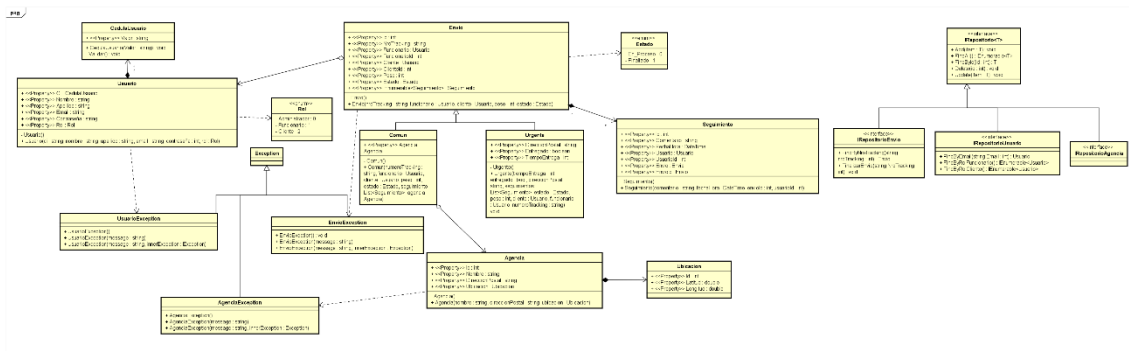
LogicaAccesoDatos



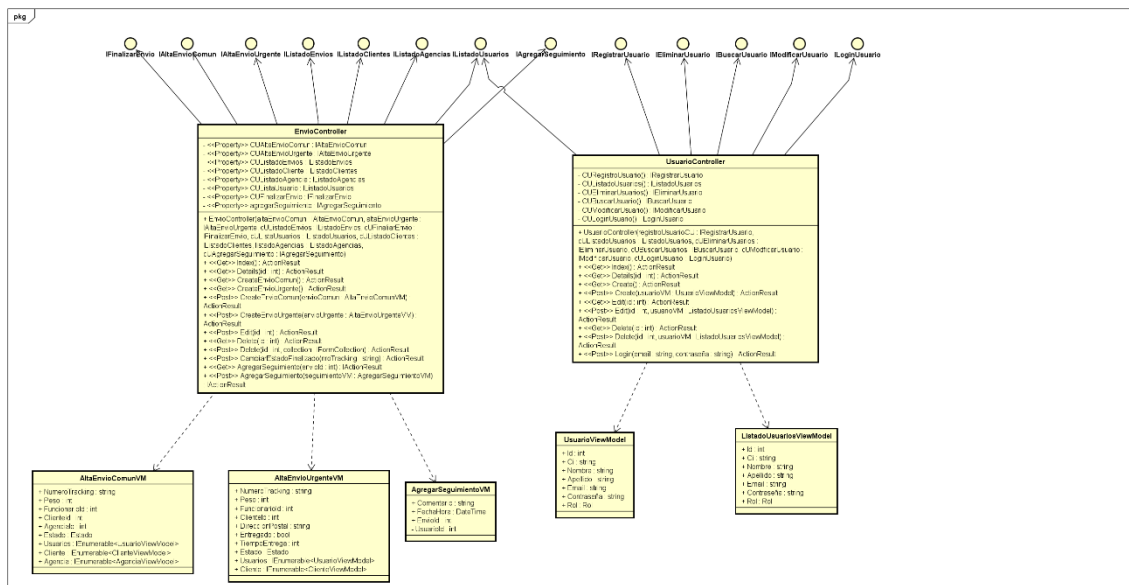
LogicaAplicacion



LogicaNegocio



M.V.C



WebAPI

