



Entrega Obligatoria  
Programación III  
Prof.: Lucas López



Mateo Martinatto – 311047



Kevin González – 290552

Mateo Martinatto

Kevin González

Fecha de Entrega: 20/06/2024

Contenido

Casos de Uso ..... 3

Diagrama de Clases ..... 4

    Lógica de Negocio ..... 4

    Acceso a Datos ..... 5

    Lógica Aplicación ..... 5

    WebApp Anterior(Deprecated)..... 6

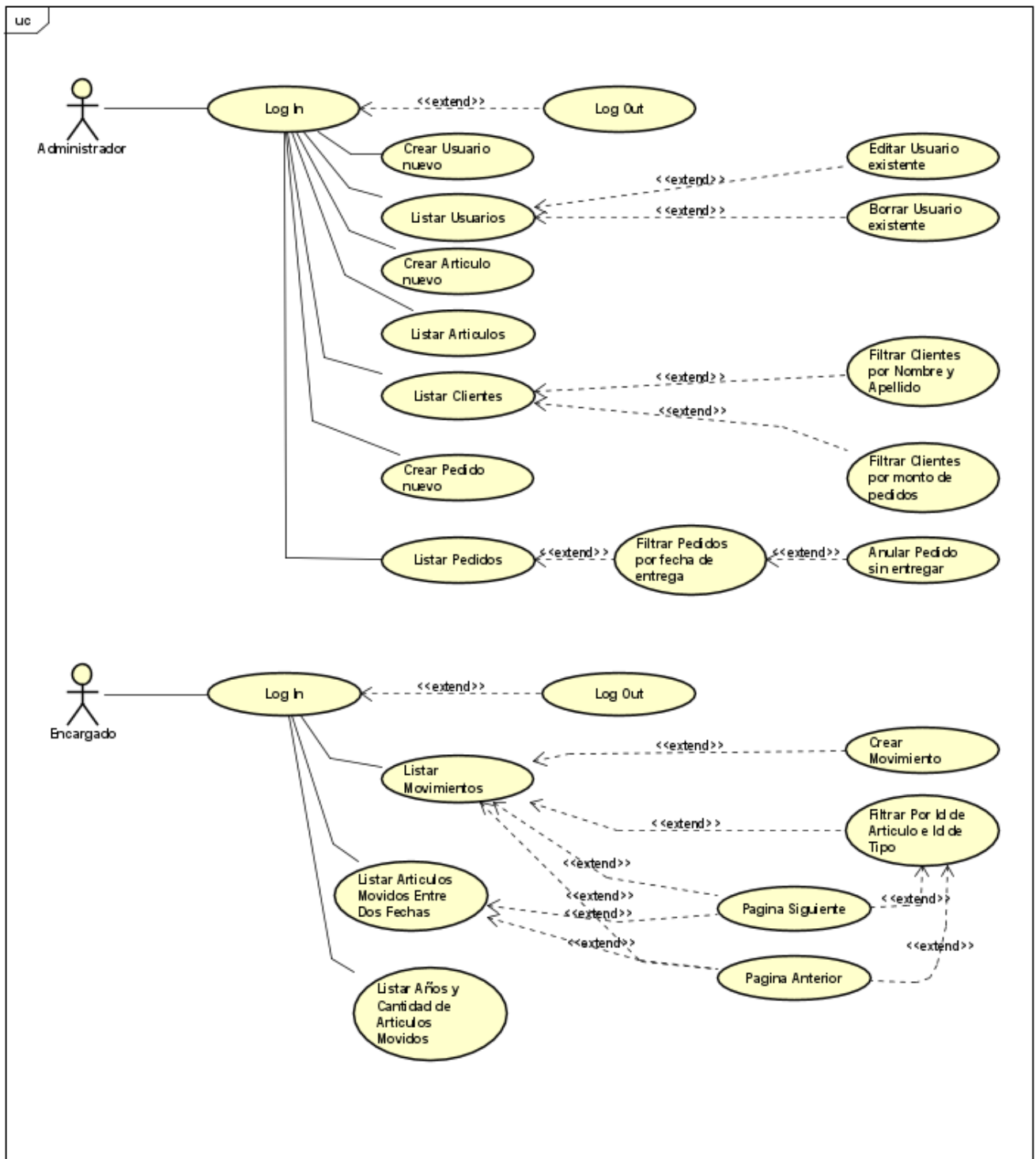
    ApiRest (ServerSide) ..... 6

    Api y Web (ClientSide) ..... 6

Uso de Inteligencia Artificial Generativa ..... 7

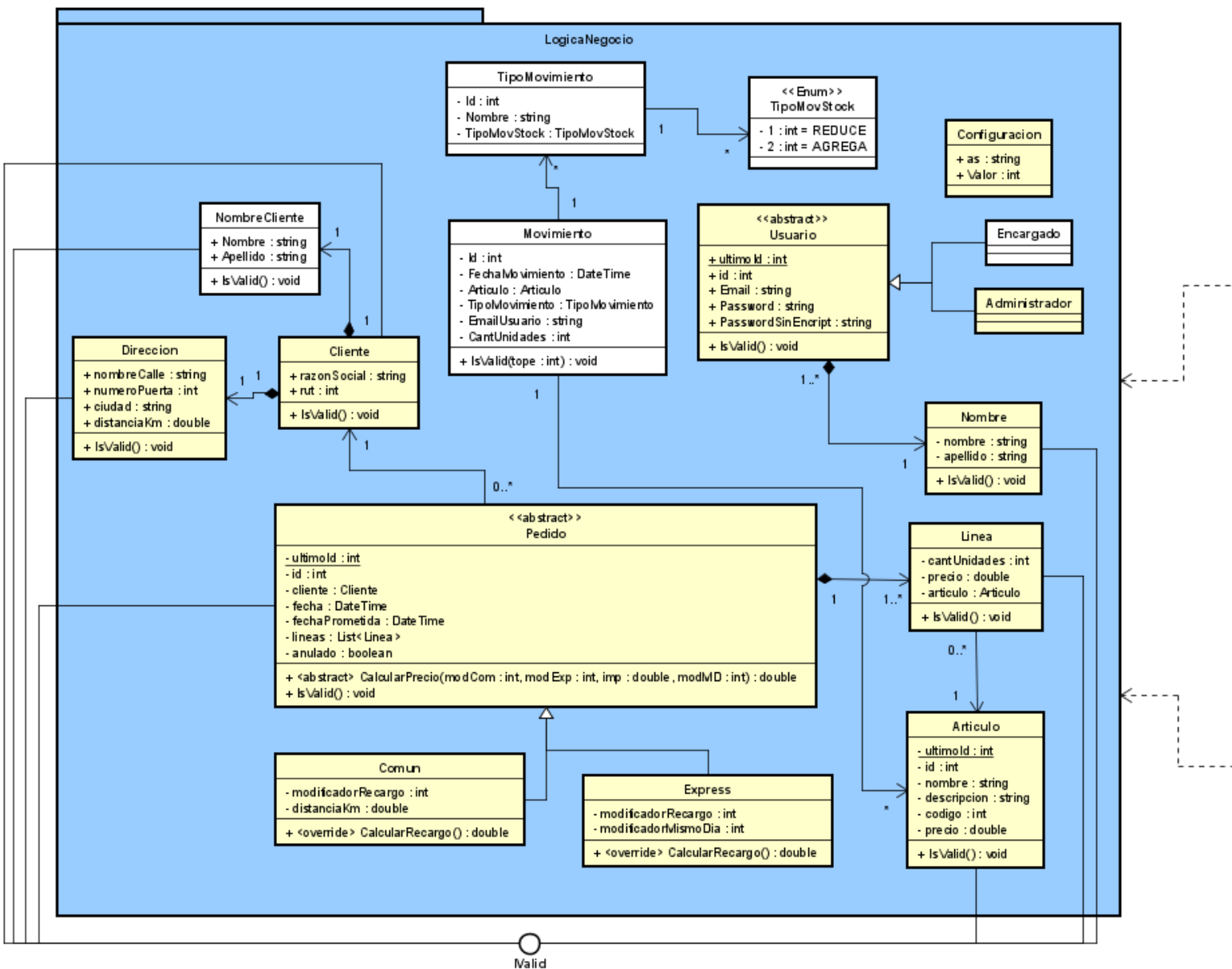
Código Fuente.....14

## Casos de Uso

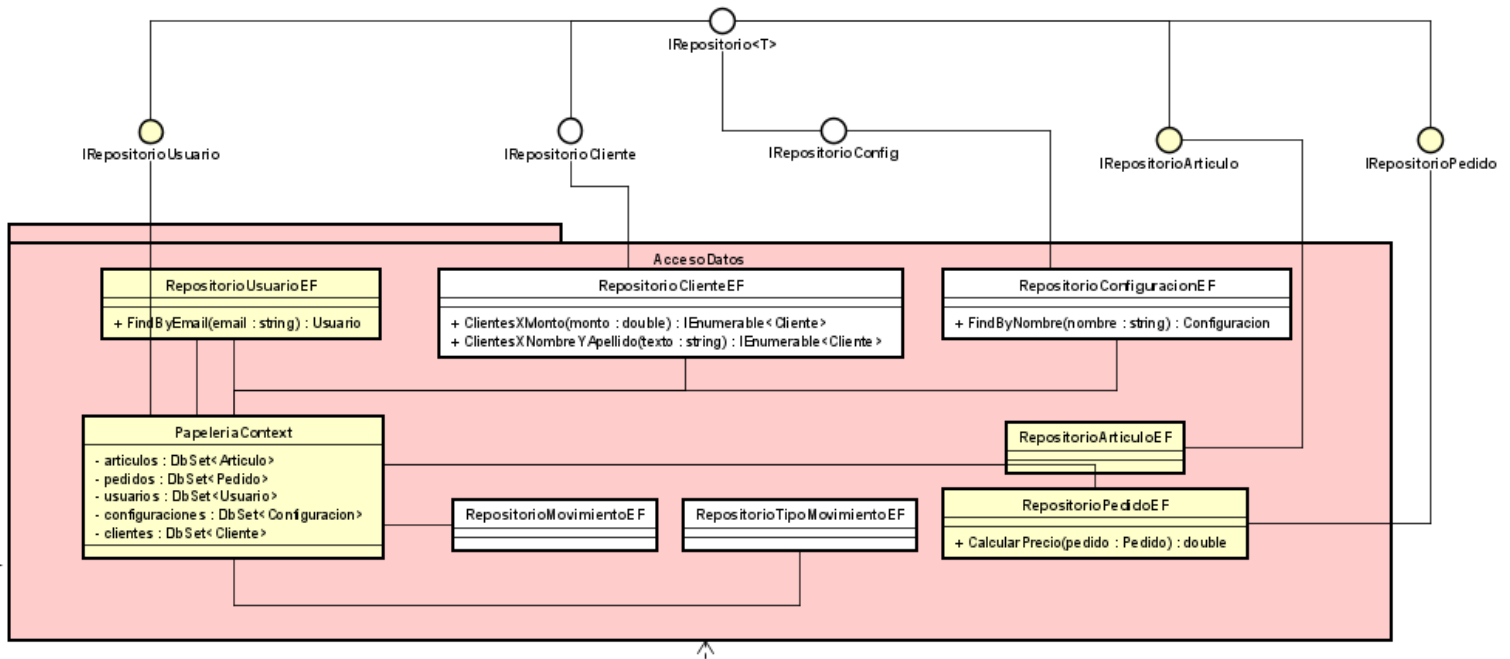


# Diagrama de Clases

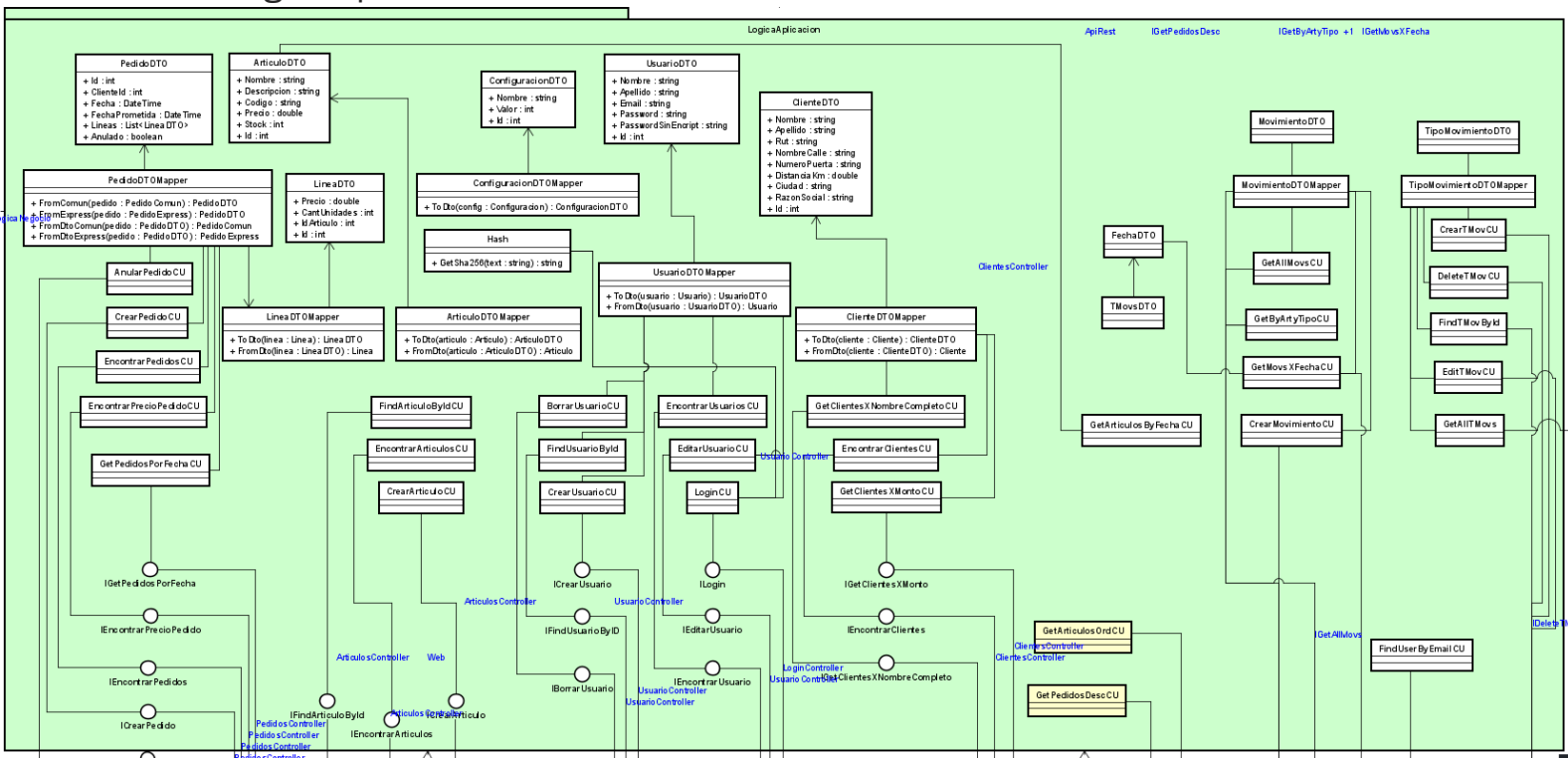
## Lógica de Negocio



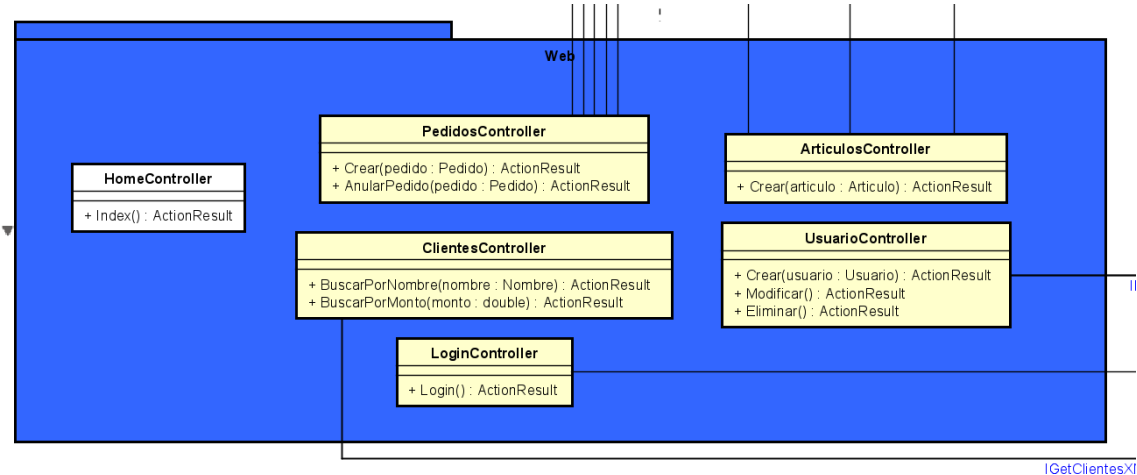
## Acceso a Datos



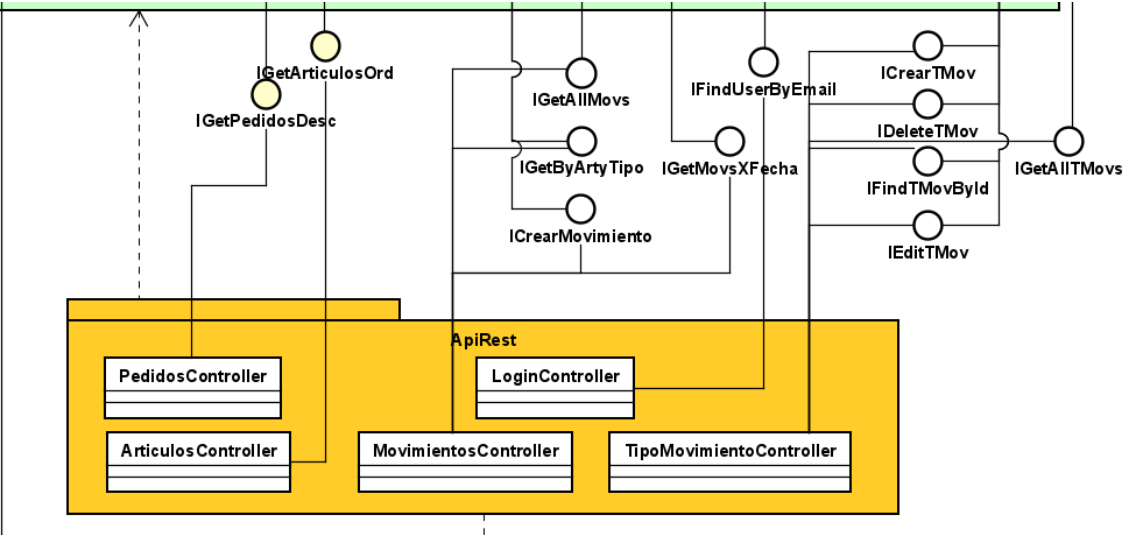
## Lógica Aplicación



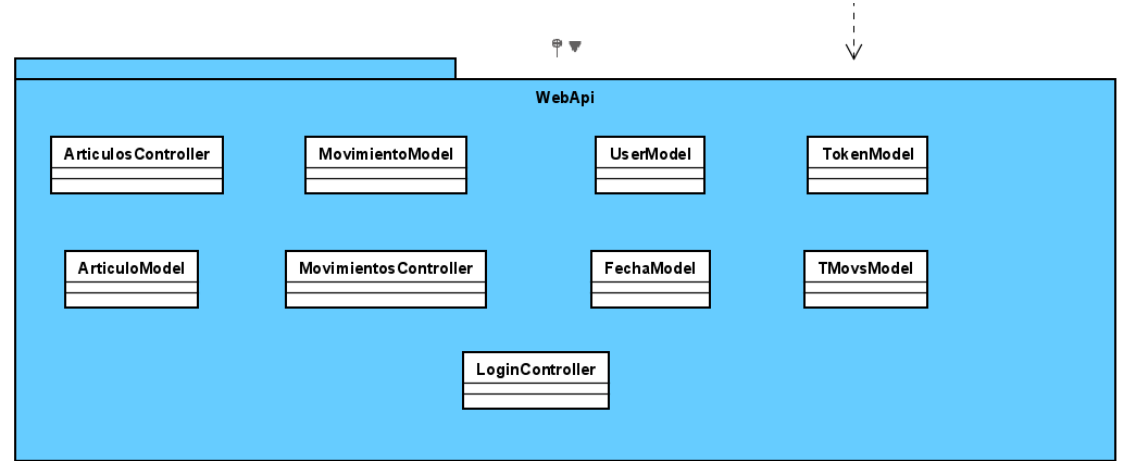
WebApp Anterior(Deprecated)



ApiRest (ServerSide)



Api y Web (ClientSide)



## Uso de Inteligencia Artificial Generativa

Se utilizó la inteligencia artificial generativa de OpenAI conocida como ChatGPT para dos prompts específicos:

- Creación de los datos de prueba:
- Utilizando el siguiente Script:

```
USE [PapeleriaObligatorio] GO /***** Object: Table [dbo].[Articulos] Script
Date: 19/06/2024 19:54:02 / SET ANSI_NULLS ON GO SET
QUOTED_IDENTIFIER ON GO CREATE TABLE [dbo].[Articulos]( [Id] [int]
IDENTITY(1,1) NOT NULL, [Nombre] nvarchar NOT NULL, [Codigo] nvarchar NOT
NULL, [Descripcion] nvarchar NOT NULL, [Precio] [float] NOT NULL, CONSTRAINT
[PK_Articulos] PRIMARY KEY CLUSTERED ( [Id] ASC )WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY] GO / Object: Table [dbo].[Clientes] Script Date:
19/06/2024 19:54:02 / SET ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON
GO CREATE TABLE [dbo].[Clientes]( [Id] [int] IDENTITY(1,1) NOT NULL,
[RazonSocial] nvarchar NOT NULL, [Rut] nvarchar NOT NULL,
[NombreCliente_Nombre] nvarchar NOT NULL, [NombreCliente_Apellido]
nvarchar NOT NULL, [Direccion_NombreCalle] nvarchar NOT NULL,
[Direccion_NumeroPuerta] nvarchar NOT NULL, [Direccion_Ciudad] nvarchar
NOT NULL, [Direccion_DistanciaKm] [float] NOT NULL, CONSTRAINT
[PK_Clientes] PRIMARY KEY CLUSTERED ( [Id] ASC )WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY] GO / Object: Table [dbo].[Configuraciones] Script
Date: 19/06/2024 19:54:02 / SET ANSI_NULLS ON GO SET
QUOTED_IDENTIFIER ON GO CREATE TABLE [dbo].[Configuraciones]( [Id] [int]
IDENTITY(1,1) NOT NULL, [Nombre] nvarchar NOT NULL, [Valor] [int] NOT NULL,
CONSTRAINT [PK_Configuraciones] PRIMARY KEY CLUSTERED ( [Id] ASC )WITH
(PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY] GO / Object: Table [dbo].[Linea] Script Date:
19/06/2024 19:54:02 / SET ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON
GO CREATE TABLE [dbo].[Linea]( [Id] [int] IDENTITY(1,1) NOT NULL, [Precio]
[float] NOT NULL, [CantUnidades] [int] NOT NULL, [Articulold] [int] NOT NULL,
[Pedidold] [int] NULL, CONSTRAINT [PK_Linea] PRIMARY KEY CLUSTERED ( [Id]
ASC )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY] GO
/ Object: Table [dbo].[Movimientos] Script Date: 19/06/2024 19:54:02 / SET
ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON GO CREATE TABLE
[dbo].[Movimientos]( [Id] [int] IDENTITY(1,1) NOT NULL, [FechaMovimiento]
datetime2 NOT NULL, [Articulold] [int] NOT NULL, [TipoMovimientold] [int] NOT
NULL, [CantUnidades] [int] NOT NULL, [EmailUsuario] nvarchar NOT NULL,
CONSTRAINT [PK_Movimientos] PRIMARY KEY CLUSTERED ( [Id] ASC )WITH
```

```

(PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY] GO / Object: Table [dbo].[Pedidos] Script Date:
19/06/2024 19:54:02 / SET ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON
GO CREATE TABLE [dbo].[Pedidos]( [Id] [int] IDENTITY(1,1) NOT NULL, [Clienteld]
[int] NOT NULL, [Fecha] datetime2 NOT NULL, [FechaPrometida] datetime2 NOT
NULL, [Anulado] [bit] NOT NULL, [Discriminator] nvarchar NOT NULL,
CONSTRAINT [PK_Pedidos] PRIMARY KEY CLUSTERED ( [Id] ASC )WITH
(PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY] GO /
Object: Table [dbo].[TipoMovimientos] Script Date: 19/06/2024 19:54:02 / SET
ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON GO CREATE TABLE
[dbo].[TipoMovimientos]( [Id] [int] IDENTITY(1,1) NOT NULL, [Nombre] nvarchar
NOT NULL, [TipoMovStock] [int] NOT NULL, CONSTRAINT [PK_TipoMovimientos]
PRIMARY KEY CLUSTERED ( [Id] ASC )WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY] GO /
Object: Table [dbo].[Usuarios] Script Date: 19/06/2024 19:54:02 *****/ SET
ANSI_NULLS ON GO SET QUOTED_IDENTIFIER ON GO CREATE TABLE
[dbo].[Usuarios]( [Id] [int] IDENTITY(1,1) NOT NULL, [Email] nvarchar NOT NULL,
[Password] nvarchar NOT NULL, [PasswordSinEncript] nvarchar NOT NULL,
[NombreCompleto_Nombre] nvarchar NOT NULL, [NombreCompleto_Apellido]
nvarchar NOT NULL, [Discriminator] nvarchar NOT NULL, CONSTRAINT
[PK_Usuarios] PRIMARY KEY CLUSTERED ( [Id] ASC )WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY] ) ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY] GO ALTER TABLE [dbo].[Movimientos] ADD DEFAULT
(N'') FOR [EmailUsuario] GO ALTER TABLE [dbo].[Usuarios] ADD DEFAULT (N'')
FOR [Discriminator] GO ALTER TABLE [dbo].[Linea] WITH CHECK ADD
CONSTRAINT [FK_Linea_Articulos_Articulold] FOREIGN KEY([Articulold])
REFERENCES [dbo].[Articulos] ([Id]) ON DELETE CASCADE GO ALTER TABLE
[dbo].[Linea] CHECK CONSTRAINT [FK_Linea_Articulos_Articulold] GO ALTER
TABLE [dbo].[Linea] WITH CHECK ADD CONSTRAINT
[FK_Linea_Pedidos_Pedidold] FOREIGN KEY([Pedidold]) REFERENCES
[dbo].[Pedidos] ([Id]) GO ALTER TABLE [dbo].[Linea] CHECK CONSTRAINT
[FK_Linea_Pedidos_Pedidold] GO ALTER TABLE [dbo].[Movimientos] WITH
CHECK ADD CONSTRAINT [FK_Movimientos_Articulos_Articulold] FOREIGN
KEY([Articulold]) REFERENCES [dbo].[Articulos] ([Id]) ON DELETE CASCADE GO
ALTER TABLE [dbo].[Movimientos] CHECK CONSTRAINT
[FK_Movimientos_Articulos_Articulold] GO ALTER TABLE [dbo].[Movimientos]
WITH CHECK ADD CONSTRAINT
[FK_Movimientos_TipoMovimientos_TipoMovimientold] FOREIGN
KEY([TipoMovimientold]) REFERENCES [dbo].[TipoMovimientos] ([Id]) ON
DELETE CASCADE GO ALTER TABLE [dbo].[Movimientos] CHECK CONSTRAINT
[FK_Movimientos_TipoMovimientos_TipoMovimientold] GO ALTER TABLE
[dbo].[Pedidos] WITH CHECK ADD CONSTRAINT [FK_Pedidos_Clientes_Clienteld]
FOREIGN KEY([Clienteld]) REFERENCES [dbo].[Clientes] ([Id]) ON DELETE

```



```
CASCADE GO ALTER TABLE [dbo].[Pedidos] CHECK CONSTRAINT  
[FK_Pedidos_Clientes_Clienteld] GO
```

Hazme INSERTs con datos de prueba con ese script. Necesito POR LO MENOS 80 Artículos, 30 Movimientos, y 10 registros para todas las demás tablas.

Respuesta:

```
USE [PapeleriaObligatorio];  
GO
```

-- Insertando datos de prueba en la tabla Artículos

```
INSERT INTO [dbo].[Articulos] (Nombre, Codigo, Descripcion, Precio)  
VALUES
```

```
('Cuaderno A4', 'CUA4', 'Cuaderno tamaño A4 con 100 hojas', 3.50),  
( 'Bolígrafo Azul', 'BOLAZ', 'Bolígrafo de tinta azul', 0.50),  
( 'Lápiz HB', 'LAPHB', 'Lápiz grafito HB', 0.25),  
( 'Goma de Borrar', 'GOMA1', 'Goma de borrar blanca', 0.30),  
( 'Marcador Permanente', 'MARKP', 'Marcador permanente negro', 1.20),  
( 'Resaltador Amarillo', 'RESA', 'Resaltador amarillo', 0.80),  
( 'Tijeras', 'TIJ', 'Tijeras de acero inoxidable', 2.00),  
( 'Pegamento en Barra', 'PEGB', 'Pegamento en barra 20g', 0.75),  
( 'Cartulina Blanca', 'CARB', 'Cartulina blanca tamaño A3', 0.50),  
( 'Grapadora', 'GRAP', 'Grapadora metálica', 5.00),  
( 'Cinta Adhesiva', 'CINT', 'Cinta adhesiva transparente', 1.00),  
( 'Corrector Líquido', 'CORL', 'Corrector líquido blanco', 1.25),  
( 'Carpeta Archivadora', 'CARP', 'Carpeta archivadora con anillas', 4.50),  
( 'Rotulador Rojo', 'ROTR', 'Rotulador de tinta roja', 0.70),  
( 'Bloc de Notas', 'BLON', 'Bloc de notas adhesivas', 1.50),  
( 'Papel Fotocopia A4', 'PAPA4', 'Papel de fotocopia tamaño A4', 3.00),  
( 'Calculadora Científica', 'CALC', 'Calculadora científica con funciones  
avanzadas', 15.00),  
( 'Regla 30cm', 'REG30', 'Regla de plástico 30 cm', 0.50),  
( 'Compás', 'COMP', 'Compás metálico', 2.00),  
( 'Estuche', 'EST', 'Estuche escolar', 3.00),  
( 'Mochila Escolar', 'MOCH', 'Mochila escolar con compartimentos', 25.00),  
( 'Cuaderno de Bocetos', 'CUAB', 'Cuaderno de bocetos tamaño A4', 4.00),  
( 'Pegamento Líquido', 'PEGL', 'Pegamento líquido 100ml', 1.00),  
( 'Sacapuntas', 'SAC', 'Sacapuntas metálico', 0.50),  
( 'Lápices de Colores', 'LAPC', 'Set de 12 lápices de colores', 3.00),  
( 'Agenda 2024', 'AGEN24', 'Agenda año 2024', 7.00),  
( 'Plastilina', 'PLAS', 'Set de plastilina de colores', 2.50),  
( 'Pinceles', 'PINC', 'Set de pinceles para pintar', 3.50),  
( 'Tempera', 'TEMP', 'Set de temperas de colores', 4.00),  
( 'Paleta para Pintar', 'PALP', 'Paleta de plástico para mezclar pintura', 1.50),  
( 'Portaminas', 'PORT', 'Portaminas con 3 minas de repuesto', 2.50),  
( 'Minas 0.5mm', 'MIN05', 'Set de minas 0.5mm para portaminas', 1.00),  
( 'Borrador para Pizarra', 'BORP', 'Borrador de fieltro para pizarra blanca', 1.25),  
( 'Marcadores para Pizarra', 'MARP', 'Set de 4 marcadores para pizarra blanca',  
5.00),  
( 'Tiza Blanca', 'TIZB', 'Caja de tiza blanca', 1.00),  
( 'Tiza de Colores', 'TIZC', 'Caja de tiza de colores', 1.50),
```

('Carpetas de Plástico', 'CARPP', 'Set de 10 carpetas de plástico transparentes', 3.00),  
 ('Archivador Metálico', 'ARCHM', 'Archivador metálico de 4 cajones', 50.00),  
 ('Cajas de Archivo', 'CAJAR', 'Set de 5 cajas de archivo', 10.00),  
 ('Rotulador Permanente', 'ROTP', 'Rotulador permanente negro', 1.00),  
 ('Pluma Estilográfica', 'PLUM', 'Pluma estilográfica con cartuchos', 12.00),  
 ('Cartuchos de Tinta', 'CART', 'Set de 5 cartuchos de tinta para pluma', 2.00),  
 ('Cuaderno de Matemáticas', 'CUAM', 'Cuaderno cuadriculado para matemáticas', 3.00),  
 ('Libreta de Notas', 'LIBN', 'Libreta de notas tamaño A6', 2.00),  
 ('Set de Geometría', 'SETG', 'Set de geometría con regla, compás y transportador', 3.50),  
 ('Plastilina de Colores', 'PLACO', 'Set de plastilina de colores', 2.50),  
 ('Tijeras de Punta Redonda', 'TIJR', 'Tijeras de punta redonda para niños', 1.50),  
 ('Cinta Correctora', 'CINC', 'Cinta correctora', 2.00),  
 ('Papel de Embalar', 'PAPEM', 'Rollo de papel de embalar', 3.00),  
 ('Papel de Regalo', 'PAPRG', 'Rollo de papel de regalo', 2.00),  
 ('Envoltorio para Libros', 'ENVL', 'Set de 10 envoltorios para libros', 5.00),  
 ('Caja de Lápices', 'CAJAL', 'Caja de 24 lápices de colores', 6.00),  
 ('Archivador de Palanca', 'ARCHP', 'Archivador de palanca tamaño A4', 4.00),  
 ('Bolsa de Plástico', 'BOLP', 'Bolsa de plástico con cierre hermético', 1.50),  
 ('Bolígrafo Negro', 'BOLN', 'Bolígrafo de tinta negra', 0.50),  
 ('Bolígrafo Rojo', 'BOLR', 'Bolígrafo de tinta roja', 0.50),  
 ('Marcador Fluorescente', 'MARF', 'Marcador fluorescente', 1.00),  
 ('Papel Mantequilla', 'PAPM', 'Rollo de papel mantequilla', 2.50),  
 ('Libreta de Dibujo', 'LIBD', 'Libreta de dibujo tamaño A4', 4.00),  
 ('Separadores de Cartulina', 'SEPC', 'Set de 10 separadores de cartulina', 1.50),  
 ('Cinta Adhesiva de Doble Cara', 'CINDC', 'Cinta adhesiva de doble cara', 3.00),  
 ('Estuche de Lápices', 'ESTL', 'Estuche con 24 lápices de colores', 10.00),  
 ('Carpeta con Separadores', 'CARSE', 'Carpeta con separadores tamaño A4', 5.00),  
 ('Archivador A-Z', 'ARCHAZ', 'Archivador A-Z tamaño A4', 6.00),  
 ('Set de Sobres', 'SETS', 'Set de 50 sobres blancos', 3.00),  
 ('Borrador de Plástico', 'BORPL', 'Borrador de plástico blanco', 0.30),  
 ('Grapas', 'GRAPA', 'Caja de grapas', 1.00),  
 ('Cuaderno de Espiral', 'CUAE', 'Cuaderno de espiral tamaño A5', 2.50),  
 ('Bloc de Dibujo', 'BLOD', 'Bloc de dibujo tamaño A3', 5.00),  
 ('Papel Crepé', 'PAPC', 'Set de 10 hojas de papel crepé', 2.00),  
 ('Bolígrafo Verde', 'BOLV', 'Bolígrafo de tinta verde', 0.50),  
 ('Cuaderno de Música', 'CUAMU', 'Cuaderno de música con pentagramas', 3.50);

– Insertando datos de prueba en la tabla Clientes

```

INSERT INTO [dbo].[Clientes] (RazonSocial, Rut, NombreCliente_Nombre,
NombreCliente_Apellido, Direccion_NombreCalle, Direccion_NumeroPuerta,
Direccion_Ciudad, Direccion_DistanciaKm)
VALUES
('Papelería El Escriba', '12345678-9', 'Juan', 'Pérez', 'Calle Falsa', '123',
'Montevideo', 15.5),
('Oficinas Express', '87654321-0', 'Ana', 'González', 'Av. Libertador', '456', 'Salto',
200.0),
('Librería Central', '11223344-5', 'Carlos', 'Rodríguez', 'Calle Real', '789',
'Paysandú', 250.0),

```

('La Oficina Moderna', '55667788-9', 'Laura', 'Fernández', 'Calle Principal', '321',  
 'Maldonado', 120.0),  
 ('Papeles y Más', '99887766-5', 'María', 'López', 'Avenida Siempre Viva', '987',  
 'Riviera', 300.0),  
 ('Librería Escolar', '44556677-8', 'José', 'García', 'Calle Comercio', '654',  
 'Canelones', 50.0),  
 ('Tienda de Papelería', '33445566-7', 'Lucía', 'Martínez', 'Av. Central', '543', 'San  
 José', 80.0),  
 ('El Mundo del Papel', '22334455-6', 'Luis', 'Hernández', 'Calle 8 de Octubre',  
 '210', 'Durazno', 160.0),  
 ('Librería Estrella', '66778899-0', 'Marta', 'Gómez', 'Calle Nueva', '147', 'Colonia',  
 130.0),  
 ('Papelería y Regalos', '88990011-2', 'Pedro', 'Díaz', 'Avenida Principal', '369',  
 'Florida', 90.0);

– Insertando datos de prueba en la tabla Configuraciones

INSERT INTO [dbo].[Configuraciones] (Nombre, Valor)

VALUES

('DescuentoMayoreo', 10),  
 ('Impuesto', 21),  
 ('EnvioGratis', 50),  
 ('StockMinimo', 5),  
 ('TiempoEntrega', 7),  
 ('HoraCierre', 18),  
 ('HoraApertura', 9),  
 ('DiasDeTrabajo', 5),  
 ('NumeroMaximoPedidos', 100),  
 ('PromocionMes', 20);

– Insertando datos de prueba en la tabla Usuarios

INSERT INTO [dbo].[Usuarios] (Email, Password, PasswordSinEncrypt, NombreCompleto\_Nombre, NombreCompleto\_Apellido, Discriminator)

VALUES

('juan.perez@escriba.com', 'password123', 'password123', 'Juan', 'Pérez', ''),  
 ('ana.gonzalez@oficinas.com', 'oficinas456', 'oficinas456', 'Ana', 'González', ''),  
 ('carlos.rodriguez@libreria.com', 'libreria789', 'libreria789', 'Carlos', 'Rodríguez',  
 ''),  
 ('laura.fernandez@moderna.com', 'moderna321', 'moderna321', 'Laura',  
 'Fernández', ''),  
 ('maria.lopez@papeles.com', 'papeles987', 'papeles987', 'María', 'López', ''),  
 ('jose.garcia@escolar.com', 'escolar654', 'escolar654', 'José', 'García', ''),  
 ('lucia.martinez@papeleria.com', 'papeleria543', 'papeleria543', 'Lucía',  
 'Martínez', ''),  
 ('luis.hernandez@mundo.com', 'mundo210', 'mundo210', 'Luis', 'Hernández', ''),  
 ('marta.gomez@estrella.com', 'estrella147', 'estrella147', 'Marta', 'Gómez', ''),  
 ('pedro.diaz@regalos.com', 'regalos369', 'regalos369', 'Pedro', 'Díaz', '');

– Insertando datos de prueba en la tabla TipoMovimientos

INSERT INTO [dbo].[TipoMovimientos] (Nombre, TipoMovStock)

VALUES

('Entrada', 1),  
 ('Salida', -1),

('Devolución', 1),  
( 'Venta', -1),  
( 'Ajuste de Inventario', 0),  
( 'Transferencia', 0),  
( 'Donación', -1),  
( 'Destrucción', -1),  
( 'Producción', 1),  
( 'Reparación', 1);

– Insertando datos de prueba en la tabla Pedidos

INSERT INTO [dbo].[Pedidos] (Clienteld, Fecha, FechaPrometida, Anulado, Discriminator)

VALUES

(1, '2024-06-01 10:00:00', '2024-06-10 10:00:00', 0, ''),  
(2, '2024-06-02 11:00:00', '2024-06-11 11:00:00', 0, ''),  
(3, '2024-06-03 12:00:00', '2024-06-12 12:00:00', 0, ''),  
(4, '2024-06-04 13:00:00', '2024-06-13 13:00:00', 0, ''),  
(5, '2024-06-05 14:00:00', '2024-06-14 14:00:00', 0, ''),  
(6, '2024-06-06 15:00:00', '2024-06-15 15:00:00', 0, ''),  
(7, '2024-06-07 16:00:00', '2024-06-16 16:00:00', 0, ''),  
(8, '2024-06-08 17:00:00', '2024-06-17 17:00:00', 0, ''),  
(9, '2024-06-09 18:00:00', '2024-06-18 18:00:00', 0, ''),  
(10, '2024-06-10 19:00:00', '2024-06-19 19:00:00', 0, '');

– Insertando datos de prueba en la tabla Movimientos

INSERT INTO [dbo].[Movimientos] (FechaMovimiento, Articulold, TipoMovimientold, CantUnidades, EmailUsuario)

VALUES

('2024-06-01 10:00:00', 1, 1, 50, 'juan.perez@escriba.com'),  
( '2024-06-02 11:00:00', 2, 2, 20, 'ana.gonzalez@oficinas.com'),  
( '2024-06-03 12:00:00', 3, 1, 30, 'carlos.rodriguez@libreria.com'),  
( '2024-06-04 13:00:00', 4, 2, 10, 'laura.fernandez@moderna.com'),  
( '2024-06-05 14:00:00', 5, 1, 40, 'maria.lopez@papeles.com'),  
( '2024-06-06 15:00:00', 6, 2, 15, 'jose.garcia@escolar.com'),  
( '2024-06-07 16:00:00', 7, 1, 25, 'lucia.martinez@papeleria.com'),  
( '2024-06-08 17:00:00', 8, 2, 12, 'luis.hernandez@mundo.com'),  
( '2024-06-09 18:00:00', 9, 1, 35, 'marta.gomez@estrella.com'),  
( '2024-06-10 19:00:00', 10, 2, 22, 'pedro.diaz@regalos.com'),  
( '2024-06-11 10:00:00', 11, 1, 45, 'juan.perez@escriba.com'),  
( '2024-06-12 11:00:00', 12, 2, 18, 'ana.gonzalez@oficinas.com'),  
( '2024-06-13 12:00:00', 13, 1, 32, 'carlos.rodriguez@libreria.com'),  
( '2024-06-14 13:00:00', 14, 2, 14, 'laura.fernandez@moderna.com'),  
( '2024-06-15 14:00:00', 15, 1, 38, 'maria.lopez@papeles.com'),  
( '2024-06-16 15:00:00', 16, 2, 16, 'jose.garcia@escolar.com'),  
( '2024-06-17 16:00:00', 17, 1, 27, 'lucia.martinez@papeleria.com'),  
( '2024-06-18 17:00:00', 18, 2, 13, 'luis.hernandez@mundo.com'),  
( '2024-06-19 18:00:00', 19, 1, 34, 'marta.gomez@estrella.com'),  
( '2024-06-20 19:00:00', 20, 2, 21, 'pedro.diaz@regalos.com'),  
( '2024-06-21 10:00:00', 21, 1, 49, 'juan.perez@escriba.com'),  
( '2024-06-22 11:00:00', 22, 2, 20, 'ana.gonzalez@oficinas.com'),  
( '2024-06-23 12:00:00', 23, 1, 31, 'carlos.rodriguez@libreria.com'),  
( '2024-06-24 13:00:00', 24, 2, 11, 'laura.fernandez@moderna.com'),

```
('2024-06-25 14:00:00', 25, 1, 41, 'maria.lopez@papeles.com'),  
( '2024-06-26 15:00:00', 26, 2, 17, 'jose.garcia@escolar.com'),  
( '2024-06-27 16:00:00', 27, 1, 29, 'lucia.martinez@papeleria.com'),  
( '2024-06-28 17:00:00', 28, 2, 13, 'luis.hernandez@mundo.com'),  
( '2024-06-29 18:00:00', 29, 1, 36, 'marta.gomez@estrella.com'),  
( '2024-06-30 19:00:00', 30, 2, 19, 'pedro.diaz@regalos.com');
```

Los datos fueron modificados en algunas partes, por ejemplo, el Enum de TipoMovStock no logró reconocerlo, por lo que el valor -1 se cambió a 1, y el valor 1 a 2.

- Investigar una manera de exportar el código fuente del proyecto de forma segura y fácil:

i want to export an entire source code from a visual studio solution to a text file,  
how can i do that?

To export the entire source code from a Visual Studio solution to a text file, you can use a PowerShell script. Here's a step-by-step guide to achieve this:

1. **Open PowerShell:** Press Win + X and select "Windows PowerShell" from the menu.
2. **Navigate to the Solution Directory:** Use the `cd` command to navigate to the directory where your Visual Studio solution is located. For example:

```
powershell  
Copy code  
cd C:\path\to\your\solution
```

3. **Run the Script:** Use the following PowerShell script to concatenate all the source code files into a single text file. This script will search for files with common source code extensions (like `.cs`, `.cpp`, `.h`, etc.) and append their content to a file named `AllSourceCode.txt`.

```
powershell  
Copy code  
# Define the output file  
$outputFile = "AllSourceCode.txt"  
  
# Clear the output file if it exists  
if (Test-Path $outputFile) {  
    Remove-Item $outputFile  
}  
  
# Define the file extensions to include  
$extensions = @("*.cs", "*.cpp", "*.h", "*.js", "*.html",  
    "*.css", "*.ts", "*.vb", "*.cshtml", "*.xaml")  
  
# Get all the files with the specified extensions  
foreach ($ext in $extensions) {  
    Get-ChildItem -Recurse -Filter $ext | ForEach-Object {  
        Add-Content -Path $outputFile -Value "=== Start of  
$($_.FullName) ==="  
        Add-Content -Path $outputFile -Value (Get-Content  
$_.FullName)
```

```

        Add-Content -Path $outputFile -Value "=== End of
$(($_.FullName) ==="
        Add-Content -Path $outputFile -Value "`n"
    }
}

```

Eso fue lo que se utilizó para traer el código fuente, el cual está en el siguiente punto.

## Código Fuente

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Pa
peleriaContext.cs ===

```

```

using Papeleria.LogicaNegocio;

using LogicaNegocio.Entidades;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using Azure.Core.GeoJson;

using System.Security.Cryptography;

using Papeleria.LogicaNegocio.Entidades;

namespace AccesoDatos.EntityFramework
{
    public class PapeleriaContext : DbContext
    {
        public DbSet<Usuario> Usuarios { get; set; }

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

        public DbSet<Administrador> Admins { get; set; }

        public DbSet<Encargado> Encargados { get; set; }

        public DbSet<TipoMovimiento> TipoMovimientos { get; set; }

        public DbSet<Movimiento> Movimientos { get; set; }

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

```

```

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{

optionsBuilder.UseSqlServer(@"SERVER=(localdb)\MsSqlLocalDb;DATABASE=PapeleriaObligatorio;Integrated Security=true;");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)
{

    modelBuilder.Entity<PedidoComun>();
    modelBuilder.Entity<PedidoExpress>();

}

}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\PapeleriaContext.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Repositorios\RepositorioArticuloEF.cs ===

```

using Azure;
using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;

```

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

```
namespace AccesoDatos.EntityFramework.Repositorios
```

```
{
```

```
    public class RepositorioArticuloEF : IRepositoryArticulo
```

```
    {
```

```
        private PapeleriaContext _context;
```

```
        public RepositorioArticuloEF()
```

```
        {
```

```
            this._context = new PapeleriaContext();
```

```
        }
```

```
        public bool Add(Articulo aAgrega
```

```
        {
```

```
            try
```

```
            {
```

```
                this._context.Articulos.Add(aAgrega
```

```
                this._context.SaveChanges();
```

```
                return true;
```

```
            }
```

```
            catch (Exception ex)
```

```
            {
```

```
                throw ex;
```

```
            }
```

```
        }
```

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

```
        {
```

```
            return this._context.Articulos;
```

```
        }
```

```
        public Articulo FindByID(int id)
```

```
        {
```

```
            return this._context.Articulos.Where(articulo => articulo.Id == id).FirstOrDefault();
```



```
}
```

```
public IEnumerable<Articulo> GetMovidosByFecha(DateTime startdate, DateTime  
enddate, int pag, int size)  
{  
    return this._context.Movimientos.Where(m => m.FechaMovimiento >= startdate &&  
m.FechaMovimiento <= enddate).Select(m => m.Articulo).Distinct()  
        .Skip((pag - 1) * size).Take(size).ToList();  
}
```

```
public bool Remove(int id)  
{  
    try  
    {  
        Articulo aBorrar = FindByID(id);  
        this._context.Articulos.Remove(aBorrar);  
        this._context.SaveChanges();  
        return true;  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
}
```

```
public bool Update(Articulo aModificar)  
{  
    try  
    {  
        this._context.Articulos.Update(aModificar);  
        this._context.SaveChanges();  
        return true;  
    }  
}
```

```

        catch (Exception ex)
        {
            throw ex;
        }
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Repositorios\RepositorioArticuloEF.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Repositorios\RepositorioClienteEF.cs ===

```

using AccesoDatos.EntityFramework;
using Microsoft.EntityFrameworkCore;
using LogicaNegocio.Entidades;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaNegocio.ValueObjects;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.AccesoDatos.EntityFramework.Repositorios
{
    public class RepositorioClienteEF : IRepositoryCliente
    {
        private PapeleriaContext _context;
        public RepositorioClienteEF()
        {
            this._context = new PapeleriaContext();

```

```

    }
    public bool Add(Cliente aAgregar)
    {
        try
        {
            this._context.Clientes.Add(aAgregar);
            this._context.SaveChanges();
            return true;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    public IEnumerable<Cliente> ClientesXmonto(double monto)
    {
        List<Cliente> clientes = new List<Cliente>();
        int iva = this._context.Configuraciones
            .Where(config => config.Nombre == "IVA")
            .FirstOrDefault().Valor;
        int modRecargoCom = this._context.Configuraciones
            .Where(config => config.Nombre == "ModRecComun")
            .FirstOrDefault().Valor;
        int modRecargoExp = this._context.Configuraciones
            .Where(config => config.Nombre == "ModRecExpress")
            .FirstOrDefault().Valor;
        int modMismoDia = this._context.Configuraciones
            .Where(config => config.Nombre == "ModMismoDia")
            .FirstOrDefault().Valor;
        IEnumerable<Pedido> pedidos = _context.Pedidos
            .Include(pedido => pedido.Lineas)
            .AsEnumerable()

```

```
                .Where(pedido => pedido.CalcularPrecio(iva, modRecargoCom,
modRecargoExp, modMismoDia) >= monto)
                .ToList();
```

```
        foreach (Pedido pedido in pedidos)
        {
            if (!clientes.Contains(pedido.ClienteObj)) clientes.Add(pedido.ClienteObj);
        }
        return clientes;
    }
}
```

```
public IEnumerable<Cliente> ClientesXnombreYapellido(string especifica)
{
    return this._context.Clientes.Where(cliente =>
cliente.NombreCliente.Nombre.Contains(especifica)
    || cliente.NombreCliente.Apellido.Contains(especifica)).ToList();
}
```

```
public IEnumerable<Cliente> FindAll()
{
    return this._context.Clientes;
}
```

```
public Cliente FindByID(int id)
{
    return this._context.Clientes.Where(cliente => cliente.Id == id).FirstOrDefault();
}
```

```
public bool Remove(int id)
{
    try
    {
        Cliente aBorrar = FindByID(id);
```

```

        this._context.Clientes.Remove(aBorrar);
        this._context.SaveChanges();
        return true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public bool Update(Cliente aModificar)
{
    try
    {
        this._context.Clientes.Update(aModificar);
        this._context.SaveChanges();
        return true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositoryoClienteEF.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositoryoConfiguracionEF.cs ===

using AccesoDatos.EntityFramework;

```

```
using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.AccesoDatos.EntityFramework.Repositorios
{
    public class RepositorioConfiguracionEF : IRepositoryConfig
    {
        PapeleriaContext _context;

        public RepositorioConfiguracionEF()
        {
            this._context = new PapeleriaContext();
        }

        public bool Add(Configuracion aAgregar)
        {
            throw new NotImplementedException();
        }

        public IEnumerable<Configuracion> FindAll()
        {
            throw new NotImplementedException();
        }

        public Configuracion FindByID(int id)
        {
            throw new NotImplementedException();
        }
    }
}
```

```

        public Configuracion FindByNombre(string nombre)
        {
            return this._context.Configuraciones.Where(config => config.Nombre ==
nombre).FirstOrDefault();
        }

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

        public bool Update(Configuracion aModificar)
        {
            throw new NotImplementedException();
        }
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
 shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re  
 positorios\RepositorioConfiguracionEF.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
 shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re  
 positorios\RepositorioMovimientoEF.cs ===

```

using AccesoDatos.EntityFramework;
using LogicaNegocio.Entidades;
using Microsoft.EntityFrameworkCore;
using Papeleria.LogicaNegocio.Entidades;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;

```

```

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

namespace Papeleria.AccesoDatos.EntityFramework.Repositorios
{
    public class RepositorioMovimientoEF : IRepositoryMovimiento
    {
        PapeleriaContext _context;

        public RepositorioMovimientoEF()
        {
            this._context = new PapeleriaContext();
        }

        public bool Add(Movimiento aAgregar)
        {
            try
            {
                Artículo elArticulo = this._context.Articulos.Where(a => a.Id ==
aAgregar.Articuloid).FirstOrDefault();

                TipoMovimiento elTipo = this._context.TipoMovimientos.Where(t => t.Id ==
aAgregar.TipoMovimientoid).FirstOrDefault();

                Usuario elUsuario = this._context.Usuarios.Where(u => u.Email ==
aAgregar.EmailUsuario).FirstOrDefault();

                aAgregar.IsValid(this._context.Configuraciones.Where(c => c.Nombre ==
"Tope").FirstOrDefault().Valor);

                this._context.Movimientos.Add(aAgregar);

                this._context.SaveChanges();

                return true;
            }
            catch (MovException ex)
            {
                throw ex;
            }
        }
    }
}

```



```

        catch (Exception ex)
        {
            throw ex;
        }
    }

    public IEnumerable<Movimiento> FindAll()
    {
        return this._context.Movimientos.Include(m => m.TipoMovimiento).Include(m =>
m.Articulo);
    }

    public Movimiento FindByID(int id)
    {
        return this._context.Movimientos.Where(t => t.Id == id).FirstOrDefault();
    }

    public IEnumerable<Movimiento> GetMovs(int pag, int size)
    {
        return this._context.Movimientos.Include(m =>
m.Articulo).Include(m => m.TipoMovimiento).Skip((pag - 1) * size).Take(size).ToList();
    }

    public IEnumerable<Movimiento> GetByArtyTipo(int articuloId, int tipoMovId, int pag,
int size)
    {
        Artículo articulo = this._context.Articulos.Where(a => a.Id ==
articuloId).FirstOrDefault();

        TipoMovimiento tipo = this._context.TipoMovimientos.Where(t => t.Id ==
tipoMovId).FirstOrDefault();

        return this._context.Movimientos.Include(tipo => tipo.Articulo).Include(tipo =>
tipo.TipoMovimiento)

            .Where(mov => mov.TipoMovimiento == tipo && mov.Articulo == articulo)

            .OrderByDescending(mov => mov.FechaMovimiento).ThenBy(mov =>
mov.CantUnidades).Skip((pag - 1) * size).Take(size).ToList();
    }

```

```

    }

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

    public bool Update(Movimiento aModificar)
    {
        throw new NotImplementedException();
    }
}

}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositoryMovimientoEF.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositoryPedidoEF.cs ===

```

```

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

namespace AccesoDatos.EntityFramework.Repositorios
{
    public class RepositoryPedidoEF : IRepositoryPedido
    {

```

```

PapeleriaContext _context;

public RepositorioPedidoEF()
{
    this._context = new PapeleriaContext();
}

public bool Add(Pedido aAgregar)
{
    try
    {
        Articulo elArticulo;
        foreach(Linea linea in aAgregar.Lineas)
        {
            elArticulo = this._context.Articulos.Where(articulo => articulo.Id ==
linea.Articuloid).FirstOrDefault();

            this._context.Pedidos.Add(aAgregar);
            this._context.SaveChanges();
            return true;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

public IEnumerable<Pedido> FindAll()
{
    return this._context.Pedidos.Include(pedido => pedido.Lineas);
}

public Pedido FindByID(int id)
{

```

```
        return this._context.Pedidos.Where(pedido => pedido.Id == id).FirstOrDefault();  
    }  
}
```

```
public bool Remove(int id)  
{  
    try  
    {  
        Pedido aBorrar = FindByID(id);  
        this._context.Pedidos.Remove(aBorrar);  
        this._context.SaveChanges();  
        return true;  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
}
```

```
public bool Update(Pedido aModificar)  
{  
    try  
    {  
        this._context.Pedidos.Update(aModificar);  
        this._context.SaveChanges();  
        return true;  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
}
```

```
public double CalcularPrecio(Pedido pedido)
```

```

    {
        return pedido.CalcularPrecio((this._context.Configuraciones.
            Where(config => config.Nombre == "IVA").FirstOrDefault()).Valor,
            (this._context.Configuraciones.
                Where(config => config.Nombre == "ModRecComun").FirstOrDefault()).Valor,
            (this._context.Configuraciones.
                Where(config => config.Nombre == "ModRecExpress").FirstOrDefault()).Valor,
            (this._context.Configuraciones.
                Where(config => config.Nombre == "ModMismoDia").FirstOrDefault()).Valor);
    }

}

}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositorioPedidoEF.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositorioTipoMovimientoEF.cs ===

```

```

using AccesoDatos.EntityFramework;
using LogicaNegocio.Entidades;
using Papeleria.LogicaNegocio.Entidades;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.AccesoDatos.EntityFramework.Repositorios
{

```

```

public class RepositorioTipoMovimientoEF : IRepositoryTipoMovimiento
{
    private PapeleriaContext _context;
    public RepositorioTipoMovimientoEF()
    {
        this._context = new PapeleriaContext();
    }

    public bool Add(TipoMovimiento aAgregar)
    {
        try
        {
            aAgregar.IsValid();
            this._context.TipoMovimientos.Add(aAgregar);
            this._context.SaveChanges();
            return true;
        }
        catch (TMovException ex)
        {
            throw ex;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    public IEnumerable<TipoMovimiento> FindAll()
    {
        return this._context.TipoMovimientos;
    }

    public TipoMovimiento FindByID(int id)

```

```

{
    return this._context.TipoMovimientos.Where(t => t.Id == id).FirstOrDefault();
}

public bool Remove(int id)
{
    try
    {

        TipoMovimiento aBorrar = FindByID(id);

        IEnumerable<Movimiento> movimientos = this._context.Movimientos.Where(m
=> m.TipoMovimiento.Id == aBorrar.Id).ToList();

        if (movimientos.Count() == 0)
        {
            this._context.TipoMovimientos.Remove(aBorrar);
            this._context.SaveChanges();
            return true;
        }
        else return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public bool Update(TipoMovimiento aModificar)
{
    try
    {

        IEnumerable<Movimiento> movimientos = this._context.Movimientos.Where(m
=> m.TipoMovimiento.Id == aModificar.Id).ToList();

        if (movimientos.Count() == 0)

```

```

        {
            this._context.TipoMovimientos.Update(aModificar);
            this._context.SaveChanges();
            return true;
        }
        else return false;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Repositorios\RepositorioTipoMovimientoEF.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Repositorios\RepositorioUsuarioEF.cs ===

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection.Metadata.Ecma335;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

```

```

namespace AccesoDatos.EntityFramework
{

```



```

public class RepositorioUsuarioEF : IRepositoryUsuario
{
    private PapeleriaContext _context;

    public RepositorioUsuarioEF()
    {
        this._context = new PapeleriaContext();
    }

    public bool Add(Usuario aAgregar)
    {
        try
        {
            this._context.Usuarios.Add(aAgregar);
            this._context.SaveChanges();
            return true;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    public IEnumerable<Usuario> FindAll()
    {
        return this._context.Usuarios;
    }

    public Usuario FindByEmail(string email)
    {
        try
        {
            return this._context.Usuarios.Where(admin => admin.Email ==
email).FirstOrDefault();
        }
    }
}

```

```
        catch (Exception ex) { throw ex; }  
    }
```

```
public Usuario FindByID(int id)  
{  
    return this._context.Usuarios.Where(admin => admin.Id == id).FirstOrDefault();  
}
```

```
public bool Remove(int id)  
{  
    try  
    {  
        Usuario aBorrar = FindByID(id);  
        this._context.Usuarios.Remove(aBorrar);  
        this._context.SaveChanges();  
        return true;  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
}
```

```
public bool Update(Usuario aModificar)  
{  
    try  
    {  
        this._context.Usuarios.Update(aModificar);  
        this._context.SaveChanges();  
        return true;  
    }  
    catch (Exception ex)  
    {
```

```

        throw ex;
    }
}

}

}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\AccesoDatos\EntityFramework\Re
positorios\RepositoryUsuarioEF.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Program.cs ===

```

```

using AccesoDatos.EntityFramework;
using AccesoDatos.EntityFramework.Repositorios;
using ApplicationLogic.UseCases.TeamsUCs;
using LogicaNegocio.InterfacesRepositorio;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.IdentityModel.Tokens;
using Microsoft.OpenApi.Models;
using Papeleria.AccesoDatos;
using Papeleria.AccesoDatos.EntityFramework.Repositorios;
using Papeleria.LogicaAplicacion.CasosDeUso.Administradores;
using Papeleria.LogicaAplicacion.CasosDeUso.Articulos;
using Papeleria.LogicaAplicacion.CasosDeUso.Movimientos;
using Papeleria.LogicaAplicacion.CasosDeUso.Pedidos;
using Papeleria.LogicaAplicacion.CasosDeUso.TMov;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using Swashbuckle.AspNetCore.Filters;
using System.Text;

```

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

builder.Services.AddScoped<IRepositorioArticulo, RepositorioArticuloEF>();
builder.Services.AddScoped<IRepositorioPedido, RepositorioPedidoEF>();
builder.Services.AddScoped<IRepositorioTipoMovimiento, RepositorioTipoMovimientoEF>();
builder.Services.AddScoped<IRepositorioConfig, RepositorioConfiguracionEF>();
builder.Services.AddScoped<IRepositorioMovimiento, RepositorioMovimientoEF>();
builder.Services.AddScoped<IRepositorioUsuario, RepositorioUsuarioEF>();

builder.Services.AddScoped<IEncontrarArticulosOrd, EncontrarArticulosOrdCU>();
builder.Services.AddScoped<IGetPedidosDesc, GetPedidosDescCU>();
builder.Services.AddScoped<IGetAllTMov, GetAllTMovCU>();
builder.Services.AddScoped<ICrearTMov, CrearTMovCU>();
builder.Services.AddScoped<IFindTMovById, FindTMovByIdCU>();
builder.Services.AddScoped<IDeleteTMov, DeleteTMovCU>();
builder.Services.AddScoped<IUpdateTMov, UpdateTMovCU>();
builder.Services.AddScoped<ICrearMovimiento, CrearMovimientoCU>();
builder.Services.AddScoped<IGetAllMovs, GetAllMovsCU>();
builder.Services.AddScoped<IGetByArtyTipo, GetByArtyTipoCU>();
builder.Services.AddScoped<IGetArticulosByFecha, GetArticulosByFechaCU>();
builder.Services.AddScoped<IGetMovsXFecha, GetMovsXFechaCU>();
builder.Services.AddScoped<IFindUserByEmail, FindUserByEmailCU>();
builder.Services.AddControllers();

var Clave = "ZWRpw6fDo28gZW0gY29tcHVOYWVvcme=";
```

```

builder.Services.AddAuthentication(auth =>
{
    auth.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

    auth.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
})
.AddJwtBearer(auth =>
{
    auth.RequireHttpsMetadata = false;

    auth.SaveToken = true;

    auth.TokenValidationParameters = new
Microsoft.IdentityModel.Tokens.TokenValidationParameters
    {
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.ASCII.GetBytes(Clave)),
        ValidateIssuer = false,
        ValidateAudience = false
    };
});

var ruta = System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Papeleria.ApiRest.xml");

builder.Services.AddSwaggerGen(
    opciones =>
    {
        opciones.AddSecurityDefinition("oauth2", new
Microsoft.OpenApi.Models.OpenApiSecurityScheme()
        {
            Description = "Autorización estándar mediante esquema Bearer",
            In = ParameterLocation.Header,
            Name = "Authorization",
            Type = SecuritySchemeType.ApiKey
        });
        opciones.OperationFilter<SecurityRequirementsOperationFilter>();
    }
);

```

```

    opciones.IncludeXmlComments(ruta);
    opciones.SwaggerDoc("v1", new Microsoft.OpenApi.Models.OpenApiInfo
    {
        Title = "Papeleria Obligatorio",
        Description = "Aplicacion para administrar el deposito de la papeleria.",
        Contact = new Microsoft.OpenApi.Models.OpenApiContact
        {
            Email = "mm.mateomartinatto@gmail.com"
        },
        Version = "v1"
    });
}
);

```

```

var app = builder.Build();

```

```

// Configure the HTTP request pipeline.

```

```

if (app.Environment.IsDevelopment())

```

```

{
    app.UseSwagger();
    app.UseSwaggerUI();
}

```

```

app.UseHttpsRedirection();

```

```

app.UseAuthentication();

```

```

app.UseAuthorization();

```

```

app.MapControllers();

```

```

app.Run();

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Program.cs ===

```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\ArticulosContr  
oller.cs ===
```

```
using LogicaNegocio.Entidades;  
using LogicaNegocio.InterfacesRepositorio;  
using Microsoft.AspNetCore.Authorization;  
using Microsoft.AspNetCore.Mvc;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;  
using System.Net;
```

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

```
namespace ApiRest.Controllers  
{  
    [Route("api/Articulos")]  
    [ApiController]  
    [Authorize]  
    public class ArticulosController : ControllerBase  
    {  
        private IEncontrarArticulosOrd _encontrarArticulosOrd;  
        private IGetArticulosByFecha _getArticulosByFecha;  
  
        public ArticulosController(IEncontrarArticulosOrd encontrarArticulosOrd,  
        IGetArticulosByFecha getArticulosByFecha)  
        {  
            _encontrarArticulosOrd = encontrarArticulosOrd;  
            _getArticulosByFecha = getArticulosByFecha;  
        }  
        /// <summary>
```

```

    /// Metodo para traer todos los articulos.
    /// </summary>
    /// <returns>Articulos ordenados por nombre ascendentemente</returns>
    // GET api/<ArticulosController>
    [HttpGet("")]
    public ActionResult<IEnumerable<ArticuloDTO>> Get()
    {
        try
        {
            IEnumerable<ArticuloDTO> articulos = _encontrarArticulosOrd.GetArticulosOrd();
            if (articulos == null) return NoContent();
            return Ok(articulos);
        }
        catch (Exception ex)
        {
            return BadRequest(ex.Message);
        }
    }
    /// <summary>
    /// Metodo para filtrar articulos que fueron usados en al menos un movimiento entre
    dos fechas.
    /// </summary>
    /// <param name="startdate">Fecha de inicio</param>
    /// <param name="enddate">Fecha de fin</param>
    /// <param name="pageNumber">Numero de paginado</param>
    /// <returns>Articulos que fueron movidos entre las dos fechas.</returns>

    [HttpGet("GetByFechas/Page/{pageNumber}/startdate={startdate}/enddate={enddate}")]
    public ActionResult<IEnumerable<ArticuloDTO>> GetByFechas(DateTime startdate,
    DateTime enddate, int pageNumber)
    {
        if (pageNumber < 1) { return BadRequest("Pagina invalida."); }

        IEnumerable<ArticuloDTO> arts =
        this._getArticulosByFecha.GetArticulosByFecha(startdate, enddate, pageNumber);

```



```

        if (arts.Count() == 0) { return NoContent(); }

        return Ok(arts);
    }
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\ArticulosContr
oller.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\LoginControlle
r.cs ===

```

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Authorization;
using Papeleria.LogicaAplicacion;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.CasosDeUso.Administradores;

```

```

namespace Papeleria.ApiRest.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class LoginController : ControllerBase
    {
        private IFindUserByEmail _getUser;

        public LoginController(IFindUserByEmail getUser) { _getUser = getUser; }

        /// <summary>
        /// Metodo para permitir inicio de sesion y obtener un jwt para uso de la api
        /// </summary>
        /// <param name="usuario">nombre de usuario y contraseña</param>
    }
}

```

```

/// <returns>Token y datos del usuario</returns>
[HttpPost]
[AllowAnonymous]
[Route("login")]
public ActionResult<UsuarioDTO> Login([FromBody] UsuarioDTO usuario)
{
    try
    {
        ManejadorJwt handler = new ManejadorJwt(_getUser);
        var usr = handler.ObtenerUsuario(usuario.Email);
        if (usr == null || usr.PasswordSinEncrypt != usuario.PasswordSinEncrypt)
            return Unauthorized("Credenciales inválidas. Reintente");
        var token = ManejadorJwt.GenerarToken(usr);
        string rol = "Admin";
        if (usr.IsEncargado == true) rol = "Encargado";
        return Ok(new
        {
            Token = token,
            Usuario = usr,
            Rol = rol

        });
    }
    catch (Exception ex)
    {
        return Unauthorized(new
        {
            Message = "Se produjo un error.Intente nuevamente" });
    }
}
}

```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\LoginControlle  
r.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\MovimientosC  
ontroller.cs ===
```

```
using Microsoft.AspNetCore.Authorization;  
  
using Microsoft.AspNetCore.Mvc;  
  
using Papeleria.LogicaAplicacion.DTOs;  
  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento;  
  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;
```

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

```
namespace Papeleria.ApiRest.Controllers  
{  
    [Route("api/Movimientos")]  
    [ApiController]  
    [Authorize]  
    public class MovimientosController : ControllerBase  
    {  
  
        private ICrearMovimiento _crearMov;  
        private IGetAllMovs _getMovs;  
        private IGetByArtyTipo _getByArtyTipo;  
        private IGetMovsXFecha _getMovsXFecha;  
  
        public MovimientosController(ICrearMovimiento crearMov, IGetAllMovs getAllMovs,  
        IGetByArtyTipo getByArtyTipo, IGetMovsXFecha getMovsXFecha)  
        {  
            this._crearMov = crearMov;  
            this._getMovs = getAllMovs;  
        }  
    }  
}
```

```

        this._getByArtyTipo = getByArtyTipo;
        this._getMovsXFecha = getMovsXFecha;
    }

    /// <summary>
    /// Metodo para traer todos los movimientos.
    /// </summary>
    /// <param name="pageNumber">Parametro para el numero de pagina.</param>
    /// <returns>Todos los movimientos, su articulo y su tipo de movimiento.</returns>

    [HttpGet("Page/{pageNumber}")]
    [ProducesResponseType(StatusCodes.Status200OK)]
    [ProducesResponseType(StatusCodes.Status204NoContent)]
    [ProducesResponseType(StatusCodes.Status400BadRequest)]
    public ActionResult<IEnumerable<MovimientoDTO>> GetMovs(int pageNumber)
    {
        if(pageNumber < 1) { return BadRequest("Pagina invalida."); }
        IEnumerable<MovimientoDTO> movs = this._getMovs.GetAllMovs(pageNumber);
        if(movs.Count() == 0) { return NoContent(); }
        return Ok(movs);
    }

    /// <summary>
    /// Metodo para filtrar los movimientos por un articulo y un tipo de movimiento
    /// </summary>
    /// <param name="articulold">ID del articulo</param>
    /// <param name="tipold">ID del tipo de movimiento</param>
    /// <param name="pageNumber">numero de pagina</param>
    /// <returns>Todos los movimientos del mismo tipo y mismo articulo.</returns>
    [HttpGet("GetByArtyTipo/Page/{pageNumber}/Articulo={articulold}/Tipo={tipold}")]
    [ProducesResponseType(StatusCodes.Status200OK)]
    [ProducesResponseType(StatusCodes.Status204NoContent)]
    [ProducesResponseType(StatusCodes.Status400BadRequest)]
    public ActionResult<IEnumerable<MovimientoDTO>> GetByArtyTipo(int articulold, int
tipold, int pageNumber)

```

```

    {
        if (pageNumber < 1) { return BadRequest("Pagina invalida."); }

        IEnumerable<MovimientoDTO> movs = this._getByArtyTipo.GetByArtyTipo(articulold,
tipold, pageNumber);

        if (movs.Count() == 0) { return NoContent(); }

        return Ok(movs);
    }

    /// <summary>

    /// Metodo para traer las cantidades movidas agrupadas por año, y dentro de año por
tipo de movimiento.

    /// </summary>

    /// <returns>Cantidades movidas agrupadas por año, y dentro de año por tipo de
movimiento.</returns>

    [HttpGet("GetMovsXFecha")]
    [ProducesResponseType(StatusCodes.Status200OK)]
    [ProducesResponseType(StatusCodes.Status204NoContent)]
    [ProducesResponseType(StatusCodes.Status400BadRequest)]
    public ActionResult<IEnumerable<FechaDTO>> GetMovsXFecha()
    {
        IEnumerable<FechaDTO> movs = this._getMovsXFecha.GetMovsXFecha();

        if (movs.Count() == 0) { return NoContent(); }

        return Ok(movs);
    }

    /// <summary>

    /// Metodo para registrar un movimiento en la base de datos.

    /// </summary>

    /// <param name="MovDTO">El articulo que se mueve, el tipo de movimiento, y la
cantidad de unidades movidas.</param>

    /// <returns></returns>

    [HttpPost("")]
    [ProducesResponseType(StatusCodes.Status201Created)]
    [ProducesResponseType(StatusCodes.Status400BadRequest)]

    public ActionResult<MovimientoDTO> CreateMovimiento([FromBody] MovimientoDTO
MovDTO)

```

```

    {
        try
        {
            this._crearMov.CrearMovimiento(MovDTO);
            return Created("api/Movimientos", MovDTO);
        }
        catch (Exception ex)
        {
            return BadRequest(new { mensaje = ex.Message });
        }
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\MovimientosC  
ontroller.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\PedidosContro  
ller.cs ===

```

using Microsoft.AspNetCore.Mvc;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;

```

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

```

namespace ApiRest.Controllers
{
    [Route("api/Pedidos")]
    [ApiController]
    public class PedidosController : ControllerBase
    {

```

```

private IGetPedidosDesc _getPedidosDesc;
public PedidosController(IGetPedidosDesc getPedidosDesc)
{
    _getPedidosDesc = getPedidosDesc;
}

// GET: api/<PedidosController>
[HttpGet(Name = "GetAllPedidos")]
public ActionResult<PedidoDTO> Get()
{
    try
    {
        IEnumerable<PedidoDTO> pedidos = _getPedidosDesc.GetPedidosDesc();
        if (pedidos == null) return NoContent();
        return Ok(pedidos);
    }
    catch (Exception ex)
    {
        return BadRequest(ex.Message);
    }
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\PedidosContro
ller.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\TipoMovimien
toController.cs ===

```

```

using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;

```

```

using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;

namespace Papeleria.ApiRest.Controllers
{
    [Route("api/TipoMovimientos")]
    [ApiController]
    public class TipoMovimientoController : ControllerBase
    {
        //TODO: Hacer los summaries de las apis
        private IGetAllTMov _getAllTMov;
        private ICrearTMov _crearTMov;
        private IFindTMovById _getTMovById;
        private IDeleteTMov _deleteTMov;
        private IUpdateTMov _updateTMov;

        public TipoMovimientoController(IGetAllTMov getAllTMov, ICrearTMov crearTMov,
            IFindTMovById getTMovById, IDeleteTMov deleteTMov, IUpdateTMov updateTMov)
        {
            _getAllTMov = getAllTMov;
            _crearTMov = crearTMov;
            _getTMovById = getTMovById;
            _deleteTMov = deleteTMov;
            _updateTMov = updateTMov;
        }

        /// <summary>
        /// Metodo para traer todos los tipos de movimiento
        /// </summary>
        /// <returns>Lista con tipos de movimiento</returns>
        [HttpGet("")]
        public ActionResult<IEnumerable<TipoMovimientoDTO>> Get()
        {
            return Ok(this._getAllTMov.GetAllTMov());
        }
    }
}

```



```

    }

    /// <summary>
    /// Metodo para traer un tipo de movimiento dado un id
    /// </summary>
    /// <param name="id">id de tipo de movimiento</param>
    /// <returns>Nombre y Valor del tipo de movimiento</returns>
    [HttpGet("{id}")]
    public ActionResult<TipoMovimientoDTO> GetByld(int id)
    {
        return Ok(this._getTMovByld.FindByld(id));
    }

    /// <summary>
    /// Metodo para crear un tipo de movimiento
    /// </summary>
    /// <param name="tMovDTO">El tipo de movimiento, incluye nombre, y si es un
    ingreso o egreso</param>
    /// <returns>Codigo 201 Created si el tipo fue creado.</returns>
    [HttpPost("")]
    [ProducesResponseType(StatusCodes.Status201Created)]
    [ProducesResponseType(StatusCodes.Status400BadRequest)]
    public ActionResult<TipoMovimientoDTO> CreateMovimiento([FromBody]
TipoMovimientoDTO tMovDTO)
    {
        try
        {
            this._crearTMov.CrearTMov(tMovDTO);
            return Created("api/TipoMovimientos", tMovDTO);
        }
        catch (Exception ex)
        {
            return BadRequest(ex.Message);
        }
    }
}

```

```

/// <summary>
/// Metodo para Borrar un tipo de movimiento del sistema
/// </summary>
/// <param name="id">id de un tipo de movimiento</param>
/// <returns>200 OK si fue borrado</returns>
[HttpDelete("{id}")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
public ActionResult<TipoMovimientoDTO> DeleteTipo(int id)
{
    try
    {
        this._deleteTMov.DeleteTMov(id);
        return Ok("Tipo Movimiento " + id + " borrado correctamente");
    }
    catch (Exception ex)
    {
        return BadRequest(ex.Message);
    }
}

/// <summary>
/// Metodo para editar un tipo de movimiento ya existente
/// </summary>
/// <param name="tMovDTO">Un tipo de movimiento, con los datos que se haya
enviado para editar.</param>
/// <returns>200 OK si fue editado</returns>
[HttpPut("{id}")]
[ProducesResponseType(StatusCodes.Status200OK)]
[ProducesResponseType(StatusCodes.Status400BadRequest)]
public ActionResult<TipoMovimientoDTO> UpdateTipo([FromBody] TipoMovimientoDTO
tMovDTO)
{
    try

```

```

        {
            this._updateTMov.UpdateTMov(tMovDTO);
            return Ok();
        }
        catch (Exception ex)
        {
            return BadRequest(ex.Message);
        }
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\ApiRest\Controllers\TipoMovimien  
toController.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\ManejadorJwt.cs  
===

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text;
using System.Security.Claims;
using System.IdentityModel.Tokens.Jwt;
using Microsoft.IdentityModel.Tokens;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.CasosDeUso.Administradores;

namespace Papeleria.LogicaAplicacion
{

```

```

public class ManejadorJwt
{
    private IFindUserByEmail _getUser;

    public ManejadorJwt(IFindUserByEmail getUser) { _getUser = getUser; }

    public static string GenerarToken(UsuarioDTO usuarioDto)
    {
        var tokenHandler = new JwtSecurityTokenHandler();

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

        var clave = Encoding.ASCII.GetBytes("ZWRpw6fDo28gZW0gY29tcHVOYWVWcmE=");

        //Se incluye un claim para el rol

        var tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = new ClaimsIdentity(new Claim[]
            {
                new Claim(ClaimTypes.Email, usuarioDto.Email)
            }),
            Expires = DateTime.UtcNow.AddMonths(1),

            SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(clave),
                SecurityAlgorithms.HmacSha256Signature)
        };

        var token = tokenHandler.CreateToken(tokenDescriptor);

        return tokenHandler.WriteToken(token);
    }
}

```

```

        public UsuarioDTO ObtenerUsuario(string username)
        {
            {
                var usuario = this._getUser.FindUserByEmail(username);
                return usuario;
            }
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\ManejadorJwt.cs
===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\BorrarAdminCU.cs ===

```

```

using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.Excepciones;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class BorrarAdminCU : IBorrarAdmin
    {
        private IRepositoryUsuario _repositorioAdministrador;
    }
}

```

```

    public BorrarAdminCU(IRepositorioUsuario repositorioAdministrador)
    {
        this._repositorioAdministrador = repositorioAdministrador;
    }

    public void BorrarAdmin(int id)
    {
        this._repositorioAdministrador.Remove(id);
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\BorrarAdminCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\CrearAdminCU.cs ===

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class CrearAdminCU : ICrearAdmin
    {

```

```

private IRepositorioUsuario _repositorioAdmin;
public CrearAdminCU(IRepositorioUsuario repositorioAdmin)
{
    _repositorioAdmin = repositorioAdmin;
}
public void CrearAdmin(UsuarioDTO adminDto)
{
    Administrador admin = UsuarioDTOMapper.FromDtoAdmin(adminDto);
    Hash hash = new Hash();
    try
    {
        admin.IsValid();
        admin.PasswordSinEncrypt = admin.Password;
        admin.Password = hash.GetHashSha256(adminDto.Password);
        this._repositorioAdmin.Add(admin);
    }
    catch(Exception ex)
    {
        throw ex;
    }
}
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\CrearAdminCU.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\EditarAdminCU.cs ===

```

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;

```

```

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using Papeleria.LogicaAplicacion.Mappers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class EditarAdminCU : IEditarAdmin
    {
        private IRepositoryUsuario _repositorioAdmin;
        public EditarAdminCU(IRepositoryUsuario repositorioAdmin)
        {
            _repositorioAdmin = repositorioAdmin;
        }
        public bool EditarAdmin(UsuarioDTO aModificar)
        {
            Administrador admin = UsuarioDTOMapper.FromDtoAdmin(aModificar);
            Hash hash = new Hash();
            try
            {
                admin.Password = aModificar.PasswordSinEncrypt;
                admin.PasswordSinEncrypt = aModificar.PasswordSinEncrypt;
                admin.IsValid();
                admin.Password = hash.GetHashSha256(aModificar.PasswordSinEncrypt);
                return this._repositorioAdmin.Update(admin);
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
    }
}

```



```

    }
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Administradores\EditarAdminCU.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Administradores\EncontrarAdminsCU.cs ===

```

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using Papeleria.LogicaAplicacion.Mappers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class EncontrarAdminsCU : IEncontrarAdmins
    {
        private IRepositoryUsuario _repositorioAdministrador;

        public EncontrarAdminsCU(IRepositoryUsuario repositorioAdministrador)
        {
            this._repositorioAdministrador = repositorioAdministrador;
        }

        public IEnumerable<UsuarioDTO> FindAllAdmins()
        {

```

```

        IEnumerable<Usuario> administradores = this._repositorioAdministrador.FindAll();
        List<UsuarioDTO> administradoresDTO = new List<UsuarioDTO>();
        foreach(Administrador admin in administradores)
        {
            administradoresDTO.Add(UsuarioDTOMapper.FromAdmin(admin));
        }
        return administradoresDTO;
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad  
ministradores\EncontrarAdminsCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad  
ministradores\FindAdminByIdCU.cs ===

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using Papeleria.LogicaAplicacion.Mappers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class FindAdminByIdCU : IFindAdminById
    {
        private IRepositoryUsuario _repositorioAdmin;
    }
}

```

```

        public FindAdminByIdCU(IRepositorioUsuario repositorioAdmin)
        {
            _repositorioAdmin = repositorioAdmin;
        }

        public UsuarioDTO FindAdminById(int id)
        {
            return
UsuarioDTOMapper.FromAdmin((Administrador)this._repositorioAdmin.FindById(id));
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\FindAdminByIdCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad
ministradores\FindUserByEmailCU.cs ===

```

```

using LogicaNegocio.Entidades;

using LogicaNegocio.InterfacesRepositorio;

using Papeleria.LogicaAplicacion.DTOs;

using Papeleria.LogicaAplicacion.Mappers;

using Papeleria.LogicaNegocio.Entidades;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class FindUserByEmailCU : IFindUserByEmail
    {

```

```

private IRepositoryUsuario _repositorioAdministrador;
public FindUserByEmailCU(IRepositoryUsuario repositorioAdministrador)
{
    this._repositorioAdministrador = repositorioAdministrador;
}
public UsuarioDTO FindUserByEmail(string email)
{
    Usuario elUsuario = this._repositorioAdministrador.FindByEmail(email);
    if (elUsuario is Encargado)
    {
        UsuarioDTO usuario = UsuarioDTOMapper.FromEncargado(elUsuario as
Encargado);
        usuario.IsEncargado = true;
        return usuario;
    }
    if (elUsuario is Administrador) return UsuarioDTOMapper.FromAdmin(elUsuario as
Administrador);
    return null;
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad  
ministradores\FindUserByEmailCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad  
ministradores>LoginCU.cs ===

```

using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Reflection.Metadata;
using Papeleria.LogicaNegocio.Excepciones;
using LogicaNegocio.Entidades;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public class LoginCU : ILogin
    {
        private IRepositoryUsuario _repositorioAdmin;
        public LoginCU(IRepositoryUsuario repositorioAdmin)
        {
            _repositorioAdmin = repositorioAdmin;
        }
        public bool Login(string email, string password)
        {
            Usuario admin = this._repositorioAdmin.FindByEmail(email);

            if (admin != null)
            {
                Hash hash = new Hash();
                if(hash.GetHashSha256(password) == admin.Password) return true;
                //Hashea la contraseña.
                //Verifica si es la misma que ya está en la BD
                //Si es así, devuelve true.
                else return false;
            }
            else return false;
        }
    }
}

```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ad  
ministradores\LoginCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Arti  
culos\CrearArticuloCU.cs ===
```

```
using LogicaNegocio.Entidades;  
using LogicaNegocio.InterfacesRepositorio;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;  
using Papeleria.LogicaAplicacion.Mappers;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Articulos
```

```
{  
    public class CrearArticuloCU : ICrearArticulo  
    {  
        private IRepositoryArticulo _repositorioArticulo;  
        public CrearArticuloCU(IRepositoryArticulo repositorioArticulo) {  
            _repositorioArticulo = repositorioArticulo;  
        }  
        public void CrearArticulo(ArticuloDTO aCrear)  
        {  
            try  
            {  
                Articulo articulo = ArticuloDTOMapper.FromDto(aCrear);  
                articulo.IsValid();  
            }  
        }  
    }  
}
```

```

        _repositorioArticulo.Add(articulo);
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Articulos\CrearArticuloCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Articulos\EncontrarArticulosCU.cs ===

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;
using Papeleria.LogicaAplicacion.Mappers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Articulos
{
    public class EncontrarArticulosCU : IEncontrarArticulos
    {
        private IRepositoryArticulo _repositorioArticulo;

        public EncontrarArticulosCU(IRepositoryArticulo repositorioArticulo)
    }
}

```

```

    {
        _repositorioArticulo = repositorioArticulo;
    }

    public IEnumerable<ArticuloDTO> EncontrarArticulos()
    {
        IEnumerable<Articulo> articulos = this._repositorioArticulo.FindAll();
        List<ArticuloDTO> articulosDTO = new List<ArticuloDTO>();
        foreach (Articulo articulo in articulos)
        {
            articulosDTO.Add(ArticuloDTOMapper.ToDto(articulo));
        }
        return articulosDTO;
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Arti
culos\EncontrarArticulosCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Arti
culos\EncontrarArticulosOrdCU.cs ===

```

```

using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;
using Papeleria.LogicaAplicacion.Mappers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```



namespace Papeleria.LogicaAplicacion.CasosDeUso.Articulos

```
{  
    public class EncontrarArticulosOrdCU : IEncontrarArticulosOrd  
    {  
        private IRepositoryArticulo _repositorioArticulo;  
        public EncontrarArticulosOrdCU(IRepositoryArticulo repositorioArticulo)  
        {  
            _repositorioArticulo = repositorioArticulo;  
        }  
  
        public IEnumerable<ArticuloDTO> GetArticulosOrd()  
        {  
            return this._repositorioArticulo.FindAll().OrderBy(articulo =>  
articulo.Nombre).Select(articulo => ArticuloDTOMapper.ToDto(articulo));  
        }  
    }  
}
```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Arti  
culos\EncontrarArticulosOrdCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Arti  
culos\FindArticuloByIdCU.cs ===

```
using LogicaNegocio.InterfacesRepositorio;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;  
using Papeleria.LogicaAplicacion.Mappers;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Articulos
```

```
{  
    public class FindArticuloByIdCU : IEncontrarXIdArticulo  
    {  
        private IRepositoryArticulo _repositorioArticulo;  
        public FindArticuloByIdCU(IRepositoryArticulo repositorioArticulo)  
        {  
            _repositorioArticulo = repositorioArticulo;  
        }  
  
        public ArticuloDTO FindById(int id)  
        {  
            return ArticuloDTOMapper.ToDto(_repositorioArticulo.FindById(id));  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Articulos\FindArticuloByIdCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Articulos\GetArticulosByFechaCU.cs ===
```

```
using LogicaNegocio.InterfacesRepositorio;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;  
using Papeleria.LogicaAplicacion.Mappers;  
using Papeleria.LogicaNegocio.InterfacesRepositorio;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Articulos
```

```
{
```

```
    public class GetArticulosByFechaCU : IGetArticulosByFecha
```

```
    {
```

```
        private IRepositoryArticulo _repositorioArticulo;
```

```
        private IRepositoryConfig _repositorioConfig;
```

```
        public GetArticulosByFechaCU(IRepositoryArticulo repositorioArticulo,  
IRepositoryConfig repositorioConfig)
```

```
        {
```

```
            _repositorioArticulo = repositorioArticulo;
```

```
            this._repositorioConfig = repositorioConfig;
```

```
            _repositorioConfig = repositorioConfig;
```

```
        }
```

```
        public IEnumerable<ArticuloDTO> GetArticulosByFecha(DateTime startdate, DateTime  
enddate, int pag)
```

```
        {
```

```
            int size = this._repositorioConfig.FindByNombre("PageSize").Valor;
```

```
            return this._repositorioArticulo.GetMovidosByFecha(startdate, enddate, pag,  
size).Select(a => ArticuloDTOMapper.ToDto(a));
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Art  
iculos\GetArticulosByFechaCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Clie  
ntes\EncontrarClientesCU.cs ===
```

```
using LogicaNegocio.Entidades;
```

```
using LogicaNegocio.InterfacesRepositorio;
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Cientes
```

```
{
    public class EncontrarClientesCU : IEncontrarClientes
    {
        private IRepositoryCliente _repositorioCliente;
        public EncontrarClientesCU(IRepositoryCliente repositorioCliente)
        {
            this._repositorioCliente = repositorioCliente;
        }
        public IEnumerable<ClienteDTO> FindAllClientes()
        {
            IEnumerable<Cliente> clientes= this._repositorioCliente.FindAll();
            List<ClienteDTO> clientesDTO= new List<ClienteDTO>();
            foreach (Cliente cliente in clientes)
            {
                clientesDTO.Add(ClienteDTOMapper.ToDto(cliente));
            }
            return clientesDTO;
        }
    }
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Clie
ntes\EncontrarClientesCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Clie  
ntes\GetClientesXmontoCU.cs ===
```

```
using LogicaNegocio.Entidades;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;  
using Papeleria.LogicaAplicacion.Mappers;  
using Papeleria.LogicaNegocio.InterfacesRepositorio;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Clientes
```

```
{  
    public class GetClientesXmontoCU : IGetClientesXmonto  
    {  
        private IRepositoryioCliente _repositorioCliente;  
        public GetClientesXmontoCU(IRepositoryioCliente repositorioCliente)  
        {  
            _repositorioCliente = repositorioCliente;  
        }  
  
        public IEnumerable<ClienteDTO> GetClientesXmonto(double monto)  
        {  
            IEnumerable<Cliente> clientes = _repositorioCliente.ClientesXmonto(monto);  
            List<ClienteDTO> clientesDTO = new List<ClienteDTO>();  
            foreach (Cliente cliente in clientes)  
            {  
                clientesDTO.Add(ClienteDTOMapper.ToDto(cliente));  
            }  
        }  
    }  
}
```

```

    }
    return clientesDTO;
}
}
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Clie
ntes\GetClientesXmontoCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Clie
ntes\GetClientesXnombreYapellidoCU.cs ===

```

```

using LogicaNegocio.Entidades;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Clientes
{
    public class GetClientesXnombreYapellidoCU : IGetClientesXnombreYapellido
    {
        private IRepositoryCliente _repositorioCliente;

        public GetClientesXnombreYapellidoCU(IRepositoryCliente repositorioCliente)
        {
            _repositorioCliente = repositorioCliente;
        }

        public IEnumerable<ClienteDTO> GetClientesXnombreYapellido(string especifica)

```

```

        {
            IEnumerable<Cliente> clientes =
            _repositorioCliente.ClientesXnombreYapellido(especifica);

            List<ClienteDTO> clientesDTO = new List<ClienteDTO>();

            foreach (Cliente cliente in clientes)
            {
                clientesDTO.Add(ClienteDTOMapper.ToDto(cliente));
            }

            return clientesDTO;
        }
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shì\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Clie  
ntes\GetClientesXnombreYapellidoCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shì\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo  
vimientos\CrearMovimientoCU.cs ===

```

using Papeleria.LogicaAplicacion.DTOs;

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento;

using Papeleria.LogicaAplicacion.Mappers;

using Papeleria.LogicaNegocio.Entidades;

using Papeleria.LogicaNegocio.InterfacesRepositorio;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Movimientos
{
    public class CrearMovimientoCU : ICrearMovimiento

```

```

{
    private IRepositoryMovimiento _repositorioMovimiento;
    public CrearMovimientoCU(IRepositoryMovimiento repositorioMovimiento)
    {
        this._repositorioMovimiento = repositorioMovimiento;
    }
    public void CrearMovimiento(MovimientoDTO MovDTO)
    {
        Movimiento mov = MovimientoDTOMapper.FromDto(MovDTO);
        this._repositorioMovimiento.Add(mov);
    }
}

}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo
vimientos\CrearMovimientoCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo
vimientos\GetAllMovsCU.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Movimientos
{

```



```

public class GetAllMovsCU : IGetAllMovs
{
    private IRepositoryMovimiento _repositorioMovimiento;
    private IRepositoryConfig _repositorioConfig;

    public GetAllMovsCU(IRepositoryMovimiento repositorioMovimiento,
        IRepositoryConfig repositorioConfig)
    {
        this._repositorioMovimiento = repositorioMovimiento;
        this._repositorioConfig = repositorioConfig;
    }

    public IEnumerable<MovimientoDTO> GetAllMovs(int pag)
    {
        int size = this._repositorioConfig.FindByNombre("PageSize").Valor;
        return this._repositorioMovimiento.GetMovs(pag, size).Select(m =>
            MovimientoDTOMapper.ToDto(m));
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo
vimientos\GetAllMovsCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo
vimientos\GetByArtyTipoCU.cs ===

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento;
using Papeleria.LogicaAplicacion.Mappers;

```

```

using Papeleria.LogicaNegocio.InterfacesRepositorio;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Movimientos
{
    public class GetByArtyTipoCU : IGetByArtyTipo
    {
        private IRepositoryMovimiento _repositorioMovimiento;
        private IRepositoryConfig _repositorioConfig;

        public GetByArtyTipoCU(IRepositoryMovimiento repositorioMovimiento,
            IRepositoryConfig repositorioConfig)
        {
            this._repositorioMovimiento = repositorioMovimiento;
            this._repositorioConfig = repositorioConfig;
        }

        public IEnumerable<MovimientoDTO> GetByArtyTipo(int articuloId, int tipoId, int pag)
        {
            int size = this._repositorioConfig.FindByNombre("PageSize").Valor;

            return this._repositorioMovimiento.GetByArtyTipo(articuloId, tipoId, pag,
                size).Select(m => MovimientoDTOMapper.ToDto(m));
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo
vimientos\GetByArtyTipoCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo
vimientos\GetMovsXFechaCU.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;

```

```

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

namespace Papeleria.LogicaAplicacion.CasosDeUso.Movimientos
{
    public class GetMovsXFechaCU : IGetMovsXFecha
    {
        private IRepositorioMovimiento _repositorioMovimiento;

        public GetMovsXFechaCU(IRepositorioMovimiento repositorioMovimiento)
        {
            this._repositorioMovimiento = repositorioMovimiento;
        }

        public IEnumerable<FechaDTO> GetMovsXFecha()
        {
            IEnumerable<MovimientoDTO> movs =
this._repositorioMovimiento.FindAll().Select(m => MovimientoDTOMapper.ToDto(m));

            return movs.GroupBy(m => m.FechaMovimiento.Year).Select(movsPorFecha => new
FechaDTO
            {
                Fecha = movsPorFecha.Key,
                Total = movsPorFecha.Sum(m => m.CantUnidades),
                Movs = movsPorFecha.GroupBy(tmov => tmov.TipoMov.Nombre)
                .Select(tm => new TMovsDTO
                {
                    Nombre = tm.Key,
                    Total = tm.Sum(m => m.CantUnidades)
                }).ToList()
            });
        }
    }
}

```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Mo  
vimientos\GetMovsXFechaCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\AnularPedidoCU.cs ===
```

```
using LogicaNegocio.Entidades;  
using LogicaNegocio.InterfacesRepositorio;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Pedidos
```

```
{  
    public class AnularPedidoCU : IAnularPedido  
    {  
        private IRepositoryPedido _repositorioPedido;  
        public AnularPedidoCU(IRepositoryPedido repositorioPedido)  
        {  
            _repositorioPedido = repositorioPedido;  
        }  
        public void AnularPedido(int id)  
        {  
            try  
            {  
                Pedido elPedido = _repositorioPedido.FindByID(id);  
                elPedido.Anulado = true;  
                _repositorioPedido.Update(elPedido);  
            }  
            catch { }  
        }  
    }  
}
```

```

        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Pedidos\AnularPedidoCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Pedidos\CrearPedidoCU.cs ===

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Pedidos
{
    public class CrearPedidoCU : ICrearPedido
    {

```

```

private IRepositoryPedido _repositorioPedido;
private IRepositoryConfig _repositorioConfig;

public CrearPedidoCU(IRepositoryPedido repositorioPedido, IRepositoryConfig
repositorioConfig)
{
    _repositorioPedido = repositorioPedido;
    _repositorioConfig = repositorioConfig;
}

public void CrearPedido(PedidoDTO pedido, Boolean esExpress)
{
    if (esExpress)
    {
        try
        {
            Pedido elPedido = PedidoDTOMapper.FromDtoExpress(pedido);
            elPedido.Fecha = DateTime.Today;

            elPedido.IsValid(this._repositorioConfig.FindByNombre("PlazoEstipulado").Valor);
            this._repositorioPedido.Add(elPedido);
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
    else
    {
        try
        {
            Pedido elPedido = PedidoDTOMapper.FromDtoComun(pedido);
            elPedido.Fecha = DateTime.Today;

            elPedido.IsValid(this._repositorioConfig.FindByNombre("PlazoEstipuladoCom").Valor);

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped
idos\CrearPedidoCU.cs ===

```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped
idos\EncontrarPedidosCU.cs ===
```

```
using LogicaNegocio.Entidades;

using LogicaNegocio.InterfacesRepositorio;

using Papeleria.LogicaAplicacion.DTOs;

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;

using Papeleria.LogicaAplicacion.Mappers;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Pedidos
{
    public class EncontrarPedidosCU : IEncontrarPedidos
    {
        private IRepositorioPedido _repositorioPedido;
```

```

public EncontrarPedidosCU(IRepositorioPedido repositorioPedido)
{
    _repositorioPedido = repositorioPedido;
}

public IEnumerable<PedidoDTO> EncontrarPedidos()
{
    IEnumerable<Pedido> pedidos = this._repositorioPedido.FindAll();
    List<PedidoDTO> pedidosDto = new List<PedidoDTO>();
    foreach (Pedido pedido in pedidos)
    {
        if (pedido is PedidoComun)
            pedidosDto.Add(PedidoDTOMapper.FromComun(pedido as PedidoComun));
        else pedidosDto.Add(PedidoDTOMapper.FromExpress(pedido as PedidoExpress));
    }
    return pedidosDto;
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shì\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\EncontrarPedidosCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shì\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\EncontrarPrecioPedidoCU.cs ===

```

using LogicaNegocio.Entidades;
using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using System;

```



```

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

namespace Papeleria.LogicaAplicacion.CasosDeUso.Pedidos
{
    public class EncontrarPrecioPedidoCU : IEncontrarPrecioPedido
    {
        private IRepositoryPedido _repositorioPedido;
        private IRepositoryCliente _repositorioCliente;

        public EncontrarPrecioPedidoCU(IRepositoryPedido repositorioPedido,
        IRepositoryCliente repositorioCliente)
        {
            this._repositorioPedido = repositorioPedido;
            this._repositorioCliente = repositorioCliente;
        }

        public double EncontrarPrecioPedido(PedidoDTO pedido, Boolean esExpress)
        {
            if (pedido == null) return 0;
            if (esExpress)
            {
                PedidoExpress elPedido = PedidoDTOMapper.FromDtoExpress(pedido);
                elPedido.ClienteObj = _repositorioCliente.FindByID(elPedido.ClienteId);
                return _repositorioPedido.CalcularPrecio(elPedido);
            }
            else
            {
                PedidoComun elPedido = PedidoDTOMapper.FromDtoComun(pedido);
                elPedido.ClienteObj = _repositorioCliente.FindByID(elPedido.ClienteId);
                return _repositorioPedido.CalcularPrecio(elPedido);
            }
        }
    }
}

```

```
}  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\EncontrarPrecioPedidoCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\GetPedidosDescCU.cs ===
```

```
using LogicaNegocio.Entidades;  
using LogicaNegocio.InterfacesRepositorio;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;  
using Papeleria.LogicaAplicacion.Mappers;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.Pedidos  
{  
    public class GetPedidosDescCU : IGetPedidosDesc  
    {  
        private IRepositoryPedido _repositorioPedido;  
        public GetPedidosDescCU(IRepositoryPedido repositorioPedido)  
        {  
            _repositorioPedido = repositorioPedido;  
        }  
  
        public IEnumerable<PedidoDTO> GetPedidosDesc()  
        {
```

```

        IEnumerable<Pedido> pedidos = _repositorioPedido.FindAll().Where(pedido =>
pedido.Anulado).OrderByDescending(pedido => pedido.Fecha);

        List<PedidoDTO> pedidosDto = new List<PedidoDTO>();

        foreach (Pedido pedido in pedidos)
        {
            if (pedido is PedidoComun)
pedidosDto.Add(PedidoDTOMapper.FromComun(pedido as PedidoComun));
            else pedidosDto.Add(PedidoDTOMapper.FromExpress(pedido as PedidoExpress));
        }

        return pedidosDto;
    }
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\GetPedidosDescCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\GetPedidosPorFechaCU.cs ===

```

using LogicaNegocio.Entidades;

using LogicaNegocio.InterfacesRepositorio;

using Papeleria.LogicaAplicacion.DTOs;

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;

using Papeleria.LogicaAplicacion.Mappers;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.CasosDeUso.Pedidos
{
    public class GetPedidosPorFechaCU : IGetPedidosPorFecha

```

```

{
    private IRepositoryPedido _repositorioPedido;
    public GetPedidosPorFechaCU(IRepositoryPedido repositorioPedido)
    {
        this._repositorioPedido = repositorioPedido;
    }

    public IEnumerable<PedidoDTO> GetPedidosPorFecha(DateTime fecha)
    {
        IEnumerable<Pedido> pedidos = _repositorioPedido.FindAll().Where(pedido =>
pedido.FechaPrometida.Date == fecha.Date);

        List<PedidoDTO> pedidosDto = new List<PedidoDTO>();

        foreach (Pedido pedido in pedidos)
        {
            if (pedido is PedidoComun)
pedidosDto.Add(PedidoDTOMapper.FromComun(pedido as PedidoComun));
            else pedidosDto.Add(PedidoDTOMapper.FromExpress(pedido as PedidoExpress));
        }

        return pedidosDto;
    }
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\Ped  
idos\GetPedidosPorFechaCU.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM  
ov\CrearTMovCU.cs ===

```

using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;
using Papeleria.LogicaAplicacion.Mappers;
using Papeleria.LogicaNegocio.Entidades;
using Papeleria.LogicaNegocio.InterfacesRepositorio;

```

```

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

namespace Papeleria.LogicaAplicacion.CasosDeUso.TMov
{
    public class CrearTMovCU : ICrearTMov
    {
        private IRepositorioTipoMovimiento _repositorioTipoMovimiento;

        public CrearTMovCU(IRepositorioTipoMovimiento repositorioTipoMovimiento)
        {
            this._repositorioTipoMovimiento = repositorioTipoMovimiento;
        }

        public void CrearTMov(TipoMovimientoDTO tMovDTO)
        {
            TipoMovimiento tipo = TipoMovimientoDTOMapper.FromDto(tMovDTO);
            this._repositorioTipoMovimiento.Add(tipo);
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TMov\CrearTMovCU.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TMov\DeleteTMovCU.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesRepositorio;

```

```

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

namespace Papeleria.LogicaAplicacion.CasosDeUso.TMov
{
    public class DeleteTMovCU : IDeleteTMov
    {
        private IRepositorioTipoMovimiento _repositorioTipoMovimiento;
        public DeleteTMovCU(IRepositorioTipoMovimiento repositorioTipoMovimiento)
        {
            this._repositorioTipoMovimiento = repositorioTipoMovimiento;
        }

        public void DeleteTMov(int id)
        {
            try
            {
                this._repositorioTipoMovimiento.Remove(id);
            }
            catch (TMovException e) { throw; }
            catch (Exception e) { throw; }
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM
ov\DeleteTMovCU.cs ===

```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM  
ov\FindTMovByIdCU.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;  
using Papeleria.LogicaAplicacion.Mappers;  
using Papeleria.LogicaNegocio.InterfacesRepositorio;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.TMov  
{  
    public class FindTMovByIdCU : IFindTMovById  
    {  
        private IRepositoryTipoMovimiento _repositorioTipoMovimiento;  
        public FindTMovByIdCU(IRepositoryTipoMovimiento repositorioTipoMovimiento)  
        {  
            this._repositorioTipoMovimiento = repositorioTipoMovimiento;  
        }  
  
        public TipoMovimientoDTO FindById(int id)  
        {  
            return TipoMovimientoDTOMapper.ToDto(_repositorioTipoMovimiento.FindById(id));  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM  
ov\FindTMovByIdCU.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM  
ov\GetAllTMovCU.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;  
using Papeleria.LogicaAplicacion.Mappers;  
using Papeleria.LogicaNegocio.InterfacesRepositorio;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ApplicationLogic.UseCases.TeamsUCs  
{  
    public class GetAllTMovCU : IGetAllTMov  
    {  
        private IRepositoryTipoMovimiento _repositorioTipoMovimiento;  
        public GetAllTMovCU(IRepositoryTipoMovimiento repositorioTipoMovimiento)  
        {  
            this._repositorioTipoMovimiento = repositorioTipoMovimiento;  
        }  
        public IEnumerable<TipoMovimientoDTO> GetAllTMov()  
        {  
            return this._repositorioTipoMovimiento.FindAll().Select(tipo =>  
TipoMovimientoDTOMapper.ToDto(tipo));  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM  
ov\GetAllTMovCU.cs ===
```



```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM  
ov\UpdateTMovCU.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov;  
using Papeleria.LogicaAplicacion.Mappers;  
using Papeleria.LogicaNegocio.Excepciones;  
using Papeleria.LogicaNegocio.InterfacesRepositorio;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.CasosDeUso.TMov
```

```
{  
    public class UpdateTMovCU : IUpdateTMov  
    {  
        private IRepositoryTipoMovimiento _repositorioTipoMovimiento;  
        public UpdateTMovCU(IRepositoryTipoMovimiento repositorioTipoMovimiento)  
        {  
            this._repositorioTipoMovimiento = repositorioTipoMovimiento;  
        }  
        public void UpdateTMov(TipoMovimientoDTO tipoMovimiento)  
        {  
            try  
            {
```

```
                this._repositorioTipoMovimiento.Update(TipoMovimientoDTOMapper.FromDto(tipoMovimie  
                nto));
```

```
            }  
            catch (TMovException){ throw; }  
            catch (Exception ex) { throw; }
```

```

    }
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\CasosDeUso\TM
ov\UpdateTMovCU.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\ArticuloDT
O.cs ===

```

```

using LogicaNegocio.Entidades;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.DTOs
{
    public class ArticuloDTO
    {
        public int Id { get; set; }

        public static int Ultimold { get; set; }

        public string Nombre { get; set; }

        public string Codigo { get; set; }

        public string Descripcion { get; set; }

        public double Precio { get; set; }

        public int Stock { get; set; }

        public ArticuloDTO() { }

        public ArticuloDTO(Articulo articulo)
        {
            if(articulo != null)
            {

```

```

        Id = articulo.Id;
        Descripcion = articulo.Descripcion;
        Precio = articulo.Precio;
       Codigo = articulo.Codigo;
        Nombre = articulo.Nombre;
    }
}
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\ArticuloDT
O.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\ClienteDTO
.cs ===

```

```

using LogicaNegocio.Entidades;
using LogicaNegocio.ValueObjects;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.DTOs

```

```

{
    public class ClienteDTO
    {
        public int Id;
        public string Rut;
        public string RazonSocial;
        public string Nombre;
        public string Apellido;
    }
}

```

```

public string nombreCalle;
public string numeroPuerta;
public string ciudad;
public double distanciaKm;
public ClienteDTO() { }
public ClienteDTO(Cliente cliente)
{
    if (cliente != null)
    {
        this.Id = cliente.Id;
        this.RazonSocial = cliente.RazonSocial;
        this.Rut = cliente.Rut;
        this.Nombre = cliente.NombreCliente.Nombre;
        this.Apellido = cliente.NombreCliente.Apellido;
        this.nombreCalle = cliente.Direccion.NombreCalle;
        this.numeroPuerta = cliente.Direccion.NumeroPuerta;
        this.ciudad = cliente.Direccion.Ciudad;
        this.distanciaKm = cliente.Direccion.DistanciaKm;
    }
}
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\ClienteDTO
.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\Configurac
ionDTO.cs ===

```

```

using LogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;

```

```

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

namespace Papeleria.LogicaAplicacion.DTOs
{
    public class ConfiguracionDTO
    {
        public int Id { get; set; }
        public string Nombre { get; set; }
        public int Valor { get; set; }
        public ConfiguracionDTO(Configuracion config)
        {
            Id = config.Id;
            Nombre = config.Nombre;
            Valor = config.Valor;
        }
    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\Configurac
ionDTO.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\FechaDTO.
cs ===

```

```

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

```

```

namespace Papeleria.LogicaAplicacion.DTOs
{

```

```

public class FechaDTO
{
    public int Fecha { get; set; }

    public List<TMovsDTO> Movs { get; set; }

    public int Total { get; set; }
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\FechaDTO.
cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\LineaDTO.
cs ===

```

```

using LogicaNegocio.Entidades;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.DTOs

```

```

{
    public class LineaDTO
    {
        public int Id { get; set; }

        public double Precio { get; set; }

        public int CantUnidades { get; set; }

        public int ArticuloId { get; set; }
    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\LineaDTO.
cs ===

```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\Movimient  
oDTO.cs ===
```

```
using LogicaNegocio.Entidades;  
  
using Papeleria.LogicaAplicacion.Mappers;  
  
using Papeleria.LogicaNegocio.Entidades;  
  
using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel.DataAnnotations.Schema;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.DTOs  
{  
    public class MovimientoDTO  
    {  
        public int Id { get; set; }  
        public DateTime FechaMovimiento { get; set; }  
        public int Articulold { get; set; }  
        public ArtículoDTO? Artículo { get; set; }  
        public TipoMovimientoDTO? TipoMov { get; set; }  
        public int TipoMovimientold { get; set; }  
        public string EmailUsuario { get; set; }  
        public int CantUnidades { get; set; }  
  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\Movimient  
oDTO.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\PedidoDTO  
.cs ===
```

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

```
namespace Papeleria.LogicaAplicacion.DTOs
```

```
{  
    public class PedidoDTO  
    {  
        public static int UltimoId { get; set; }  
        public int Id { get; set; }  
        public int ClienteId { get; set; }  
        public DateTime Fecha { get; set; }  
        public DateTime FechaPrometida { get; set; }  
        public List<LineaDTO> Lineas { get; set; }  
        public Boolean Anulado { get; set; }  
  
    }  
}
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\PedidoDTO  
.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\TipoMovim  
ientoDTO.cs ===
```

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



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

```
namespace Papeleria.LogicaAplicacion.DTOs
```

```
{
    public class TipoMovimientoDTO
    {
        public int Id { get; set; }
        public string Nombre { get; set; }
        public TipoMovStock Tipo { get; set; }

    }
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\TipoMovim
ientoDTO.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\TMovsDTO
.cs ===
```

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

```
namespace Papeleria.LogicaAplicacion.DTOs
```

```
{
    public class TMovsDTO
    {
```

```

        public string Nombre { get; set; }

        public int Total { get; set; }

    }

}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\TMovsDTO
.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\UsuarioDT
O.cs ===

```

```

using LogicaNegocio.Entidades;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.DTOs

```

```

{
    [Index(nameof(Email), IsUnique = true)]

    public class UsuarioDTO
    {
        public int Id { get; set; }
        public string? Nombre { get; set; }
        public string? Apellido { get; set; }
        public string Email { get; set; }
        public string? Password { get; set; }
        public string PasswordSinEncrypt { get; set; }
        public bool? IsEncargado { get; set; }
    }
}

```

```
}  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\DTOs\UsuarioDT  
O.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Encriptado\Hash  
.cs ===
```

```
using System;  
using System.Security.Cryptography;  
using System.Text;  
  
public class Hash  
{  
    public Hash() { }  
    public string GetHashSha256(string text)  
    {  
        byte[] bytes = Encoding.UTF8.GetBytes(text);  
        using (SHA256 hashAlgorithm = SHA256.Create())  
        {  
            byte[] hash = hashAlgorithm.ComputeHash(bytes);  
            string hashString = string.Empty;  
            foreach (byte x in hash)  
            {  
                hashString += String.Format("{0:x2}", x);  
            }  
            return hashString;  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Encriptado\Hash  
.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Administrador\IBorrarAdmin.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador
```

```
{
```

```
    public interface IBorrarAdmin
```

```
    {
```

```
        void BorrarAdmin(int id);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Administrador\IBorrarAdmin.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Administrador\ICrearAdmin.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador
```

```

{
    public interface ICrearAdmin
    {
        void CrearAdmin(UsuarioDTO aCrear);
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\ICrearAdmin.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IEditarAdmin.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador

```

```

{
    public interface IEditarAdmin
    {
        bool EditarAdmin(UsuarioDTO aModificar);
    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IEditarAdmin.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IEncontrarAdmins.cs ===

```

```

using LogicaNegocio.Entidades;
using Papeleria.LogicaAplicacion.DTOs;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador
{
    public interface IEncontrarAdmins
    {
        IEnumerable<UsuarioDTO> FindAllAdmins();
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IEncontrarAdmins.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IFindAdminByld.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador
{
    public interface IFindAdminByld
    {

```

```

        UsuarioDTO FindAdminById(int id);
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IFindAdminById.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IFindUserByEmail.cs ===

using Papeleria.LogicaAplicacion.DTOs;

```

```

namespace Papeleria.LogicaAplicacion.CasosDeUso.Administradores
{
    public interface IFindUserByEmail
    {
        UsuarioDTO FindUserByEmail(string email);
    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\IFindUserByEmail.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\ILogin.cs ===

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador
{

```

```

public interface ILogin
{
    bool Login(string email, string password);
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Administrador\ILogin.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Articulo\ICrearArticulo.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo

```

```

{
    public interface ICrearArticulo
    {
        void CrearArticulo(ArticuloDTO aCrear);
    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Articulo\ICrearArticulo.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Articulo\IEncontrarArticulos.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;

```



```

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

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo
{
    public interface IEncontrarArticulos
    {
        IEnumerable<ArticuloDTO> EncontrarArticulos();
    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Articulo\IEncontrarArticulos.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Articulo\IEncontrarArticulosOrd.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo
{
    public interface IEncontrarArticulosOrd
    {
        IEnumerable<ArticuloDTO> GetArticulosOrd();
    }
}

```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Articulo\IEncontrarArticulosOrd.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Articulo\IEncontrarXIdArticulo.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo
```

```
{
```

```
    public interface IEncontrarXIdArticulo
```

```
    {
```

```
        ArticuloDTO FindById(int id);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Articulo\IEncontrarXIdArticulo.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Articulo\IGetArticulosByFecha.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo
```

```
{
```

```
    public interface IGetArticulosByFecha
```

```
    {
```

```
        IEnumerable<ArticuloDTO> GetArticulosByFecha(DateTime startdate, DateTime  
enddate, int pag);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Articulo\IGetArticulosByFecha.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Cliente\IEncontrarClientes.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente
```

```
{
```

```
    public interface IEncontrarClientes
```

```
    {
```

```
        IEnumerable<ClienteDTO> FindAllClientes();
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Cliente\IEncontrarClientes.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Cliente\IGetClientesXmonto.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente
```

```
{
```

```
    public interface IGetClientesXmonto
```

```
    {
```

```
        IEnumerable<ClienteDTO> GetClientesXmonto(double monto);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Cliente\IGetClientesXmonto.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Cliente\IGetClientesXnombreYapellido.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

```
using System.Collections;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente
```

```
{
```

```
    public interface IGetClientesXnombreYapellido
```

```
    {
```

```
        IEnumerable<ClienteDTO> GetClientesXnombreYapellido(string especifica);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Cliente\IGetClientesXnombreYapellido.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\ICrearMovimiento.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento
```

```
{
```

```
    public interface ICrearMovimiento
```

```
    {
```

```
        void CrearMovimiento(MovimientoDTO movimiento);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\ICrearMovimiento.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\IGetAllMovs.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento
```

```
{  
    public interface IGetAllMovs  
    {  
        IEnumerable<MovimientoDTO> GetAllMovs(int pag);  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\IGetAllMovs.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\IGetByArtyTipo.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento
```

```
{  
    public interface IGetByArtyTipo  
    {  
        IEnumerable<MovimientoDTO> GetByArtyTipo(int articulold, int tipold, int pag);  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\IGetByArtyTipo.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\IGetMovsXFecha.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Movimiento
```

```
{  
    public interface IGetMovsXFecha  
    {  
        public IEnumerable<FechaDTO> GetMovsXFecha();  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Movimiento\IGetMovsXFecha.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\IAnularPedido.cs ===
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido
```

```
{
```

```
    public interface IAnularPedido
```

```
    {
```

```
        void AnularPedido(int id);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\IAnularPedido.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\ICrearPedido.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido
```

```
{
```

```
    public interface ICrearPedido
```

```
    {
```

```
        void CrearPedido(PedidoDTO pedido, Boolean esExpress);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\ICrearPedido.cs ===
```



```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\IEncontrarPedidos.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido
```

```
{
```

```
    public interface IEncontrarPedidos
```

```
    {
```

```
        IEnumerable<PedidoDTO> EncontrarPedidos();
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\IEncontrarPedidos.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\Pedido\IEncontrarPrecioPedido.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido
```

```
{
```

```
    public interface IEncontrarPrecioPedido
```

```

    {
        double EncontrarPrecioPedido(PedidoDTO pedido, Boolean esExpress);
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Pedido\IEncontrarPrecioPedido.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Pedido\IGetPedidosDesc.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo

```

```

{
    public interface IGetPedidosDesc
    {
        IEnumerable<PedidoDTO> GetPedidosDesc();
    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Pedido\IGetPedidosDesc.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\Pedido\IGetPedidosPorFecha.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;

```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido
```

```
{
```

```
    public interface IGetPedidosPorFecha
```

```
    {
```

```
        IEnumerable<PedidoDTO> GetPedidosPorFecha(DateTime fecha);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasosDeUso\Pedido\IGetPedidosPorFecha.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasosDeUso\TMov\ICrearTMov.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov
```

```
{
```

```
    public interface ICrearTMov
```

```
    {
```

```
        public void CrearTMov(TipoMovimientoDTO tMovDTO);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasosDeUso\TMov\ICrearTMov.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasosDeUso\TMov\IDeleteTMov.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov
```

```
{
```

```

public interface IDeleteTMov
{
    void DeleteTMov(int id);
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\TMov\IDeleteTMov.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\TMov\IFindTMovById.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;

```

```

namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov

```

```

{
    public interface IFindTMovById
    {
        TipoMovimientoDTO FindById(int id);
    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\TMov\IFindTMovById.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos
DeUso\TMov\IGetAllTMov.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;

```

```

using Papeleria.LogicaNegocio.Entidades;

```

```

using System;

```

```

using System.Collections.Generic;

```

```

using System.Linq;

```

```

using System.Text;

```

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

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov
```

```
{
```

```
    public interface IGetAllTMov
```

```
    {
```

```
        public IEnumerable<TipoMovimientoDTO> GetAllTMov();
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\TMov\IGetAllTMov.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\TMov\IUpdateTMov.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```
namespace Papeleria.LogicaAplicacion.InterfacesCasosDeUso.TMov
```

```
{
```

```
    public interface IUpdateTMov
```

```
    {
```

```
        void UpdateTMov(TipoMovimientoDTO tipoMovimiento);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\InterfacesCasos  
DeUso\TMov\IUpdateTMov.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Articulo  
DTOMapper.cs ===
```

```
using LogicaNegocio.Entidades;
```

```
using Papeleria.LogicaAplicacion.DTOs;
```

```

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

namespace Papeleria.LogicaAplicacion.Mappers
{
    public class ArticuloDTOMapper
    {
        public static ArticuloDTO ToDto(Articulo articulo)
        {
            return new ArticuloDTO(articulo);
        }

        public static Articulo FromDto(ArticuloDTO articuloDTO)
        {
            if (articuloDTO == null)
            {
                throw new NotImplementedException();
            }

            return new Articulo(articuloDTO.Id, articuloDTO.Nombre, articuloDTO.Codigo,
            articuloDTO.Descripcion, articuloDTO.Precio);
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Articulo
DTOMapper.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Cliente
DTOMapper.cs ===

```

```

using LogicaNegocio.Entidades;
using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaNegocio.Excepciones;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaAplicacion.Mappers
{
    public class ClienteDTOMapper
    {
        public static ClienteDTO ToDto(Cliente cliente)
        {
            return new ClienteDTO(cliente);
        }

        public static Cliente FromDto(ClienteDTO clienteDTO)
        {
            if (clienteDTO == null)
            {
                throw new ClienteNoValidoException("Cliente invalido.");
            }

            return new Cliente(clienteDTO.Id, clienteDTO.RazonSocial, clienteDTO.Rut,
clienteDTO.Nombre, clienteDTO.Apellido,
                clienteDTO.nombreCalle, clienteDTO.numeroPuerta, clienteDTO.ciudad,
clienteDTO.distanciaKm);
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Cliente
DTOMapper.cs ===

```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Configu  
racionDTOMapper.cs ===
```

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using LogicaNegocio.Entidades;  
using Papeleria.LogicaAplicacion.DTOs;
```

```
namespace Papeleria.LogicaAplicacion.Mappers  
{  
    public class ConfiguracionDTOMapper  
    {  
        public ConfiguracionDTO ToDto(Configuracion config)  
        {  
            return new ConfiguracionDTO(config);  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Configu  
racionDTOMapper.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\LineaD  
TOMapper.cs ===
```

```
using LogicaNegocio.Entidades;  
using Papeleria.LogicaAplicacion.DTOs;  
using System;  
using System.Collections.Generic;
```



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

namespace Papeleria.LogicaAplicacion.Mappers
{
    public class LineaDTOMapper
    {
        public static LineaDTO toDTO(Linea linea)
        {
            return new LineaDTO
            {
                Id = linea.Id,
                Precio = linea.Precio,
                CantUnidades = linea.CantUnidades,
                ArticuloId = linea.ArticuloId
            };
        }

        public static Linea FromDTO(LineaDTO linea)
        {
            return new Linea
            {
                Id = linea.Id,
                Precio = linea.Precio,
                CantUnidades = linea.CantUnidades,
                ArticuloId = linea.ArticuloId,
            };
        }
    }
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\LineaD  
TOMapper.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Movimi  
entoDTOMapper.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaNegocio.Entidades;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.Mappers
```

```
{  
    public class MovimientoDTOMapper  
    {  
        public static Movimiento FromDto(MovimientoDTO movimientoDTO)  
        {  
            return new Movimiento  
            {  
                Id = movimientoDTO.Id,  
                FechaMovimiento = movimientoDTO.FechaMovimiento,  
                ArticuloId = movimientoDTO.ArticuloId,  
                TipoMovimientoId = movimientoDTO.TipoMovimientoId,  
                EmailUsuario = movimientoDTO.EmailUsuario,  
                CantUnidades = movimientoDTO.CantUnidades  
            };  
        }  
    }
```

```
    public static MovimientoDTO ToDto(Movimiento m)
```

```

    {
        return new MovimientoDTO
        {
            Id = m.Id,
            FechaMovimiento = m.FechaMovimiento,
            ArticuloId = m.ArticuloId,
            TipoMovimientoId = m.TipoMovimientoId,
            Articulo = ArticuloDTOMapper.ToDto(m.Articulo),
            TipoMov = TipoMovimientoDTOMapper.ToDto(m.TipoMovimiento),
            EmailUsuario = m.EmailUsuario,
            CantUnidades = m.CantUnidades
        };
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\MovimientoDTOMapper.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\PedidoDTOMapper.cs ===

```

```

using LogicaNegocio.Entidades;
using Papeleria.LogicaAplicacion.DTOs;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaAplicacion.Mappers
{
    public class PedidoDTOMapper

```

```

{
    public static PedidoDTO FromComun(PedidoComun pedido)
    {
        if (pedido == null) return null;
        return new PedidoDTO
        {
            Id = pedido.Id,
            ClientId = pedido.ClientId,
            Fecha = pedido.Fecha,
            FechaPrometida = pedido.FechaPrometida,
            Lineas = pedido.Lineas.Select(linea => LineaDTOMapper.toDTO(linea)).ToList(),
            Anulado = pedido.Anulado,
        };
    }
    public static PedidoDTO FromExpress(PedidoExpress pedido)
    {
        if (pedido == null) return null;
        return new PedidoDTO
        {
            Id = pedido.Id,
            ClientId = pedido.ClientId,
            Fecha = pedido.Fecha,
            FechaPrometida = pedido.FechaPrometida,
            Lineas = pedido.Lineas.Select(linea => LineaDTOMapper.toDTO(linea)).ToList(),
            Anulado = pedido.Anulado,
        };
    }
    public static PedidoComun FromDtoComun(PedidoDTO pedido)
    {
        if (pedido != null)
        {
            return new PedidoComun
            {

```

```

        Id = pedido.Id,
        ClientId = pedido.ClientId,
        Fecha = pedido.Fecha,
        FechaPrometida = pedido.FechaPrometida,
        Lineas = pedido.Lineas.Select(linea =>
LineaDTOMapper.FromDTO(linea)).ToList(),
        Anulado = pedido.Anulado,
    };
}
else throw new NotImplementedException();
}
public static PedidoExpress FromDtoExpress(PedidoDTO pedido)
{
    if (pedido != null)
    {
        return new PedidoExpress
        {
            Id = pedido.Id,
            ClientId = pedido.ClientId,
            Fecha = pedido.Fecha,
            FechaPrometida = pedido.FechaPrometida,
            Lineas = pedido.Lineas.Select(linea =>
LineaDTOMapper.FromDTO(linea)).ToList(),
            Anulado = pedido.Anulado,
        };
    }
    else throw new NotImplementedException();
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
 shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Pedido  
 DTOMapper.cs ===

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\TipoMo  
vimientoDTOMapper.cs ===
```

```
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaNegocio.Entidades;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaAplicacion.Mappers
```

```
{  
    public class TipoMovimientoDTOMapper  
    {  
        public static TipoMovimiento FromDto(TipoMovimientoDTO tipoMovimientoDTO)  
        {  
            return new TipoMovimiento  
            {  
                Id = tipoMovimientoDTO.Id,  
                Nombre = tipoMovimientoDTO.Nombre,  
                TipoMovStock = tipoMovimientoDTO.Tipo  
            };  
        }  
    }  
}
```

```
public static TipoMovimientoDTO ToDto(TipoMovimiento tipoMovimiento)  
{  
    if (tipoMovimiento == null) return null;  
    return new TipoMovimientoDTO  
    {  
        Id = tipoMovimiento.Id,  
        Nombre = tipoMovimiento.Nombre,  
    }  
}
```

```

        Tipo = tipoMovimiento.TipoMovStock
    };
}
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\TipoMo
vimientoDTOMapper.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Usuario
DTOMapper.cs ===

```

```

using Papeleria.LogicaAplicacion.DTOs;
using LogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Papeleria.LogicaNegocio.Entidades;
using Papeleria.LogicaNegocio.ValueObjects;
using static System.Runtime.InteropServices.JavaScript.JSType;
using LogicaNegocio.ValueObjects;

```

```

namespace Papeleria.LogicaAplicacion.Mappers
{
    public class UsuarioDTOMapper
    {
        public static UsuarioDTO FromAdmin(Administrador admin)
        {
            if (admin == null) return null;
            return new UsuarioDTO
            {

```

```

        Id = admin.Id,
        Nombre = admin.NombreCompleto.Nombre,
        Apellido = admin.NombreCompleto.Apellido,
        Email = admin.Email,
        Password = admin.Password,
        PasswordSinEncript = admin.PasswordSinEncript
    };
}

public static UsuarioDTO FromEncargado(Encargado enc)
{
    if (enc == null) return null;
    return new UsuarioDTO
    {
        Id = enc.Id,
        Nombre = enc.NombreCompleto.Nombre,
        Apellido = enc.NombreCompleto.Apellido,
        Email = enc.Email,
        Password = enc.Password,
        PasswordSinEncript = enc.PasswordSinEncript
    };
}

public static Encargado FromDtoEnc(UsuarioDTO encdto)
{
    if (encdto == null) return null;
    return new Encargado
    {
        Id = encdto.Id,
        NombreCompleto = new NombreCompleto(encdto.Nombre, encdto.Apellido),
        Email = encdto.Email,
        Password = encdto.Password,
        PasswordSinEncript = encdto.PasswordSinEncript
    };
}

```



```

    }
    public static Administrador FromDtoAdmin(UsuarioDTO admindto)
    {
        if (admindto == null) return null;
        return new Administrador
        {
            Id = admindto.Id,
            NombreCompleto = new NombreCompleto(admindto.Nombre, admindto.Apellido),
            Email = admindto.Email,
            Password = admindto.Password,
            PasswordSinEncript = admindto.PasswordSinEncript
        };
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\Mappers\Usuario
DTOMapper.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\obj\Debug\net8.
0\NETCoreApp,Version=v8.0.AssemblyAttributes.cs ===

```

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v8.0",
FrameworkDisplayName = ".NET 8.0")]
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\obj\Debug\net8.
0\NETCoreApp,Version=v8.0.AssemblyAttributes.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\obj\Debug\net8.  
0\Papeleria.LogicaAplicacion.AssemblyInfo.cs ===
```

```
//-----  
// <auto-generated>  
// Este código fue generado por una herramienta.  
// Versión de runtime:4.0.30319.42000  
//  
// Los cambios en este archivo podrían causar un comportamiento incorrecto y se  
perderán si  
// se vuelve a generar el código.  
// </auto-generated>  
//-----
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly: System.Reflection.AssemblyCompanyAttribute("Papeleria.LogicaAplicacion")]
```

```
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
```

```
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
```

```
[assembly:  
System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+d70b4dd48c049d4f3ba3  
b7a575c3e880ae3b8b47")]
```

```
[assembly: System.Reflection.AssemblyProductAttribute("Papeleria.LogicaAplicacion")]
```

```
[assembly: System.Reflection.AssemblyTitleAttribute("Papeleria.LogicaAplicacion")]
```

```
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]
```

```
// Generado por la clase WriteCodeFragment de MSBuild.
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\obj\Debug\net8.  
0\Papeleria.LogicaAplicacion.AssemblyInfo.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\obj\Debug\net8.  
0\Papeleria.LogicaAplicacion.GlobalUsings.g.cs ===
```

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

```
global using global::System;
```

```
global using global::System.Collections.Generic;
```

```
global using global::System.IO;
```

```
global using global::System.Linq;
```

```
global using global::System.Net.Http;
```

```
global using global::System.Threading;
```

```
global using global::System.Threading.Tasks;
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaAplicacion\obj\Debug\net8.  
0\Papeleria.LogicaAplicacion.GlobalUsings.g.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Adminis  
trador.cs ===
```

```
    using LogicaNegocio.ValueObjects;
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaNegocio.Entidades
```

```
{
```

```
    public class Administrador : Usuario
```

```
    {
```

```
        public Administrador() { }
```

```
        public Administrador(int id, string nombre, string apellido, string email, string  
password)
```

```
        {
```

```
            Id = id;
```

```

        NombreCompleto = new NombreCompleto(nombre, apellido);
        this.Email = email;
        this.Password = password;
    }

    public Administrador(string nombre, string apellido, string email, string password) {
        NombreCompleto = new NombreCompleto(nombre, apellido);
        this.Email = email;
        this.Password = password;
    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Adminis
trador.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Articulo.
cs ===

```

```

using Microsoft.EntityFrameworkCore;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesEntidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace LogicaNegocio.Entidades
{
    [Index(nameof(Nombre), IsUnique = true)]
    public class Articulo : IValid
    {
        public int Id { get; set; }
    }
}

```

```

public static int UltimoId { get; set; }

public string Nombre { get; set; }
public stringCodigo { get; set; }
public string Descripcion { get; set; }
public double Precio { get; set; }
public Articulo() { }
public Articulo(int id, string nombre, string codigo, string descripcion, double precio)
{
    Id = id;
    Nombre = nombre;
    Codigo = codigo;
    Descripcion = descripcion;
    Precio = precio;
}
public Articulo( string nombre, string codigo, string descripcion, double precio)
{
    Nombre = nombre;
    Codigo = codigo;
    Descripcion = descripcion;
    Precio = precio;
}
public void IsValid()
{
    if (Nombre == "" && Nombre.Length < 10 || Nombre.Length > 200) throw new
ArticuloNoValidoException("Nombre del articulo invalido.");
    if (Codigo == "") throw new ArticuloNoValidoException("Codigo invalido.");
    if (Descripcion == "" && Descripcion.Length < 5) throw new
ArticuloNoValidoException("Descripcion no valida");
    if (Precio <= 0) throw new ArticuloNoValidoException("Precio no valido");
}
}
}

```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Articulo.  
cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Cliente.  
cs ===
```

```
using LogicaNegocio.ValueObjects;  
using Papeleria.LogicaNegocio.Excepciones;  
using Papeleria.LogicaNegocio.InterfacesEntidades;  
using Papeleria.LogicaNegocio.ValueObjects;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace LogicaNegocio.Entidades
```

```
{  
    public class Cliente : IValid  
    {  
        public int Id { get; set; }  
        public static int Ultimold { get; set; }  
        public string RazonSocial { get; set; }  
        public string Rut { get; set; }  
        public NombreCliente NombreCliente { get; set; }  
        public Direccion Direccion { get; set; }  
  
        public Cliente() { }  
        public Cliente(int id, string razonSocial, string rut, string nombre, string apellido, string  
nombreCalle, string numeroPuerta, string ciudad, double distanciaKm)  
        {  
            Id = id;
```

```

        RazonSocial = razonSocial;

        Rut = rut;

        NombreCliente = new NombreCliente(nombre, apellido);

        Direccion = new Direccion(nombreCalle, numeroPuerta, ciudad, distanciaKm);
    }

    public Cliente(string razonSocial, string rut, string nombre, string apellido, string
nombreCalle, string numeroPuerta, string ciudad, double distanciaKm)
    {
        RazonSocial = razonSocial;

        Rut = rut;

        NombreCliente = new NombreCliente(nombre, apellido);

        Direccion = new Direccion(nombreCalle, numeroPuerta, ciudad, distanciaKm);
    }

    public void IsValid()
    {
        if (RazonSocial == null) throw new ClienteNoValidoException("Razon Social no
valida");

        if (Rut.Length != 12 || !int.TryParse(Rut, out int x)) throw new
ClienteNoValidoException("RUT no valido. Asegurese de que sea de 12 digitos y no tenga
letras.");

        NombreCliente.IsValid();

        Direccion.IsValid();
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Cliente.
cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Configur
acion.cs ===

using Papeleria.LogicaNegocio.InterfacesEntidades;

using System;

```

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

```
namespace LogicaNegocio.Entidades
```

```
{
    public class Configuracion
    {
        public int Id { get; set; }
        public string Nombre { get; set; }
        public int Valor { get; set; }
        public Configuracion() { }
        public Configuracion(int id, string nombre, int valor)
        {
            Id = id;
            Nombre = nombre;
            Valor = valor;
        }
        public Configuracion(string nombre, int valor)
        {
            Nombre = nombre;
            Valor = valor;
        }
    }
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Configur
acion.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Encarga
do.cs ===
```



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

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

```
{
    public class Encargado : Usuario
    {

    }
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Encarga
do.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Linea.cs
===
```

```
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesEntidades;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace LogicaNegocio.Entidades
```

```
{
    public class Linea : IValid
```

```

{
    public int Id { get; set; }
    public double Precio { get; set; }
    public int CantUnidades { get; set; }
    [ForeignKey(nameof(Articulo))] public int ArticuloId { get; set; }
    public Articulo ArticuloObj { get; set; }

    public Linea() { }
    public Linea(int id, int stock, Articulo articuloObj, int cantUnidades)
    {
        Id = id;
        CantUnidades = cantUnidades;
        ArticuloObj = articuloObj;
        Precio = articuloObj.Precio * cantUnidades;
    }
    public Linea(double precio, int stock, Articulo articuloObj, int cantUnidades)
    {
        Precio = precio;
        CantUnidades = cantUnidades;
        ArticuloObj = articuloObj;
    }

    public void IsValid()
    {
        if (ArticuloObj == null) throw new LineaNoValidaException("Articulo no es valido.");
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Linea.cs
===

```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Movimie  
nto.cs ===
```

```
using LogicaNegocio.Entidades;  
using Papeleria.LogicaNegocio.Excepciones;  
using Papeleria.LogicaNegocio.InterfacesEntidades;  
using System;  
using System.Collections.Generic;  
using System.ComponentModel.DataAnnotations.Schema;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.Entidades  
{  
    public class Movimiento  
    {  
        public int Id { get; set; }  
        public DateTime FechaMovimiento { get; set; }  
        [ForeignKey(nameof(Articulo))] public int Articuloid { get; set; }  
        public Articulo Articulo { get; set; }  
        [ForeignKey(nameof(TipoMovimiento))] public int TipoMovimientoid { get; set; }  
        public TipoMovimiento TipoMovimiento { get; set; }  
        public string EmailUsuario { get; set; }  
        public int CantUnidades { get; set; }  
  
        public void IsValid(int tope)  
        {  
            if (CantUnidades < 1) throw new MovimientoException("Cantidad invalida");  
            if (CantUnidades > tope) throw new MovimientoException("Cantidad supera tope");  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Movimie  
nto.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Pedido.  
cs ===
```

```
using System;
```

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

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaNegocio.Entidades
```

```
{
```

```
    public abstract class Pedido
```

```
    {
```

```
        public static int Ultimold { get; set; }
```

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

```
        [ForeignKey(nameof(Cliente))] public int Clienteld { get; set; }
```

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

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

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

```
        public List<Linea> Lineas { get; set; }
```

```
        public Boolean Anulado { get; set; }
```

```
        public Pedido() { }
```

```
        public abstract double CalcularPrecio(double impuesto, int modificadorRecargo, int  
modificadorRecargoExp, int modificadorMismoDia);
```

```
        public abstract void IsValid(int fechaPrometida);
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Pedido.  
cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\PedidoC  
omun.cs ===
```

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

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaNegocio.Entidades
```

```
{
```

```
    public class PedidoComun : Pedido
```

```
    {
```

```
        public static int ModificadorRecargo { get; set; }
```

```
        public PedidoComun(){}
```

```
        public PedidoComun(int id, Cliente obj, DateTime fechaPrometida)
```

```
        {
```

```
            Id = id;
```

```
            ClienteObj = obj;
```

```
            FechaPrometida = fechaPrometida;
```

```
            Lineas = new List<Linea>();
```

```
            Anulado = false;
```

```
        }
```

```
        public PedidoComun(Cliente obj, DateTime fechaPrometida)
```

```
        {
```

```
            ClienteObj = obj;
```

```
            FechaPrometida = fechaPrometida;
```

```

        Lineas = new List<Linea>();
        Anulado = false;

    }

    public override double CalcularPrecio(double impuesto, int modificadorRecargo, int
modificadorRecargoExp, int modificadorMismoDia)
    {
        {
            double suma = 0;
            foreach (Linea linea in Lineas)
            {
                suma += linea.Precio;
            }
            if (ClienteObj.Direccion.DistanciaKm > 100) suma += suma * modificadorRecargo
/ 100;
            suma += suma * impuesto / 100;
            return suma;
        }
    }

    public override void IsValid(int fechaPrometida)
    {
        if (this.FechaPrometida.Day - DateTime.Today.Day < fechaPrometida) throw new
PedidoNoValidoException("Fecha no valida.");
        if (this.FechaPrometida < DateTime.Today) throw new
PedidoNoValidoException("Fecha no valida.");
        if (Lineas.Count == 0) throw new PedidoNoValidoException("Debe tener al menos
una linea.");
    }

}

}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\PedidoC
omun.cs ===

```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\PedidoE  
xpress.cs ===
```

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

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace LogicaNegocio.Entidades
```

```
{
```

```
    public class PedidoExpress : Pedido
```

```
    {
```

```
        public static int ModificadorRecargo { get; set; }
```

```
        public static int ModificadorMismoDia { get; set; }
```

```
        public PedidoExpress() { }
```

```
        public PedidoExpress(int id, Cliente obj, DateTime fechaPrometida)
```

```
        {
```

```
            Id = id;
```

```
            ClienteObj = obj;
```

```
            FechaPrometida = fechaPrometida;
```

```
            Lineas = new List<Linea>();
```

```
            Anulado = false;
```

```
        }
```

```
        public PedidoExpress(Cliente obj, DateTime fechaPrometida)
```

```
        {
```

```
            ClienteObj = obj;
```

```
            FechaPrometida = fechaPrometida;
```

```
            Lineas = new List<Linea>();
```

```
            Anulado = false;
```

```

    }

    public override double CalcularPrecio(double impuesto, int modificadorRecargo, int
modificadorRecargoExp, int modificadorMismoDia)
    {
        {
            double suma = 0;
            foreach (Linea linea in Lineas)
            {
                suma += linea.Precio;
            }

            if (FechaPrometida.Day == DateTime.Today.Day) suma += suma *
modificadorMismoDia / 100;

            else suma += suma * modificadorRecargoExp / 100;

            suma += suma * impuesto / 100;

            return suma;
        }
    }

    public override void IsValid(int fechaPrometida)
    {
        if (this.FechaPrometida.Day - DateTime.Today.Day > fechaPrometida) throw new
PedidoNoValidoException("Fecha no valida.");

        if (this.FechaPrometida < DateTime.Today) throw new
PedidoNoValidoException("Fecha no valida.");

        if (Lineas.Count == 0) throw new PedidoNoValidoException("Debe tener al menos
una linea.");
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\PedidoE
xpress.cs ===

```



```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\TipoMov  
imiento.cs ===
```

```
using Microsoft.EntityFrameworkCore;  
using Papeleria.LogicaNegocio.Excepciones;  
using Papeleria.LogicaNegocio.InterfacesEntidades;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

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

```
{  
    [Index(nameof(Nombre), IsUnique = true)]  
    public class TipoMovimiento : IValid  
    {  
        public int Id { get; set; }  
        public string Nombre { get; set; }  
        public TipoMovStock TipoMovStock { get; set; }  
  
        public void IsValid()  
        {  
            if (Nombre == null) throw new TMovException("Error en el nombre");  
        }  
    }  
}
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\TipoMov  
imiento.cs ===
```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\TipoMov  
Stock.cs ===
```

```

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

namespace Papeleria.LogicaNegocio.Entidades
{
    public enum TipoMovStock
    {
        REDUCE = 1,
        AGREGA = 2
    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\TipoMov
Stock.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Usuario.
cs ===

```

```

using LogicaNegocio.ValueObjects;
using Microsoft.EntityFrameworkCore;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesEntidades;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

```

namespace LogicaNegocio.Entidades

{

[Index(nameof(Email), IsUnique = true)]

public abstract class Usuario : IValid

{

public int Id { get; set; }

[Required(ErrorMessage = "Requerido.")]

[EmailAddress]

public string Email { get; set; }

[Required(ErrorMessage = "Requerido.")]

[RegularExpression(@"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[.,!])[a-zA-Z\d.,!]{6,}\$",

ErrorMessage = "La contraseña debe tener al menos una letra en minúscula, mayúscula, un número, y uno de estos símbolos: . ; , ! ")

[StringLength(200, MinimumLength = 6, ErrorMessage = "La Contraseña debe ser de al menos 6 caracteres")]

public string Password { get; set; }

[Required(ErrorMessage = "Requerido.")]

[RegularExpression(@"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[.,!])[a-zA-Z\d.,!]{6,}\$",

ErrorMessage = "La contraseña debe tener al menos una letra en minúscula, mayúscula, un número, y uno de estos símbolos: . ; , ! ")

[StringLength(200, MinimumLength = 6, ErrorMessage = "La Contraseña debe ser de al menos 6 caracteres")]

public string PasswordSinEncrypt { get; set; }

public static int UltimoId { get; set; }

[Required(ErrorMessage = "Requerido.")]

[RegularExpression(@"^[a-zA-Z]([-a-zA-Z ']\*(<![ '])\$)", ErrorMessage = "El nombre o apellido es inválido.")]

public NombreCompleto NombreCompleto { get; set; }

public void IsValid()

{

if (this != null)

```

        {
            if (Email == null) throw new AdministradorNoValidoException("Email requerido.");

            if (Password == null) throw new AdministradorNoValidoException("Password
requerido.");

            if (NombreCompleto.Nombre == null) throw new
AdministradorNoValidoException("Nombre requerido.");

            if (NombreCompleto.Apellido == null) throw new
AdministradorNoValidoException("Apellido requerido.");

            if (!Regex.IsMatch(Password, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[.,;!])[a-zA-
Z\d.,;!]{6,}$"))
            {
                throw new AdministradorNoValidoException("La contraseña debe tener al
menos una letra en minúscula, mayúscula, un número, y uno de estos símbolos: . ; , ! ");
            }

            NombreCompleto.IsValid();
        }

        else throw new AdministradorNoValidoException("Admin no válido");
    }
}
}

```

=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Entidades\Usuario.  
cs ===

=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Admi  
nistradorNoValidoException.cs ===

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

```

namespace Papeleria.LogicaNegocio.Excepciones

```

{
    public class AdministradorNoValidoException : Exception
    {
        public AdministradorNoValidoException()
        {
        }

        public AdministradorNoValidoException(string? message) : base(message)
        {
        }

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

        protected AdministradorNoValidoException(SerializationInfo info, StreamingContext
        context) : base(info, context)
        {
        }
    }
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
 shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Admi  
 nistradorNoValidoException.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
 shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Articu  
 loNoValidoException.cs ===

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

```

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

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    public class ArticuloNoValidoException : Exception
```

```
    {
```

```
        public ArticuloNoValidoException()
```

```
        {
```

```
        }
```

```
        public ArticuloNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public ArticuloNoValidoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected ArticuloNoValidoException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Articu  
loNoValidoException.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Client  
eNoValidoException.cs ===
```

```
using System;
```

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

```

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

namespace Papeleria.LogicaNegocio.Excepciones
{
    public class ClienteNoValidoException : Exception
    {
        public ClienteNoValidoException()
        {
        }

        public ClienteNoValidoException(string? message) : base(message)
        {
        }

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

        protected ClienteNoValidoException(SerializationInfo info, StreamingContext context) :
        base(info, context)
        {
        }
    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Client
eNoValidoException.cs ===

```

```
=== Start of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Direc  
cionNoValidaException.cs ===
```

```
using System.Runtime.Serialization;
```

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    [Serializable]
```

```
    internal class DireccionNoValidaException : Exception
```

```
    {
```

```
        public DireccionNoValidaException()
```

```
        {
```

```
        }
```

```
        public DireccionNoValidaException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public DireccionNoValidaException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected DireccionNoValidaException(SerializationInfo info, StreamingContext  
context) : base(info, context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\Desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Direc  
cionNoValidaException.cs ===
```



=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Linea  
NoValidaException.cs ===

```
using System;
```

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

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    public class LineaNoValidaException : Exception
```

```
    {
```

```
        public LineaNoValidaException()
```

```
        {
```

```
        }
```

```
        public LineaNoValidaException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public LineaNoValidaException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected LineaNoValidaException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Linea  
NoValidaException.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\MovE  
xception.cs ===
```

```
using System.Runtime.Serialization;
```

```
namespace Papeleria.AccesoDatos.EntityFramework.Repositorios
```

```
{
```

```
    [Serializable]
```

```
    public class MovException : Exception
```

```
    {
```

```
        public MovException()
```

```
        {
```

```
        }
```

```
        public MovException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public MovException(string? message, Exception? innerException) : base(message,  
innerException)
```

```
        {
```

```
        }
```

```
        protected MovException(SerializationInfo info, StreamingContext context) : base(info,  
context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\MovE  
xception.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Movi  
mientoException.cs ===
```

```
using System.Runtime.Serialization;
```

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    [Serializable]
```

```
    internal class MovimientoException : Exception
```

```
    {
```

```
        public MovimientoException()
```

```
        {
```

```
        }
```

```
        public MovimientoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public MovimientoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected MovimientoException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Movi  
mientoException.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Nomb  
reClienteNoValidoException.cs ===
```

```
using System.Runtime.Serialization;
```

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    [Serializable]
```

```
    internal class NombreClienteNoValidoException : Exception
```

```
    {
```

```
        public NombreClienteNoValidoException()
```

```
        {
```

```
        }
```

```
        public NombreClienteNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public NombreClienteNoValidoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected NombreClienteNoValidoException(SerializationInfo info, StreamingContext  
context) : base(info, context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Nomb  
reClienteNoValidoException.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Nomb  
reCompletoNoValidoException.cs ===
```

```
using System.Runtime.Serialization;
```

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    [Serializable]
```

```
    internal class NombreCompletoNoValidoException : Exception
```

```
    {
```

```
        public NombreCompletoNoValidoException()
```

```
        {
```

```
        }
```

```
        public NombreCompletoNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public NombreCompletoNoValidoException(string? message, Exception?  
innerException) : base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected NombreCompletoNoValidoException(SerializationInfo info, StreamingContext  
context) : base(info, context)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Nomb  
reCompletoNoValidoException.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\Pedid  
oNoValidoException.cs ===
```

```
using System;
```

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

```
using System.Linq;
```

```
using System.Runtime.Serialization;
```

```
using System.Text;
```

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

```
namespace Papeleria.LogicaNegocio.Excepciones
```

```
{
```

```
    public class PedidoNoValidoException : Exception
```

```
    {
```

```
        public PedidoNoValidoException()
```

```
        {
```

```
        }
```

```
        public PedidoNoValidoException(string? message) : base(message)
```

```
        {
```

```
        }
```

```
        public PedidoNoValidoException(string? message, Exception? innerException) :  
base(message, innerException)
```

```
        {
```

```
        }
```

```
        protected PedidoNoValidoException(SerializationInfo info, StreamingContext context) :  
base(info, context)
```

```
        {
```

```

    }
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\PedidoNoValidoException.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\TMovException.cs ===

```

```

using System.Runtime.Serialization;

namespace Papeleria.LogicaNegocio.Excepciones
{
    [Serializable]
    public class TMovException : Exception
    {
        public TMovException()
        {
        }

        public TMovException(string? message) : base(message)
        {
        }

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

        protected TMovException(SerializationInfo info, StreamingContext context) : base(info, context)
        {
        }
    }
}

```

```

    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\Excepciones\TMov
Exception.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesEntidade
s\IValid.cs ===

```

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

namespace Papeleria.LogicaNegocio.InterfacesEntidades

```

```

{
    public interface IValid
    {
        public void IsValid();
    }
}

```

```

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesEntidade
s\IValid.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor
io\IRepositorio.cs ===

```

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```



```
namespace LogicaNegocio.InterfacesRepositorio
```

```
{  
    public interface IRepositoryo<T> where T : class  
    {  
        IEnumerable<T> FindAll();  
        T FindByID(int id);  
        bool Add(T aAgregar);  
        bool Remove(int id);  
        bool Update(T aModificar);  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositorio.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositorioArticulo.cs ===
```

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

```
namespace LogicaNegocio.InterfacesRepositorio
```

```
{  
    public interface IRepositoryioArticulo : IRepositoryo<Articulo>  
    {  
        IEnumerable<Articulo> GetMovidosByFecha(DateTime startdate, DateTime enddate,  
int pag, int size);  
    }  
}
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioArticulo.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioCliente.cs ===
```

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

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

```
{  
    public interface IRepositorioCliente : IRepositorio<Cliente>  
    {  
        public IEnumerable<Cliente> ClientesXnombreYapellido(string especifica);  
        public IEnumerable<Cliente> ClientesXmonto(double monto);  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioCliente.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioConfig.cs ===
```

```
using LogicaNegocio.Entidades;  
  
using LogicaNegocio.InterfacesRepositorio;  
  
using System;
```

```

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

namespace Papeleria.LogicaNegocio.InterfacesRepositorio
{
    public interface IRepositoryConfig : IRepository<Configuracion>
    {
        Configuracion FindByNombre(string nombre);
    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositoryConfig.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositoryMovimiento.cs ===

```

```

using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Papeleria.LogicaNegocio.InterfacesRepositorio
{
    public interface IRepositoryMovimiento : IRepository<Movimiento>
    {
        IEnumerable<Movimiento> GetMovs(int pag, int size);
        IEnumerable<Movimiento> GetByArtyTipo(int articuloId, int tipoMovId, int pag, int size);
    }
}

```

```
}  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioMovimiento.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioPedido.cs ===
```

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

```
namespace LogicaNegocio.InterfacesRepositorio  
{  
    public interface IRepositorioPedido : IRepositorio<Pedido>  
    {  
        public double CalcularPrecio(Pedido pedido);  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioPedido.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositor  
io\IRepositorioTipoMovimiento.cs ===
```

```
using LogicaNegocio.InterfacesRepositorio;  
using Papeleria.LogicaNegocio.Entidades;  
using System;  
using System.Collections.Generic;
```

```

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

namespace Papeleria.LogicaNegocio.InterfacesRepositorio
{
    public interface IRepositoryTipoMovimiento : IRepository<TipoMovimiento>
    {
    }
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositoryTipoMovimiento.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositoryUsuario.cs ===

```

```

using LogicaNegocio.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorio
{
    public interface IRepositoryUsuario : IRepository<Usuario>
    {
        Usuario FindByEmail(string email);
    }
}

```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\InterfacesRepositorio\IRepositorioUsuario.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\obj\Debug\net8.0\  
.NETCoreApp,Version=v8.0.AssemblyAttributes.cs ===
```

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v8.0",  
FrameworkDisplayName = ".NET 8.0")]
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\obj\Debug\net8.0\  
.NETCoreApp,Version=v8.0.AssemblyAttributes.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\obj\Debug\net8.0\  
Papeleria.LogicaNegocio.AssemblyInfo.cs ===
```

```
//-----
```

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

```
// Este código fue generado por una herramienta.
```

```
// Versión de runtime:4.0.30319.42000
```

```
//
```

```
// Los cambios en este archivo podrían causar un comportamiento incorrecto y se  
perderán si
```

```
// se vuelve a generar el código.
```

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

```
//-----
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly: System.Reflection.AssemblyCompanyAttribute("Papeleria.LogicaNegocio")]
```

```
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly:
System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+d70b4dd48c049d4f3ba3
b7a575c3e880ae3b8b47")]
[assembly: System.Reflection.AssemblyProductAttribute("Papeleria.LogicaNegocio")]
[assembly: System.Reflection.AssemblyTitleAttribute("Papeleria.LogicaNegocio")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]
```

```
// Generado por la clase WriteCodeFragment de MSBuild.
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\obj\Debug\net8.0\
Papeleria.LogicaNegocio.AssemblyInfo.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\obj\Debug\net8.0\
Papeleria.LogicaNegocio.GlobalUsings.g.cs ===
```

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

```
global using global::System;
global using global::System.Collections.Generic;
global using global::System.IO;
global using global::System.Linq;
global using global::System.Net.Http;
global using global::System.Threading;
global using global::System.Threading.Tasks;
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\obj\Debug\net8.0\
Papeleria.LogicaNegocio.GlobalUsings.g.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\ValueObjects\Direc
cion.cs ===
```

```
using Microsoft.EntityFrameworkCore;
```

```

using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesEntidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.ValueObjects
{
    [Owned]
    public class Direccion : IValid
    {
        public string NombreCalle { get; private set; }
        public string NumeroPuerta { get; private set; }
        public string Ciudad { get; private set; }
        public double DistanciaKm { get; private set; }

        public Direccion(string nombreCalle, string numeroPuerta, string ciudad, double
distanciaKm)
        {
            NombreCalle = nombreCalle;
            NumeroPuerta = numeroPuerta;
            Ciudad = ciudad;
            DistanciaKm = distanciaKm;
        }

        public void IsValid()
        {
            if (DistanciaKm < 0) throw new DireccionNoValidaException("Distancia negativa.");
            if (!int.TryParse(NumeroPuerta, out int x)) throw new
DireccionNoValidaException("Numero de puerta invalido");
        }
    }
}

```



```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\ValueObjects\Direc  
cion.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\ValueObjects\Nom  
breCliente.cs ===
```

```
using Microsoft.EntityFrameworkCore;  
using Papeleria.LogicaNegocio.Excepciones;  
using Papeleria.LogicaNegocio.InterfacesEntidades;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Text.RegularExpressions;  
using System.Threading.Tasks;
```

```
namespace Papeleria.LogicaNegocio.ValueObjects
```

```
{
```

```
    [Owned]
```

```
    public class NombreCliente : IValid
```

```
    {
```

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

```
        public string Apellido { get; private set; }
```

```
        public NombreCliente() { }
```

```
        public NombreCliente(string nombre, string apellido)
```

```
        {
```

```
            this.Nombre = nombre;
```

```
            this.Apellido = apellido;
```

```
        }
```

```
        public void IsValid()
```

```
        {
```

```

        if (!Regex.IsMatch(Nombre, @"^[a-zA-Z] [-a-zA-Z ']* (?<![ '])$")) throw new
NombreClienteNoValidoException("Nombre no es valido");

        if (!Regex.IsMatch(Apellido, @"^[a-zA-Z] [-a-zA-Z ']* (?<![ '])$")) throw new
NombreClienteNoValidoException("Apellido no es valido");

    }

}

}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\ValueObjects\Nom
breCliente.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\ValueObjects\Nom
breCompleto.cs ===

```

```

using Microsoft.EntityFrameworkCore;
using Papeleria.LogicaNegocio.Excepciones;
using Papeleria.LogicaNegocio.InterfacesEntidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

```

```

namespace LogicaNegocio.ValueObjects
{
    [Owned]
    public class NombreCompleto : IValid
    {
        public string Nombre { get; private set; }
        public string Apellido { get; private set; }
        public NombreCompleto()
        {
            Nombre = "Sin Nombre";

```

```

        Apellido = "Sin Apellido";
    }

    public NombreCompleto(string nombre, string apellido) {
        this.Nombre = nombre;
        this.Apellido = apellido;
    }

    public void IsValid()
    {
        if (!Regex.IsMatch(Nombre, @"^[a-zA-Z][-a-zA-Z ']*(?<![ '])$")) throw new
NombreCompletoNoValidoException("Nombre no es valido");

        if (!Regex.IsMatch(Apellido, @"^[a-zA-Z][-a-zA-Z ']*(?<![ '])$")) throw new
NombreCompletoNoValidoException("Apellido no es valido");
    }
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\LogicaNegocio\ValueObjects\Nom
breCompleto.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Program.cs ===

using Papeleria.AccesoDatos.EntityFramework.Repositorios;
using Papeleria.LogicaAplicacion.CasosDeUso.Administradores;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using LogicaNegocio.InterfacesRepositorio;
using AccesoDatos.EntityFramework;
using Papeleria.LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;
using Papeleria.LogicaAplicacion.CasosDeUso.Clientes;
using Papeleria.LogicaAplicacion.CasosDeUso.Articulos;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;
using AccesoDatos.EntityFramework.Repositorios;

```

```

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;
using Papeleria.LogicaAplicacion.CasosDeUso.Pedidos;

namespace Papeleria.Web
{
    public class Program
    {
        public static void Main(string[] args)
        {
            var builder = WebApplication.CreateBuilder(args);

            // Add services to the container.
            builder.Services.AddControllersWithViews();
            builder.Services.AddDistributedMemoryCache();

            builder.Services.AddScoped<IRepositorioUsuario, RepositorioUsuarioEF>();
            builder.Services.AddScoped<IRepositorioConfig, RepositorioConfiguracionEF>();
            builder.Services.AddScoped<IRepositorioCliente, RepositorioClienteEF>();
            builder.Services.AddScoped<IRepositorioArticulo, RepositorioArticuloEF>();
            builder.Services.AddScoped<IRepositorioPedido, RepositorioPedidoEF>();

            builder.Services.AddScoped<ILogin, LoginCU>();
            builder.Services.AddScoped<IEncontrarArticulosOrd, EncontrarArticulosOrdCU>();
            builder.Services.AddScoped<IGetPedidosDesc, GetPedidosDescCU>();
            builder.Services.AddScoped<ICrearAdmin, CrearAdminCU>();
            builder.Services.AddScoped<IEditarAdmin, EditarAdminCU>();
            builder.Services.AddScoped<IFindAdminById, FindAdminByIdCU>();
            builder.Services.AddScoped<IEncontrarAdmins, EncontrarAdminsCU>();
            builder.Services.AddScoped<IBorrarAdmin, BorrarAdminCU>();
            builder.Services.AddScoped<IEncontrarClientes, EncontrarClientesCU>();
            builder.Services.AddScoped<IGetClientesXnombreYapellido,
GetClientesXnombreYapellidoCU>();

```

```
builder.Services.AddScoped<IGetClientesXmonto, GetClientesXmontoCU>();
builder.Services.AddScoped<ICrearArticulo, CrearArticuloCU>();
builder.Services.AddScoped<IEncontrarArticulos, EncontrarArticulosCU>();
builder.Services.AddScoped<IEncontrarPedidos, EncontrarPedidosCU>();
builder.Services.AddScoped<ICrearPedido, CrearPedidoCU>();
builder.Services.AddScoped<IEncontrarPrecioPedido, EncontrarPrecioPedidoCU>();
builder.Services.AddScoped<IEncontrarXIdArticulo, FindArticuloByIdCU>();
builder.Services.AddScoped<IAnularPedido, AnularPedidoCU>();
builder.Services.AddScoped<IGetPedidosPorFecha, GetPedidosPorFechaCU>();
```

```
builder.Services.AddSession(option =>
{
    option.IdleTimeout = TimeSpan.FromSeconds(5000);
    option.Cookie.HttpOnly = true;
    option.Cookie.IsEssential = true;
});
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production
    scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}
```

```
app.UseHttpsRedirection();
app.UseStaticFiles();
```

```

        app.UseRouting();

        app.UseAuthorization();

        app.UseSession();

        app.MapControllerRoute(
            name: "default",
            pattern: "{controller=Login}/{action=Index}/{id?}");

        app.Run();
    }
}
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Program.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\AdminC
ontroller.cs ===

```

```

using Microsoft.AspNetCore.Mvc;

using Papeleria.LogicaAplicacion.DTOs;

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;

namespace Papeleria.Web.Controllers
{
    public class AdminController : Controller
    {
        private ICrearAdmin _crearAdmin;
        private IEditarAdmin _editarAdmin;
        private IEncontrarAdmins _encontrarAdmins;
        private IFindAdminById _findAdminById;
        private IBorrarAdmin _borrarAdmin;

        public AdminController(ICrearAdmin crearAdmin, IEditarAdmin editarAdmin,

```

```

        IEncontrarAdmins encontrarAdmins, IFindAdminById findAdminById, IBorrarAdmin
borrarAdmin)
    {
        this._encontrarAdmins = encontrarAdmins;
        this._crearAdmin = crearAdmin;
        this._editarAdmin = editarAdmin;
        this._findAdminById = findAdminById;
        this._borrarAdmin = borrarAdmin;
    }

// GET: AdminController
public ActionResult Index()
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
    }
    return View(_encontrarAdmins.FindAllAdmins());
}

// GET: AdminController/Create
public ActionResult Create(string mensaje)
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
    }
    ViewBag.mensaje = mensaje;
    return View();
}

// POST: AdminController/Create
[HttpPost]

```

```

[ValidateAntiForgeryToken]
public ActionResult Create(UsuarioDTO adminDto)
{
    try
    {
        this._crearAdmin.CrearAdmin(adminDto);
        return RedirectToAction(nameof(Index));
    }
    catch(Exception ex)
    {
        return RedirectToAction("Create", new {mensaje = ex.Message});
    }
}

// GET: AdminController/Edit/5
public ActionResult Edit(int id)
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
    });
    }

    UsuarioDTO admin = this._findAdminById.FindAdminById(id);
    return View(admin);
}

// POST: AdminController/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(UsuarioDTO adminDto)
{
    this._editarAdmin.EditarAdmin(adminDto);
    return View(adminDto);
}

```



```

    }

    // GET: AdminController/Delete/5
    public ActionResult Delete(int id)
    {
        if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
        {
            return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
        });
        }
        UsuarioDTO admin = this._findAdminById.FindAdminById(id);
        return View(admin);
    }

```

```

    // POST: AdminController/Delete/5
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Delete(UsuarioDTO admin, int id)
    {
        this._borrarAdmin.BorrarAdmin(id);
        return RedirectToAction("Index");
    }
}

```

=== End of C:\Users\Usuario\Desktop\ORT & Other  
sh\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\AdminC  
ontroller.cs ===

=== Start of C:\Users\Usuario\Desktop\ORT & Other  
sh\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\Articulo  
sController.cs ===

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

```

```

using Papeleria.LogicaAplicacion.DTOs;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;

namespace Papeleria.Web.Controllers
{
    public class ArticulosController : Controller
    {

        private IEncontrarArticulos _encontrarArticulos;
        private ICrearArticulo _crearArticulo;

        public ArticulosController(IEncontrarArticulos encontrarArticulos, ICrearArticulo
crearArticulo)
        {
            _encontrarArticulos = encontrarArticulos;
            _crearArticulo = crearArticulo;
        }

        // GET: ArticulosController
        public ActionResult Index()
        {
            if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
            {
                return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
            }

            return View(this._encontrarArticulos.EncontrarArticulos());
        }

        // GET: ArticulosController/Create
        public ActionResult Create()
        {
            if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
            {

```

```

        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
    });
    }
    return View();
}

```

```

// POST: ArticulosController/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(ArticuloDTO articulo)
{
    try
    {
        this._crearArticulo.CrearArticulo(articulo);
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
sh\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\ArticulosController.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
sh\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\ClientesController.cs ===

```

using Microsoft.AspNetCore.Mvc;

using Papeleria.LogicaAplicacion.DTOs;

using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;

```

```

namespace Papeleria.Web.Controllers
{
    public class ClientesController : Controller
    {

        private IEncontrarClientes _encontrarClientes;
        private IGetClientesXnombreYapellido _xnombreYapellido;
        private IGetClientesXmonto _xMonto;

        public ClientesController(IEncontrarClientes encontrarClientes,
            IGetClientesXnombreYapellido xnombreYapellido, IGetClientesXmonto xMonto)
        {
            _encontrarClientes = encontrarClientes;
            _xnombreYapellido = xnombreYapellido;
            _xMonto = xMonto;
        }

        // GET: ClientesController
        public ActionResult Index(string mensaje, string filtro)
        {
            if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
            {
                return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
            });
            }

            IEnumerable<ClienteDTO> toShow = new List<ClienteDTO>();
            ViewBag.Mensaje = mensaje;
            ViewBag.Clientes = this._encontrarClientes.FindAllClientes();

            if (string.IsNullOrEmpty(filtro))

```

```

{
    toShow = _encontrarClientes.FindAllClientes();
}
if (filtro == "PorNombre")
{
    string nombreCliente = (string)TempData["NombreCliente"];
    toShow = this._xnombreYapellido.GetClientesXnombreYapellido(nombreCliente);
}
if(filtro == "PorMonto")
{
    double monto;
    if(double.TryParse(TempData["Monto"].ToString(), out monto)) toShow =
this._xMonto.GetClientesXmonto(monto);
}
return View(toShow);
}

```

```

[HttpPost]
public ActionResult FiltrarPorNombre (string nombreCliente)
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
    }
    if (nombreCliente == null)
    {
        return RedirectToAction("Index");
    }
    TempData["NombreCliente"] = nombreCliente;
    return RedirectToAction("Index", new { filtro = "PorNombre" });
}

```

```

[HttpPost]
public ActionResult FiltrarPorMonto (double monto)
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
    });
    }
    if (monto <= 0)
    {
        return RedirectToAction("Index", new { mensaje = "Monto invalido." });
    }
    TempData["Monto"] = monto.ToString();
    return RedirectToAction("Index", new { filtro = "PorMonto" });
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\ClientesController.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\HomeController.cs ===

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace Papeleria.Web.Controllers
{
    public class HomeController : Controller
    {
        // GET: HomeController

```

```

public ActionResult Index(string mensaje)
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "No tienes acceso" });
    }
    else
    {
        ViewBag.Mensaje = mensaje;
        if (HttpContext.Session.GetString("rol") == "Miembro")
        {
            ViewBag.usuario = (HttpContext.Session.GetString("usuario"));
        }
        return View();
    }
}
}
}

=== End of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\HomeC
ontroller.cs ===

```

```

=== Start of C:\Users\Usuario\Desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers>LoginC
ontroller.cs ===

```

```

using LogicaNegocio.InterfacesRepositorio;
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Administrador;
using Microsoft.AspNetCore.Mvc;

```

```

namespace Papeleria.Web.Controllers
{
    public class LoginController : Controller
    {

```

```

private IRepositoryUsuario _repositorioAdmins;
private ILogin _loginCU;
public LoginController(IRepositoryUsuario repositorioAdmins,
    ILogin login)
{
    this._repositorioAdmins = repositorioAdmins;
    this._loginCU = login;
}
// GET: LoginController
public ActionResult Index(string mensaje)
{
    ViewBag.mensaje = mensaje;
    return View();
}

public ActionResult Login(string email, string password)
{
    if (this._loginCU.Login(email, password))
    {
        HttpContext.Session.SetString("usuario", email);
        return RedirectToAction("Index", "Home");
    }

    return RedirectToAction("Index", new { mensaje = "Nombre de usuario o contraseña incorrecta." });

    //manda esto para el CULogin, que primero encripta la contraseña, la desencripta
    en la capa de datos,

    //la verifica con los usuarios de la BD, y devuelve true o false.
}

public IActionResult Logout()
{
    HttpContext.Session.SetString("usuario", "");
    return RedirectToAction("Index");
}

```



```
}  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\LoginC  
ontroller.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\Pedido  
sController.cs ===
```

```
using Microsoft.AspNetCore.Mvc;  
using Microsoft.CodeAnalysis.Elfi.Serialization;  
using Papeleria.LogicaAplicacion.DTOs;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Articulo;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Cliente;  
using Papeleria.LogicaAplicacion.InterfacesCasosDeUso.Pedido;
```

```
namespace Papeleria.Web.Controllers
```

```
{  
    public class PedidosController : Controller  
    {  
        private ICrearPedido _crearPedido;  
        private IEncontrarPedidos _encontrarPedidos;  
        private IEncontrarClientes _encontrarClientes;  
        private IEncontrarArticulos _encontrarArticulos;  
        private IEncontrarPrecioPedido _encontrarPrecioPedido;  
        private IEncontrarXIdArticulo _encontrarXIdArticulo;  
        private IAnularPedido _anularPedido;  
        private IGetPedidosPorFecha _getPedidosPorFecha;  
        private static PedidoDTO tempPedido;  
        private static List<ArticuloDTO> tempArticulos;  
  
        public PedidosController(ICrearPedido crearPedido,  
            IEncontrarArticulos encontrarArticulos, IEncontrarPedidos encontrarPedidos,
```

```
        IEncontrarClientes encontrarClientes, IEncontrarPrecioPedido  
        encontrarPrecioPedido,
```

```
        IEncontrarXIdArticulo encontrarXIdArticulo, IAnularPedido anularPedido,  
        IGetPedidosPorFecha getPedidosPorFecha)
```

```
{  
    _crearPedido = crearPedido;  
    _encontrarArticulos = encontrarArticulos;  
    _encontrarPedidos = encontrarPedidos;  
    _encontrarClientes = encontrarClientes;  
    _encontrarPrecioPedido = encontrarPrecioPedido;  
    _encontrarXIdArticulo = encontrarXIdArticulo;  
    _anularPedido = anularPedido;  
    _getPedidosPorFecha = getPedidosPorFecha;  
}
```

```
// GET: PedidosController
```

```
public ActionResult Index(string filtro, string mensaje)
```

```
{  
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))  
    {  
        return RedirectToAction("Index", "Login", new { mensaje = "Necesita hacer login" });  
    }  
    IEnumerable<PedidoDTO> aMostrar = new List<PedidoDTO>();  
    if (string.IsNullOrEmpty(filtro))  
    {  
        aMostrar = this._encontrarPedidos.EncontrarPedidos();  
        ViewBag.Filtro = "Ninguno";  
    }  
    if (filtro == "PorFecha")  
    {  
        DateTime FechaPrometida =  
DateTime.Parse(TempData["FechaPrometida"].ToString());  
        aMostrar = this._getPedidosPorFecha.GetPedidosPorFecha(FechaPrometida);  
        if(aMostrar.Count() == 0)
```

```

        {
            return RedirectToAction("Index", new { filtro = "", mensaje = "No hay pedidos sin
entregar con esa fecha." });
        }
        ViewBag.Filtro = "Fecha";
    }
    ViewBag.Mensaje = mensaje;
    ViewBag.Clientes = this._encontrarClientes.FindAllClientes();
    ViewBag.Articulos = this._encontrarArticulos.EncontrarArticulos();
    return View(aMostrar);
}

```

// GET: PedidosController/Create

```

public ActionResult Create(Boolean esExpress)
{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
    }
    ViewBag.Clientes = this._encontrarClientes.FindAllClientes();
    ViewBag.Articulos = this._encontrarArticulos.EncontrarArticulos();
    ViewBag.PrecioPedido = 0;
    if (tempPedido != null)
    {
        ViewBag.Lineas = tempPedido.Lineas;
        ViewBag.PrecioPedido =
this._encontrarPrecioPedido.EncontrarPrecioPedido(tempPedido, esExpress);
    }
    return View();
}

```

// POST: PedidosController/Create

[HttpPost]

[ValidateAntiForgeryToken]

```
public ActionResult Create(PedidoDTO pedido, Boolean esExpress)
{
    try
    {
        if(tempPedido != null && tempPedido.Lineas.Count > 0)
        {
            pedido.Lineas = tempPedido.Lineas;
        }
        this._crearPedido.CrearPedido(pedido, esExpress);
        tempPedido = null;
        tempArticulos = null;
        return RedirectToAction(nameof(Index));
    }
    catch (Exception ex)
    {
        return RedirectToAction("Index", "Pedidos", new {mensaje = ex.Message});
    }
}
```

[HttpPost]

[ValidateAntiForgeryToken]

```
public ActionResult AddLinea(int articuloId, int cantUnidades, int idCliente)
{
    ArticuloDTO articulo = _encontrarXIdArticulo.FindById(articuloId);
    if (tempArticulos == null)
    {
        tempArticulos = new List<ArticuloDTO>();
        if (articulo.Stock >= cantUnidades && cantUnidades > 0)
        {
            articulo.Stock -= cantUnidades;
            tempArticulos.Add(articulo);
        }
    }
}
```

```

        LineaDTO linea = new LineaDTO { Articulold = articulold, CantUnidades =
cantUnidades, Precio = articulo.Precio * cantUnidades };

        if (tempPedido == null)
        {
            tempPedido = new PedidoDTO { Lineas = new List<LineaDTO>() };
        }

        tempPedido.Clienteld = idCliente;
        tempPedido.Lineas.Add(linea);
        return this.RedirectToAction(nameof(Create));
    }
    else return this.RedirectToAction(nameof(Create));
}

else {
    foreach (ArticuloDTO unArticulo in tempArticulos)
    {
        if (unArticulo.Id == articulold)
        {
            if (unArticulo.Stock >= cantUnidades && cantUnidades > 0)
            {
                unArticulo.Stock = unArticulo.Stock - cantUnidades;

                LineaDTO linea = new LineaDTO { Articulold = articulold, CantUnidades =
cantUnidades, Precio = articulo.Precio * cantUnidades };

                if (tempPedido == null)
                {
                    tempPedido = new PedidoDTO { Lineas = new List<LineaDTO>() };
                }

                tempPedido.Lineas.Add(linea);
                return this.RedirectToAction(nameof(Create));
            }
            else return this.RedirectToAction(nameof(Create));
        }
    }
}

if (articulo.Stock >= cantUnidades && cantUnidades > 0)

```

```

    {
        tempArticulos.Add(articulo);

        LineaDTO linea = new LineaDTO { ArticuloId = articuloId, CantUnidades =
cantUnidades, Precio = articulo.Precio * cantUnidades };

        if (tempPedido == null)
        {
            tempPedido = new PedidoDTO { Lineas = new List<LineaDTO>() };
        }

        tempPedido.Lineas.Add(linea);

        return this.RedirectToAction(nameof(Create));
    }
    else return this.RedirectToAction(nameof(Create));
}
}

```

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult AnularPedido(int idPedido)

```

{
    if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
    {
        return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
    }

    try
    {
        this._anularPedido.AnularPedido(idPedido);

        return this.RedirectToAction(nameof(Index));
    }

    catch (Exception ex)
    {
        return this.RedirectToAction("Index", "Pedidos", new {mensaje = "Error al anular" });
    }
}

```

```

    }
    [HttpPost]
    public ActionResult FiltrarPorFecha(DateTime FechaPrometida)
    {
        if (string.IsNullOrEmpty(HttpContext.Session.GetString("usuario")))
        {
            return RedirectToAction("Index", "Login", new { mensaje = "Por favor inicie sesion"
});
        }
        if (FechaPrometida < DateTime.Today)
        {
            return RedirectToAction("Index", new { mensaje = "Fecha invalida." });
        }
        TempData["FechaPrometida"] = FechaPrometida.ToString();
        return RedirectToAction("Index", new { filtro = "PorFecha" });
    }
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Controllers\PedidosController.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Models\ErrorViewModel.cs ===

```

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

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

```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\Models\ErrorView  
Model.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\  
.NETCoreApp,Version=v8.0.AssemblyAttributes.cs ===
```

```
// <autogenerated />
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:  
global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v8.0",  
FrameworkDisplayName = ".NET 8.0")]
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\  
.NETCoreApp,Version=v8.0.AssemblyAttributes.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\  
Papeleria.Web.AssemblyInfo.cs ===
```

```
//-----
```

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

```
// Este código fue generado por una herramienta.
```

```
// Versión de runtime:4.0.30319.42000
```

```
//
```

```
// Los cambios en este archivo podrían causar un comportamiento incorrecto y se  
perderán si
```

```
// se vuelve a generar el código.
```

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

```
//-----
```

```
using System;
```

```
using System.Reflection;
```



```
[assembly: System.Reflection.AssemblyCompanyAttribute("Papeleria.Web")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly:
System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+d70b4dd48c049d4f3ba3
b7a575c3e880ae3b8b47")]
[assembly: System.Reflection.AssemblyProductAttribute("Papeleria.Web")]
[assembly: System.Reflection.AssemblyTitleAttribute("Papeleria.Web")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]
```

```
// Generado por la clase WriteCodeFragment de MSBuild.
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\
Papeleria.Web.AssemblyInfo.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\
Papeleria.Web.GlobalUsings.g.cs ===
```

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

```
global using global::Microsoft.AspNetCore.Builder;
global using global::Microsoft.AspNetCore.Hosting;
global using global::Microsoft.AspNetCore.Http;
global using global::Microsoft.AspNetCore.Routing;
global using global::Microsoft.Extensions.Configuration;
global using global::Microsoft.Extensions.DependencyInjection;
global using global::Microsoft.Extensions.Hosting;
global using global::Microsoft.Extensions.Logging;
global using global::System;
global using global::System.Collections.Generic;
global using global::System.IO;
global using global::System.Linq;
global using global::System.Net.Http;
global using global::System.Net.Http.Json;
```

```
global using global::System.Threading;
```

```
global using global::System.Threading.Tasks;
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\  
Papeleria.Web.GlobalUsings.g.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\  
Papeleria.Web.RazorAssemblyInfo.cs ===
```

```
//-----
```

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

```
// Este código fue generado por una herramienta.
```

```
// Versión de runtime:4.0.30319.42000
```

```
//
```

```
// Los cambios en este archivo podrían causar un comportamiento incorrecto y se  
perderán si
```

```
// se vuelve a generar el código.
```

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

```
//-----
```

```
using System;
```

```
using System.Reflection;
```

```
[assembly:
```

```
Microsoft.AspNetCore.Mvc.ApplicationParts.ProvideApplicationPartFactoryAttribute("Microso  
ft.AspNetCore.Mvc.ApplicationParts.ConsolidatedAssemblyApplicationPartFact" +
```

```
"ory, Microsoft.AspNetCore.Mvc.Razor")]
```

```
// Generado por la clase WriteCodeFragment de MSBuild.
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\ObligatorioMartinattoGonzalezM3C\Papeleria.Web\obj\Debug\net8.0\  
Papeleria.Web.RazorAssemblyInfo.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Program.cs ===
```

```
var builder = WebApplication.CreateBuilder(args);
```

```
// Add services to the container.
```

```
builder.Services.AddControllersWithViews();
```

```
builder.Services.AddSession(options =>
```

```
{
```

```
    options.IdleTimeout = TimeSpan.FromSeconds(300);
```

```
    options.Cookie.HttpOnly = true;
```

```
    options.Cookie.IsEssential = true;
```

```
});
```

```
var app = builder.Build();
```

```
// Configure the HTTP request pipeline.
```

```
if (!app.Environment.IsDevelopment())
```

```
{
```

```
    app.UseExceptionHandler("/Home/Error");
```

```
    // The default HSTS value is 30 days. You may want to change this for production  
    scenarios, see https://aka.ms/aspnetcore-hsts.
```

```
    app.UseHsts();
```

```
}
```

```
app.UseHttpsRedirection();
```

```
app.UseStaticFiles();
```

```
app.UseRouting();
```

```
app.UseAuthorization();
```

```
app.UseSession();
```

```
app.MapControllerRoute(
```

```
    name: "default",
```

```
pattern: "{controller=Login}/{action=Login}/{id?}");
```

```
app.Run();
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Program.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\ArticuloControll  
er.cs ===
```

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

```
using Newtonsoft.Json;
```

```
using System.Net.Http.Headers;
```

```
using WebDeposito.Models;
```

```
namespace WebDeposito.Controllers
```

```
{
```

```
    public class ArticuloController : Controller
```

```
    {
```

```
        private HttpClient _client;
```

```
        private string _baseUrl;
```

```
        private static int actualPage;
```

```
        public ArticuloController()
```

```
        {
```

```
            _client = new HttpClient();
```

```
            _baseUrl = "http://localhost:5091/api/Articulos/";
```

```
        }
```

```
        public IActionResult Index(DateTime startdate, DateTime enddate)
```

```
        {
```

```
            string token = HttpContext.Session.GetString("token");
```

```
            if (string.IsNullOrEmpty(token) || HttpContext.Session.GetString("rol") !=  
"Encargado")
```

```
            {
```

```
                return RedirectToAction("Login", "Login", new { mensaje = "No Autorizado." });
```

```

    }

    _client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue(
        "Bearer",
        token
    );

    if (actualPage < 1) { actualPage = 1; }

    HttpRequestMessage solicitud =
        new HttpRequestMessage(HttpMethod.Get, new Uri(_baseUrl + "GetByFechas/" +
            "Page/" + actualPage
            + "/startdate=" + startdate.ToString("yyyy-MM-dd") + "/enddate=" +
            enddate.ToString("yyyy-MM-dd")));

    Task<HttpResponseMessage> respuesta = _client.SendAsync(solicitud);
    respuesta.Wait();

    if (respuesta.Result.IsSuccessStatusCode)
    {
        if (respuesta.Result.StatusCode == System.Net.HttpStatusCode.NoContent)
        {
            ViewBag.mensaje = "No hay resultados.";
            actualPage = 1;
            return View();
        }

        var objetoComoTexto = respuesta.Result.Content.ReadAsStringAsync().Result;

        var articulos =
            JsonConvert.DeserializeObject<IEnumerable<ArticuloModel>>(objetoComoTexto);

        return View(articulos);
    }

    return View();
}

```

```

public IActionResult GetFechas(string mensaje)
{
    if (!string.IsNullOrEmpty(HttpContext.Session.GetString("token")) &&
        HttpContext.Session.GetString("rol") == "Encargado")

```

```

    {
        ViewBag.mensaje = mensaje;
        return View();
    }

    return RedirectToAction("Login", "Login", new { mensaje = "No autorizado" });
}

[HttpPost]
public IActionResult GetFechas(DateTime start, DateTime end)
{
    if (!string.IsNullOrEmpty(HttpContext.Session.GetString("token")) &&
        HttpContext.Session.GetString("rol") == "Encargado")
    {
        if (start > end || start == null || end == null) return RedirectToAction("Index", new
        { mensaje = "Fechas no validas." });

        return RedirectToAction("Index", new { startdate = start, enddate = end });
    }

    return RedirectToAction("Login", "Login", new { mensaje = "No autorizado" });
}

[HttpPost]
public ActionResult Next()
{
    try
    {
        string mensaje = "";
        actualPage++;
        if (actualPage < 1)
        {
            actualPage = 1;
            mensaje = "Solo números positivos.";
        }

        return RedirectToAction("Index", new { message = mensaje });
    }
}

```

```

    }
    catch (Exception ex)
    {
        return RedirectToAction("Index", "Movimiento", new { mensaje = "Hubo un error" });
    }

}

[HttpPost]
public ActionResult Previous()
{

    try
    {
        string message = "";
        actualPage--;
        if (actualPage < 1)
        {
            actualPage = 1;
            message = "Only postive numbers allowed";
        }

        return RedirectToAction("Index", new { message = message });
    }
    catch (Exception ex)
    {
        return RedirectToAction("Index", "Movimiento", new { mensaje = "Hubo un error" });
    }
}
}

```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\ArticuloControll  
er.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\LoginController.  
cs ===
```

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

```
using Newtonsoft.Json;
```

```
using NuGet.Common;
```

```
using System.Text;
```

```
using WebDeposito.Models;
```

```
namespace WebDeposito.Controllers
```

```
{
```

```
    public class LoginController : Controller
```

```
    {
```

```
        public IActionResult Login(string mensaje)
```

```
        {
```

```
            ViewBag.mensaje = mensaje;
```

```
            return View();
```

```
        }
```

```
        [HttpPost]
```

```
        public IActionResult Login(string email, string password)
```

```
        {
```

```
            HttpClient cliente = new HttpClient();
```

```
            HttpRequestMessage solicitud =
```

```
                new HttpRequestMessage(HttpMethod.Post, new  
Uri("http://localhost:5091/api/Login/login"));
```

```
            UserModel model = new UserModel();
```

```
            model.Email = email;
```

```
            model.PasswordSinEncript = password;
```



```

        var json = JsonConvert.SerializeObject(model);

        HttpContent contenido = new StringContent(json, Encoding.UTF8,
"application/json");

        solicitud.Content = contenido;

        Task<HttpResponseMessage> respuesta = cliente.SendAsync(solicitud);

        respuesta.Wait();

        if (respuesta.Result.IsSuccessStatusCode)
        {
            var objetoComoTexto = respuesta.Result.Content.ReadAsStringAsync().Result;
            var user = JsonConvert.DeserializeObject<TokenModel>(objetoComoTexto);
            HttpContext.Session.SetString("email", user.Usuario.Email);
            HttpContext.Session.SetString("token", user.Token);
            HttpContext.Session.SetString("rol", user.Rol);
            return RedirectToAction("Index", "Movimiento");
        }

        return RedirectToAction("Login", new { mensaje = "Username or password incorrect"
});
    }

```

```

public IActionResult Logout()
{
    HttpContext.Session.SetString("token", "");
    HttpContext.Session.SetString("user", "");
    return RedirectToAction("Login", "Login", new {mensaje = "Bye bye" });
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
 shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\LoginController.  
 cs ===

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\MovimientoCon  
troller.cs ===
```

```
using Microsoft.AspNetCore.Mvc;  
using Newtonsoft.Json;  
using System.Net.Http.Headers;  
using System.Text;  
using WebDeposito.Models;
```

```
namespace WebDeposito.Controllers
```

```
{  
    public class MovimientoController : Controller  
    {  
        private HttpClient _client;  
        private string _baseUrl;  
        private static int actualPage;  
        public MovimientoController()  
        {  
            _client = new HttpClient();  
            _baseUrl = "http://localhost:5091/api/Movimientos/";  
        }  
    }  
}
```

```
    public ActionResult Index(string mensaje, string filtro)  
    {  
  
        string token = HttpContext.Session.GetString("token");  
        if (string.IsNullOrEmpty(token) || HttpContext.Session.GetString("rol") !=  
"Encargado")  
        {  
            return RedirectToAction("Login", "Login", new { mensaje = "No Autorizado." });  
        }  
  
        _client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue(  
            "Bearer",
```

```

        token
    );
    if (actualPage < 1) { actualPage = 1; }
    if (string.IsNullOrEmpty(filtro))
    {
        HttpRequestMessage solicitud =
            new HttpRequestMessage(HttpMethod.Get, new Uri(_baseUrl + "page/" +
actualPage));
        Task<HttpResponseMessage> respuesta = _client.SendAsync(solicitud);
        respuesta.Wait();
        if (respuesta.Result.IsSuccessStatusCode)
        {
            if (respuesta.Result.StatusCode == System.Net.HttpStatusCode.NoContent)
            {
                ViewBag.mensaje = "No hay resultados.";
                actualPage = 1;
                return View();
            }
            var objetoComoTexto = respuesta.Result.Content.ReadAsStringAsync().Result;
            var movs =
                JsonConvert.DeserializeObject<IEnumerable<MovimientoModel>>(objetoComoTexto);
            return View(movs);
        }
    }
    if(filtro == "PorArtyTipo")
    {
        int idArticulo;
        int.TryParse(TempData["idArticulo"].ToString(), out idArticulo);
        int idTipo;
        int.TryParse(TempData["idTipo"].ToString(), out idTipo);
        HttpRequestMessage solicitud =
            new HttpRequestMessage(HttpMethod.Get, new Uri(_baseUrl + "GetByArtyTipo" +
"/Page/" + actualPage + "/Articulo="+idArticulo+"/Tipo="+idTipo));
    }

```

```

Task<HttpResponseMessage> respuesta = _client.SendAsync(solicitud);
respuesta.Wait();
if (respuesta.Result.IsSuccessStatusCode)
{
    if (respuesta.Result.StatusCode == System.Net.HttpStatusCode.NoContent)
    {
        ViewBag.mensaje = "No hay resultados.";
        actualPage = 1;
        return View();
    }
    var objetoComoTexto = respuesta.Result.Content.ReadAsStringAsync().Result;
    var movs =
JsonConvert.DeserializeObject<IEnumerable<MovimientoModel>>(objetoComoTexto);
    return View(movs);
}

}

return View();

}

```

[HttpPost]

public ActionResult FiltrarPorArtyTipo(int idArticulo, int idTipo)

```

{
    if (idArticulo == null || idTipo == null || idArticulo == 0 || idTipo == 0)
    {
        return RedirectToAction("Index");
    }
    TempData["idArticulo"] = idArticulo;
    TempData["idTipo"] = idTipo;
    return RedirectToAction("Index", new {filtro = "PorArtyTipo" });
}

```

[HttpPost]

public ActionResult Next()

```
{  
  
    try  
    {  
        string mensaje = "";  
        actualPage++;  
        if (actualPage < 1)  
        {  
            actualPage = 1;  
            mensaje = "Solo números positivos.";  
        }  
  
        return RedirectToAction("Index", new { message = mensaje });  
    }  
    catch (Exception ex)  
    {  
        return RedirectToAction("Index", "Movimineto", new { mensaje = "Hubo un error" });  
    }  
  
}
```

[HttpPost]

public ActionResult Previous()

```
{  
  
    try  
    {  
        string message = "";  
        actualPage--;
```

```

        if (actualPage < 1)
        {
            actualPage = 1;
            message = "Only postive numbers allowed";
        }

        return RedirectToAction("Index", new { message = message });
    }
    catch (Exception ex)
    {
        return RedirectToAction("Index", "Movimineto", new { mensaje = "Hubo un error" });
    }
}

// GET: TeamController/Create
public ActionResult Create(string mensaje)
{
    string token = HttpContext.Session.GetString("token");
    if (string.IsNullOrEmpty(token) || HttpContext.Session.GetString("rol") !=
"Encargado")
    {
        return RedirectToAction("Login", "Login", new { mensaje = "No Autorizado." });
    }

    _client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue(
        "Bearer",
        token
    );

    HttpRequestMessage solicitud =
        new HttpRequestMessage(HttpMethod.Get, new
Uri("http://localhost:5091/api/Articulos"));

    Task<HttpResponseMessage> respuesta = _client.SendAsync(solicitud);
    respuesta.Wait();

```

```

        if (respuesta.Result.IsSuccessStatusCode)
        {
            var objetoComoTexto = respuesta.Result.Content.ReadAsStringAsync().Result;

            IEnumerable<ArticuloModel> articulos =
            JsonConvert.DeserializeObject<IEnumerable<ArticuloModel>>(objetoComoTexto);

            ViewBag.Articulos = articulos;
        }

        HttpRequestMessage solicitudM =
            new HttpRequestMessage(HttpMethod.Get, new
            Uri("http://localhost:5091/api/TipoMovimientos"));

        Task<HttpResponseMessage> respuestaM = _client.SendAsync(solicitudM);

        respuestaM.Wait();

        if (respuestaM.Result.IsSuccessStatusCode)
        {
            var objetoComoTexto = respuestaM.Result.Content.ReadAsStringAsync().Result;

            IEnumerable<TipoMovimientoModel> tipos =
            JsonConvert.DeserializeObject<IEnumerable<TipoMovimientoModel>>(objetoComoTexto);

            ViewBag.TipoMovs = tipos;
        }

        ViewBag.Email = HttpContext.Session.GetString("email");

        ViewBag.mensaje = mensaje;

        return View();
    }

    // POST: TeamController/Create
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create(MovimientoModel mov)
    {
        string token = HttpContext.Session.GetString("token");
    }

```

```

        if (string.IsNullOrEmpty(token) || HttpContext.Session.GetString("rol") !=
"Encargado")
        {
            return RedirectToAction("Login", "Login", new { mensaje = "No Autorizado." });
        }

        _client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue(
            "Bearer",
            token
        );

        try
        {
            HttpRequestMessage solicitud = new HttpRequestMessage(HttpMethod.Post,
new Uri(_baseUrl));

            string json = JsonConvert.SerializeObject(mov);

            HttpContent contenido = new StringContent(json, Encoding.UTF8,
"application/json");

            solicitud.Content = contenido;

            Task<HttpResponseMessage> respuesta = _client.SendAsync(solicitud);

            respuesta.Wait();

            if (respuesta.Result.IsSuccessStatusCode)
            {
                return View();
            }
            else
            {
                string mensaje = "Datos invalidos. Revise la cantidad de unidades e intente
nuevamente.";

                return RedirectToAction("Create", new { mensaje = mensaje });
            }
        }
        catch (Exception e)
        {

```



```

        return RedirectToAction("Create", new { mensaje = "Hubo un error." });
    }
}
}
}

```

=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\MovimientoCon  
troller.cs ===

=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\ReporteControll  
er.cs ===

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Newtonsoft.Json;
using System.Net.Http.Headers;
using WebDeposito.Models;

```

```

namespace WebDeposito.Controllers

```

```

{
    public class ReporteController : Controller
    {
        private HttpClient _client;
        private string _baseUrl;
        public ReporteController()
        {
            _client = new HttpClient();
            _baseUrl = "http://localhost:5091/api/Movimientos/GetMovsXFecha";
        }
        // GET: ReporteController
        public ActionResult Index(string mensaje)
        {

```

```

        string token = HttpContext.Session.GetString("token");
        if (string.IsNullOrEmpty(token) || HttpContext.Session.GetString("rol") !=
"Encargado")
        {
            return RedirectToAction("Login", "Login", new { mensaje = "No Autorizado." });
        }
        _client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue(
            "Bearer",
            token
        );
        ViewBag.mensaje = mensaje;

        HttpRequestMessage solicitud =
            new HttpRequestMessage(HttpMethod.Get, new Uri(_baseUrl));
        Task<HttpResponseMessage> respuesta = _client.SendAsync(solicitud);
        respuesta.Wait();
        if (respuesta.Result.IsSuccessStatusCode)
        {
            if (respuesta.Result.StatusCode == System.Net.HttpStatusCode.NoContent)
            {
                ViewBag.mensaje = "No hay resultados.";
                return View();
            }
            var objetoComoTexto = respuesta.Result.Content.ReadAsStringAsync().Result;
            var movs =
                JsonConvert.DeserializeObject<IEnumerable<FechaModel>>(objetoComoTexto);
            return View(movs);
        }

        return View();
    }
}

```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Controllers\ReporteControl  
er.cs ===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\ArticuloModel.cs  
===
```

```
namespace WebDeposito.Models  
{  
    public class ArticuloModel  
    {  
        public int Id { get; set; }  
        public string Nombre { get; set; }  
        public stringCodigo { get; set; }  
        public string Descripcion { get; set; }  
        public double Precio { get; set; }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\ArticuloModel.cs  
===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\ErrorViewModel.cs  
===
```

```
namespace WebDeposito.Models  
{  
    public class ErrorViewModel  
    {  
        public string? RequestId { get; set; }  
  
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\ErrorViewModel.cs  
===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\FechaModel.cs  
===
```

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

```
namespace WebDeposito.Models
```

```
{  
    public class FechaModel  
    {  
        public int Fecha { get; set; }  
        public List<TMovsModel> Movs { get; set; }  
        public int Total { get; set; }  
    }  
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\FechaModel.cs  
===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other  
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\MovimientoModel.cs  
s ===
```

```
namespace WebDeposito.Models
```

```
{  
    public class MovimientoModel  
    {  
        public int Id { get; set; }  
        public DateTime FechaMovimiento { get; set; }  
        public int ArticuloId { get; set; }  
        public int TipoMovimientoid { get; set; }  
        public ArticuloModel? Articulo { get; set; }  
    }  
}
```

```

        public TipoMovimientoModel? TipoMov { get; set; }

        public string EmailUsuario { get; set; }

        public int CantUnidades { get; set; }

    }
}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\MovimientoModel.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\TipoMovimientoModel.cs ===

```

```

namespace WebDeposito.Models
{
    public class TipoMovimientoModel
    {
        public int Id { get; set; }

        public string Nombre { get; set; }

        public int TipoMovStock { get; set; }

    }
}

```

```

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\TipoMovimientoModel.cs ===

```

```

=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\TMovsModel.cs
===

```

```

using Microsoft.AspNetCore.Mvc;

```

```

namespace WebDeposito.Models
{
    public class TMovsModel
    {

```

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

        public int Total { get; set; }

    }

}

=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\TMovsModel.cs
===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\TokenModel.cs
===
```

```
namespace WebDeposito.Models
```

```
{

    public class TokenModel

    {

        public UserModel Usuario { get; set; }

        public string Token { get; set; }

        public string Rol { get; set; }

    }

}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\TokenModel.cs
===
```

```
=== Start of C:\Users\Usuario\desktop\ORT & Other
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\UserModel.cs ===
```

```
namespace WebDeposito.Models
```

```
{

    public class UserModel

    {

        public string Email { get; set; }

        public string PasswordSinEncript { get; set; }

    }

}
```

```
}
```

```
}
```

```
=== End of C:\Users\Usuario\desktop\ORT & Other
```

```
shi\obligatorioM3C\WebDeposito\WebDeposito\WebDeposito\Models\UserModel.cs ===
```