

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO 2

Programación 3

DOCUMENTACIÓN



Maverick López - 335912

M3A

Docente: Liliana Pino

Analista en Tecnologías de la Información

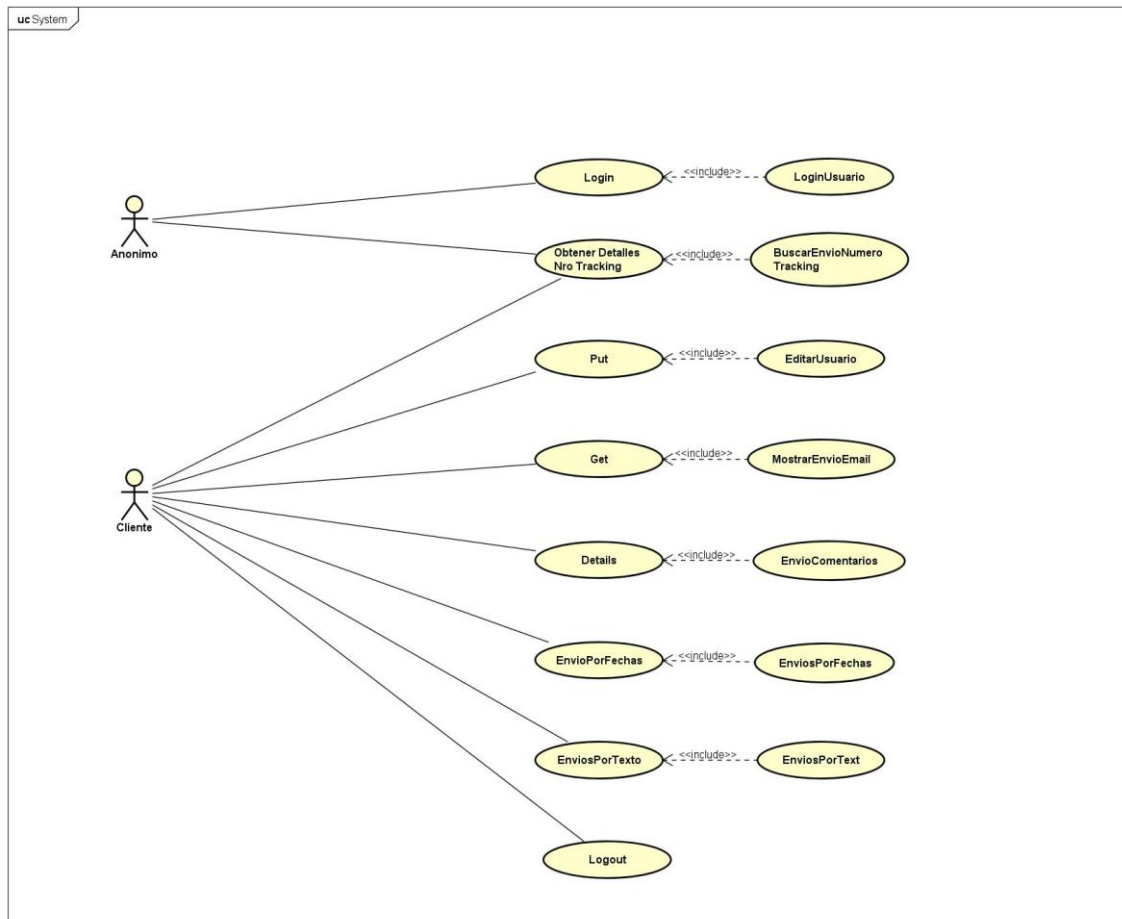
26-06-2025

Índice de contenidos

Contenidos

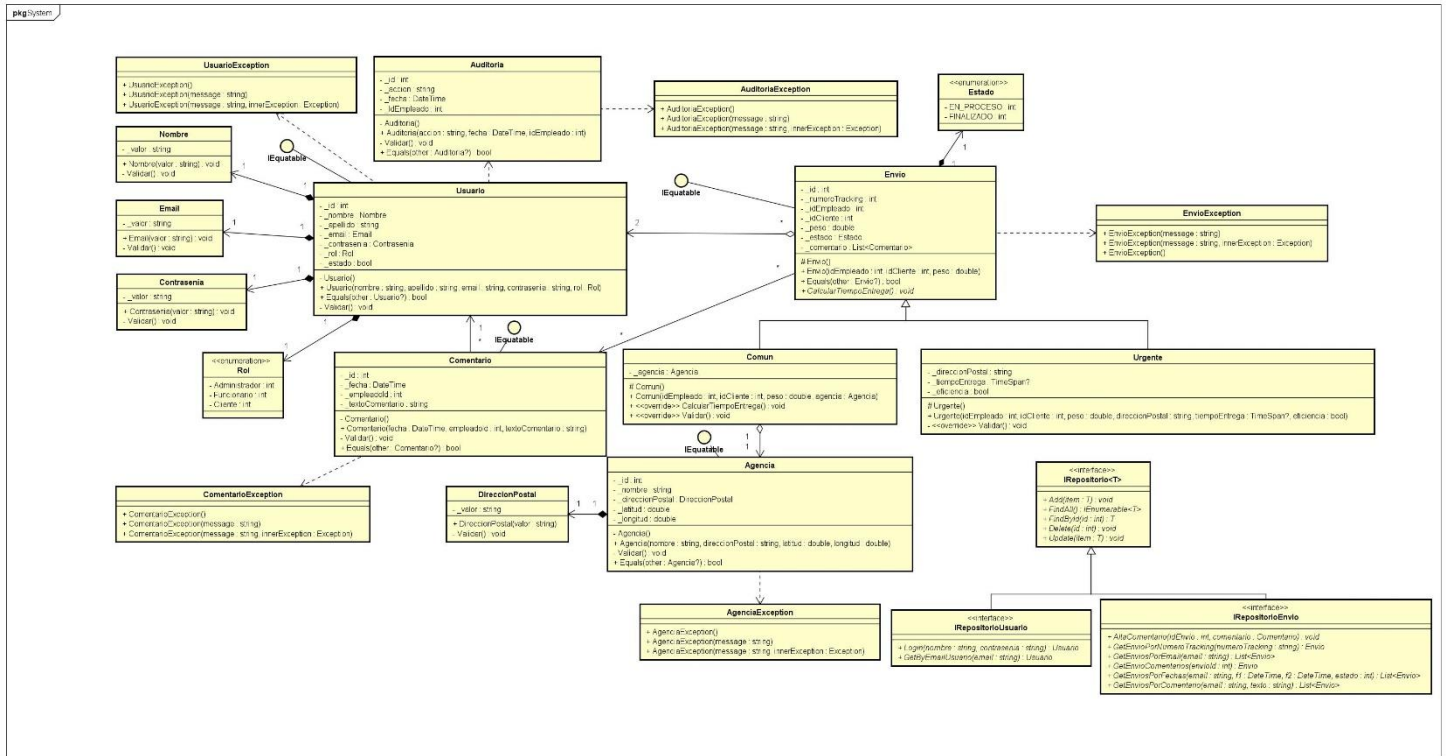
1 Diagrama de Casos de Uso.....	3
2 Diagrama de clases UML	4
2.1 Entidades de Negocio	4
2.2 Compartido	4
2.3 Lógica Acceso Datos	5
2.4 Logica Aplicación	6
2.5 MVC	7
2.6 Web API	7
3 Código fuente de toda la Aplicación	8
3.1 API	8
3.2 MVC	61
4 Datos de prueba	83
5 Links Importantes	87
5.1 GitHub.....	87
5.2 Azure	87

1 Diagrama de Casos de Uso

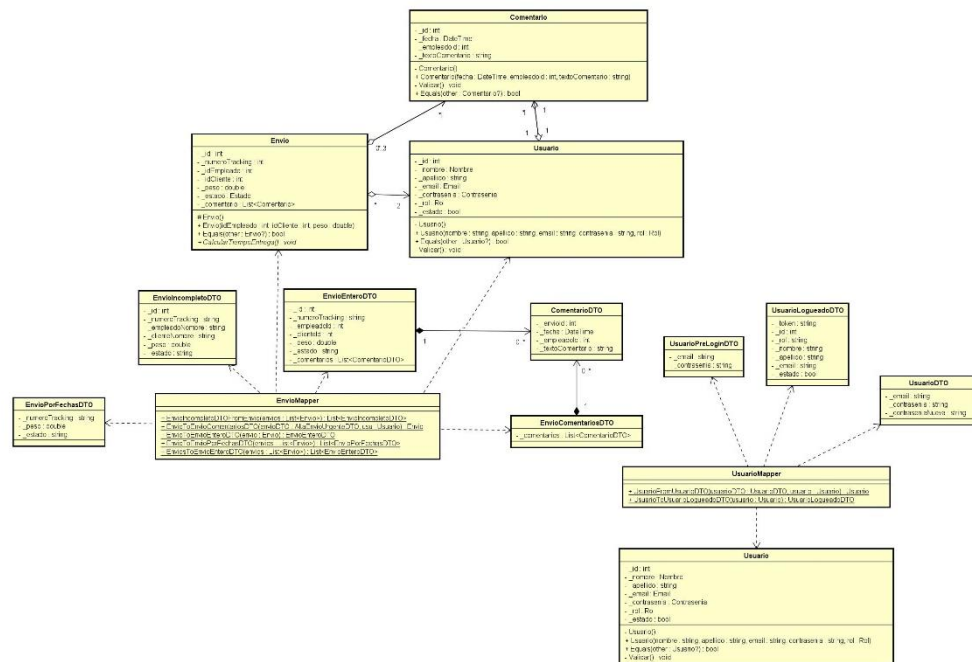


2 Diagrama de clases UML

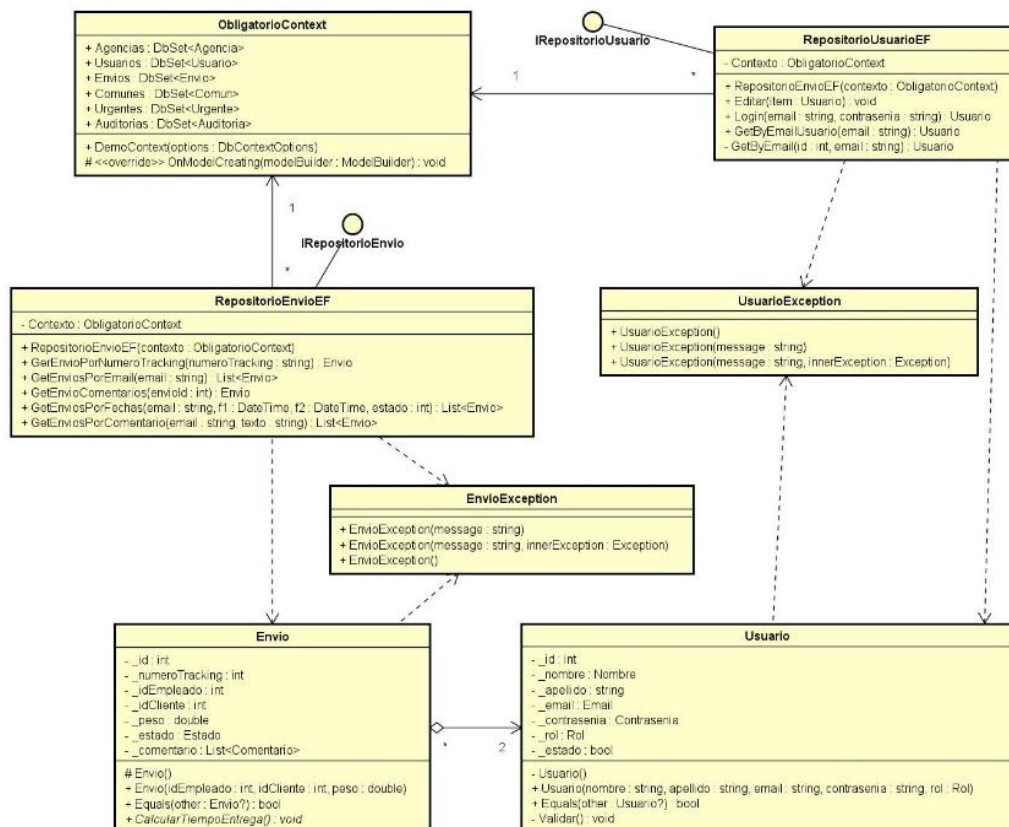
2.1 Entidades de Negocio



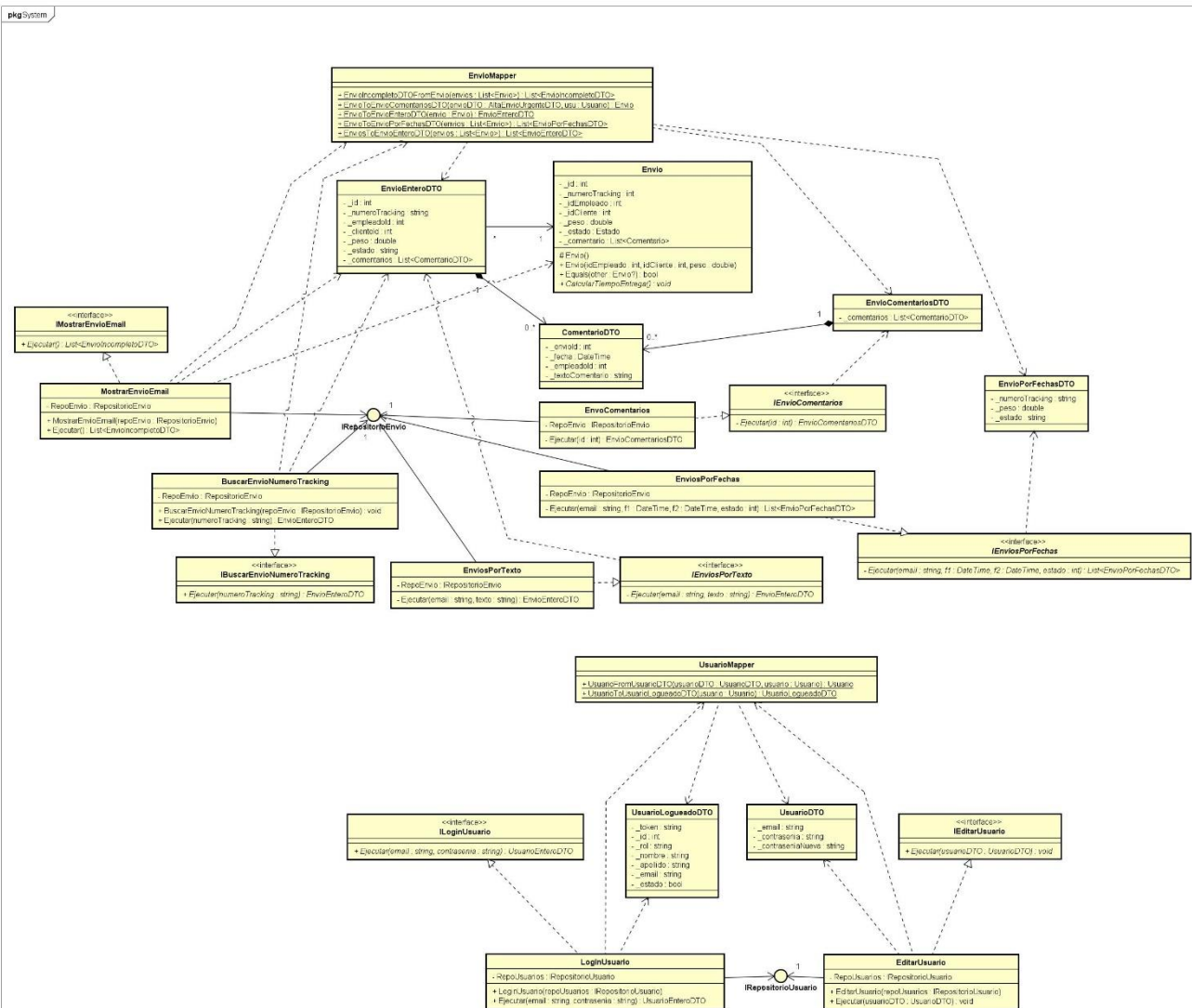
2.2 Compartido



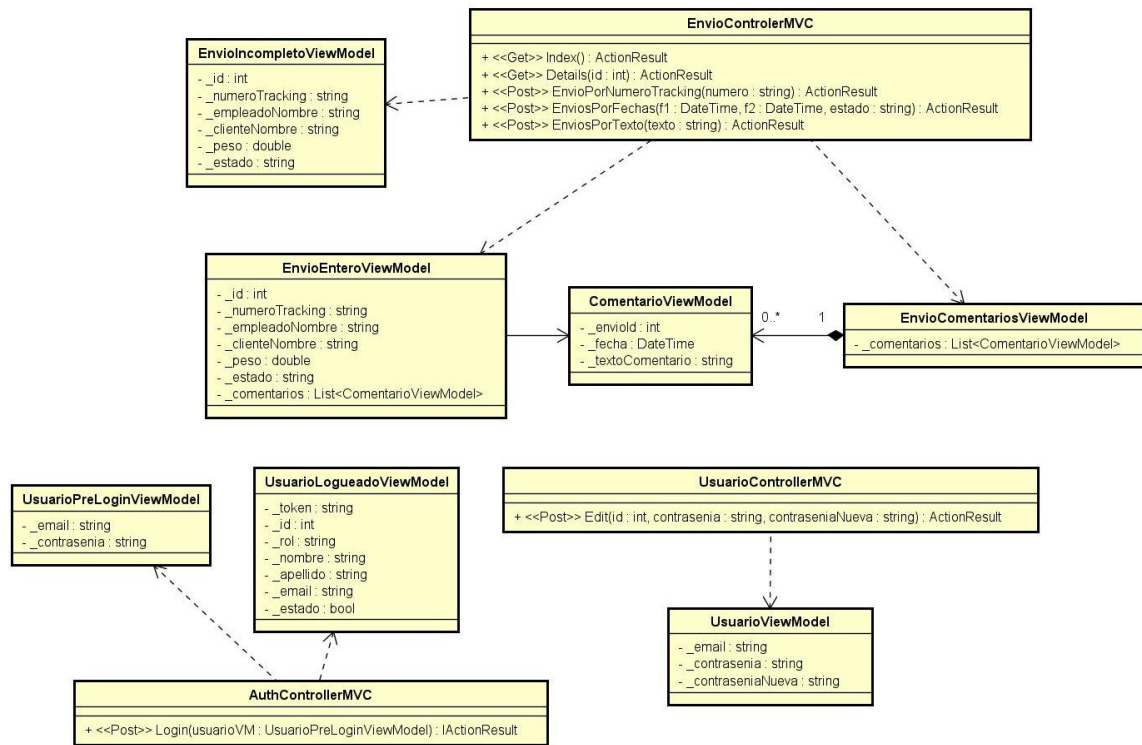
2.3 Lógica Acceso Datos



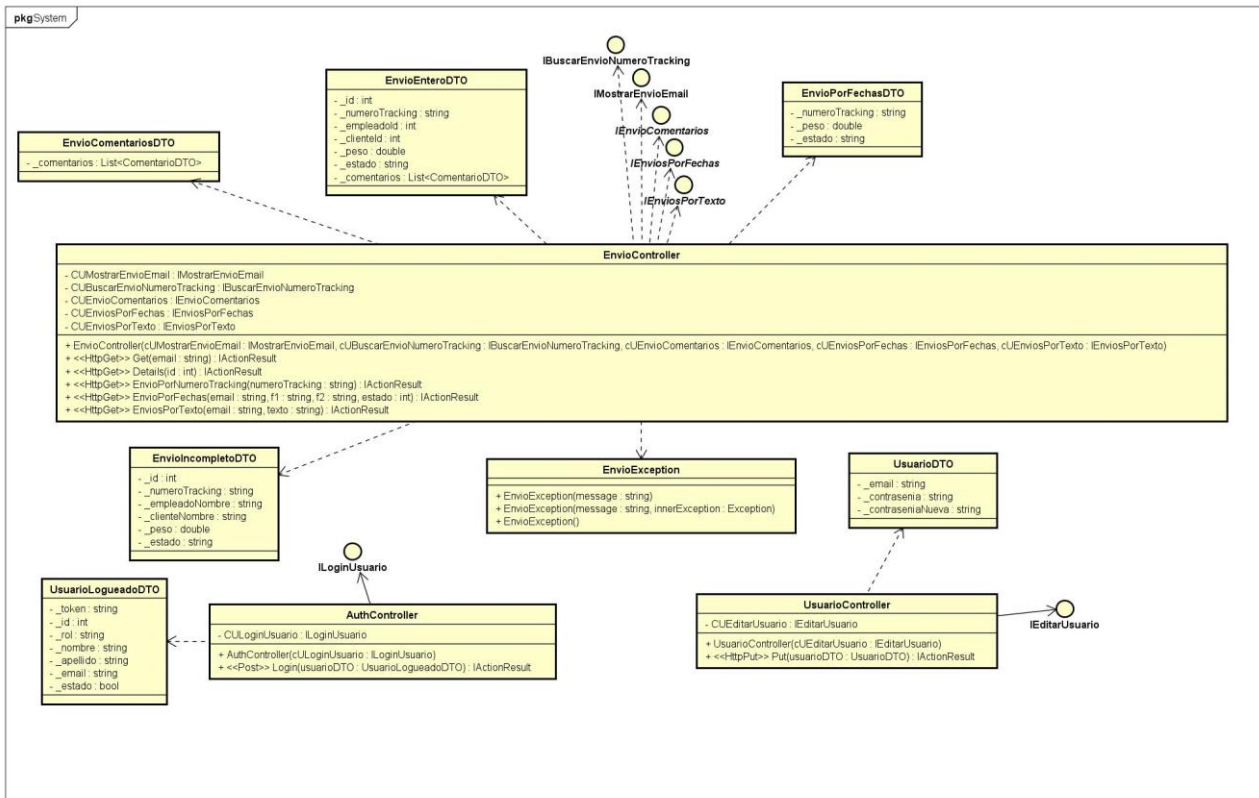
2.4 Logica Aplicación



2.5 MVC



2.6 Web API



3 Código fuente de toda la Aplicación

3.1 API

```
*****
Archivo: ObligatorioContext.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\ObligatorioContext.cs
*****

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

namespace LogicaAccesoDatos
{
    public class ObligatorioContext : DbContext
    {
        public DbSet<Agencia> Agencias { get; set; }
        public DbSet<Usuario> Usuarios { get; set; }
        public DbSet<Envio> Envios { get; set; }
        public DbSet<Comun> Comunidades { get; set; }
        public DbSet<Urgente> Urgentes { get; set; }
        public DbSet<Auditoria> Auditorias { get; set; }

        public ObligatorioContext(DbContextOptions options) : base(options)
        {
        }

        //Introducing FOREIGN KEY constraint 'FK_Curso_Usuario_Usuariold' on table 'Curso' may cause cycles or
        multiple cascade paths.
        //Specify ON DELETE NO ACTION or ON UPDATE NO ACTION, or modify other FOREIGN KEY constraints.
        //Could not create constraint or index. See previous errors.

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Envio>().HasOne(c =>
c.Empleado).WithMany().OnDelete(DeleteBehavior.NoAction);
            modelBuilder.Entity<Envio>().HasOne(c => c.Cliente).WithMany().OnDelete(DeleteBehavior.NoAction);
            base.OnModelCreating(modelBuilder);
        }
    }
}

*****
Archivo: ConflictException.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\ExcepcionesGenericas\Conflict
Exception.cs
*****

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

namespace Compartido.ExcepcionesGenericas
```



```

{
    public class ConflictException:Exception
    {
        public ConflictException() { }
        public ConflictException(string? message, Exception? innerException) :
            base(message, innerException)
        {}
        public ConflictException(string message):
            base(message)
        {}
    }
}

```

Archivo: EnvioMapper.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\Mappers\EnvioMapper.cs

```

using Compartido.DTOs.ComentarioDTOs;
using Compartido.DTOs.EnvioDTOs;
using Compartido.DTOs.UsuarioDTOs;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;

namespace Compartido.Mappers
{
    public class EnvioMapper
    {
        public static List<EnvioIncompletoDTO> EnvioIncompletoDTOFromEnvio(List<Envio> envios)
        {
            List<EnvioIncompletoDTO> mostrarEnviosDTO = new List<EnvioIncompletoDTO>();

            foreach (Envio e in envios)
            {
                EnvioIncompletoDTO mostrarEnvioDTO = new EnvioIncompletoDTO()
                {
                    Id = e.Id,
                    NumeroTracking = e.NumeroTracking,
                    EmpleadoNombre = e.Empleado.Nombre.Valor,
                    ClienteNombre = e.Cliente.Nombre.Valor,
                    Peso = e.Peso,
                    Estado = e.Estado.ToString()
                };

                mostrarEnviosDTO.Add(mostrarEnvioDTO);
            }
            return mostrarEnviosDTO;
        }

        public static EnvioComentariosDTO EnvioToEnvioComentariosDTO(Envio envio)
        {
            EnvioComentariosDTO envioDTO = new EnvioComentariosDTO()
            {
                Comentarios = envio.Comentario.Select(x => new ComentarioDTO
                {

```

```

        Enviold = envio.Id,
        Fecha = x.Fecha,
        EmpleadoId = x.EmpleadoId,
        TextoComentario = x.TextoComentario
    }).ToList()
};

return envioDTO;
}

public static EnvioEnteroDTO EnvioToEnvioEnteroDTO(Envio envio)
{
    EnvioEnteroDTO envioDTO = new EnvioEnteroDTO()
    {
        Id = envio.Id,
        NumeroTracking = envio.NumeroTracking,
        EmpleadoNombre = envio.Empleado.Nombre.Valor,
        ClienteNombre = envio.Cliente.Nombre.Valor,
        Peso = envio.Peso,
        Estado = envio.Estado.ToString(),
        Comentarios = envio.Comentario.Select(x => new ComentarioDTO
        {
            Enviold = envio.Id,
            Fecha = x.Fecha,
            EmpleadoId = x.EmpleadoId,
            TextoComentario = x.TextoComentario
        }).ToList()
    };
    return envioDTO;
}

public static List<EnvioPorFechasDTO> EnvioToEnvioPorFechasDTO(List<Envio> envios)
{
    List<EnvioPorFechasDTO> enviosPorFechasDTO = new List<EnvioPorFechasDTO>();

    foreach (Envio e in envios)
    {
        EnvioPorFechasDTO envioPorFechasDTO = new EnvioPorFechasDTO()
        {
            NumeroTracking = e.NumeroTracking,
            Peso = e.Peso,
            Estado = e.Estado.ToString()
        };

        enviosPorFechasDTO.Add(envioPorFechasDTO);
    }
    return enviosPorFechasDTO;
}

public static List<EnvioEnteroDTO> EnviosToEnvioEnteroDTO(List<Envio> envios)
{
    List<EnvioEnteroDTO> enviosDTO = new List<EnvioEnteroDTO>();
    foreach (Envio e in envios)
    {
        EnvioEnteroDTO envioDTO = new EnvioEnteroDTO()
        {
            Id = e.Id,
            NumeroTracking = e.NumeroTracking,
            EmpleadoNombre = e.Empleado.Nombre.Valor,
            ClienteNombre = e.Cliente.Nombre.Valor,
            Peso = e.Peso,
            Estado = e.Estado.ToString()
        }
    }
}

```

```

        };
        enviosDTO.Add(envioDTO);
    }

    return enviosDTO;
}
}
}

```

Archivo: UsuarioMapper.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\Mappers\UsuarioMapper.cs

```

using Compartido.DTOs.UsuarioDTOs;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Compartido.Mappers

```

```

{
    public class UsuarioMapper
    {
        public static Usuario UsuarioFromUsuarioDTO(UsuarioDTO usuarioDTO, Usuario usuario)
        {
            if (usuarioDTO == null)
            {
                throw new ArgumentNullException("Datos incorrectos");
            }

            return new Usuario(usuario.Nombre.Valor, usuario.Apellido, usuarioDTO.Email,
usuarioDTO.ContraseniaNueva, usuario.Rol);
        }
    }
}

```

```

        public static UsuarioLogueadoDTO UsuarioToUsuarioLogueadoDTO(Usuario usuario)
        {
            if (usuario != null)
            {
                UsuarioLogueadoDTO usuarioEnteroDTO = new UsuarioLogueadoDTO()
                {
                    Id = usuario.Id,
                    Rol = usuario.Rol.ToString(),
                    Nombre = usuario.Nombre.Valor,
                    Apellido = usuario.Apellido,
                    Email = usuario.Email.Valor,
                    Estado = usuario.Estado
                };
                return usuarioEnteroDTO;
            }
            else
            {
                throw new UsuarioException("Credenciales Incorrectas");
            }
        }
    }
}
}

```

Archivo: 20250519115046_Primer.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Migrations\20250519115046_Primer.cs

using System;

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace LogicaAccesoDatos.Migrations

{

/// <inheritdoc />

public partial class Primera : Migration

{

/// <inheritdoc />

protected override void Up(MigrationBuilder migrationBuilder)

{

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),

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

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

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

},

constraints: table =>

{

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

});

migrationBuilder.CreateTable(

name: "Auditorias",

columns: table => new

{

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

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

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

Fecha = table.Column<DateTime>(type: "datetime2", nullable: false),

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

},

constraints: table =>

{

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

});

migrationBuilder.CreateTable(

name: "Usuarios",

columns: table => new

{

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

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

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

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

Estado = table.Column<bool>(type: "bit", nullable: false),

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

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

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

},

constraints: table =>

```

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

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),
        IdEmpleado = table.Column<int>(type: "int", nullable: false),
        IdCliente = table.Column<int>(type: "int", nullable: false),
        Peso = table.Column<double>(type: "float", nullable: false),
        Estado = table.Column<int>(type: "int", nullable: false),
        Discriminator = table.Column<string>(type: "nvarchar(8)", maxLength: 8, nullable: false),
        AgencialId = table.Column<int>(type: "int", nullable: true),
        DireccionPostal = table.Column<string>(type: "nvarchar(max)", nullable: true),
        TiempoEntrega = table.Column<TimeSpan>(type: "time", nullable: true),
        Eficiencia = table.Column<bool>(type: "bit", nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Envios", x => x.Id);
        table.ForeignKey(
            name: "FK_Envios_Agencias_AgencialId",
            column: x => x.AgencialId,
            principalTable: "Agencias",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_Envios_Usuarios_IdCliente",
            column: x => x.IdCliente,
            principalTable: "Usuarios",
            principalColumn: "Id");
        table.ForeignKey(
            name: "FK_Envios_Usuarios_IdEmpleado",
            column: x => x.IdEmpleado,
            principalTable: "Usuarios",
            principalColumn: "Id");
    });

migrationBuilder.CreateTable(
    name: "Comentario",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Fecha = table.Column<DateTime>(type: "datetime2", nullable: false),
        IdEmpleado = table.Column<int>(type: "int", nullable: false),
        TextoComentario = table.Column<string>(type: "nvarchar(max)", nullable: false),
        Envioid = table.Column<int>(type: "int", nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Comentario", x => x.Id);
        table.ForeignKey(
            name: "FK_Comentario_Envios_Envioid",
            column: x => x.Envioid,
            principalTable: "Envios",
            principalColumn: "Id");
    });

```

```

        migrationBuilder.CreateIndex(
            name: "IX_Comentario_EnvioId",
            table: "Comentario",
            column: "EnvioId");

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

        migrationBuilder.CreateIndex(
            name: "IX_Envios_IdCliente",
            table: "Envios",
            column: "IdCliente");

        migrationBuilder.CreateIndex(
            name: "IX_Envios_IdEmpleado",
            table: "Envios",
            column: "IdEmpleado");
    }

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

        migrationBuilder.DropTable(
            name: "Comentario");

        migrationBuilder.DropTable(
            name: "Envios");

        migrationBuilder.DropTable(
            name: "Agencias");

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

*****
Archivo: 20250519115046_Primer.Designer.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Migrations\20250519115046_Primer.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(ObligatorioContext))]

```

```

[Migration("20250519115046_Primer")]
partial class Primera
{
    /// <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<double>("Latitud")
                .HasColumnType("float");

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

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

            b.ComplexProperty<Dictionary<string, object>>("DireccionPostal",
"LogicaNegocio.EntidadesNegocio.Agencia.DireccionPostal#DireccionPostal", b1 =>
            {
                b1.IsRequired();

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

            b.HasKey("Id");

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

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

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

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

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

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

```

```

        b.HasKey("Id");

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

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

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

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

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

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

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

    b.HasKey("Id");

    b.HasIndex("Envioid");

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

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

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

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

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

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

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

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

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

```



```

        b.HasKey("Id");

        b.HasIndex("IdCliente");

        b.HasIndex("IdEmpleado");

        b.ToTable("Envios");

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

        b.UseTphMappingStrategy();
    });

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<bool>("Estado")
        .HasColumnType("bit");

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

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

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

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

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

    b.ComplexProperty<Dictionary<string, object>>("Nombre",
"LogicaNegocio.EntidadesNegocio.Usuario.Nombre#Nombre", 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>("Eficiencia")
        .HasColumnType("bit");

    b.Property<TimeSpan?>("TiempoEntrega")
        .HasColumnType("time");

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

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comentario", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Envio", null)
        .WithMany("Comentario")
        .HasForeignKey("EnvioId");
});

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

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

    b.Navigation("Cliente");

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

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("Comentario");
    });
#pragma warning restore 612, 618
    }
}
}

*****
Archivo: 20250519182707_Segunda.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Migrations\20250519182
707_Segunda.cs
*****
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace LogicaAccesoDatos.Migrations
{
    /// <inheritdoc />
    public partial class Segunda : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.RenameColumn(
                name: "IdEmpleado",
                table: "Comentario",
                newName: "Empleadold");
        }

        /// <inheritdoc />
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.RenameColumn(
                name: "Empleadold",
                table: "Comentario",
                newName: "IdEmpleado");
        }
    }
}

*****
Archivo: 20250519182707_Segunda.Designer.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Migrations\20250519182
707_Segunda.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(ObligatorioContext))]
    [Migration("20250519182707_Segunda")]
    partial class Segunda
    {
        /// <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<double>("Latitud")
                    .HasColumnType("float");

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

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

                b.ComplexProperty<Dictionary<string, object>>("DireccionPostal",
"LogicaNegocio.EntidadesNegocio.Agencia.DireccionPostal#DireccionPostal", b1 =>
                {
                    b1.IsRequired();

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

                b.HasKey("Id");

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

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

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

                b.Property<string>("Accion")

```

```

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

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

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

b.HasKey("Id");

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

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

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

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

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

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

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

    b.HasKey("Id");

    b.HasIndex("Enviold");

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

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

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

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

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

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

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

```

```

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

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

        b.HasKey("Id");

        b.HasIndex("IdCliente");

        b.HasIndex("IdEmpleado");

        b.ToTable("Envios");

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

        b.UseTphMappingStrategy();
    });

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<bool>("Estado")
        .HasColumnType("bit");

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

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

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

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

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

    b.ComplexProperty<Dictionary<string, object>>("Nombre",
"LogicaNegocio.EntidadesNegocio.Usuario.Nombre#Nombre", 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>("Eficiencia")
        .HasColumnType("bit");

    b.Property<TimeSpan?>("TiempoEntrega")
        .HasColumnType("time");

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

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comentario", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Envio", null)
        .WithMany("Comentario")
        .HasForeignKey("EnvioId");
});

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

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

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

```

```

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

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("Comentario");
});
#pragma warning restore 612, 618
}
}
}

*****
Archivo: ObligatorioContextModelSnapshot.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Migrations\ObligatorioContextModelSnapshot.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(ObligatorioContext))]
    partial class ObligatorioContextModelSnapshot : 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<double>("Latitud")

```



```

        .HasColumnType("float");

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

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

b.ComplexProperty<Dictionary<string, object>>("DireccionPostal",
"LogicaNegocio.EntidadesNegocio.Agencia.DireccionPostal#DireccionPostal", b1 =>
{
    b1.IsRequired();

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

b.HasKey("Id");

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

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

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

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

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

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

    b.HasKey("Id");

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

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

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

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

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

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

```

```

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

        b.HasKey("Id");

        b.HasIndex("Enviold");

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

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

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

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

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

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

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

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

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

    b.HasKey("Id");

    b.HasIndex("IdCliente");

    b.HasIndex("IdEmpleado");

    b.ToTable("Envios");

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

    b.UseTphMappingStrategy();
});

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<bool>("Estado")
    .HasColumnType("bit");

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

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

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

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

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

b.ComplexProperty<Dictionary<string, object>>("Nombre",
"LogicaNegocio.EntidadesNegocio.Usuario.Nombre#Nombre", 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>("Eficiencia")
            .HasColumnType("bit");

        b.Property<TimeSpan?>("TiempoEntrega")
            .HasColumnType("time");

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

modelBuilder.Entity("LogicaNegocio.EntidadesNegocio.Comentario", b =>
{
    b.HasOne("LogicaNegocio.EntidadesNegocio.Envio", null)
        .WithMany("Comentario")
        .HasForeignKey("EnvioId");
});

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

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

    b.Navigation("Cliente");

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

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("Comentario");
});
#pragma warning restore 612, 618
    }
}
}

*****
Archivo: RepositorioEnvioEF.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Repositorios\Repositorio
EnvioEF.cs
*****
using System;

```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.InterfaceRepositorios;
using Microsoft.EntityFrameworkCore;

namespace LogicaAccesoDatos.Repositorios
{
    public class RepositorioEnvioEF : IRepositorioEnvio
    {
        private ObligatorioContext Contexto { get; set; }
        public RepositorioEnvioEF(ObligatorioContext contexto)
        {
            Contexto = contexto;
        }

        public void Alta(Envio item)
        {
            throw new NotImplementedException();
        }

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

        public void Editar(Envio item)
        {
            throw new NotImplementedException();
        }

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

        public Envio GetPorId(int id)
        {
            throw new NotImplementedException();
        }

        public void AltaComentario(int idEnvio, Comentario comentario)
        {
            throw new NotImplementedException();
        }

        public Envio GetEnvioPorNumeroTracking(string numeroTracking)
        {
            return Contexto.Envios.Where(t => t.NumeroTracking == numeroTracking)
                .Include(envio => envio.Empleado)
                .Include(envio => envio.Cliente)
                .Include(envio => envio.Comentario)
                .SingleOrDefault();
        }

        public List<Envio> GetEnviosPorEmail(string email)
        {
            return Contexto.Envios.Where(e => e.Cliente.Email.Valor == email)
                .Include(e => e.Empleado)
                .Include(e => e.Cliente)
        }
    }
}

```

```

        .Include(e => e.Comentario)
        .OrderBy(e => e.Comentario[0].Fecha)
        .ToList();
    }

    public Envio GetEnvioComentarios(int enviold)
    {
        return Contexto.Envios.Where(e => e.Id == enviold).Include(e => e.Comentario).SingleOrDefault();
    }

    public List<Envio> GetEnviosPorFechas(string email, DateTime f1, DateTime f2, int estado)
    {
        return Contexto.Envios.Include(e => e.Comentario).Where(e => e.Cliente.Email.Valor == email
            && e.Comentario.Any(c => c.Fecha >= f1 && c.Fecha <= f2)
            && (int)e.Estado == estado)
            .OrderBy(e => e.NumeroTracking)
            .ToList();
    }

    public List<Envio> GetEnviosPorComentario(string email, string texto)
    {
        return Contexto.Envios.Include(e => e.Comentario)
            .Where(e => e.Cliente.Email.Valor == email
            && e.Comentario.Any(c => c.TextoComentario.Contains(texto)))
            .Include(e => e.Empleado)
            .Include(e => e.Cliente)
            .OrderBy(e => e.Comentario[0].Fecha)
            .ToList();
    }
}

```

Archivo: RepositorioUsuarioEF.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAccesoDatos\Repositorios\Repositorio
 UsuarioEF.cs

```

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

namespace LogicaAccesoDatos.Repositorios
{
    public class RepositorioUsuarioEF : IRepositoryUsuario
    {
        private ObligatorioContext Contexto { get; set; }

        public RepositorioUsuarioEF(ObligatorioContext contexto)
        {
            Contexto = contexto;
        }

        public void Alta(Usuario item)
        {
            throw new NotImplementedException();
        }
    }
}

```

```

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

public void Editar(Usuario item)
{
    Usuario usuarioBuscado = GetByEmail(item.Id, item.Email.Valor);

    if (usuarioBuscado == null)
    {
        Contexto.ChangeTracker.Clear();
        Contexto.Usuarios.Update(item);
        Contexto.SaveChanges();
    }
    else
    {
        throw new UsuarioException("Usuario ya existente");
    }
}

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

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

public Usuario Login(string email, string contrasenia)
{
    Usuario ret = Contexto.Usuarios.Where(c => c.Email.Valor == email
        && c.Contrasenia.Valor == contrasenia).SingleOrDefault();

    if (ret == null)
    {
        throw new UsuarioException("Datos incorrectos");
    }

    return ret;
}

public Usuario GetByEmailUsuario(string email)
{
    Usuario ret = Contexto.Usuarios.Where(c => c.Email.Valor == email).SingleOrDefault();
    return ret;
}

private Usuario GetByEmail(int id, string email)
{
    Usuario ret = Contexto.Usuarios.Where(c => c.Id != id && c.Email.Valor == email).SingleOrDefault();

    if (ret != null)
    {
        throw new UsuarioException("Email incorrecto");
    }

    return ret;
}
}

```

```
}
```

```
*****
```

Archivo: Agencia.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Agencia.cs

```
*****
```

```
using LogicaNegocio.ExcepcionesEntidades;
```

```
using LogicaNegocio.ValueObject.Agencia;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaNegocio.EntidadesNegocio
```

```
{
```

```
    public class Agencia : IEquatable<Agencia>
```

```
    {
```

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

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

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

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

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

```
        private Agencia() { }
```

```
        public Agencia(string nombre, string direccionPostal, double latitud, double longitud)
```

```
        {
```

```
            Nombre = nombre;
```

```
            DireccionPostal = new DireccionPostal(direccionPostal);
```

```
            Latitud = latitud;
```

```
            Longitud = longitud;
```

```
            Validar();
```

```
        }
```

```
        private void Validar()
```

```
        {
```

```
            if (string.IsNullOrEmpty(Nombre)
```

```
                || string.IsNullOrWhiteSpace(Nombre)
```

```
                || string.IsNullOrEmpty(DireccionPostal.Value)
```

```
                || string.IsNullOrWhiteSpace(DireccionPostal.Value)
```

```
                || Latitud == 0
```

```
                || Longitud == 0
```

```
            )
```

```
            {
```

```
                throw new AgenciaException("No pueden haber datos vacios");
```

```
            }
```

```
        }
```

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

```
        {
```

```
            return Id == other.Id;
```

```
        }
```

```
    }
```

```
}
```

```
*****
```

Archivo: Auditoria.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Auditoria.c

s

```
*****
```

```
using LogicaNegocio.ExcepcionesEntidades;
```

```
using System;
```

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



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

```
namespace LogicaNegocio.EntidadesNegocio
{
    public class Auditoria : IEquatable<Auditoria>
    {
        public int Id { get; set; }
        public string Accion { get; set; }
        public DateTime Fecha { get; set; }
        public int IdEmpleado { get; set; }

        private Auditoria() { }
        public Auditoria(string accion, DateTime fecha, int idEmpleado)
        {
            Accion = accion;
            Fecha = fecha;
            IdEmpleado = idEmpleado;
            Validar();
        }
        private void Validar()
        {
            if (string.IsNullOrEmpty(Accion) || string.IsNullOrWhiteSpace(Accion))
            {
                throw new AuditoriaException("La accion no debe estar vacia");
            }
        }

        public bool Equals(Auditoria? other)
        {
            return Id == other.Id;
        }
    }
}
```

Archivo: Comentario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Comentari
o.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 Comentario : IEquatable<Comentario>
    {
        public int Id { get; set; }
        public DateTime Fecha { get; set; }
        public int EmpleadoId { get; set; }
        public string TextoComentario { get; set; }

        private Comentario() { }
        public Comentario(DateTime fecha, int empleadoId, string textoComentario)
        {
            Fecha = fecha;
            EmpleadoId = empleadoId;
        }
    }
}
```

```

        TextoComentario = textoComentario;
        Validar();
    }

    private void Validar()
    {
        if (string.IsNullOrEmpty(TextoComentario) || string.IsNullOrWhiteSpace(TextoComentario))
        {
            throw new ComentarioException("El texto no puede estar vacio");
        }
    }

    public bool Equals(Comentario? other)
    {
        return Id == other.Id;
    }
}

*****
Archivo: Comun.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\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; }

        protected Comun():base() { }

        public Comun(int idEmpleado, int idCliente, double peso, Agencia agencia) : base(idEmpleado, idCliente,
peso)
        {
            Agencia = agencia;
        }

        public override void CalcularTiempoEntrega()
        {
        }

        public override void Validar()
        {
            if (Agencia == null)
            {
                throw new EnvioException("Agencia no encontrada");
            }
        }
    }
}

*****
Archivo: Envio.cs
Carpeta:
C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\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;

namespace LogicaNegocio.EntidadesNegocio
{
    public abstract class Envio : IEquatable<Envio>
    {
        private static int _ultimoTracking = 120;
        public int Id { get; set; }
        public string NumeroTracking { get; set; }

        public Usuario Empleado { get; set; }
        [ForeignKey("Empleado")]
        public int IdEmpleado { get; set; }

        public Usuario Cliente { get; set; }
        [ForeignKey("Cliente")]
        public int IdCliente { get; set; }

        public double Peso { get; set; }
        public Estado Estado { get; set; }
        public List<Comentario> Comentario { get; set; } = new List<Comentario>();

        protected Envio()
        {
            NumeroTracking = _ultimoTracking.ToString();
            Estado = Estado.EN_PROCESO;
            _ultimoTracking += _ultimoTracking / 3;
        }

        public Envio(int idEmpleado, int idCliente, double peso)
        {
            NumeroTracking = _ultimoTracking.ToString();
            IdEmpleado = idEmpleado;
            IdCliente = idCliente;
            Peso = peso;
            Estado = Estado.EN_PROCESO;
            _ultimoTracking += _ultimoTracking / 3;
        }

        public bool Equals(Envio? other)
        {
            {
                return Id == other.Id;
            }
        }

        public abstract void CalcularTiempoEntrega();

        public abstract void Validar();
    }
}
```

Archivo: Estado.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Estado.cs

```
using System;
```

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

```
namespace LogicaNegocio.EntidadesNegocio
{
    public enum Estado
    {
        EN_PROCESO,
        FINALIZADO
    }
}
```

Archivo: Rol.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Rol.cs

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

```
namespace LogicaNegocio.EntidadesNegocio
{
    public enum Rol
    {
        Administrador,
        Funcionario,
        Cliente
    }
}
```

Archivo: Urgente.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Urgente.cs

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

```
namespace LogicaNegocio.EntidadesNegocio
{
    public class Urgente : Envio
    {
        public string DireccionPostal { get; set; }
        public TimeSpan? TiempoEntrega { get; set; }
        public bool Eficiencia { get; set; }
```

```
        protected Urgente() : base() { }
```

```
        public Urgente(int idEmpleado, int idCliente, double peso, string direccionPostal, TimeSpan?
tiempoEntrega, bool eficiencia) : base(idEmpleado, idCliente, peso)
```

```
        {
            DireccionPostal = direccionPostal;
            TiempoEntrega = tiempoEntrega;
```

```

        Eficiencia = eficiencia;
    }

    public override void CalcularTiempoEntrega()
    {
        DateTime ingreso = DateTime.Now;
        DateTime finalizado = DateTime.Now;
        foreach (Comentario c in Comentario)
        {
            if (c.TextoComentario == "Ingresado en agencia de origen")
            {
                ingreso = c.Fecha;
            }
            else if (c.TextoComentario == "Finalizado")
            {
                finalizado = c.Fecha;
            }
        }

        TiempoEntrega = finalizado - ingreso;

        if (TiempoEntrega < new TimeSpan(24, 0, 0))
        {
            Eficiencia = true;
        }
        else
        {
            Eficiencia = false;
        }
    }

    public override void Validar()
    {
        if (string.IsNullOrEmpty(DireccionPostal) || string.IsNullOrWhiteSpace(DireccionPostal))
        {
            throw new EnvioException("La direccion postal es obligatoria");
        }
    }
}
}
}

```

Archivo: Usuario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\EntidadesNegocio\Usuario.cs

```

using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.ValueObject.Usuario;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Principal;
using System.Text;
using System.Threading.Tasks;

```

```

namespace LogicaNegocio.EntidadesNegocio
{
    public class Usuario : IEquatable<Usuario>
    {
        public int Id { get; set; }
        public Nombre Nombre { get; set; }
        public string Apellido { get; set; }
        public Email Email { get; set; }
    }
}

```

```

    public Contraseña Contraseña { get; set; }
    public Rol Rol { get; set; }
    public bool Estado { get; set; }

    private Usuario() { }
    public Usuario(string nombre, string apellido, string email, string contraseña, Rol rol)
    {
        Nombre = new Nombre(nombre);
        Apellido = apellido;
        Email = new Email(email);
        Contraseña = new Contraseña(contraseña);
        Rol = rol;
        Estado = true;
        Validar();
    }

    private void Validar()
    {
        if (string.IsNullOrEmpty(Apellido) || string.IsNullOrWhiteSpace(Apellido))
        {
            throw new UsuarioException("No pueden haber datos vacios");
        }
    }

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

```

Archivo: AgenciaException.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\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: AuditoriaException.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\ExcepcionesEntidades\AuditoriaException.cs

```

using System;
using System.Collections.Generic;

```

```

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

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

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

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

    }
}

```

Archivo: ComentarioException.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\ExcepcionesEntidades\ComentarioException.cs

```

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

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

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

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

    }

}

```

Archivo: EnvioException.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\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\Usuario\ObligatorioMaverickLopez335912\Obligatorio\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\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\InterfaceRepositorios\IRepositorio.cs

```

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

```

```

namespace LogicaNegocio.InterfaceRepositorios

```

```

{
    public interface IRepository<T>
    {
        void Alta(T item);
        IEnumerable<T> GetAll();
        T GetPorId(int id);
        void Baja(int id);
        void Editar(T item);
    }
}

```

Archivo: IRepositoryEnvio.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\InterfaceRepositorios\IRepositorioEnvio.cs

```

using LogicaNegocio.EntidadesNegocio;
using System;
using System.Collections.Generic;

```



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

```
namespace LogicaNegocio.InterfaceRepositorios
{
    public interface IRepositoryioEnvio : IRepositoryio<Envio>
    {
        public void AltaComentario(int idEnvio, Comentario comentario);
        public Envio GetEnvioPorNumeroTracking(string numeroTracking);
        public List<Envio> GetEnviosPorEmail(string email);
        public Envio GetEnvioComentarios(int envioid);
        public List<Envio> GetEnviosPorFechas(string email, DateTime f1, DateTime f2, int estado);
        public List<Envio> GetEnviosPorComentario(string email, string texto);
    }
}
```

Archivo: IRepositoryioUsuario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\InterfaceRepositorios\IRepositoryioUsuario.cs

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

```
namespace LogicaNegocio.InterfaceRepositorios
{
    public interface IRepositoryioUsuario : IRepositoryio<Usuario>
    {
        public Usuario Login(string nombre, string contrasenia);
        public Usuario GetByEmailUsuario(string email);
    }
}
```

Archivo: AuthController.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\WebAPI\Controllers\AuthController.cs

```
using Compartido.DTOs.UsuarioDTOs;
using LogicaAplicacion.InterfaceCasosUso.UsuarioCU;
using LogicaNegocio.ExcepcionesEntidades;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore.Metadata.Internal;
using WebAPI.Autenticacion;
```

```
namespace WebAPI.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class AuthController : ControllerBase
    {
        public ILoginUsuario CULogin { get; set; }
        public AuthController(ILoginUsuario cULogin)
        {
            CULogin = cULogin;
        }
    }
}
```

```

[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
[HttpPost]
public IActionResult Login(UsuarioPreLoginDTO usuarioDTO)
{
    try
    {
        if (usuarioDTO == null)
        {
            return BadRequest("Datos incorrectos");
        }

        UsuarioLogueadoDTO usuarioLogueadoDTO = CULogin.Ejecutar(usuarioDTO.Email,
usuarioDTO.Contrasenia);

        if (usuarioLogueadoDTO != null)
        {
            usuarioLogueadoDTO.Token = ManejadorToken.CrearToken(usuarioLogueadoDTO,
usuarioDTO.Email);
            return Ok(usuarioLogueadoDTO);
        }
        else
        {
            return BadRequest("Datos incorrectos");
        }
    }
    catch (UsuarioException ex)
    {
        return NotFound(ex.Message);
    }
    catch (Exception ex)
    {
        return StatusCode(500, "Error");
    }
}
}
}

```

Archivo: EnvioController.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\WebAPI\Controllers\EnvioController.cs

```

using Compartido.DTOs.EnvioDTOs;
using LogicaAplicacion.InterfaceCasosUso.EnvioCU;
using LogicaNegocio.ExcepcionesEntidades;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using static System.Runtime.InteropServices.JavaScript.JSType;

```

namespace WebAPI.Controllers

```

{
    [Route("api/[controller]")]
    [ApiController]
    public class EnvioController : ControllerBase
    {
        private IMostrarEnvioEmail CUMostrarEnvioEmail { get; set; }
        private IEnvioComentarios CUEnvioComentarios { get; set; }
        private IBuscarEnvioNumeroTracking CUBuscarEnvioNumeroTracking { get; set; }
    }
}

```

```

private IEnviosPorFechas CUEnviosPorFechas { get; set; }
private IEnviosPorTexto CUEnviosPorTexto { get; set; }

public EnvioController(IMostrarEnvioEmail cUMostrarEnvioEmail,
    IEnvioComentarios cUEnvioComentarios,
    IBuscarEnvioNumeroTracking cUBuscarEnvioNumeroTracking,
    IEnviosPorFechas cUEnviosPorFechas,
    IEnviosPorTexto cUEnviosPorTexto)
{
    CUMostrarEnvioEmail = cUMostrarEnvioEmail;
    CUEnvioComentarios = cUEnvioComentarios;
    CUBuscarEnvioNumeroTracking = cUBuscarEnvioNumeroTracking;
    CUEnviosPorFechas = cUEnviosPorFechas;
    CUEnviosPorTexto = cUEnviosPorTexto;
}

[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
// GET: api/<EnvioController>
[Authorize(Roles = "Cliente")]
[HttpGet("BuscarPorEmail/{email}")]
public IActionResult Get(string email)
{
    try
    {
        if (email == null)
        {
            return BadRequest("El email ingresado no es valido");
        }

        List<EnvioIncompletoDTO> listaEnviosDTO = CUMostrarEnvioEmail.Ejecutar(email);

        if (listaEnviosDTO != null)
        {
            return Ok(listaEnviosDTO);
        }
        else
        {
            return NotFound("No existen envios");
        }
    }
    catch (EnvioException ex)
    {
        return NotFound(ex.Message);
    }
    catch (Exception ex)
    {
        return StatusCode(500, "Error");
    }
}

[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
[Authorize(Roles = "Cliente")]
[HttpGet("Detalles/{id}")]
public IActionResult Details(int id)
{

```

```

try
{
    if (id == null)
    {
        return BadRequest("El id recibido no es valido");
    }

    EnvioComentariosDTO envioDTO = CUEnvioComentarios.Ejecutar(id);

    if (envioDTO != null)
    {
        return Ok(envioDTO);
    }
    else
    {
        return NotFound("No existen envios");
    }
}
catch (EnvioException ex)
{
    return NotFound(ex.Message);
}
catch (Exception ex)
{
    return StatusCode(500, "Error");
}
}

```

```

[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
// GET api/<EnvioController>/5
[HttpGet("BuscarPorNumeroTracking/{numeroTracking}")]
public IActionResult EnvioPorNumeroTracking(string numeroTracking)
{
    try
    {
        if (numeroTracking == null)
        {
            return BadRequest("El Numero de tracking recibido no es valido");
        }

        EnvioEnteroDTO envioDTO = CUBuscarEnvioNumeroTracking.Ejecutar(numeroTracking);

        if (envioDTO != null)
        {
            return Ok(envioDTO);
        }
        else
        {
            return BadRequest("Datos incorrectos");
        }
    }
    catch (EnvioException ex)
    {
        return NotFound(ex.Message);
    }
    catch (Exception ex)
    {
        return StatusCode(500, "Error");
    }
}

```

```

    }
}

[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
[Authorize(Roles = "Cliente")]
[HttpGet("EnvioPorFechas/{email}/{f1}/{f2}/{estado}")]
public IActionResult EnvioPorFechas(string email, string f1, string f2, int estado)
{
    try
    {
        if (f1 == null || f2 == null)
        {
            return BadRequest("Las fechas recibidas no son validas");
        }

        DateTime.TryParse(f1, out DateTime fecha1);
        DateTime.TryParse(f2, out DateTime fecha2);

        List<EnvioPorFechasDTO> envioDTO = CUEnviosPorFechas
            .Ejecutar(email, fecha1, fecha2, estado);

        if (envioDTO != null)
        {
            return Ok(envioDTO);
        }
        else
        {
            return BadRequest("Datos incorrectos");
        }
    }
    catch (EnvioException ex)
    {
        return NotFound(ex.Message);
    }
    catch (Exception ex)
    {
        return StatusCode(500, "Error");
    }
}

```

```

[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(StatusCodes.Status500InternalServerError)]
[Authorize(Roles = "Cliente")]
[HttpGet("EnviosPorTexto/{email}/{texto}")]
public IActionResult EnviosPorTexto(string email, string texto)
{
    try
    {
        if (string.IsNullOrEmpty(texto) || string.IsNullOrEmpty(email))
        {
            return BadRequest("El texto recibido no es valido");
        }

        List<EnvioEnteroDTO> envioDTO = CUEnviosPorTexto.Ejecutar(email, texto);

        if (envioDTO != null)

```

```

        {
            return Ok(envioDTO);
        }
        else
        {
            return BadRequest("Datos incorrectos");
        }
    }
    catch (EnvioException ex)
    {
        return NotFound(ex.Message);
    }
    catch (Exception ex)
    {
        return StatusCode(500, "Error");
    }
}
}
}

```

Archivo: UsuarioController.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\WebAPI\Controllers\UsuarioController.cs

```

using Compartido.DTOs.UsuarioDTOs;
using Compartido.ExcepcionesGenericas;
using LogicaAplicacion.InterfaceCasosUso.UsuarioCU;
using LogicaNegocio.ExcepcionesEntidades;
using Microsoft.AspNetCore.Authorization;
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

```

{
    [Authorize(Roles = "Cliente")]
    [Route("api/[controller]")]
    [ApiController]
    public class UsuarioController : ControllerBase
    {
        public IEditarUsuario CUEditarUsuario { get; set; }

        public UsuarioController(IEditarUsuario cUEditarUsuario)
        {
            CUEditarUsuario = cUEditarUsuario;
        }

        [ProducesResponseType(StatusCodes.Status200OK)]
        [ProducesResponseType(StatusCodes.Status400BadRequest)]
        [ProducesResponseType(StatusCodes.Status401Unauthorized)]
        [ProducesResponseType(StatusCodes.Status404NotFound)]
        [ProducesResponseType(StatusCodes.Status409Conflict)]
        [ProducesResponseType(StatusCodes.Status500InternalServerError)]
        // PUT api/<UsuarioController>/5
        [HttpPut]
        public IActionResult Put([FromBody] UsuarioDTO usuarioDTO)
        {
            try
            {
                if (usuarioDTO == null)
                {

```

```

        return BadRequest("Datos incorrectos");
    }

    CUEditarUsuario.Ejecutar(usuarioDTO);
    return Ok();
}
catch (UsuarioException ex)
{
    return BadRequest(ex.Message);
}
catch (ConflictException ex)
{
    return Conflict(ex.Message);
}
catch (Exception ex)
{
    return StatusCode(500, "Error interno");
}
}
}
}

```

Archivo: ManejadorToken.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\WebAPI\JWT\ManejadorToken.cs

```

using Compartido.DTOs.UsuarioDTOs;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;

namespace WebAPI.Autenticacion
{
    public class ManejadorToken
    {
        internal static string CrearToken(UsuarioLogueadoDTO usuarioDTO, string email)
        {
            // Poner misma clave que en el program para que ande
            byte[] clave = Encoding.ASCII.GetBytes("ZWRpw6fDo28gZW0gY29tcHV0YWVWRvcme=");

            JwtSecurityTokenHandler tokenHandler = new JwtSecurityTokenHandler();

            //clave secreta, generalmente se incluye en el archivo de configuración
            //Debe ser un vector de bytes

            //Se incluye un claim para el email
            SecurityTokenDescriptor tokenDescriptor = new SecurityTokenDescriptor
            {
                Subject = new ClaimsIdentity(new Claim[]
                {
                    new Claim(ClaimTypes.Email, email),
                    new Claim(ClaimTypes.Role, usuarioDTO.Rol),
                }),
                Expires = DateTime.UtcNow.AddMinutes(30),
                SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(clave),
                    SecurityAlgorithms.HmacSha256Signature)
            };

            var token = tokenHandler.CreateToken(tokenDescriptor);

```

```

        return tokenHandler.WriteToken(token);
    }
}
}

```

Archivo: ComentarioDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\ComentarioDTOs\ComentarioDTO.cs

```

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

```

```

namespace Compartido.DTOs.ComentarioDTOs

```

```

{
    public class ComentarioDTO
    {
        public int Enviold { get; set; }
        public DateTime Fecha { get; set; }
        public int EmpleadoId { get; set; }
        public string TextoComentario { get; set; }
    }
}

```

Archivo: EnvioComentariosDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\EnvioDTOs\EnvioComentariosDTO.cs

```

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

```

```

namespace Compartido.DTOs.EnvioDTOs

```

```

{
    public class EnvioComentariosDTO
    {
        public List<ComentarioDTO> Comentarios { get; set; }
    }
}

```

Archivo: EnvioEnteroDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\EnvioDTOs\EnvioEnteroDTO.cs

```

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

```



```

namespace Compartido.DTOs.EnvioDTOs
{
    public class EnvioEnteroDTO
    {
        public int Id { get; set; }
        public string NumeroTracking { get; set; }
        public string EmpleadoNombre { get; set; }
        public string ClienteNombre { get; set; }
        public double Peso { get; set; }
        public string Estado { get; set; }
        public List<ComentarioDTO> Comentarios { get; set; }
    }
}

```

Archivo: EnvioIncompletoDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\EnvioDTOs\EnvioIncompletoDTO.cs

```

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

```

```

namespace Compartido.DTOs.EnvioDTOs
{
    public class EnvioIncompletoDTO
    {
        public int Id { get; set; }
        public string NumeroTracking { get; set; }
        public string EmpleadoNombre { get; set; }
        public string ClienteNombre { get; set; }
        public double Peso { get; set; }
        public string Estado { get; set; }
    }
}

```

Archivo: EnvioPorFechasDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\EnvioDTOs\EnvioPorFechasDTO.cs

```

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

```

```

namespace Compartido.DTOs.EnvioDTOs
{
    public class EnvioPorFechasDTO
    {
        public string NumeroTracking { get; set; }
        public double Peso { get; set; }
        public string Estado { get; set; }
    }
}

```

Archivo: UsuarioDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\UsuarioDTOs\UsuarioDTO.cs

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

```
namespace Compartido.DTOs.UsuarioDTOs
{
    public class UsuarioDTO
    {
        public string Email { get; set; }
        public string Contraseña { get; set; }
        public string ContraseñaNueva { get; set; }
    }
}
```

Archivo: UsuarioLogueadoDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\UsuarioDTOs\UsuarioLogueadoDTO.cs

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

```
namespace Compartido.DTOs.UsuarioDTOs
{
    public class UsuarioLogueadoDTO
    {
        public string Token { get; set; }
        public int Id { get; set; }
        public string Rol { get; set; }
        public string Nombre { get; set; }
        public string Apellido { get; set; }
        public string Email { get; set; }
        public bool Estado { get; set; }
    }
}
```

Archivo: UsuarioPreLoginDTO.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\Compartido\DTOs\UsuarioDTOs\UsuarioPreLoginDTO.cs

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

```
namespace Compartido.DTOs.UsuarioDTOs
```

```
{
    public class UsuarioPreLoginDTO
    {
        public string Email { get; set; }

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

Archivo: BuscarEnvioNumeroTracking.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\BuscarEnvioNumeroTracking.cs

```
using Compartido.DTOs.EnvioDTOs;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.InterfaceRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class BuscarEnvioNumeroTracking : IBuscarEnvioNumeroTracking
    {
        private IRepositoryEnvio RepoEnvio { get; set; }
        public BuscarEnvioNumeroTracking(IRepositoryEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }
        public EnvioEnteroDTO Ejecutar(string numeroTracking)
        {
            Envio envio = RepoEnvio.GetEnvioPorNumeroTracking(numeroTracking);

            if (envio == null)
            {
                throw new EnvioException("Envio no encontrado");
            }

            return EnvioMapper.EnvioToEnvioEnteroDTO(envio);
        }
    }
}
```

Archivo: EnvioComentarios.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\EnvioComentarios.cs

```
using Compartido.DTOs.EnvioDTOs;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfaceRepositorios;
using System;
using System.Collections.Generic;
```

```

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

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class EnvioComentarios : IEnvioComentarios
    {
        private IRepositorioEnvio RepoEnvio { get; set; }
        public EnvioComentarios(IRepositorioEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }
        public EnvioComentariosDTO Ejecutar(int id)
        {
            Envio envio = RepoEnvio.GetEnvioComentarios(id);

            return EnvioMapper.EnvioToEnvioComentariosDTO(envio);
        }
    }
}

```

Archivo: EnviosPorFechas.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\EnviosPorFechas.cs

```

using Compartido.DTOs.EnvioDTOs;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfaceRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class EnviosPorFechas : IEnviosPorFechas
    {
        private IRepositorioEnvio RepoEnvio { get; set; }
        public EnviosPorFechas(IRepositorioEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }
        public List<EnvioPorFechasDTO> Ejecutar(string email, DateTime f1, DateTime f2, int estado)
        {
            List<Envio> envios = RepoEnvio.GetEnviosPorFechas(email, f1, f2, estado);

            return EnvioMapper.EnvioToEnvioPorFechasDTO(envios);
        }
    }
}

```

Archivo: EnviosPorTexto.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\EnviosPorTexto.cs

```

using Compartido.DTOs.EnvioDTOs;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.InterfaceRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class EnviosPorTexto : IEnviosPorTexto
    {
        private IRepositoryEnvio RepoEnvio { get; set; }
        public EnviosPorTexto(IRepositoryEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }
        public List<EnvioEnteroDTO> Ejecutar(string email, string texto)
        {
            List<Envio> envios = RepoEnvio.GetEnviosPorComentario(email, texto);

            if (envios == null)
            {
                throw new EnvioException("No hay envios");
            }

            return EnvioMapper.EnviosToEnvioEnteroDTO(envios);
        }
    }
}

```

Archivo: MostrarEnvioEmail.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\EnvioCU\MostrarEnvioEmail.cs

```

using Compartido.DTOs.EnvioDTOs;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.InterfaceRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.ImplementacionCasosUso.EnvioCU
{
    public class MostrarEnvioEmail : IMostrarEnvioEmail
    {
        private IRepositoryEnvio RepoEnvio { get; set; }

        public MostrarEnvioEmail(IRepositoryEnvio repoEnvio)
        {
            RepoEnvio = repoEnvio;
        }
    }
}

```

```

public List<EnvioIncompletoDTO> Ejecutar(string email)
{
    List<Envio> envios = RepoEnvio.GetEnviosPorEmail(email);
    if (envios == null || envios.Count == 0)
    {
        throw new ArgumentException("No hay envios");
    }

    List<EnvioIncompletoDTO> enviosDTO = EnvioMapper.EnvioIncompletoDTOFromEnvio(envios);
    return enviosDTO;
}
}
}

```

Archivo: EditarUsuario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\EditarUsuario.cs

```

using Compartido.DTOs.UsuarioDTOs;
using Compartido.ExcepcionesGenericas;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.UsuarioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.InterfaceRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace LogicaAplicacion.ImplementacionCasosUso.UsuarioCU

```

```

{
    public class EditarUsuario : IEditarUsuario
    {
        private IRepositoryUsuario repoUsuario { get; set; }
        public EditarUsuario(IRepositoryUsuario repoUsuario)
        {
            this.repoUsuario = repoUsuario;
        }

        public void Ejecutar(UsuarioDTO usuarioDTO)
        {
            Usuario buscado = repoUsuario.GetByEmailUsuario(usuarioDTO.Email);

            if (buscado == null)
            {
                throw new UsuarioException("Datos incorrectos");
            }

            if (buscado.Contrasenia.Valor != usuarioDTO.Contrasenia)
            {
                throw new ConflictException("Datos incorrectos");
            }

            Usuario usuario = UsuarioMapper.UsuarioFromUsuarioDTO(usuarioDTO, buscado);
            usuario.Id = buscado.Id;
            repoUsuario.Editar(usuario);
        }
    }
}

```

Archivo: LoginUsuario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\ImplementacionCasosUso\UsuarioCU\LoginUsuario.cs

```
using Compartido.DTOs.UsuarioDTOs;
using Compartido.Mappers;
using LogicaAplicacion.InterfaceCasosUso.UsuarioCU;
using LogicaNegocio.EntidadesNegocio;
using LogicaNegocio.ExcepcionesEntidades;
using LogicaNegocio.InterfaceRepositorios;
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)
        {
            this.repoUsuario = repoUsuario;
        }
        public UsuarioLogueadoDTO Ejecutar(string email, string contrasenia)
        {
            Usuario usuario = repoUsuario.Login(email, contrasenia);

            if (usuario == null)
            {
                throw new UsuarioException("Credenciales incorrectas");
            }

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

Archivo: IBuscarEnvioNumeroTracking.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\EnvioCU\IBuscarEnvioNumeroTracking.cs

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

```
namespace LogicaAplicacion.InterfaceCasosUso.EnvioCU
{
    public interface IBuscarEnvioNumeroTracking
    {
        EnvioEnteroDTO Ejecutar(string numeroTracking);
    }
}
```

Archivo: IEnvioComentarios.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\EnvioC
U\IEnvioComentarios.cs

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

```
namespace LogicaAplicacion.InterfaceCasosUso.EnvioCU
{
    public interface IEnvioComentarios
    {
        EnvioComentariosDTO Ejecutar(int id);
    }
}
```

Archivo: IEnviosPorFechas.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\EnvioC
U\IEnviosPorFechas.cs

```
using Compartido.DTOs.EnvioDTOs;
using LogicaAplicacion.ImplementacionCasosUso.EnvioCU;
using LogicaNegocio.EntidadesNegocio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace LogicaAplicacion.InterfaceCasosUso.EnvioCU
{
    public interface IEnviosPorFechas
    {
        List<EnvioPorFechasDTO> Ejecutar(string email, DateTime f1, DateTime f2, int estado);
    }
}
```

Archivo: IEnviosPorTexto.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\EnvioC
U\IEnviosPorTexto.cs

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

```
namespace LogicaAplicacion.InterfaceCasosUso.EnvioCU
{
    public interface IEnviosPorTexto
    {
        List<EnvioEnteroDTO> Ejecutar(string email, string texto);
    }
}
```



```
}  
}
```

Archivo: IMostrarEnvioEmail.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\EnvioC
U\IMostrarEnvioEmail.cs

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

```
namespace LogicaAplicacion.InterfaceCasosUso.EnvioCU  
{  
    public interface IMostrarEnvioEmail  
    {  
        List<EnvioIncompletoDTO> Ejecutar(string email);  
    }  
}
```

Archivo: IEditarUsuario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\Usuari
oCU\IEditarUsuario.cs

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

```
namespace LogicaAplicacion.InterfaceCasosUso.UsuarioCU  
{  
    public interface IEditarUsuario  
    {  
        public void Ejecutar(UsuarioDTO usuarioDTO);  
    }  
}
```

Archivo: ILoginUsuario.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaAplicacion\InterfaceCasosUso\Usuari
oCU\ILoginUsuario.cs

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

```
namespace LogicaAplicacion.InterfaceCasosUso.UsuarioCU  
{  
    public interface ILoginUsuario  
    {
```

```

        UsuarioLogueadoDTO Ejecutar(string email, string contrasenia);
    }
}

```

Archivo: DireccionPostal.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\ValueObject\Agencia\DireccionPostal.cs

```

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

```

```

namespace LogicaNegocio.ValueObject.Agencia

```

```

{
    [ComplexType]
    public record DireccionPostal
    {
        public string Valor { get; init; }

        public DireccionPostal(string valor)
        {
            Valor = valor;
            Validar();
        }

        private void Validar()
        {
            bool tieneLetra = false;
            for (int i = 0; i < Valor.Length; i++)
            {
                if (!char.IsLetterOrDigit(Valor[i]) && Valor[i] != ' ')
                    tieneLetra = true; // Si encuentra un carácter no permitido
            }

            if (string.IsNullOrEmpty(Valor) || string.IsNullOrWhiteSpace(Valor))
            {
                throw new AgenciaException("La direccion postal es obligatoria");
            }
            if (Valor.Length != 5)
            {
                throw new AgenciaException("La direccion postal debe poseer solo 5 dígitos");
            }
            if (tieneLetra == true)
            {
                throw new AgenciaException("La direccion postal debe poseer numeros");
            }
        }
    }
}

```

Archivo: Contrasenias.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\ValueObject\Usuario\Contrasenias.cs

```

using LogicaNegocio.ExcepcionesEntidades;

```

```

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

namespace LogicaNegocio.ValueObject.Usuario
{
    [ComplexType]
    public record Contraseña
    {
        public string Valor { get; init; }

        public Contraseña(string valor)
        {
            Valor = valor;
            Validar();
        }

        private void Validar()
        {
            if (string.IsNullOrEmpty(Valor) || string.IsNullOrWhiteSpace(Valor))
            {
                throw new UsuarioException("El contraseña es obligatoria");
            }

            if (Valor.Length < 8)
            {
                throw new UsuarioException("La contraseña debe minimo tener 8 caracteres");
            }
        }
    }
}

```

Archivo: Email.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\ValueObject\Usuario\Email.cs

```

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

namespace LogicaNegocio.ValueObject.Usuario
{
    [ComplexType]
    public record Email
    {
        public string Valor { get; init; }

        public Email(string valor)
        {
            Valor = valor;
            Validar();
        }

        private void Validar()

```

```

{
    bool tArroba = false;
    bool tPunto = false;
    for (int i = 0; i < Valor.Length; i++)
    {
        if (Valor[i] == '@')
            tArroba = true;
        if (Valor[i] == '.')
            tPunto = true;
    }

    if (string.IsNullOrEmpty(Valor) || string.IsNullOrWhiteSpace(Valor))
    {
        throw new UsuarioException("El email es obligatorio");
    }

    if (tArroba != true || tPunto != true || Valor.Length < 5)
    {
        throw new UsuarioException("El email debe contener un arroba, un punto y minimo 5 caracteres");
    }
}
}
}

```

Archivo: Nombre.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\Obligatorio\LogicaNegocio\ValueObject\Usuario\Nombre.cs

```

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

namespace LogicaNegocio.ValueObject.Usuario
{
    [ComplexType]
    public record Nombre
    {
        public string Valor { get; init; }

        public Nombre(string valor)
        {
            Valor = valor;
            Validar();
        }

        private void Validar()
        {
            if (string.IsNullOrEmpty(Valor) || string.IsNullOrWhiteSpace(Valor))
            {
                throw new UsuarioException("El nombre es obligatorio");
            }
        }
    }
}

```

3.2 MVC

Archivo: AuthController.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Controllers\AuthController.cs

using Microsoft.AspNetCore.Mvc;

using MVC.Models.Usuario;

using Newtonsoft.Json;

namespace MVC.Controllers

{

public class AuthController : Controller

{

private string url = "";

public AuthController(IConfiguration configuration)

{

url = configuration.GetValue<string>("urlBase") + "/Auth";

}

// GET: AuthController/Login

public IActionResult Login()

{

if (HttpContext.Session.GetInt32("Logueadold") == null)

{

return View();

}

else

{

return RedirectToAction("NoAutorizado", "Auth");

}

}

// POST: AuthController/HttpPost/Login/5

[HttpPost]

public IActionResult Login(UsuarioPreLoginViewModel usuarioVM)

{

try

{

if (ModelState.IsValid)

{

HttpClient cliente = new HttpClient();

Task<HttpResponseMessage> tarea = cliente.PostAsJsonAsync(url, usuarioVM);

tarea.Wait();

HttpResponseMessage respuesta = tarea.Result;

HttpContent contenido = respuesta.Content;

Task<string> body = contenido.ReadAsStringAsync();

body.Wait();

string datos = body.Result;

if (respuesta.IsSuccessStatusCode)

{

UsuarioLogueadoViewModel usuLogueadoVM =

JsonConvert.DeserializeObject<UsuarioLogueadoViewModel>(datos);

if (usuLogueadoVM.Rol != "Cliente")

{

return RedirectToAction("NoAutorizado", "Auth");

}

if (usuLogueadoVM.Estado == false)

{

```

        return RedirectToAction("NoAutorizado", "Auth");
    }

    HttpContext.Session.SetString("Token", usuLogueadoVM.Token);
    HttpContext.Session.SetInt32("Logueadold", usuLogueadoVM.Id);
    HttpContext.Session.SetString("LogueadoRol", usuLogueadoVM.Rol);
    HttpContext.Session.SetString("LogueadoEmail", usuLogueadoVM.Email);
    HttpContext.Session.SetString("LogueadoNombre", usuLogueadoVM.Nombre);
    HttpContext.Session.SetString("LogueadoApellido", usuLogueadoVM.Apellido);
    return RedirectToAction("Index", "Home");
}
else
{
    ViewBag.msg = datos;
}
}
else
{
    ViewBag.msg = "Datos incorrectos";
}
}
catch (Exception ex)
{
    ViewBag.msg = "Error";
}
return View();
}

// GET: AuthController/Logout
public IActionResult Logout()
{
    if (HttpContext.Session.GetInt32("Logueadold") != null)
    {
        HttpContext.Session.Clear();
        return RedirectToAction("Login", "Auth");
    }
    else
    {
        return RedirectToAction("NoAutorizado", "Auth");
    }
}

// GET: AuthController/NoAutorizado
[HttpGet]
public IActionResult NoAutorizado()
{
    return View();
}

}
}

```

Archivo: EnvioController.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Controllers\EnvioController.cs

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.CodeAnalysis.CSharp.Syntax;
using Microsoft.VisualStudio.Web.CodeGenerators.Mvc.Templates.BlazorIdentity.Pages.Manage;
using MVC.Models.Comentario;
using MVC.Models.Envio;
using MVC.Models.Usuario;

```

```

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

namespace MVC.Controllers
{
    public class EnvioController : Controller
    {
        private string url = "";

        public EnvioController(IConfiguration configuration)
        {
            url = configuration.GetValue<string>("urlBase") + "/Envio";
        }

        // GET: EnvioController
        public ActionResult Index()
        {
            if (HttpContext.Session.GetInt32("Logueadold") != null)
            {
                List<EnvioIncompletoViewModel> envioVM = new List<EnvioIncompletoViewModel>();

                try
                {
                    string email = HttpContext.Session.GetString("LogueadoEmail");
                    HttpClient cliente = new HttpClient();
                    cliente.DefaultRequestHeaders.Authorization = new
System.Net.Http.Headers.AuthenticationHeaderValue("Bearer", HttpContext.Session.GetString("Token"));
                    Task<HttpResponseMessage> tarea = cliente.GetAsync(url + "/BuscarPorEmail/" + email);
                    tarea.Wait();
                    HttpResponseMessage respuesta = tarea.Result;
                    HttpContent contenido = respuesta.Content;
                    Task<string> body = contenido.ReadAsStringAsync();
                    body.Wait();
                    string datos = body.Result;

                    if (respuesta.IsSuccessStatusCode)
                    {
                        envioVM = JsonConvert.DeserializeObject<List<EnvioIncompletoViewModel>>(datos);
                    }
                    else
                    {
                        ViewBag.msg = datos;
                    }
                }
                catch (Exception ex)
                {
                    ViewBag.Mensaje = "Datos incorrectos";
                }
                return View(envioVM);
            }
            else
            {
                return RedirectToAction("NoAutorizado", "Auth");
            }
        }

        public ActionResult Details(int id)
        {
            if (HttpContext.Session.GetInt32("Logueadold") != null)
            {
                EnvioComentariosViewModel envioVM = new EnvioComentariosViewModel();
                try

```

```

        {
            if (int.IsNegative(id))
            {
                ViewBag.msg = "Datos incorrectos";
            }
            else
            {
                HttpClient cliente = new HttpClient();
                cliente.DefaultRequestHeaders.Authorization =
new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
HttpContext.Session.GetString("Token"));
                Task<HttpResponseMessage> tarea = cliente.GetAsync(url + "/Detalles/" + id);
                tarea.Wait();
                HttpResponseMessage respuesta = tarea.Result;
                HttpContent contenido = respuesta.Content;
                Task<string> body = contenido.ReadAsStringAsync();
                body.Wait();
                string datos = body.Result;

                if (respuesta.IsSuccessStatusCode)
                {
                    envioVM = JsonConvert.DeserializeObject<EnvioComentariosViewModel>(datos);
                }
                else
                {
                    ViewBag.msg = datos;
                }
            }
        }
        catch (Exception ex)
        {
            ViewBag.msg = "Error";
        }
        return View(envioVM);
    }
    else
    {
        return RedirectToAction("NoAutorizado", "Auth");
    }
}

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

[HttpPost]
public ActionResult EnvioPorNumeroTracking(string numero)
{
    EnvioEnteroViewModel envioVM = new EnvioEnteroViewModel();
    try
    {
        if (string.IsNullOrEmpty(numero))
        {
            ViewBag.msg = "Datos incorrectos";
        }
        else
        {
            HttpClient cliente = new HttpClient();
            Task<HttpResponseMessage> tarea = cliente.GetAsync(url + "/BuscarPorNumeroTracking/" +
numero);
            tarea.Wait();
            HttpResponseMessage respuesta = tarea.Result;

```



```

        HttpContent contenido = respuesta.Content;
        Task<string> body = contenido.ReadAsStringAsync();
        body.Wait();
        string datos = body.Result;

        if (respuesta.IsSuccessStatusCode)
        {
            envioVM = JsonConvert.DeserializeObject<EnvioEnteroViewModel>(datos);
        }
        else
        {
            ViewBag.msg = datos;
        }
    }
}
catch (Exception ex)
{
    ViewBag.msg = "Error";
}
return View(envioVM);
}

public ActionResult EnviosPorFechas()
{
    if (HttpContext.Session.GetInt32("Logueadold") != null)
    {
        return View();
    }
    else
    {
        return RedirectToAction("NoAutorizado", "Auth");
    }
}

[HttpPost]
public ActionResult EnviosPorFechas(DateTime f1, DateTime f2, string estado)
{
    if (HttpContext.Session.GetInt32("Logueadold") != null)
    {
        List<EnvioEnteroViewModel> envioVM = new List<EnvioEnteroViewModel>();
        try
        {
            if (string.IsNullOrEmpty(estado))
            {
                ViewBag.msg = "Datos incorrectos";
            }
            else
            {
                string email = HttpContext.Session.GetString("LogueadoEmail");
                HttpClient cliente = new HttpClient();
                cliente.DefaultRequestHeaders.Authorization =
                new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
                HttpContext.Session.GetString("Token"));
                Task<HttpResponseMessage> tarea = cliente.GetAsync(url + "/EnvioPorFechas/" + email + "/" +
                f1.ToString("yyyy-MM-dd") + "/" + f2.ToString("yyyy-MM-dd") + "/" + estado);
                tarea.Wait();
                HttpResponseMessage respuesta = tarea.Result;
                HttpContent contenido = respuesta.Content;
                Task<string> body = contenido.ReadAsStringAsync();
                body.Wait();
                string datos = body.Result;

                if (respuesta.IsSuccessStatusCode)

```

```

        {
            envioVM = JsonConvert.DeserializeObject<List<EnvioEnteroViewModel>>(datos);
        }
        else
        {
            ViewBag.msg = datos;
        }
    }
}
catch (Exception ex)
{
    ViewBag.msg = "Error";
}
return View(envioVM);
}
else
{
    return RedirectToAction("NoAutorizado", "Auth");
}
}

public ActionResult EnviosPorTexto()
{
    if (HttpContext.Session.GetInt32("Logueadold") != null)
    {
        return View();
    }
    else
    {
        return RedirectToAction("NoAutorizado", "Auth");
    }
}

[HttpPost]
public ActionResult EnviosPorTexto(string texto)
{
    if (HttpContext.Session.GetInt32("Logueadold") != null)
    {
        List<EnvioEnteroViewModel> envioVM = new List<EnvioEnteroViewModel>();
        try
        {
            if (string.IsNullOrEmpty(texto))
            {
                ViewBag.msg = "Datos incorrectos";
            }
            else
            {
                string email = HttpContext.Session.GetString("LogueadoEmail");
                HttpClient cliente = new HttpClient();
                cliente.DefaultRequestHeaders.Authorization =
new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
HttpContext.Session.GetString("Token"));
                Task<HttpResponseMessage> tarea = cliente.GetAsync(url + "/EnviosPorTexto/" + email + "/" +
texto);
                tarea.Wait();
                HttpResponseMessage respuesta = tarea.Result;
                HttpContent contenido = respuesta.Content;
                Task<string> body = contenido.ReadAsStringAsync();
                body.Wait();
                string datos = body.Result;

                if (respuesta.IsSuccessStatusCode)
                {

```

```

        envioVM = JsonConvert.DeserializeObject<List<EnvioEnteroViewModel>>(datos);
    }
    else
    {
        ViewBag.msg = datos;
    }
}
}
catch (Exception ex)
{
    ViewBag.msg = "Error";
}
return View(envioVM);
}
else
{
    return RedirectToAction("NoAutorizado", "Auth");
}
}
}
}
}

```

Archivo: HomeController.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\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()
        {
            int? lid = HttpContext.Session.GetInt32("Logueadold");
            if (lid != null)
            {
                string LogueadoNombre = HttpContext.Session.GetString("LogueadoNombre");
                string LogueadoApellido = HttpContext.Session.GetString("LogueadoApellido");
                ViewBag.msgBienvenida = $"Hola {LogueadoNombre + " " + LogueadoApellido}";
            }
            else
            {
                ViewBag.msgBienvenida = $"Hola, inicie sesi n porfavor";
            }
            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\Usuario\ObligatorioMaverickLopez335912\MVC\Controllers\UsuarioController.cs

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using MVC.Models.Usuario;
using static System.Runtime.InteropServices.JavaScript.JSType;
using System.Security.Policy;
using System.Net.Http.Json;

```

namespace MVC.Controllers

```

{
    public class UsuarioController : Controller
    {
        private string url = "";

        public UsuarioController(IConfiguration configuration)
        {
            url = configuration.GetValue<string>("urlBase") + "/Usuario";
        }
        // GET: UsuarioController/Edit/5
        public ActionResult Edit()
        {
            if (HttpContext.Session.GetInt32("Logueadold") != null)
            {
                return View();
            }
            else
            {
                return RedirectToAction("NoAutorizado", "Auth");
            }
        }

        // POST: UsuarioController/Edit/5
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit(int id, string contrasenia, string contraseniaNueva)
        {
            if (HttpContext.Session.GetInt32("Logueadold") != null)
            {
                try
                {
                    if (ModelState.IsValid)
                    {
                        UsuarioViewModel usuVM = new UsuarioViewModel
                        {
                            Email = HttpContext.Session.GetString("LogueadoEmail"),
                            Contrasenia = contrasenia,
                            ContraseniaNueva = contraseniaNueva
                        };

                        HttpClient cliente = new HttpClient();
                        cliente.DefaultRequestHeaders.Authorization =
                            new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
                                HttpContext.Session.GetString("Token"));
                    }
                }
            }
        }
    }
}

```

```

        Task<HttpResponseMessage> tarea = cliente.PutAsJsonAsync(url, usuVM);
        tarea.Wait();
        HttpResponseMessage respuesta = tarea.Result;

        if (respuesta.IsSuccessStatusCode)
        {
            ViewBag.msg = "Datos cambiados correctamente";
            return View();
        }
        else
        {
            HttpContent contenido = respuesta.Content;
            Task<string> body = contenido.ReadAsStringAsync();
            body.Wait();
            ViewBag.msg = body.Result;
        }
    }
    else
    {
        ViewBag.msg = "Datos incorrecto";
    }
}
catch (Exception ex)
{
    ViewBag.msg = "Error en los datos";
}
return View();
}
else
{
    return RedirectToAction("NoAutorizado","Auth");
}
}
}
}
}

```

Archivo: ErrorViewModel.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\ErrorViewModel.cs

namespace MVC.Models

```

{
    public class ErrorViewModel
    {
        public string? RequestId { get; set; }

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

```

Archivo: ComentarioViewModel.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Comentario\ComentarioViewModel.cs

namespace MVC.Models.Comentario

```

{
    public class ComentarioViewModel
    {
        public int Envioid { get; set; }
        public DateTime Fecha { get; set; }
        public string TextoComentario { get; set; }
    }
}

```

```
}
```

```
*****
```

Archivo: EnvioComentariosViewModel.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Envio\EnvioComentariosViewModel.cs

```
*****
```

```
using MVC.Models.Comentario;
```

```
namespace MVC.Models.Envio
```

```
{
```

```
    public class EnvioComentariosViewModel
```

```
    {
```

```
        public List<ComentarioViewModel> Comentarios { get; set; } = new List<ComentarioViewModel>();
```

```
    }
```

```
}
```

```
*****
```

Archivo: EnvioEnteroViewModel.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Envio\EnvioEnteroViewModel.cs

```
*****
```

```
using MVC.Models.Comentario;
```

```
namespace MVC.Models.Envio
```

```
{
```

```
    public class EnvioEnteroViewModel
```

```
    {
```

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

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

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

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

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

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

```
        public List<ComentarioViewModel> Comentarios { get; set; } = new List<ComentarioViewModel>();
```

```
    }
```

```
}
```

```
*****
```

Archivo: EnvioIncompletoViewModel.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Envio\EnvioIncompletoViewModel.cs

```
*****
```

```
namespace MVC.Models.Envio
```

```
{
```

```
    public class EnvioIncompletoViewModel
```

```
    {
```

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

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

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

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

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

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

```
    }
```

```
}
```

```
*****
```

Archivo: UsuarioLogueadoViewModel.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Usuario\UsuarioLogueadoViewModel.cs

```
*****
```

```
namespace MVC.Models.Usuario
```

```
{
```

```
    public class UsuarioLogueadoViewModel
```

```

    {
        public string Token { get; set; }
        public int Id { get; set; }
        public string Rol { get; set; }
        public string Nombre { get; set; }
        public string Apellido { get; set; }
        public string Email { get; set; }
        public bool Estado { get; set; }
    }
}

```

Archivo: UsuarioPreLoginViewModel.cs

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Usuario\UsuarioPreLoginViewModel.cs

```

namespace MVC.Models.Usuario
{
    public class UsuarioPreLoginViewModel
    {
        public string Email { get; set; }
        public string Contraseña { get; set; }

    }
}

```

Archivo: UsuarioViewModel.cs

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Models\Usuario\UsuarioViewModel.cs

using Microsoft.AspNetCore.Mvc.Rendering;

```

namespace MVC.Models.Usuario
{
    public class UsuarioViewModel
    {
        public string Email { get; set; }
        public string Contraseña { get; set; }
        public string ContraseñaNueva { get; set; }

    }
}

```

Archivo: _ViewImports.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views_ViewImports.cshtml

```

@using MVC
@using MVC.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

```

Archivo: _ViewStart.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views_ViewStart.cshtml

```

@{
    Layout = "_Layout";
}

```

Archivo: Login.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Auth\Login.cshtml

@model MVC.Models.Usuario.UsuarioPreLoginViewModel

```
@{
    ViewData["Title"] = "Login";
}
```

<h1>Login</h1>

<hr />

<div class="row">

<div class="col-md-4">

<form asp-action="Login">

<div asp-validation-summary="ModelOnly" class="text-danger"></div>

<div class="form-group">

<label asp-for="Email" class="control-label"></label>

<input asp-for="Email" class="form-control" />

</div>

<div class="form-group">

<label asp-for="Contrasenia" class="control-label"></label>

<input asp-for="Contrasenia" class="form-control" />

</div>

<div class="form-group">

<input type="submit" value="Iniciar Sesion" class="btn btn-primary" />

</div>

<p>@ViewBag.msg</p>

</form>

</div>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("_ValidationScriptsPartial");}

}

Archivo: NoAutorizado.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Auth\NoAutorizado.cshtml

<h1>No tiene permisos</h1>

<a asp-action="Index" asp-controller="Home">Volver al Home

Archivo: Details.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Envio\Details.cshtml

@model MVC.Models.Envio.EnvioComentariosViewModel

```
@{
    ViewData["Title"] = "Details";
}
```

<h1>Details</h1>

<div>

<h4>EnvioComentariosViewModel</h4>

<hr />

<dl class="row">

</dl>

<p>@ViewBag.msg</p>

<table class="table">

<thead>


```

                <tr>
                    <th>
                        @Html.DisplayNameFor(model =>
model.Comentarios[0].Enviold)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model =>
model.Comentarios[0].Fecha)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model =>
model.Comentarios[0].TextoComentario)
                    </th>
                </tr>
            </thead>
            <tbody>
                @foreach (var item in Model.Comentarios)
                {
                    <tr>
                        <td>
                            @Html.DisplayFor(modelItem => item.Enviold)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem => item.Fecha)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem =>
item.TextoComentario)
                        </td>
                    </tr>
                }
            </tbody>
        </table>
    </div>
    <div>
        <a asp-action="Index">Back to List</a>
    </div>

```

Archivo: EnvioPorNumeroTracking.cshtml

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Envio\EnvioPorNumeroTracking.cshtml

```

@{
    ViewData["Title"] = "Buscar envio por numero de tracking";
}

```

```

<h1>Buscar envio por numero de tracking</h1>

```

```

<hr />

```

```

<div class="row">

```

```

    <div class="col-md-4">

```

```

        <form method="post">

```

```

            <div class="form-group">

```

```

                <label class="control-label"></label>

```

```

                <input type="text" name="numero" class="form-control" />

```

```

            </div>

```

```

            <div class="form-group">

```

```

                <input type="submit" value="Buscar Envio" class="btn btn-primary" />

```

```

            </div>

```

```

            <p>@ViewBag.msg</p>

```

```

        </form>

```

```
</div>
</div>
```

```
@if (Model != null && Model.NumeroTracking != null)
```

```
{
    <div>
        <h4>EnvioEnteroViewModel</h4>
        <hr />
        <dl class="row">
            <dt class="col-sm-2">
                Id
            </dt>
            <dd class="col-sm-10">
                @Model.Id
            </dd>
            <dt class="col-sm-2">
                Numero de Tracking
            </dt>
            <dd class="col-sm-10">
                @Model.NumeroTracking
            </dd>
            <dt class="col-sm-2">
                Empleado
            </dt>
            <dd class="col-sm-10">
                @Model.EmpleadoNombre
            </dd>
            <dt class="col-sm-2">
                Cliente
            </dt>
            <dd class="col-sm-10">
                @Model.ClienteNombre
            </dd>
            <dt class="col-sm-2">
                Peso
            </dt>
            <dd class="col-sm-10">
                @Model.Peso
            </dd>
            <dt class="col-sm-2">
                Estado
            </dt>
            <dd class="col-sm-10">
                @Model.Estado
            </dd>
        </dl>
    </div>

```

```
@foreach (var item in Model.Comentarios)
{
```

```
    <dt class="col-sm-2">
        Fecha
    </dt>
    <dd class="col-sm-10">
        @item.Fecha
    </dd>
    <dt class="col-sm-2">
        Comentario
    </dt>
    <dd class="col-sm-10">
        @item.TextoComentario
    </dd>

```

```
}
}
```

Archivo: EnviosPorFechas.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Envio\EnviosPorFechas.cshtml

```
@{
    ViewData["Title"] = "Buscar envio por numero de tracking";
}

<h1>Buscar envios</h1>

<hr />
<div class="row">
    <div class="col-md-4">
        <form method="post">
            <div class="form-group">
                <label class="control-label"></label>
                <input type="date" name="f1" class="form-control" />
            </div>
            <div class="form-group">
                <label class="control-label"></label>
                <input type="date" name="f2" class="form-control" />
            </div>
            <div class="form-group">
                <label class="control-label"></label>
                <select name="estado">
                    <option value=0>En Proceso</option>
                    <option value=1>Finalizado</option>
                </select>
            </div>
            <div class="form-group">
                <input type="submit" value="Buscar Envios" class="btn btn-primary" />
            </div>
            <p>@ViewBag.msg</p>
        </form>
    </div>
</div>

@if (Model != null)
{
    <div>
        <h4>Envios</h4>
        <hr />
        <dl class="row">
        </dl>
        <table class="table">
            <thead>
                <tr>
                    <th>
                        <label>Numero Tracking</label>
                    </th>
                    <th>
                        <label>Peso</label>
                    </th>
                    <th>
                        <label>Estado</label>
                    </th>
                </tr>
            </thead>
            <tbody>
                @foreach (var item in Model)
                {
```

```

        <tr>
            <td>
                @item.NumeroTracking
            </td>
            <td>
                @item.Peso
            </td>
            <td>
                @item.Estado
            </td>
        </tr>
    }
</tbody>
</table>
</div>
}
*****
Archivo: EnviosPorTexto.cshtml
Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Envio\EnviosPorTexto.cshtml
*****
@model IEnumerable<MVC.Models.Envio.EnvioEnteroViewModel>

@{
    ViewData["Title"] = "EnviosPorTexto";
}

<h1>EnviosPorTexto</h1>

<div class="row">
    <div class="col-md-4">
        <form method="post">
            <div class="form-group">
                <label class="control-label"></label>
                <input type="text" name="texto" class="form-control" />
            </div>
            <div class="form-group">
                <input type="submit" value="Buscar Envios" class="btn btn-primary" />
            </div>
            <p>@ViewBag.msg</p>
        </form>
    </div>
</div>

@if (Model != null)
{
    <table class="table">
        <thead>
            <tr>
                <th>
                    @Html.DisplayNameFor(model => model.Id)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.NumeroTracking)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.EmpleadoNombre)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.ClienteNombre)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.Peso)
                </th>
            </tr>
        </thead>
    </table>
}

```

```

        <th>
            @Html.DisplayNameFor(model => model.Estado)
        </th>
    </th></th>
</tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Id)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.NumeroTracking)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EmpleadoNombre)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.ClienteNombre)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Peso)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Estado)
            </td>
            <td>
                @Html.ActionLink("Details", "Details", new { id = item.Id })
            </td>
        </tr>
    }
</tbody>
</table>

}

*****
Archivo: Index.cshtml
Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Envio\Index.cshtml
*****
@model IEnumerable<MVC.Models.Envio.EnvioIncompletoViewModel>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Id)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.NumeroTracking)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EmpleadoNombre)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ClienteNombre)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Peso)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Estado)
            </th>
            <th>
                @Html.ActionLink("Details", "Details", new { id = item.Id })
            </th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.Id)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.NumeroTracking)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.EmpleadoNombre)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.ClienteNombre)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Peso)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Estado)
                </td>
                <td>
                    @Html.ActionLink("Details", "Details", new { id = item.Id })
                </td>
            </tr>
        }
    </tbody>
</table>

```

```

                @Html.DisplayNameFor(model => model.ClienteNombre)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Peso)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Estado)
            </th>
        </th></th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Id)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.NumeroTracking)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EmpleadoNombre)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.ClienteNombre)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Peso)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Estado)
            </td>
            <td>
                @Html.ActionLink("Details", "Details", new { id = item.Id})
            </td>
        </tr>
    }
</tbody>
</table>
<p>@ViewBag.msg</p>

```

Archivo: Index.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Home\Index.cshtml

```

@{
    ViewData["Title"] = "Home Page";
}

```

```

<div class="text-center">
    <p>Bienvenido</p>
    <p>@ViewBag.msgBienvenida</p>
    <p>@TempData["msg"]</p>
</div>

```

Archivo: Privacy.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Home\Privacy.cshtml

```

@{
    ViewData["Title"] = "Privacy Policy";
}

```

<h1>@ViewData["Title"]</h1>

<p>Use this page to detail your site's privacy policy.</p>

Archivo: Error.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Shared\Error.cshtml

@model ErrorViewModel

@{

 ViewData["Title"] = "Error";

}

<h1 class="text-danger">Error.</h1>

<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)

{

 <p>

 Request ID: <code>@Model.RequestId</code>

 </p>

}

<h3>Development Mode</h3>

<p>

 Swapping to Development environment will display more detailed information about the error that occurred.

</p>

<p>

 The Development environment shouldn't be enabled for deployed applications.

 It can result in displaying sensitive information from exceptions to end users.

 For local debugging, enable the Development environment by setting the

ASPNETCORE_ENVIRONMENT environment variable to Development and restarting the app.

</p>

Archivo: NoAutorizado.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Shared\NoAutorizado.cshtml

<h1>No tiene permisos</h1>

<a asp-action="Index" asp-controller="Home">Volver al Home

Archivo: _Layout.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Shared_Layout.cshtml

<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="utf-8" />

 <meta name="viewport" content="width=device-width, initial-scale=1.0" />

 <title>@ViewData["Title"] - MVC</title>

 <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

 <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />

 <link rel="stylesheet" href="~/MVC.styles.css" asp-append-version="true" />

</head>

<body>

 <header>

 <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

 <div class="container-fluid">

 MVC

```

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
    <ul class="navbar-nav flex-grow-1">

        @* PRUEBA *@

        @* PRUEBA *@

        @if (Context.Session.GetInt32("Logueadold") != null)
        {
            @* Está logueado *@
            @* Es Funcionario *@
            @if (Context.Session.GetString("LogueadoRol") == "Funcionario")
            {
                <!-- SIN TERMINAR-->
            }
            @* Es Administrador *@
            else if (Context.Session.GetString("LogueadoRol") == "Administrador")
            {
                <!-- SIN TERMINAR-->
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Usuario" asp-
action="Index">Mostrar Usuarios</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Usuario" asp-action="Create">Alta
Usuario</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Agencia" asp-action="Create">Alta
Agencia</a>
                </li>
            }
            @* Es Cliente *@
            else
            {
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Usuario" asp-
action="Edit">Cambiar Contraseña</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Envio" asp-action="Index">Mis
Envios</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Envio" asp-
action="EnviosPorFechas">Envios Fechas</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-controller="Envio" asp-
action="EnviosPorTexto">Envios Texto</a>
                </li>
            }

            @* Ambos *@
            <li class="nav-item">
                <a class="nav-link text-dark" asp-area="" asp-controller="Auth" asp-
action="Logout">Logout</a>
            </li>

```



```

    }
    else
    {
        @* Está deslogueado *@
        <!-- SIN TERMINAR-->

        <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-controller="Auth" asp-action="Login">Login</a>
        </li>
    }
    @* Siempr aparece *@
    <li class="nav-item">
        <a class="nav-link text-dark" asp-area="" asp-controller="Envio" asp-
action="EnvioPorNumeroTracking">Trackear Envio</a>
    </li>

</ul>
</div>
</div>
</nav>
</header>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2025 - MVC - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
    </div>
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
@await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

Archivo: _ValidationScriptsPartial.cshtml

Carpeta:

C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Shared_ValidationScriptsPartial.cshtml

<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>

Archivo: Edit.cshtml

Carpeta: C:\Users\Usuario\ObligatorioMaverickLopez335912\MVC\Views\Usuario\Edit.cshtml

@{

 ViewData["Title"] = "Edit";

}

<h1>Modificar Contraseña</h1>

<hr />

<div class="row">

 <div class="col-md-4">

 <form method="post">

```
<div class="form-group">
  <label class="control-label" >Contrasenia: </label>
  <input type="text" name="contrasenia" class="form-control" />
</div>
<div class="form-group">
  <label class="control-label">Contrasenia Nueva: </label>
  <input type="text" name="contraseniaNueva" class="form-control" />
</div>
<div class="form-group">
  <input type="submit" value="Guardar Cambios" class="btn btn-primary" />
</div>
<p>@ViewBag.msg</p>
</form>
</div>
</div>
```

4 Datos de prueba

```
DELETE FROM [dbo].[Comentario];
```

```
DELETE FROM [dbo].[Auditorias];
```

```
DELETE FROM [dbo].[Envios];
```

```
DELETE FROM [dbo].[Usuarios];
```

```
DELETE FROM [dbo].[Agencias];
```

```
SET IDENTITY_INSERT [dbo].[Agencias] ON
```

```
INSERT INTO [dbo].[Agencias] ([Id],[Nombre], [Latitud], [Longitud],  
[DireccionPostal_Valor])
```

```
VALUES
```

```
(1,'Agencia Centro', 19.432608, -99.133209, '12345'),
```

```
(2,'Agencia Norte', 19.523615, -99.226405, '67890'),
```

```
(3,'Agencia Sur', 19.299467, -99.150148, '23456'),
```

```
(4,'Agencia Occidente', 20.676312, -103.344432, '98765'),
```

```
(5,'Agencia Oriente', 19.123456, -98.987654, '54321'),
```

```
(6,'Agencia Puebla', 19.043297, -98.202267, '11223'),
```

```
(7,'Agencia Monterrey', 25.686614, -100.316113, '44556'),
```

```
(8,'Agencia Tijuana', 32.502407, -117.038247, '78901'),
```

```
(9,'Agencia Guadalajara', 20.659698, -103.349609, '22334'),
```

```
(10,'Agencia Cancún', 21.174290, -86.846560, '99887'),
```

```
(11,'Agencia Mérida', 21.016162, -89.146395, '66778'),
```

```
(12,'Agencia Veracruz', 19.173017, -96.126566, '33445'),
```

```
(13,'Agencia León', 21.121800, -101.686852, '55443'),
```

```
(14,'Agencia Mazatlán', 23.249393, -106.411443, '88990'),
```

```
(15,'Agencia Querétaro', 20.588825, -100.396548, '44567');
```

```
SET IDENTITY_INSERT [dbo].[Agencias] OFF
```

```
SET IDENTITY_INSERT [dbo].[Usuarios] ON
```

```
INSERT INTO [dbo].[Usuarios] ([Id],[Apellido], [Rol], [Estado], [Contrasenia_Valor],  
[Email_Valor], [Nombre_Valor])
```

```
VALUES
```

```
(1,'González', 0, 1, 'Contraseña123', 'gonzalez@example.com', 'Carlos'),
```

```
(2,'Pérez', 1, 1, 'MiClaveSecreta99', 'perez@example.com', 'Ana'),
```

```
(3,'Martínez', 2, 0, 'SeguraContrasenia88', 'martinez@example.com', 'Luis'),
```

```
(4,'López', 1, 1, 'Pass1234!@', 'lopez@example.com', 'Marta'),
```

```
(5,'Sánchez', 0, 1, 'ClaveSegura2025', 'sanchez@example.com', 'José'),
```

```
(6,'Ramírez', 2, 0, 'Contraseña789!', 'ramirez@example.com', 'Elena'),
```

```
(7,'Torres', 1, 1, 'ContraseñaSecreta2023', 'torres@example.com', 'Pedro'),
```

```
(8,'Vázquez', 0, 1, 'Passw0rd1234', 'vazquez@example.com', 'Laura'),
```

```
(9,'Mendoza', 0, 0, 'Contraseñas1234', 'mendoza@example.com', 'Juan'),
```

```
(10,'Gutiérrez', 2, 1, 'MiClave@Segura9', 'gutierrez@example.com', 'Sofia'),
```

```
(11,'Morales', 1, 0, 'S3guridad88!@#', 'morales@example.com', 'Ricardo'),
```

```
(12,'Jiménez', 0, 1, 'Clave123!@456', 'jimenez@example.com', 'Isabel'),
```

```
(13,'Hernández', 2, 1, 'SecurePassword2024', 'hernandez@example.com',  
'Francisco'),
```

```
(14,'Álvarez', 1, 1, 'Contraseñ@Segura7', 'alvarez@example.com', 'Santiago'),
```

```
(15,'Ruiz', 0, 0, 'P@ssw0rd5678', 'ruiz@example.com', 'Verónica');
```

```
SET IDENTITY_INSERT [dbo].[Usuarios] OFF
```

```
SET IDENTITY_INSERT [dbo].[Envios] ON
```

```
INSERT INTO [dbo].[Envios] ([Id], [NumeroTracking], [IdEmpleado], [IdCliente],  
[Peso], [Estado], [Discriminator], [Agenciald], [DireccionPostal], [TiempoEntrega],  
[Eficiencia])
```

```
VALUES
```

```
(1, N'120', 1, 3, 85, 1, N'Comun', 1, NULL, NULL, NULL),
```

```
(2, N'213', 1, 3, 85, 0, N'Urgente', NULL, N'15451', NULL, 0),
```

(3, N'504', 1, 6, 0, 0, N'Comun', 9, NULL, NULL, NULL),
(4, N'1592', 1, 6, 52, 1, N'Urgente', NULL, N'21508', N'00:00:50.9596429', 1),
(5, N'6705', 1, 10, 85, 0, N'Comun', 6, NULL, NULL, NULL),
(6, N'37670', 1, 10, 964, 0, N'Comun', 7, NULL, NULL, NULL),
(7, N'2610', 1, 10, 8, 1, N'Urgente', NULL, N'85120', N'00:00:43.6080118', 1),
(8, N'2818', 1, 13, 13, 0, N'Urgente', NULL, N'25210', NULL, 0),
(9, N'37541', 1, 13, 96, 1, N'Comun', 5, NULL, NULL, NULL),
(10, N'6666', 1, 13, 63, 0, N'Comun', 4, NULL, NULL, NULL);
SET IDENTITY_INSERT [dbo].[Envios] OFF

SET IDENTITY_INSERT [dbo].[Comentario] ON

INSERT INTO [dbo].[Comentario] ([Id], [Fecha], [Empleadold], [TextoComentario],
[Enviold])

VALUES

(1, N'2025-05-19 11:20:37', 1, N'Ingresado en agencia de origen', 1),
(2, N'2025-05-19 11:20:41', 1, N'Ingresado en agencia de origen', 2),
(3, N'2025-05-19 11:20:48', 1, N'Ingresado en agencia de origen', 4),
(4, N'2025-05-19 11:20:52', 1, N'Ingresado en agencia de origen', 5),
(5, N'2025-05-19 11:20:54', 1, N'Ingresado en agencia de origen', 6),
(6, N'2025-05-19 11:20:57', 1, N'Ingresado en agencia de origen', 7),
(7, N'2025-05-19 11:21:00', 1, N'Ingresado en agencia de origen', 8),
(8, N'2025-05-19 11:21:03', 1, N'Ingresado en agencia de origen', 9),
(9, N'2025-05-19 11:21:06', 1, N'Ingresado en agencia de origen', 10),
(10, N'2025-05-19 11:21:25', 1, N'Ingresado en agencia de origen', 3),
(11, N'2025-05-19 11:21:36', 1, N'Finalizado', 1),
(12, N'2025-05-19 11:21:39', 1, N'Finalizado', 4),
(13, N'2025-05-19 11:21:41', 1, N'Finalizado', 7),
(14, N'2025-05-19 11:21:43', 1, N'Finalizado', 9),

(15, N'2025-05-19 11:21:48', 1, N'Demorado', 3);

SET IDENTITY_INSERT [dbo].[Comentario] OFF

SET IDENTITY_INSERT [dbo].[Auditorias] ON

INSERT INTO [dbo].[Auditorias] ([Id], [Accion], [Fecha], [IdEmpleado])

VALUES

(1, N'Nuevo usuario creado', N'2025-05-19 11:36:02', 1),

(2, N'Usuario editado', N'2025-05-19 11:37:02', 1),

(3, N'Nuevo usuario creado', N'2025-05-19 11:38:02', 1),

(4, N'Nuevo usuario creado', N'2025-05-19 11:39:02', 1),

(5, N'Usuario editado', N'2025-05-19 11:40:02', 1),

(6, N'Usuario dado de baja', N'2025-05-19 11:41:02', 1),

(7, N'Usuario editado', N'2025-05-19 11:42:02', 1),

(8, N'Nuevo usuario creado', N'2025-05-19 11:43:02', 1),

(9, N'Usuario editado', N'2025-05-19 11:44:02', 1),

(10, N'Usuario dado de baja', N'2025-05-19 11:45:02', 1);

SET IDENTITY_INSERT [dbo].[Auditorias] OFF

5 Links Importantes

5.1 GitHub

<https://github.com/Prog-P3-2025-1/ObligatorioMaverickLopez335912.git>

5.2 Azure

API:

<https://mavericklopezwebapi-fuf6e2f7gaavd6ay.centralus-01.azurewebsites.net>

MVC:

<https://mavericklopezmvc-bdbehfdcdbfwg4gq.centralus-01.azurewebsites.net>