

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO 2 PROGRAMACIÓN 3



Carolina Larroque – 254338

Grupo N3G

Analista en Tecnologías de la Información

20-06-2024

Índice

Diagrama de casos de uso	3
Diagrama de clases	4
Generación de Test Data	5
Artículos	5
Movimientos de Stock	7
Consulta sobre paginación	8
Web Api	11
Status Codes	11
Artículos	11
Pedidos	11
Usuarios	12
Tipos de Movimientos	12
Movimientos de Stock	13
Código	14
Solucion MVC	14
Solucion WebApi	33

Diagrama de casos de uso

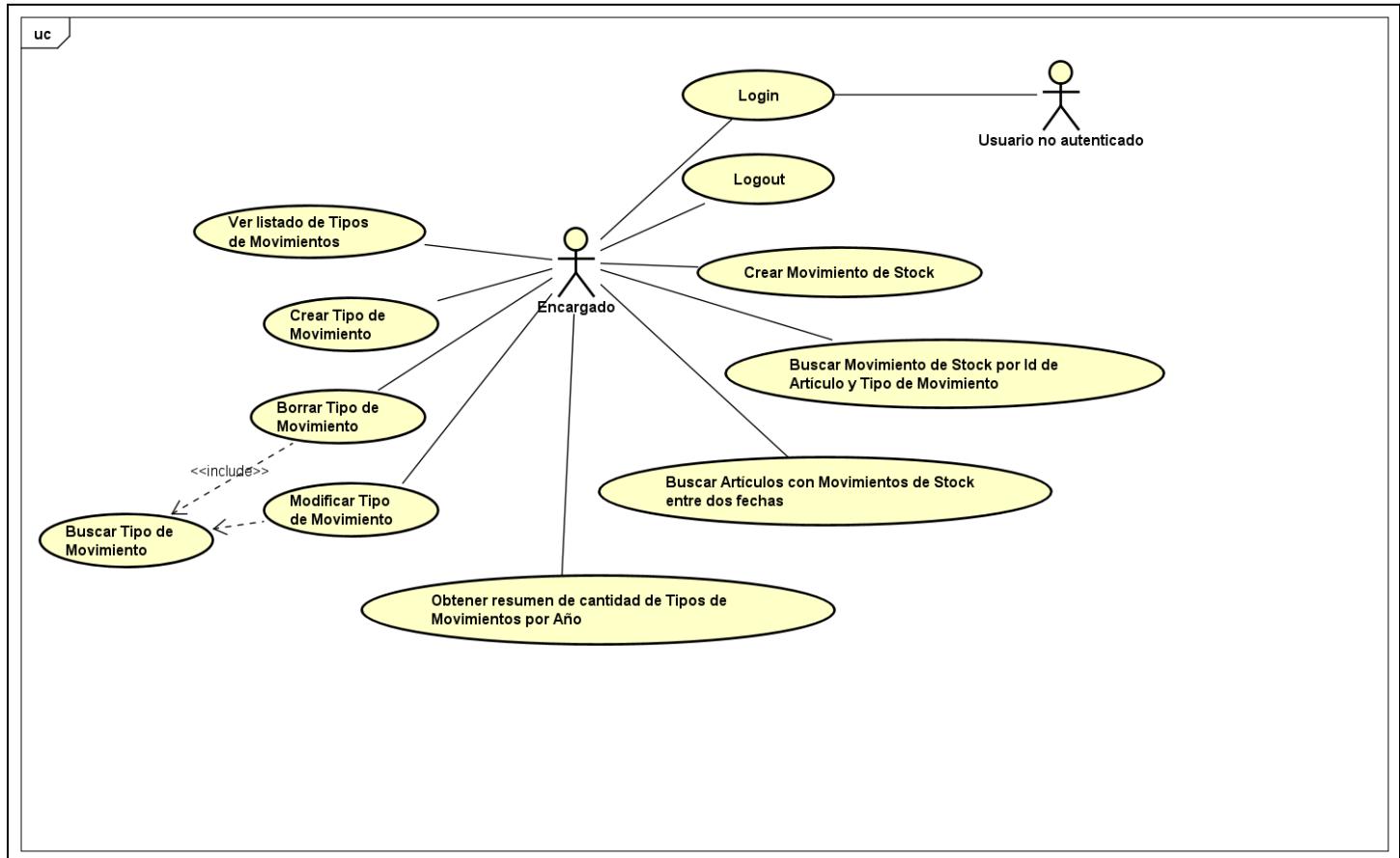
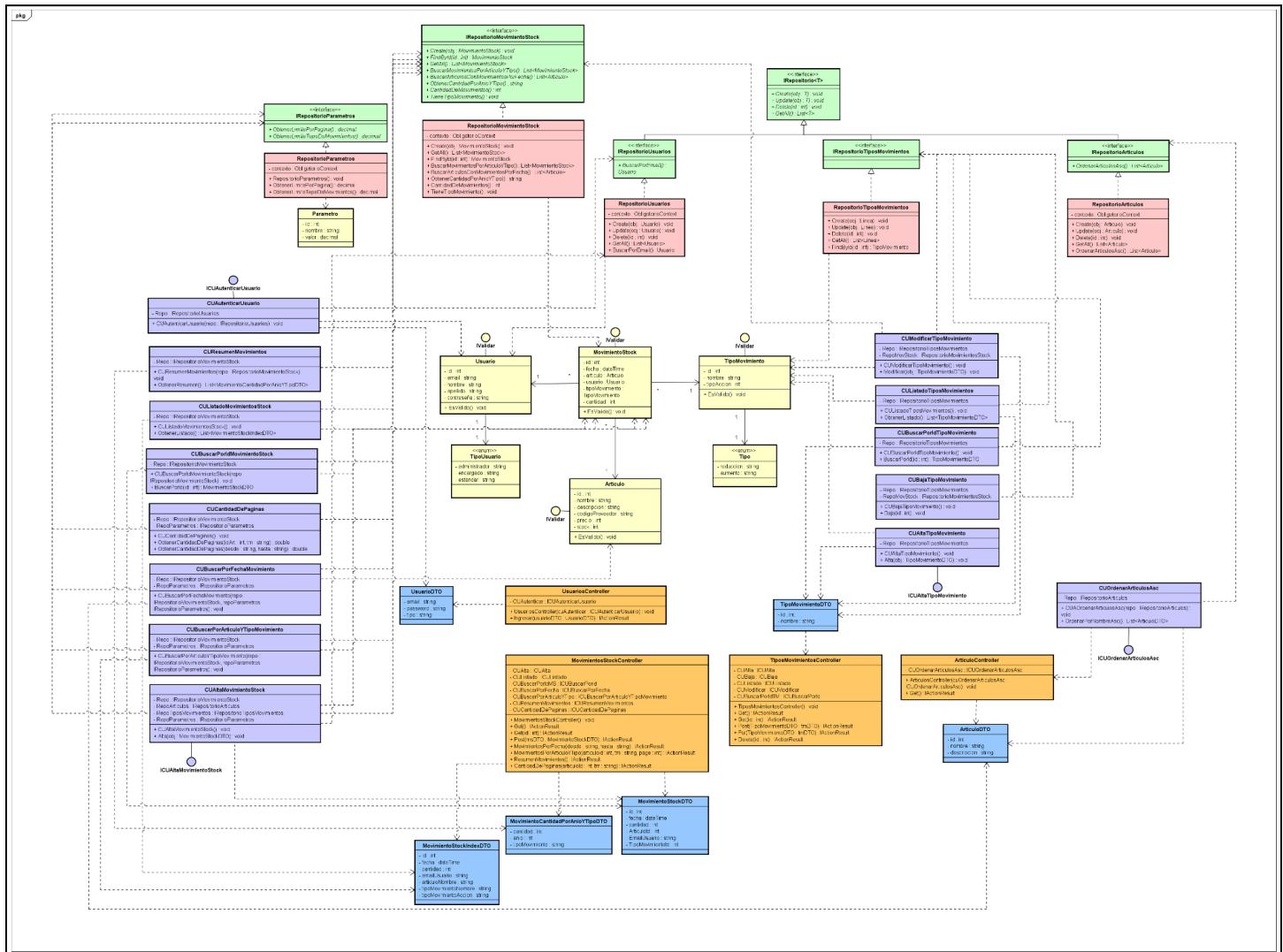


Diagrama de clases



Generación de Test Data

Artículos

Input:

Estás trabajando en el desarrollo de un sistema de ventas de insumos de papelería. Necesitas generar datos de prueba para una entidad llamada "Articulos". Los valores de las columnas "Nombre" y "CodigoProveedor" no se pueden repetir entre registros (son únicos). Necesitas generar 72 registros de artículos con Nombre, Descripcion, CodigoProveedor, Precio y Stock tomando como base el siguiente script de sql. La Descripcion debe ser corta, 50 caracteres como maximo.

```
"insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Calculadora Casio', 'Calculadora de bolsillo Casio', '1234567890123', 300, 10);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Resma A4', '500 hojas para imprimir en formato A4', '1234567890124', 250, 50);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Auriculares con Micrófono', 'Bluetooth o conexión USB, cable desmontable', '1234567890125', 3200, 20);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Silla De Escritorio', 'Soporte lumbar, altura ajustable y base giratoria n ruedas', '1234567890126', 3200, 20);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Notas Adhesivas', 'Block de notas autoadhesivas', '1234597890124', 50, 100);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Bandas Elásticas', '100 unidades', '1274597890124', 150, 90);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Engrampadora', 'Hasta 100 hojas', '1834597890124', 800, 10);  
  
insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cable HDMI', 'HD, FullHD y 4K,', '1236597890124', 200, 10);"
```

Output:

Aquí tienes los valores generados para la entidad "Articulos". Los valores de "Nombre" y "CodigoProveedor" son únicos para cada registro.

```

1 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Calculadora Casio', 'Calculadora de bolsillo Casio', '1234567890123', 300, 10);
2 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Resma A4', '500 hojas para imprimir en formato A4', '1234567890124', 250, 50);
3 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Auriculares con Micrófono', 'Bluetooth o conexión USB, cable desmontable', '1234567890125', 3200, 20);
4 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Silla De Escritorio', 'Soporte lumbar, altura ajustable', '1234567890126', 3200, 20);
5 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Notas Adhesivas', 'Bloque de notas autoadhesivas', '1234597890124', 50, 100);
6 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Bandas Elásticas', '100 unidades', '1274597890124', 150, 90);
7 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Engrapadora', 'Hasta 100 hojas', '1834597890124', 800, 10);
8 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cable HDMI', 'HD, FullHD y 4K', '1236597890124', 200, 10);
9 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Lápices HB', 'Caja de 12 lápices', '1234567890127', 100, 50);
10 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cuaderno A5', 'Cuaderno de 100 hojas', '1234567890128', 150, 30);
11 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Bolígrafos Azules', 'Paquete de 10 bolígrafos', '1234567890129', 50, 70);
12 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Tijeras', 'Tijeras de acero inoxidable', '1234567890130', 80, 40);
13 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Pegamento en Barra', 'Pegamento de 20g', '1234567890131', 40, 60);
14 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Carpetas A4', 'Carpetas con anillos', '1234567890132', 90, 30);
15 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Rotuladores', 'Set de 6 colores', '1234567890133', 120, 50);
16 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Libreta de Notas', 'Libreta de 50 hojas', '1234567890134', 70, 80);
17 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Gomas de Borrar', 'Paquete de 4 gomas', '1234567890135', 30, 100);
18 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Sacapuntas', 'Sacapuntas de metal', '1234567890136', 25, 90);
19 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Corrector Líquido', 'Corrector blanco de 20ml', '1234567890137', 45, 70);
20 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Marcadores Permanentes', 'Set de 4 colores', '1234567890138', 100, 60);
21 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Papel Fotocopia', 'Paquete de 500 hojas', '1234567890139', 250, 40);
22 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Carpetas Colgantes', 'Paquete de 10 carpetas', '1234567890140', 150, 30);
23 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Calculadora Científica', 'Calculadora con funciones científicas', '1234567890141', 400, 20);
24 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Tacos de Notas', 'Paquete de 12 tacos', '1234567890142', 100, 60);
25 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cinta Adhesiva', 'Cinta transparente 5m', '1234567890143', 70, 80);
26 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Clips', 'Caja de 100 clips', '1234567890144', 20, 90);
27 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Grapas', 'Caja de 1000 grapas', '1234567890145', 15, 100);
28 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Archivador', 'Archivador tamaño A4', '1234567890146', 200, 40);
29 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Portaminas', 'Portaminas con 10 minas', '1234567890147', 80, 50);
30 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cuaderno Espiral', 'Cuaderno de 80 hojas', '1234567890148', 90, 60);
31 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Bloc de Dibujo', 'Bloc de 30 hojas A3', '1234567890149', 150, 30);
32 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Lápices de Colores', 'Set de 24 colores', '1234567890150', 200, 40);
33 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Estuche', 'Estuche para lápices', '1234567890151', 100, 50);
34 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Calculadora Gráfica', 'Calculadora para gráficos', '1234567890152', 500, 10);
35 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Perforadora', 'Perforadora de 2 agujeros', '1234567890153', 120, 40);
36 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cinta Correctora', 'Cinta correctora de 10m', '1234567890154', 50, 80);
37 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Tijeras para Niños', 'Tijeras con punta redonda', '1234567890155', 30, 70);
38 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Notas de Colores', 'Paquete de 6 bloques', '1234567890156', 90, 60);
39 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Agenda', 'Agenda 2024', '1234567890157', 250, 30);
40 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cuaderno de Música', 'Cuaderno de 50 hojas pautadas', '1234567890158', 80, 40);
41 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Compás', 'Compás metálico', '1234567890159', 70, 50);
42 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Transportador', 'Transportador de plástico', '1234567890160', 20, 100);
43 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Set de Regla', 'Set de 3 reglas', '1234567890161', 40, 90);
44 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Pinzas para Papel', 'Paquete de 20 pinzas', '1234567890162', 50, 80);
45 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Post-it de Colores', 'Paquete de 5 colores', '1234567890163', 100, 70);
46 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Bolígrafos Negros', 'Paquete de 10 bolígrafos', '1234567890164', 50, 70);
47 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Notas Adhesivas Pequeñas', 'Notas de 5x5cm', '1234567890165', 30, 100);
48 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Clip de Papel Grande', 'Caja de 50 clips', '1234567890166', 40, 80);
49 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Grapadora Grande', 'Grapadora para 150 hojas', '1234567890167', 200, 10);
50 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Lápices Mecánicos', 'Set de 2 lápices', '1234567890168', 100, 50);
51 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cuaderno de Dibujo', 'Cuaderno de 40 hojas A4', '1234567890169', 120, 40);
52 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Goma de Borrar', 'Goma de borrar grande', '1234567890170', 30, 90);
53 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Rotuladores Fino', 'Set de 12 colores', '1234567890171', 150, 30);
54 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Carpetas Transparetes', 'Paquete de 10 carpetas', '1234567890172', 80, 60);
55 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Regla de Metal', 'Regla de 30cm', '1234567890173', 40, 50);
56 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Bolígrafos Rojos', 'Paquete de 10 bolígrafos', '1234567890174', 50, 70);
57 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Subrayadores', 'Set de 6 colores', '1234567890175', 120, 50);
58 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cuaderno con Cubierta Dura', 'Cuaderno de 100 hojas', '1234567890176', 150, 40);
59 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Grapas Grandes', 'Caja de 500 grapas', '1234567890177', 25, 90);
60 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Separadores de Carpetas', 'Set de 5 separadores', '1234567890178', 60, 80);
61 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Cinta Adhesiva de Papel', 'Cinta de 20m', '1234567890179', 50, 70);
62 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Tijeras de Cocina', 'Tijeras multiusos', '1234567890180', 100, 50);
63 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Archivador con Llave', 'Archivador de metal', '1234567890181', 300, 20);
64 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Pizarra Blanca', 'Pizarra de 90x60cm', '1234567890182', 250, 15);
65 insert into Articulos(Nombre, Descripcion, CodigoProveedor, Precio, Stock) values ('Rotuladores para Pizarra', 'Set de 4 colores', '1234567890183', 100, 50);

```

Con estos datos se completan los 72 registros solicitados para la entidad "Articulos".

Movimientos de Stock

Input:

Estás trabajando en el desarrollo de un sistema de ventas de insumos de papelería.

Necesitas generar datos de prueba para una entidad llamada "MovimientosDeStock".

Necesitas generar 30 registros de MovimientosDeStock con Fecha, ArticuloId, UsuarioId,

TipoMovimientoId y Cantidad tomando como base el siguiente script de sql.

```
"insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad)
values ('16-05-2024',1,3,1,5);
```

```
insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad)
values ('18-05-2024',2,4,1,2);"
```

Tener en cuenta que:

1. El mínimo valor de Id para ArticuloId es 1 y el máximo 15.
2. El mínimo valor de Id para UsuarioId es 3 y el máximo 8.
3. El mínimo valor de Id para TipoMovimientoId es 1 y el máximo 10.
4. La Cantidad debe ser menor o igual a 20

Output:

Aquí tienes los valores generados para la entidad "MovimientosDeStock". Los datos cumplen con las restricciones especificadas para los valores de Id y la cantidad.

```
1 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('16-05-2024', 1, 3, 1, 5);
2 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('18-05-2024', 2, 4, 1, 2);
3 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('01-06-2024', 3, 5, 2, 10);
4 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('02-06-2024', 4, 6, 3, 15);
5 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('03-06-2024', 5, 7, 4, 8);
6 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('04-06-2024', 6, 8, 5, 20);
7 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('05-06-2024', 7, 3, 6, 12);
8 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('06-06-2024', 8, 4, 7, 6);
9 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('07-06-2024', 9, 5, 8, 18);
10 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('08-06-2024', 10, 6, 9, 14);
11 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('09-06-2024', 11, 7, 10, 11);
12 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('10-06-2024', 12, 8, 1, 3);
13 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('11-06-2024', 13, 3, 2, 7);
14 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('12-06-2024', 14, 4, 3, 9);
15 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('13-06-2024', 15, 5, 4, 13);
16 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('14-06-2024', 1, 6, 5, 4);
17 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('15-06-2024', 2, 7, 6, 16);
18 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('16-06-2024', 3, 8, 7, 19);
19 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('17-06-2024', 4, 3, 8, 5);
20 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('18-06-2024', 5, 4, 9, 17);
21 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('19-06-2024', 6, 5, 10, 12);
22 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('20-06-2024', 7, 6, 1, 10);
23 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('21-06-2024', 8, 7, 2, 14);
24 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('22-06-2024', 9, 8, 3, 11);
25 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('23-06-2024', 10, 3, 4, 6);
26 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('24-06-2024', 11, 4, 5, 8);
27 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('25-06-2024', 12, 5, 6, 13);
28 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('26-06-2024', 13, 6, 7, 7);
29 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('27-06-2024', 14, 7, 8, 9);
30 insert into MovimientosDeStock(Fecha, ArticuloId, UsuarioId, TipoMovimientoId, Cantidad) values ('28-06-2024', 15, 8, 9, 5);
```

Con estos datos se completan los 30 registros solicitados para la entidad "MovimientosDeStock".

Consulta sobre paginación

Input:

You're a C# developer consuming an API with pagination in .Net8.
In your mvc view you have two inputs that send an id and a string through the api url and
display the paginated data in a <table> with pagination buttons at the bottom of the
page.
The API call has been tested in swagger and it's working correctly. But you're not being
able to connect the pagination buttons to make the new api calls.
Here's the code in your .cshtml

```
1 <ul class="pagination justify-content-center">
2     @for (int i = 1; i <= ViewBag.Paginas; i++) {
3         <li class="page-item">
4             <a class="page-link" href=@($"MovimientosStock/BuscarPorArticuloYTipo?page={i}")>
5                 @i
6             </a>
7         </li>
8     }
9 </ul>
```

Here is the Controller Method:

```
1 public ActionResult BuscarPorArticuloYTipo(int idArticulo, string tipoMovimiento, int? page) {
2     if (page == null) page = 1;
3
4     List<MovimientoStockIndexDTO> movimientosDeStock = new List<MovimientoStockIndexDTO>();
5
6     try {
7         HttpClient client = new HttpClient();
8         string url = UrlApi + $"MovimientosStock/MovimientosPorArticuloYTipo/{idArticulo}/{tipoMovimiento}/{page}";
9         client.DefaultRequestHeaders.Authorization = new
10            System.Net.Http.Headers.AuthenticationHeaderValue("Bearer", HttpContext.Session.GetString("Token"));
11         var tarea = client.GetAsync(url);
12         tarea.Wait();
13         var respuesta = tarea.Result;
14
15         HttpResponseMessage content = respuesta.Content;
16         var tarea2 = content.ReadAsStringAsync();
17         tarea2.Wait();
18         string body = tarea2.Result;
19
20         if (respuesta.IsSuccessStatusCode) {
21             movimientosDeStock = JsonConvert.DeserializeObject<List<MovimientoStockIndexDTO>>(body);
22             double cantidadPaginas = ObtenerCantidadPaginas(idArticulo, tipoMovimiento);
23             ViewBag.Paginas = Math.Ceiling(cantidadPaginas);
24             ViewBag.ArticuloId = idArticulo;
25             ViewBag.TipoMov = tipoMovimiento;
26         } else {
27             ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
28         }
29     } catch (Exception e) {
30         ViewBag.ErrorMsg = e.Message;
31     }
32
33     return View(movimientosDeStock);
34 }
```

Output:

To connect your pagination buttons to make new API calls and handle the form submission correctly, you need to ensure a few things:

1. The form action should handle both the initial search and pagination.

2. The pagination links should include the necessary query parameters (`idArticulo` and `tipoMovimiento`).
3. You should correctly handle the `page` parameter in the controller to fetch the paginated data.

Here is how you can refactor your code to achieve this:

Controller Method

First, update the `BuscarPorArticuloYTipo` method to handle both the initial search and pagination. Use the `HttpGet` attribute to allow pagination links to trigger the method correctly.

```

1 [HttpGet]
2 public ActionResult BuscarPorArticuloYTipo(int idArticulo, string tipoMovimiento, int? page)
3 {
4     if (page == null) page = 1;
5
6     List<MovimientoStockIndexDTO> movimientosDeStock = new List<MovimientoStockIndexDTO>();
7
8     try
9     {
10         HttpClient client = new HttpClient();
11         string url = UrlApi + $"MovimientosStock/MovimientosPorArticuloYTipo/{idArticulo}/{tipoMovimiento}/{page}";
12         client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
13             HttpContext.Session.GetString("Token"));
14         var tarea = client.GetAsync(url);
15         tarea.Wait();
16         var respuesta = tarea.Result;
17
18         HttpContent content = respuesta.Content;
19         var tarea2 = content.ReadAsStringAsync();
20         tarea2.Wait();
21         string body = tarea2.Result;
22
23         if (respuesta.IsSuccessStatusCode)
24         {
25             movimientosDeStock = JsonConvert.DeserializeObject<List<MovimientoStockIndexDTO>>(body);
26             double cantidadPaginas = ObtenerCantidadPaginas(idArticulo, tipoMovimiento);
27             ViewBag.Paginas = Math.Ceiling(cantidadPaginas);
28             ViewBag.ArticuloId = idArticulo;
29             ViewBag.TipoMov = tipoMovimiento;
30         }
31         else
32         {
33             ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
34         }
35     }
36     catch (Exception e)
37     {
38         ViewBag.ErrorMsg = e.Message;
39     }
40
41     return View(movimientosDeStock);
42 }
43 [HttpPost]
44 public ActionResult BuscarPorArticuloYTipo(int idArticulo, string tipoMovimiento)
45 {
46     return RedirectToAction("BuscarPorArticuloYTipo", new { idArticulo, tipoMovimiento, page = 1 });
47 }
48

```

View (.cshtml)

Update your view to include the form and pagination links correctly.

```

1 @if (ViewBag.Paginas != null && ViewBag.Paginas > 1)
2 {
3     <ul class="pagination justify-content-center">
4         @for (int i = 1; i <= ViewBag.Paginas; i++)
5         {
6             <li class="page-item @(i == ViewBag.CurrentPage ? "active" : "")">
7                 <a class="page-link" href="@Url.Action("BuscarPorArticuloYTipo", new { idArticulo = ViewBag.ArticuloId, tipoMovimiento =
8                     ViewBag.TipoMov, page = i })">
9                     @i
10                </a>
11            </li>
12        }
13    </ul>

```

Explanation

1. Controller Methods:

- The `HttpGet` method `BuscarPorArticuloYTipo` handles both the initial search and pagination.
- The `HttpPost` method redirects to the `HttpGet` method with the search parameters and sets the page to 1.

2. View:

- The form posts to the `BuscarPorArticuloYTipo` action method to initiate a search.
- The `value` attributes for the inputs are set to `ViewBag.ArticuloId` and `ViewBag.TipoMov` to retain the search values after pagination.
- The pagination links include `idArticulo`, `tipoMovimiento`, and `page` parameters to correctly fetch the paginated data.
- A conditional check ensures that pagination links are only rendered if there is more than one page of results.

This should allow your pagination links to correctly trigger new API calls and display the paginated data.

Web Api

Status Codes

Artículos

Código	Mensaje
200	Ok
404	No existen artículos
500	Ocurrió un error en el servidor

Pedidos

Código	Mensaje
200	Ok
404	No existen pedidos
500	Ocurrió un error en el servidor

Usuarios

Código	Mensaje
200	Ok
400	Datos incorrectos
404	El usuario no existe
500	Ocurrió un error inesperado en el servidor. Reintente más tarde.

Tipos de Movimientos

Código	Mensaje
200	Ok
400	<ul style="list-style-type: none">1. El id debe ser positivo2. Faltan datos requeridos para el alta3. Faltan datos requeridos para la modificación
404	El tipo de movimiento no existe
500	Ocurrió un error inesperado en el servidor. Reintente más tarde.

Movimientos de Stock

Código	Mensaje
200	Ok
400	<ol style="list-style-type: none">1. El id debe ser positivo2. Faltan datos requeridos para el alta3. El artículo es requerido para el alta4. Las fechas son requeridas5. Artículo y tipo de movimiento son requeridos.
404	<ol style="list-style-type: none">1. El movimiento de stock no existe2. No existen movimientos para las fechas seleccionadas.3. No existen movimientos para la combinación de Artículo y tipo de movimiento seleccionados
500	Ocurrió un error inesperado en el servidor. Reintente más tarde.

Código

Solucion MVC

```
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\ArticuloDTO.cs
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class ArticuloDTO {
        [Display(Name = "Id del Artículo")]
        public int Id { get; set; }
        [Display(Name = "Nombre del Artículo")]
        public string Nombre { get; set; }
        [Display(Name = "Descripción del Artículo")]
        public string Descripcion { get; set; }

    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\MovimientoCantidadPorAnioYTipo
DTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MovimientoCantidadPorAnioYTipoDTO {
        public int Cantidad { get; set; }
        public int Anio { get; set; }
        public string TipoMovimiento { get; set; }

    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\MovimientoStockDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MovimientoStockDTO {
        public int Id { get; set; }
        public DateTime Fecha { get; set; }
        public int Articuloid { get; set; }
        public string EmailUsuario { get; set; }
        public int TipоМovimiento { get; set; }
        public int Cantidad { get; set; }

        public List<ArticuloDTO> Articulos { get; set; } = new List<ArticuloDTO>();
    }
}
```

```

        public List<TipoMovimientoDTO> TiposMovimientos { get; set; } = new List<TipoMovimientoDTO>();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\MovimientoStockIndexDTO.cs
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MovimientoStockIndexDTO {
        public int Id { get; set; }
        public DateTime Fecha { get; set; }
        [Display(Name = "Articulo")]
        public string ArticuloNombre { get; set; }
        [Display(Name = "Usuario")]
        public string EmailUsuario { get; set; }
        [Display(Name = "Tipo de Movimiento")]
        public string TipoMovimientoNombre { get; set; }
        public int Cantidad { get; set; }
        [Display(Name = "Acción Tipo de Movimiento")]
        public string? TipoMovimientoAccion { get; set; } // RF-04-a
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\MovimientoStockResumenDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MovimientoStockResumenDTO {
        public List<string> TipoCantidad { get; set; } = new List<string>();
        public int CantidadTotalAnio { get; set; }
        public int Anio { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\TipoMovimientoDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class TipoMovimientoDTO {
        public int Id { get; set; }
        public string Nombre { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\UsuarioAutenticadoDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;

namespace DataTransferObjects {
    public class UsuarioAutenticadoDTO {
        public string Email { get; set; }
        public string Token { get; set; }
        public string Tipo { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\DataTransferObjects\UsuarioDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class UsuarioDTO {
        public string Email { get; set; }
        public string Password { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Controllers\HomeController.cs
using Microsoft.AspNetCore.Mvc;
using ObligatorioMVC.Models;
using System.Diagnostics;

namespace ObligatorioMVC.Controllers {
    public class HomeController : Controller {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger) {
            _logger = logger;
        }

        public IActionResult Index() {
            return View();
        }

        public IActionResult Privacy() {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error() {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Controllers\MovimientosStockController.cs
using DataTransferObjects;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Newtonsoft.Json;
using Obligatorio.Filters;

namespace ObligatorioMVC.Controllers {
    public class MovimientosStockController : Controller {

```

```

public string UrlApi;

public MovimientosStockController(IConfiguration config) {
    UrlApi = config.GetValue<string>("ApiUrl");
}

// GET: MovimientosStockController
[Privado(TipoUsuarios = "Encargado")]
public ActionResult Index() {
    List<MovimientoStockIndexDTO> movimientosDeStock = new List<MovimientoStockIndexDTO>();

    try {
        HttpClient client = new HttpClient();
        string url = UrlApi + "movimientosStock";
        client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
            HttpContext.Session.GetString("Token"));
        var tarea = client.GetAsync(url);
        tarea.Wait();
        var respuesta = tarea.Result;

        HttpContent content = respuesta.Content;
        var tarea2 = content.ReadAsStringAsyncAsync();
        tarea2.Wait();
        string body = tarea2.Result;

        if (respuesta.IsSuccessStatusCode) {
            movimientosDeStock = JsonConvert.DeserializeObject<List<MovimientoStockIndexDTO>>(body);
        } else {
            ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
        }
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }

    return View(movimientosDeStock);
}

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

//-----
//----- CREATE -----
//-----

// GET: MovimientosStockController/Create
[Privado(TipoUsuarios = "Encargado")]
public ActionResult Create() {
    MovimientoStockDTO movimientoDeStock = new MovimientoStockDTO();
    try {
        movimientoDeStock.Articulos = ObtenerArticulos();
        movimientoDeStock.TiposMovimientos = ObtenerTiposDeMovimientos();
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }
    return View(movimientoDeStock);
}

// POST: MovimientosStockController/Create

```

```

[HttpPost]
[Privado(TipoUsuarios = "Encargado")]
public ActionResult Create(MovimientoStockDTO movimientoDeStock) {
    movimientoDeStock.EmailUsuario = HttpContext.Session.GetString("Email");
    try {
        HttpClient cliente = new HttpClient();
        string url = UrlApi + "movimientosStock";
        cliente.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
        HttpContext.Session.GetString("Token"));
        var tarea = cliente.PostAsJsonAsync(url, movimientoDeStock);
        tarea.Wait();
        var respuesta = tarea.Result;

        HttpContent content = respuesta.Content;
        var tarea2 = content.ReadAsStringAsync();
        tarea2.Wait();
        string body = tarea2.Result;

        if (respuesta.IsSuccessStatusCode) {
            return RedirectToAction(nameof(Index));
        } else {
            ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
        }
    } catch {}
    movimientoDeStock.Articulos = ObtenerArticulos();
    movimientoDeStock.TiposMovimientos = ObtenerTiposDeMovimientos();
    return View(movimientoDeStock);
}

//-----
//----- BUSCAR -----
//-----

[HttpGet]
[Privado(TipoUsuarios = "Encargado")]
public ActionResult BuscarPorFecha(string desde, string hasta, int? page) {
    if (page == null) page = 1;
    List<ArticuloDTO> articulosConMovimientosDeStock = new List<ArticuloDTO>();

    try {
        HttpClient client = new HttpClient();
        string url = UrlApi + $"MovimientosStock/MovimientosPorFecha/{desde}/{hasta}/{page}";
        client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
        HttpContext.Session.GetString("Token"));
        var tarea = client.GetAsync(url);
        tarea.Wait();
        var respuesta = tarea.Result;

        HttpContent content = respuesta.Content;
        var tarea2 = content.ReadAsStringAsync();
        tarea2.Wait();
        string body = tarea2.Result;

        if (respuesta.IsSuccessStatusCode) {
            articulosConMovimientosDeStock = JsonConvert.DeserializeObject<List<ArticuloDTO>>(body);
            double cantidadPaginas = ObtenerCantidadPaginas($"MovimientosStock/CantidadDePaginasFechas/{desde}/{hasta}");
            ViewBag.Paginas = Math.Ceiling(cantidadPaginas);
            ViewBag.Desde = desde;
            ViewBag.Hasta = hasta;
        } else {
            ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
        }
    }
}

```

```

        } catch (Exception e) {
            ViewBag.ErrorMsg = e.Message;
        }

        return View(articulosConMovimientosDeStock);
    }

    [HttpPost]
    [Privado(TipoUsuarios = "Encargado")]
    public ActionResult BuscarPorFecha(string desde, string hasta) {
        return RedirectToAction("BuscarPorFecha", new { desde, hasta, page = 1 });
    }

    [HttpGet]
    [Privado(TipoUsuarios = "Encargado")]
    public ActionResult BuscarPorArticuloYTipo(int idArticulo, string tipoMovimiento, int? page) {
        if (page == null) page = 1;

        List<MovimientoStockIndexDTO> movimientosDeStock = new List<MovimientoStockIndexDTO>();

        try {
            HttpClient client = new HttpClient();
            string url = UrlApi + $"MovimientosStock/MovimientosPorArticuloYTipo/{idArticulo}/{tipoMovimiento}/{page}";
            client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
                HttpContext.Session.GetString("Token"));
            var tarea = client.GetAsync(url);
            tarea.Wait();
            var respuesta = tarea.Result;

            HttpContent content = respuesta.Content;
            var tarea2 = content.ReadAsStringAsyncAsync();
            tarea2.Wait();
            string body = tarea2.Result;

            if (respuesta.IsSuccessStatusCode) {
                movimientosDeStock = JsonConvert.DeserializeObject<List<MovimientoStockIndexDTO>>(body);
                double cantidadPaginas = ObtenerCantidadPaginas($"MovimientosStock/CantidadDePaginas/{idArticulo}/{tipoMovimiento}");
                ViewBag.Paginas = Math.Ceiling(cantidadPaginas);
                ViewBag.Articuloid = idArticulo;
                ViewBag.TipoMov = tipoMovimiento;
            } else {
                ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
            }
        } catch (Exception e) {
            ViewBag.ErrorMsg = e.Message;
        }

        return View(movimientosDeStock);
    }

    [HttpPost]
    [Privado(TipoUsuarios = "Encargado")]
    public ActionResult BuscarPorArticuloYTipo(int idArticulo, string tipoMovimiento) {
        return RedirectToAction("BuscarPorArticuloYTipo", new { idArticulo, tipoMovimiento, page = 1 });
    }

    //-----
    //----- RESUMEN -----
    //-----

    [Privado(TipoUsuarios = "Encargado")]
    public ActionResult ObtenerResumen() {

```

```

List<MovimientoCantidadPorAnioYTipoDTO> movimientosDeStock = new List<MovimientoCantidadPorAnioYTipoDTO>();
List<MovimientoStockResumenDTO> resumenMS = new List<MovimientoStockResumenDTO>();

try {
    HttpClient client = new HttpClient();
    string url = UrlApi + "MovimientosStock/ResumenMovimientos";
    client.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
HttpContext.Session.GetString("Token"));
    var tarea = client.GetAsync(url);
    tarea.Wait();
    var respuesta = tarea.Result;

    HttpContent content = respuesta.Content;
    var tarea2 = content.ReadAsStringAsyncAsync();
    tarea2.Wait();
    string body = tarea2.Result;

    if (respuesta.IsSuccessStatusCode) {
        movimientosDeStock = JsonConvert.DeserializeObject<List<MovimientoCantidadPorAnioYTipoDTO>>(body);

        resumenMS = movimientosDeStock
            .GroupBy(mse => mse.Anio)
            .Select(mse => new MovimientoStockResumenDTO {
                Anio = mse.Key,
                TipoCantidad = mse.Select(item => $"Tipo: {item.TipoMovimiento} - Cantidad: {item.Cantidad}").ToList(),
                CantidadTotalAnio = mse.Sum(item => item.Cantidad)
            })
            .ToList();
    } else {
        ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
    }
} catch (Exception e) {
    ViewBag.ErrorMsg = e.Message;
}

return View(resumenMS);
}

//-----
//----- UTILS -----
//-----

public List<ArticuloDTO> ObtenerArticulos() {
    List<ArticuloDTO> articulos = new List<ArticuloDTO>();

    HttpClient cliente = new HttpClient();
    string url = UrlApi + "articulos";
    cliente.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
HttpContext.Session.GetString("Token"));
    var tarea = cliente.GetAsync(url);
    tarea.Wait();
    var respuesta = tarea.Result;

    HttpContent content = respuesta.Content;
    var tarea2 = content.ReadAsStringAsyncAsync();
    tarea2.Wait();
    string body = tarea2.Result;

    if (respuesta.IsSuccessStatusCode) {
        articulos = JsonConvert.DeserializeObject<List<ArticuloDTO>>(body);
    }
}

```

```

        return articulos;
    }

    public List<TipoMovimientoDTO> ObtenerTiposDeMovimientos() {
        List<TipoMovimientoDTO> tiposDeMovimientos = new List<TipoMovimientoDTO>();

        HttpClient cliente = new HttpClient();
        string url = UrlApi + "tiposMovimientos";
        cliente.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
HttpContext.Session.GetString("Token"));
        var tarea = cliente.GetAsync(url);
        tarea.Wait();
        var respuesta = tarea.Result;

        HttpContent content = respuesta.Content;
        var tarea2 = content.ReadAsStringAsync();
        tarea2.Wait();
        string body = tarea2.Result;

        if (respuesta.IsSuccessStatusCode) {
            tiposDeMovimientos = JsonConvert.DeserializeObject<List<TipoMovimientoDTO>>(body);
        }
    }

    return tiposDeMovimientos;
}

public double ObtenerCantidadPaginas(string url) {
    double cantidadPaginas = 0;
    try {
        HttpClient cliente = new HttpClient();
        string requestUrl = UrlApi + url;
        cliente.DefaultRequestHeaders.Authorization = new System.Net.Http.Headers.AuthenticationHeaderValue("Bearer",
HttpContext.Session.GetString("Token"));
        var tarea = cliente.GetAsync(requestUrl);
        tarea.Wait();
        var respuesta = tarea.Result;

        HttpContent content = respuesta.Content;
        var tarea2 = content.ReadAsStringAsync();
        tarea2.Wait();
        string contenido = tarea2.Result;

        if (respuesta.IsSuccessStatusCode) {
            contenido = contenido.Replace(".", ",");
            double.TryParse(contenido, out cantidadPaginas);
        } else if ((int)respuesta.StatusCode == StatusCodes.Status400BadRequest || (int)respuesta.StatusCode ==
StatusCodes.Status500InternalServerError) {
            cantidadPaginas = -1;
        }
    } catch (Exception ex) {
        throw;
    }
    return cantidadPaginas;
}

}

//// File:  

C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Controllers\UsuariosController.cs  

using DataTransferObjects;  

using Microsoft.AspNetCore.Http;

```

```

using Microsoft.AspNetCore.Mvc;
using Newtonsoft.Json;

namespace ObligatorioMVC.Controllers {
    public class UsuariosController : Controller {
        public string UrlApi;

        public UsuariosController(IConfiguration config) {
            UrlApi = config.GetValue<string>("ApiUrl");
        }

        //-----
        //----- INGRESAR -----
        //

        public IActionResult Ingresar() {
            return View();
        }

        [HttpPost]
        public IActionResult Ingresar(UsuarioDTO usuarioDTO) {
            try {
                HttpClient client = new HttpClient();
                string url = UrlApi + "Usuarios/Ingresar";
                var tarea = client.PostAsJsonAsync(url, usuarioDTO);
                tarea.Wait();

                var respuesta = tarea.Result;
                HttpContent content = respuesta.Content;
                var tarea2 = content.ReadAsStringAsync();
                tarea2.Wait();
                string cuerpo = tarea2.Result;
                ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;

                if (respuesta.IsSuccessStatusCode) {
                    UsuarioAutenticadoDTO usuarioAutenticado = JsonConvert.DeserializeObject<UsuarioAutenticadoDTO>(cuerpo);

                    if (usuarioAutenticado != null) {
                        HttpContext.Session.SetString("Email", usuarioAutenticado.Email);
                        HttpContext.Session.SetString("Rol", usuarioAutenticado.Tipo);
                        HttpContext.Session.SetString("Token", usuarioAutenticado.Token);

                        return RedirectToAction("Index", "Home");
                    } else {
                        ViewBag.ErrorMsg = "Credenciales incorrectas";
                    }
                } else {
                    ViewBag.ErrorMsg = respuesta.Content.ReadAsStringAsync().Result;
                }
            } catch (Exception e) {
                ViewBag.ErrorMsg = e.Message;
            }

            return View();
        }

        public IActionResult CerrarSesion() {
            HttpContext.Session.Clear();
            return RedirectToAction("Index", "Home");
        }
    }
}

```

```

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Filters\PrivadoAttribute.cs
using System;
using System.Linq;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Filters;

namespace Obligatorio.Filters {
    public class PrivadoAttribute : Attribute, IActionFilter {
        public string TipoUsuarios { get; set; }

        public void OnActionExecuted(ActionExecutedContext context) {
        }

        public void OnActionExecuting(ActionExecutingContext context) {
            bool usuarioAutenticado = !string.IsNullOrEmpty(context.HttpContext.Session.GetString("Rol"));
            if (!usuarioAutenticado) {
                context.Result = new RedirectResult("/usuarios/ingresar");
            } else {
                if (!string.IsNullOrEmpty(TipoUsuarios)) {
                    if (!TipoUsuarios.Split(",").Contains(context.HttpContext.Session.GetString("Rol")))
                        context.Result = new RedirectResult("/home/error");
                }
            }
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Models\ErrorViewModel.cs
namespace ObligatorioMVC.Models {
    public class ErrorViewModel {
        public string? RequestId { get; set; }

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

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\_ViewImports.cshtml
@using ObligatorioMVC
@using ObligatorioMVC.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\_ViewStart.cshtml
@{
    Layout = "_Layout";
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Home\Index.cshtml
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://learn.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Home\Privacy.cshtml
@{
    ViewData["Title"] = "Privacy Policy";
}
<h1>@ViewData["Title"]</h1>

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

```

```

//// File:
C:\Users\caro0\Documents\ORT\TP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\MovimientosStock\BuscarPorArticuloYTipo.cshtml
@model IEnumerable<DataTransferObjects.MovimientoStockIndexDTO>

@{
    ViewData["Title"] = "Buscar Por Artículo Y Tipo de Movimiento";
}



# @ViewData["Title"]



Id del Artículo:



Tipo de Movimiento:



---


@if (Model != null) {
@if (Model.Count() > 0) {


| @Html.DisplayNameFor(model => model.Id) | @Html.DisplayNameFor(model => model.Fecha) | @Html.DisplayNameFor(model => model.ArticuloNombre) | @Html.DisplayNameFor(model => model.Cantidad) | @Html.DisplayNameFor(model => model.EmailUsuario) | @Html.DisplayNameFor(model => model.TipoMovimientoNombre) | @Html.DisplayNameFor(model => model.TipoMovimientoAccion) |
|-----------------------------------------|--------------------------------------------|-----------------------------------------------------|-----------------------------------------------|---------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|
|-----------------------------------------|--------------------------------------------|-----------------------------------------------------|-----------------------------------------------|---------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|


}
}

```

```

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Id)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Fecha)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ArticuloNombre)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Cantidad)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.EmailUsuario)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.TipoMovimientoNombre)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.TipoMovimientoAccion)
        </td>
    </tr>
}
</tbody>
</table>
} else {
    <span class="alert-danger">@ViewBag.ErrorMsg</span>
}
} else {
    <span class="alert-danger">@ViewBag.ErrorMsg</span>
}

@if (ViewBag.Paginas != null && ViewBag.Paginas > 1) {
    <ul class="pagination justify-content-center">
        @for (int i = 1; i <= ViewBag.Paginas; i++) {
            <li class="page-item">
                <a class="page-link" href="@Url.Action("BuscarPorArticuloYTipo", new { idArticulo = ViewBag.Articuloid, tipoMovimiento = ViewBag.TipoMov, page = i })">
                    @i
                </a>
            </li>
        }
    </ul>
}
//// File:  

C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\MovimientosStock\BuscarPorFec  

ha.cshtml
@model IEnumerable<DataTransferObjects.ArticuloDTO>

@{
    ViewData["Title"] = "Buscar Artículos Por Fecha de Movimiento";
}

<div class="text-center">
    <h1 class="display-5">@ViewData["Title"]</h1>
</div>

<div class="row justify-content-center">
    <div class="col-md-4">

```

```

<form asp-action="BuscarPorFecha">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <div class="form-group">
        <label class="control-label">Desde:</label>
        <input type="date" name="desde" class="form-control" value="@DateOnly.FromDateTime(DateTime.Today).ToString("yyyy-MM-dd")" />
    </div>
    <div class="form-group">
        <label class="control-label">Hasta:</label>
        <input type="date" name="hasta" class="form-control" value="@DateOnly.FromDateTime(DateTime.Today).ToString("yyyy-MM-dd")" />
    </div>
    <div class="form-group">
        <input type="submit" value="Buscar" class="btn btn-primary mt-2" />
    </div>
</form>
</div>
<hr />
@if (Model != null) {
    @if (Model.Count() > 0) {
        <table class="table">
            <thead>
                <tr>
                    <th>
                        @Html.DisplayNameFor(model => model.Id)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model => model.Nombre)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model => model.Descripcion)
                    </th>
                </tr>
            </thead>
            <tbody>
                @foreach (var item in Model) {
                    <tr>
                        <td>
                            @Html.DisplayFor(modelItem => item.Id)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem => item.Nombre)
                        </td>
                        <td>
                            @Html.DisplayFor(modelItem => item.Descripcion)
                        </td>
                    </tr>
                }
            </tbody>
        </table>
    } else {
        <span class="alert-danger">@ViewBag.ErrorMsg</span>
    }
} else {
    <span class="alert-danger">@ViewBag.ErrorMsg</span>
}

@if (ViewBag.Paginas != null && ViewBag.Paginas > 1) {
    <ul class="pagination justify-content-center">
        @for (int i = 1; i <= ViewBag.Paginas; i++) {
            <li class="page-item">
                <a class="page-link" href="@Url.Action("BuscarPorFecha", new { desde = ViewBag.Desde, hasta = ViewBag.Hasta, page = i })">

```

```

        @i
    </a>
</li>
}
</ul>
}

//// File:  

C:\Users\caro0\Documents\ORTP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\MovimientosStock\Create.cshtml  

@model DataTransferObjects.MovimientoStockDTO

{@
    ViewData["Title"] = "Crear Movimiento de Stock";
}

<div class="text-center">
    <h1 class="display-4">@ViewData["Title"]</h1>
</div>

<span class="alert-danger">@ViewBag.ErrorMsg</span>

<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Articuloid">Artículo</label>
                <select asp-for="Articuloid" class="form-select">
                    @foreach (var articulo in Model.Articulos) {
                        <option value="@articulo.Id">@articulo.Nombre</option>
                    }
                </select>
            </div>
            <div class="form-group">
                <label asp-for="TipoMovimientoid">Tipo De Movimiento</label>
                <select asp-for="TipoMovimientoid" class="form-select">
                    @foreach (var tm in Model.TiposMovimientos) {
                        <option value="@tm.Id">@tm.Nombre</option>
                    }
                </select>
            </div>
            <div class="form-group">
                <label asp-for="Cantidad" class="control-label"></label>
                <input asp-for="Cantidad" class="form-control" />
                <span asp-validation-for="Cantidad" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary mt-2" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

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

```

```

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\MovimientosStock\Index.cshtml
@model IEnumerable<DataTransferObjects.MovimientoStockIndexDTO>

@{
    ViewData["Title"] = "Movimientos de Stock";
}



# @ViewData["Title"]

@ViewBag.ErrorMsg

Crear Movimiento



Buscar Por Fecha



Buscar Por Artículo y Tipo de Movimiento



Resumen de Movimientos



| @Html.DisplayNameFor(model => model.Id) | @Html.DisplayNameFor(model => model.Fecha) | @Html.DisplayNameFor(model => model.ArticuloNombre) | @Html.DisplayNameFor(model => model.EmailUsuario) | @Html.DisplayNameFor(model => model.TipoMovimientoNombre) | @Html.DisplayNameFor(model => model.Cantidad) |
|-----------------------------------------|--------------------------------------------|-----------------------------------------------------|---------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------|
| @Html.DisplayFor(modelItem => item.Id)  |                                            |                                                     |                                                   |                                                           |                                               |


```

```

<td>
    @Html.DisplayFor(modelItem => item.Fecha)
</td>
<td>
    @Html.DisplayFor(modelItem => item.ArticuloNombre)
</td>
<td>
    @Html.DisplayFor(modelItem => item.EmailUsuario)
</td>
<td>
    @Html.DisplayFor(modelItem => item.TipoMovimientoNombre)
</td>
<td>
    @Html.DisplayFor(modelItem => item.Cantidad)
</td>
</tr>
}
</tbody>
</table>
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\MovimientosStock\ObtenerResumen.cshtml
@model IEnumerable<DataTransferObjects.MovimientoStockResumenDTO>

@{
    ViewData["Title"] = "Movimientos de Stock por Año";
}

<div class="text-center">
    <h1 class="display-5">@ViewData["Title"]</h1>
</div>

<span class="alert-danger">@ViewBag.ErrorMsg</span>

<hr />

@foreach (var item in Model) {
    <div class="card col-md-4 mt-3">
        <div class="card-header">
            <b>
                Año: @Html.DisplayFor(modelItem => item.Ano)
            </b>
        </div>
        <ul class="list-group list-group-flush">
            @foreach (var tm in item.TipoCantidad) {
                <li class="list-group-item">@Html.DisplayFor(t => tm)</li>
            }
        </ul>
        <div class="card-footer">
            <b>
                Total año: @Html.DisplayFor(modelItem => item.CantidadTotalAnio)
            </b>
        </div>
    </div>
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Shared\Error.cshtml
@model ErrorViewModel
@{
    ViewData["Title"] = "Error";
}

```

```

}

<h1 class="text-danger">Error.</h1>
<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)
{
    <p>
        <strong>Request ID:</strong> <code>@Model.RequestId</code>
    </p>
}

<h3>Development Mode</h3>
<p>
    Swapping to <strong>Development</strong> environment will display more detailed information about the error that occurred.
</p>
<p>
    <strong>The Development environment shouldn't be enabled for deployed applications.</strong>
    It can result in displaying sensitive information from exceptions to end users.
    For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE_ENVIRONMENT</strong>
    environment variable to <strong>Development</strong>
    and restarting the app.
</p>
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Shared\_Layout.cshtml
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - ObligatorioMVC</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="~/ObligatorioMVC.styles.css" asp-append-version="true" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
            <partial name="_Menu.cshtml" />
        </nav>
    </header>
    <div class="container">
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>

    <footer class="border-top footer text-muted">
        <div class="container">
            © 2024 - ObligatorioMVC
        </div>
    </footer>
    <script src="~/lib/jquery/dist/jquery.min.js"></script>
    <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
    <script src="~/js/site.js" asp-append-version="true"></script>
    @await RenderSectionAsync("Scripts", required: false)
</body>
</html>
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Shared\_Menu.cshtml
@using Microsoft.AspNetCore.Http;

<div class="container">
```

```

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index"> Obligatorio</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="navbar-collapse collapse d-sm-inline-flex">
    <ul class="navbar-nav flex-grow-1">
        <li class="nav-item">
            <a class="nav-link" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
        </li>

        <!-- Menú de Administrador -->
        @if (Context.Session.GetString("Rol") == "Encargado") {
            <li class="nav-item">
                <a class="nav-link" asp-controller="MovimientosStock" asp-action="Index">Movimientos de Stock</a>
            </li>
        }
    </ul>

    <!-- Menú de Sesión -->
    <ul class="navbar-nav flex-grow-1 justify-content-end">
        @if (Context.Session.GetString("Rol") == null) {
            <li class="nav-item">
                <a class="nav-link text-dark" asp-controller="Usuarios" asp-action="Ingresar">Ingresar</a>
            </li>
        } else {
            <li class="nav-item">
                <a class="nav-link" asp-controller="Usuarios" asp-action="CerrarSesion">Cerrar Sesión</a>
            </li>
        }
    </ul>
</div>
</div>
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Shared\_ValidationScriptsPartial.cshtml
<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>
<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_MVC\ObligatorioMVC\Views\Usuarios\Ingresar.cshtml
@model DataTransferObjects.UsuarioDTO
 @{
    ViewData["Title"] = "Ingresar";
}

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

<div class="row">
    <div class="col-md-4">
        <form asp-action="Ingresar">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Email" class="control-label"></label>
                <input asp-for="Email" class="form-control" />
                <span asp-validation-for="Email" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Password" class="control-label"></label>
                <input asp-for="Password" class="form-control" />
                <span asp-validation-for="Password" class="text-danger"></span>
            </div>
        </form>
    </div>
</div>

```

```
<div class="form-group">
    <input type="submit" value="Iniciar sesión" class="btn btn-primary mt-2" />
</div>

<span class="alert-danger">@ViewBag.Error</span>
</form>
</div>
</div>
```

Solucion WebApi

```
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\ArticuloDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class ArticuloDTO {
        public int Id { get; set; }
        public string Nombre { get; set; }
        public string Descripcion { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\ClienteDTO.cs
using LogicaNegocio.ValueObjects;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class ClienteDTO {
        public int Id { get; set; }
        public string RazonSocial { get; set; }
        public string Rut { get; set; }
        public string Direccion { get; set; }
        public int? DistanciaHastaDeposito { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MapperArticulos.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MapperArticulos {
        public static List<ArticuloDTO> ToListDto(List<Articulo> articulos) {
            return articulos.Select(a => new ArticuloDTO() {
                Id = a.Id,
                Nombre = a.Nombre,
                Descripcion = a.Descripcion
            })
            .ToList();
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MapperClientes.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```

using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MapperClientes {
        public static List<ClienteDTO> ToListDto(List<Cliente> clientes) {
            return clientes.Select(c => new ClienteDTO() {
                Id = c.Id,
                RazonSocial = c.RazonSocial,
                Rut = c.Rut,
                DistanciaHastaDeposito = c.DistanciaHastaDeposito,
                Direccion = c.Direccion.Calle.Valor + " " + c.Direccion.NumeroPuerta.Valor + ", " + c.Direccion.Ciudad.Valor
            })
            .ToList();
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MapperPedidos.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MapperPedidos {
        public static Pedido CrearEntidad(PedidoDTO pedidoDTO) {
            Pedido p;

            if (pedidoDTO.TipoPedido == "PedidoExpress") {
                p = new PedidoExpress();
            } else {
                p = new PedidoComun();
            }

            p.FechaEntrega = pedidoDTO.FechaEntrega;
            p.Estado = pedidoDTO.Estado;

            return p;
        }

        public static PedidoDTO CrearDTO(Pedido pedido) {
            PedidoDTO pedidoDTO = new PedidoDTO();

            pedidoDTO.Id = pedido.Id;
            pedidoDTO.FechaEntrega = pedido.FechaEntrega;
            pedidoDTO.Estado = pedido.Estado;

            return pedidoDTO;
        }

        public static List<PedidoNoEntregadoDTO> ToListDto(List<Pedido> pedidos) {
            return pedidos.Select(p => new PedidoNoEntregadoDTO() {
                Id = p.Id,
                FechaEntrega = p.FechaEntrega,
                RazonSocialCliente = p.Cliente.RazonSocial,
                Recargo = p.Recargo,
                Iva = p.Iva,
                Total = p.Total
            })
            .ToList();
        }
    }
}

```

```

        }
    }

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MapperUsuarios.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MapperUsuarios {
        public static UsuarioDTO CrearDTO(Usuario usuario) {
            UsuarioDTO usuarioDTO = new UsuarioDTO();

            usuarioDTO.Email = usuario.Email;
            usuarioDTO.Password = usuario.Contraseña;
            usuarioDTO.Tipo = usuario.Tipo.ToString();

            return usuarioDTO;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MovimientoCantidadPorAnioYTi
poDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MovimientoCantidadPorAnioYTipoDTO {
        public int Cantidad { get; set; }
        public int Anio { get; set; }
        public string TipoMovimiento { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MovimientoStockDTO.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class MovimientoStockDTO {
        public int Id { get; set; }
        public DateTime Fecha { get; set; }
        public int Articuloid { get; set; }
        public string EmailUsuario { get; set; }
        public int TipMovimientoid { get; set; }
        public int Cantidad { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\MovimientoStockIndexDTO.cs

```

```

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

namespace DataTransferObjects {
    public class MovimientoStockIndexDTO {
        public int Id { get; set; }
        public DateTime Fecha { get; set; }
        public string ArticuloNombre { get; set; }
        public string EmailUsuario { get; set; }
        public string TipoMovimientoNombre { get; set; }
        public int Cantidad { get; set; }
        public string? TipoMovimientoAccion { get; set; } // RF-04-a
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\PedidoDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class PedidoDTO {
        public int Id { get; set; }
        public int IdCliente { get; set; }
        public int IdArticulo { get; set; }
        public int Cantidad { get; set; }
        public DateOnly FechaEntrega { get; set; }
        public string TipoPedido { get; set; }
        public string Estado { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\PedidoNoEntregadoDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class PedidoNoEntregadoDTO {
        // DTO usado para:
        // Listado de pedidos que se pueden anular (Estado == "Pendiente")
        // Listado de pedidos anulados (Estado == "Anulado")
        public int Id { get; set; }
        public DateOnly FechaEntrega { get; set; }
        public string RazonSocialCliente { get; set; }
        public decimal Iva { get; set; }
        public decimal Recargo { get; set; }
        public decimal Total { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\TipoMovimientoDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;

namespace DataTransferObjects {
    public class TipoMovimientoDTO {
        public int Id { get; set; }
        public string Nombre { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\UsuarioAutenticadoDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class UsuarioAutenticadoDTO {
        public string Email { get; set; }
        public string Token { get; set; }
        public string Tipo { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\UsuarioDTO.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTransferObjects {
    public class UsuarioDTO {
        public string Email { get; set; }
        public string Password { get; set; }
        public string? Tipo { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\obj\Debug\net8.0\DataTransfer
Objects.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("DataTransferObjects")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+16d075e4bdc1af1f44ce24eefe9cb9e2b59eac12")]
[assembly: System.Reflection.AssemblyProductAttribute("DataTransferObjects")]
[assembly: System.Reflection.AssemblyTitleAttribute("DataTransferObjects")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\obj\Debug\net8.0\DataTransfer
Objects.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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\obj\Release\net8.0\.NETCoreAp
p, 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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\obj\Release\net8.0\DataTransfe
rObjects.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("DataTransferObjects")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+f9761fd17d5acfdc29824f17f329c90bf5a7c8da")]
[assembly: System.Reflection.AssemblyProductAttribute("DataTransferObjects")]
[assembly: System.Reflection.AssemblyTitleAttribute("DataTransferObjects")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\DataTransferObjects\obj\Release\net8.0\DataTransfe
rObjects.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;
//// File:
C:\Users\caro0\Documents\ORT\TP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAregarArticuloEnPedido.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUAregarArticuloEnPedido : ICUModificar<PedidoDTO> {
        public IRepositorioPedidos Repo { get; set; }
        public IRepositorioLineas RepoLineas { get; set; }
        public IRepositorioArticulos RepoArticulos { get; set; }
        public CUAregarArticuloEnPedido(IRepositorioPedidos repo, IRepositorioArticulos repositorioArticulos, IRepositorioLineas repoLineas) {
            Repo = repo;
            RepoArticulos = repositorioArticulos;
            RepoLineas = repoLineas;
        }
        public void Modificar(PedidoDTO obj) {
            Pedido aModificar = Repo.FindById(obj.Id);
            Articulo articulo = RepoArticulos.FindById(obj.IdArticulo);

            if (articulo != null) {
                // chequear stock
                if (ChequearStock(articulo, obj.Cantidad)) {
                    // crear la Linea
                    Linea nuevaLinea = new Linea();
                    nuevaLinea.Articulo = articulo;
                    nuevaLinea.PrecioUnitario = articulo.Precio;
                    nuevaLinea.UnidadesSolicitadas = obj.Cantidad;
                    RepoLineas.Create(nuevaLinea);

                    // agregarla al pedido
                    aModificar.Lineas.Add(nuevaLinea);
                    aModificar.Total = CalcularTotal(aModificar, aModificar.Lineas);
                } else {
                    throw new NoStockException("No hay suficiente stock del artículo seleccionado");
                }
            } else {
                throw new RegistroNoExisteException("El artículo seleccionado para el pedido no existe");
            }

            Repo.Update(aModificar);
        }

        public static bool ChequearStock(Articulo articulo, int cantidadSolicitada) {
            if (articulo.Stock >= cantidadSolicitada) {
                return true;
            } else {
                return false;
            }
        }
    }
}

```

```

    }

    public static decimal CalcularTotal(Pedido p, List<Linea> lineas) {
        decimal resultado = 0;
        decimal montoArticulos = 0;
        decimal iva = p.Iva + 1;
        decimal recargo = p.Recargo + 1;

        foreach (Linea linea in lineas) {
            montoArticulos += linea.PrecioUnitario * linea.UnidadesSolicitadas;
        }

        resultado = montoArticulos * iva * recargo;

        return resultado;
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAltaArticulo.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUAltaArticulo : ICUAlta<Articulo> {
        public IRepositorioArticulos Repo { get; set; }
        public CUAltaArticulo(IRepositorioArticulos repo) {
            Repo = repo;
        }
        public void Alta(Articulo obj) {
            Repo.Create(obj);
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAltaCliente.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUAltaCliente : ICUAlta<Cliente> {
        public IRepositorioClientes Repo { get; set; }
        public CUAltaCliente(IRepositorioClientes repo) {
            Repo = repo;
        }
        public void Alta(Cliente obj) {
            Repo.Create(obj);
        }
    }
}

```

```

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAltaMovimientoStock.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUAltaMovimientoStock : ICUAlta<MovimientoStockDTO> {
        public IRepositoryMovimientosStock Repo { get; set; }
        public IRepositoryArticulos RepoArticulos { get; set; }
        public IRepositoryUsuarios RepoUsuarios { get; set; }
        public IRepositoryTiposMovimientos RepoTiposMovimientos { get; set; }
        public IRepositoryParametros RepoParametros { get; set; }
        public CUAltaMovimientoStock(
            IRepositoryMovimientosStock repo,
            IRepositoryArticulos repoArticulos,
            IRepositoryTiposMovimientos repoTiposMovimientos,
            IRepositoryUsuarios repoUsuarios,
            IRepositoryParametros repoParametros
        ) {
            Repo = repo;
            RepoArticulos = repoArticulos;
            RepoTiposMovimientos = repoTiposMovimientos;
            RepoUsuarios = repoUsuarios;
            RepoParametros = repoParametros;
        }

        public void Alta(MovimientoStockDTO obj) {
            MovimientoStock nuevoMovimiento = new MovimientoStock();

            // obtener el límite de cantidad
            int limiteCantidad = (int) RepoParametros.ObtenerLimiteTopeDeMovimientos();

            // chequear que cantidad < parametro
            if (obj.Cantidad <= limiteCantidad) {
                Articulo articulo = RepoArticulos.FindById(obj.Articuloid);
                TipoMovimiento tm = RepoTiposMovimientos.FindById(obj.TipoMovimientoid);
                Usuario u = RepoUsuarios.BuscarPorEmail(obj.EmailUsuario);

                nuevoMovimiento.Articulo = articulo;
                nuevoMovimiento.TipoMovimiento = tm;
                nuevoMovimiento.Usuario = u;
                nuevoMovimiento.Cantidad = obj.Cantidad;

                Repo.Create(nuevoMovimiento);
                obj.Id = nuevoMovimiento.Id;
            } else {
                throw new DatosInvalidosException("La cantidad debe ser menor o igual a " + limiteCantidad);
            }
        }
    }
}

```

```

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAltaPedido.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUAltaPedido : ICUAlta<PedidoDTO> {
        public IRepositorioPedidos Repo { get; set; }
        public IRepositorioClientes RepoClientes { get; set; }
        public IRepositorioArticulos RepoArticulos { get; set; }
        public IRepositorioLineas RepoLineas { get; set; }
        public IRepositorioParametros RepoParametros { get; set; }

        public CUAltaPedido(IRepositorioPedidos repo, IRepositorioClientes repoClientes, IRepositorioArticulos repoArticulos, IRepositorioLineas
repoLineas, IRepositorioParametros repoParametros) {
            Repo = repo;
            RepoClientes = repoClientes;
            RepoArticulos = repoArticulos;
            RepoLineas = repoLineas;
            RepoParametros = repoParametros;
        }

        public void Alta(PedidoDTO obj) {
            Pedido nuevoPedido = MapperPedidos.CrearEntidad(obj);
            // TODO popular al ejecutar la app
            Pedido._iva = RepoParametros.ObtenerIva();
            Cliente cliente = RepoClientes.FindById(obj.IdCliente);
            Articulo articulo = RepoArticulos.FindById(obj.IdArticulo);
            if (cliente != null) {
                nuevoPedido.Cliente = cliente;
            } else {
                throw new RegistroNoExisteException("El cliente seleccionado para el pedido no existe");
            }

            // TODO RepoParametros.ObtenerRecargo(obj.TipoPedido);
            if (obj.TipoPedido == "PedidoExpress") {
                nuevoPedido.CalcularRecargo(0.15m, 0.1m);
            } else {
                nuevoPedido.CalcularRecargo(0.05m, 0);
            }

            if (articulo != null) {
                // chequear stock
                if (ChequearStock(articulo, obj.Cantidad)) {
                    // crear la Linea
                    Linea nuevaLinea = new Linea();
                    nuevaLinea.Articulo = articulo;
                    nuevaLinea.PrecioUnitario = articulo.Precio;
                    nuevaLinea.UnidadesSolicitadas = obj.Cantidad;
                    RepoLineas.Create(nuevaLinea);

                    // agregarla al pedido
                }
            }
        }
    }
}

```

```

        nuevoPedido.Lineas = new List<Linea> { nuevaLinea };
        nuevoPedido.Total = CalcularTotal(nuevoPedido, nuevoPedido.Lineas);
    } else {
        throw new NoStockException("No hay suficiente stock del artículo seleccionado");
    }
} else {
    throw new RegistroNoExisteException("El artículo seleccionado para el pedido no existe");
}

Repo.Create(nuevoPedido);
}

public static bool ChequearStock(Articulo articulo, int cantidadSolicitada) {
    if(articulo.Stock >= cantidadSolicitada) {
        return true;
    } else {
        return false;
    }
}

public static decimal CalcularTotal(Pedido p, List<Linea> lineas) {
    decimal resultado = 0;
    decimal montoArticulos = 0;
    decimal iva = p.Iva + 1;
    decimal recargo = p.Recargo + 1;

    foreach ( Linea linea in lineas ) {
        montoArticulos += linea.PrecioUnitario * linea.UnidadesSolicitadas;
    }

    resultado = montoArticulos * iva * recargo;

    return resultado;
}
}

//// File:  

C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAltaTipoMovimiento.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;

namespace LogicaAplicacion.CasosUso {
    public class CUAltaTipoMovimiento : CUAlta<TipoMovimientoDTO> {
        public IRepositoryTiposMovimientos Repo { get; set; }

        public CUAltaTipoMovimiento(IRepositoryTiposMovimientos repo) {
            Repo = repo;
        }

        public void Alta(TipoMovimientoDTO obj) {
            TipoMovimiento nuevo = new TipoMovimiento();
            nuevo.Id = obj.Id;
            nuevo.Nombre = obj.Nombre;

            Repo.Create(nuevo);
            obj.Id = nuevo.Id;
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\TP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAltaUsuario.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUAltaUsuario : ICUAlta<Usuario>{

        public IRepositorioUsuarios Repo { get; set; }
        public CUAltaUsuario(IRepositorioUsuarios repo) {
            Repo = repo;
        }
        public void Alta(Usuario obj) {
            byte[] key = EncryptionUtility.GenerateRandomKey();
            byte[] iv = EncryptionUtility.GenerateRandomIV();
            obj.ContraseniaEncriptada = EncryptionUtility.EncryptString(obj.Contraseña, key, iv);
            Repo.Create(obj);
        }
    }
}

public class EncryptionUtility {
    public static byte[] GenerateRandomKey() {
        using var aes = Aes.Create();
        aes.GenerateKey();
        return aes.Key;
    }

    public static byte[] GenerateRandomIV() {
        using var aes = Aes.Create();
        aes.GenerateIV();
        return aes.IV;
    }

    public static string EncryptString(string plainText, byte[] key, byte[] iv) {
        using var aes = Aes.Create();
        aes.Key = key;
        aes.IV = iv;

        var encryptor = aes.CreateEncryptor(aes.Key, aes.IV);
        using var msEncrypt = new MemoryStream();
        using (var csEncrypt = new CryptoStream(msEncrypt, encryptor, CryptoStreamMode.Write)) {
            using (var swEncrypt = new StreamWriter(csEncrypt)) {
                swEncrypt.Write(plainText);
            }
        }
        return Convert.ToString(msEncrypt.ToArray());
    }
}
//// File:
C:\Users\caro0\Documents\ORT\TP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAnularPedido.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.InterfacesRepositorios;

```

```

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

namespace LogicaAplicacion.CasosUso {
    public class CUAnularPedido : ICUAnularPedido {
        public IRepositoryPedidos Repo { get; set; }
        public CUAnularPedido(IRepositoryPedidos repo) {
            Repo = repo;
        }
        public void Anular(int id) {
            Repo.AnularPedido(id);
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUAutenticarUsuario.cs

using LogicaNegocio.Dominio;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DataTransferObjects;
using LogicaNegocio.Excepciones;

namespace LogicaAplicacion.CasosUso {
    public class CUAutenticarUsuario : ICUAutenticarUsuario {
        public IRepositoryUsuarios Repo { get; set; }

        public CUAutenticarUsuario(IRepositoryUsuarios repo) {
            Repo = repo;
        }

        public UsuarioDTO Autenticar(string email, string contraseña) {

            try {
                Usuario u = Repo.BuscarPorEmail(email, contraseña);
                return MapperUsuarios.CrearDTO(u);
            } catch (Exception e) {
                throw new DatosInvalidosException("Las credenciales son incorrectas");
            }
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBajaArticulo.cs

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

namespace LogicaAplicacion.CasosUso {

```

```

public class CUBajaArticulo : ICUBaja<Articulo> {
    public IRepositoryArticulos Repo { get; set; }
    public CUBajaArticulo(IRepositoryArticulos repo) {
        Repo = repo;
    }
    public void Baja(int id) {
        Repo.Delete(id);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBajaTipoMovimiento.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;

namespace LogicaAplicacion.CasosUso {
    public class CUBajaTipoMovimiento : ICUBaja<TipoMovimientoDTO> {
        public IRepositoryTiposMovimientos Repo { get; set; }
        public IRepositoryMovimientosStock RepoMovStock { get; set; }

        public CUBajaTipoMovimiento(IRepositoryTiposMovimientos repo, IRepositoryMovimientosStock repoMovStock) {
            Repo = repo;
            RepoMovStock = repoMovStock;
        }

        public void Baja(int id) {
            //chequear que el tipo no este siendo usado en ningun movimiento
            if (!RepoMovStock.TieneTipoMovimiento(id)) {
                Repo.Delete(id);
            } else {
                throw new DatosInvalidosException("El tipo de movimiento no puede ser eliminado porque tiene un Movimiento de Stock asociado.");
            }
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBajaUsuario.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBajaUsuario : ICUBaja<Usuario> {
        public IRepositoryUsuarios Repo { get; set; }
        public CUBajaUsuario(IRepositoryUsuarios repo) {
            Repo = repo;
        }

        public void Baja(int id) {
            Repo.Delete(id);
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarClientesPorMo
ntoPedido.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarClientesPorMontoPedido : ICUBuscarClientesPorMontoPedido {
        public IRepositoryPedidos Repo { get; set; }
        public CUBuscarClientesPorMontoPedido(IRepositoryPedidos repo) {
            Repo = repo;
        }
        public List<ClienteDTO> BuscarPorMontoPedido(decimal monto) {
            List<Cliente> clientesEncontrados = Repo.BuscarClientes(monto);
            if (clientesEncontrados.Count > 0) {
                List<ClienteDTO> clientesDTO = MapperClientes.ToListDto(clientesEncontrados);
                return clientesDTO;
            } else {
                throw new RegistroNoExisteException("No existen clientes que hayan realizado pedidos con un total mayor a ese monto.");
            }
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorArticuloYTi
poMovimiento.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorArticuloYTipoMovimiento : ICUBuscarPorArticuloYTipoMovimiento {
        public IRepositoryMovimientosStock Repo { get; set; }
        public IRepositoryParametros RepoParametros { get; set; }
        public CUBuscarPorArticuloYTipoMovimiento(IRepositoryMovimientosStock repo, IRepositoryParametros repoParametros) {
            Repo = repo;
            RepoParametros = repoParametros;
        }
        public List<MovimientoStockIndexDTO> BuscarMovimientosPorArticuloYTipo(int idArticulo, string tipoMovimiento, int pagina) {
            List<MovimientoStockIndexDTO> dtos = null;
            // obtener el límite de records por página establecido en Parametros
            int limitePorPagina = (int)RepoParametros.ObtenerLimitePorPagina();
            List<MovimientoStock> msEncontrados = Repo.BuscarMovimientosPorArticuloYTipo(idArticulo, tipoMovimiento, pagina, limitePorPagina);
        }
    }
}

```

```

if (msEncontrados.Count > 0) {
    // aplicar paginado al resultado de la query
    // tdo evaluar aplicar a nivel de repo xa no traer todos los records de la base
    //List<MovimientoStock> dtosPorPagina = msEncontrados.Skip((pagina - 1) * limitePorPagina).Take(limitePorPagina).ToList();

    dtos = msEncontrados.Select(ms => new MovimientoStockIndexDTO() {
        Id = ms.Id,
        Fecha = ms.Fecha,
        Cantidad = ms.Cantidad,
        ArticuloNombre = ms.Articulo.Nombre,
        EmailUsuario = ms.Usuario.Email,
        TipoMovimientoNombre = ms.TipoMovimiento.Nombre,
        TipoMovimientoAccion = ms.TipoMovimiento.TipoAccion.ToString()
    })
    .ToList();
}

return dtos;
}
}

}

//// File:  

C:\Users\caro\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorFechaMovimiento.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorFechaMovimiento : ICUBuscarPorFechaMovimiento {
        public IRepositoryMovimientosStock Repo { get; set; }
        public IRepositoryParametros RepoParametros { get; set; }
        public CUBuscarPorFechaMovimiento(IRepositoryMovimientosStock repo, IRepositoryParametros repoParametros) {
            Repo = repo;
            RepoParametros = repoParametros;
        }

        public List<ArticuloDTO> BuscarPorFecha(DateTime desde, DateTime hasta, int pagina) {
            List<ArticuloDTO> dtos = null;

            // obtener el límite de records por página establecido en Parametros
            int limitePorPagina = (int)RepoParametros.ObtenerLimitePorPagina();

            List<Articulo> articulosEncontrados = Repo.BuscarArticulosConMovimientosPorFecha(desde, hasta, pagina, limitePorPagina);

            if (articulosEncontrados.Count > 0) {
                dtos = MapperArticulos.ToListDto(articulosEncontrados);
            }

            return dtos;
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorFechaPedi
do.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using LogicaNegocio.Excepciones;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorFechaPedido : ICUBuscarPorFechaPedido {
        public IRepositoryPedidos Repo { get; set; }
        public CUBuscarPorFechaPedido(IRepositoryPedidos repo) {
            Repo = repo;
        }
        public List<PedidoNoEntregadoDTO> BuscarPorFechaPedido(DateOnly fecha) {
            List<PedidoNoEntregadoDTO> dtos = new List<PedidoNoEntregadoDTO>();
            List<Pedido> pedidosEncontrados = Repo.BuscarPorFechaDeEmision(fecha);
            if (pedidosEncontrados.Count > 0) {
                dtos = MapperPedidos.ToListDto(pedidosEncontrados);
            }
            //else {
            //    throw new RegistroNoExisteException("No existen pedidos para la fecha solicitada");
            //}
            return dtos;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorIdArticulo.c
s
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorIdArticulo : ICUBuscarPorId<Articulo> {
        public IRepositoryArticulos Repo { get; set; }
        public CUBuscarPorIdArticulo(IRepositoryArticulos repo) {
            Repo = repo;
        }
        public Articulo BuscarPorId(int id) {
            return Repo.FindById(id);
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorIdMovimie
ntoStock.cs
using DataTransferObjects;

```

```

using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorIdMovimientoStock : ICUBuscarPorId<MovimientoStockDTO> {
        public IRepositoryMovimientosStock Repo { get; set; }
        public CUBuscarPorIdMovimientoStock(IRepositoryMovimientosStock repo) {
            Repo = repo;
        }

        public MovimientoStockDTO BuscarPorId(int id) {
            MovimientoStockDTO dto = new MovimientoStockDTO();
            MovimientoStock msEncontrado = Repo.FindById(id);

            if (msEncontrado != null) {
                dto.Id = msEncontrado.Id;
                dto.Fecha = msEncontrado.Fecha;
            } else {
                throw new RegistroNoExisteException("El tipo de movimiento no existe");
            }

            return dto;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorIdPedido.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorIdPedido : ICUBuscarPorId<PedidoDTO> {
        public IRepositoryPedidos Repo { get; set; }

        public CUBuscarPorIdPedido(IRepositoryPedidos repo) {
            Repo = repo;
        }

        public PedidoDTO BuscarPorId(int id) {
            PedidoDTO dto = new PedidoDTO();
            Pedido pedidosEncontrados = Repo.FindById(id);
            if (pedidosEncontrados != null) {
                dto = MapperPedidos.CrearDTO(pedidosEncontrados);
            } else {
                throw new RegistroNoExisteException("El pedido no existe");
            }
        }
    }
}

```

```

        }
        return dto;
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorIdTipoMovimiento.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorIdTipoMovimiento : ICUBuscarPorId<TipoMovimientoDTO> {
        public IRepositoryTiposMovimientos Repo { get; set; }
        public CUBuscarPorIdTipoMovimiento(IRepositoryTiposMovimientos repo) {
            Repo = repo;
        }
        public TipoMovimientoDTO BuscarPorId(int id) {
            TipoMovimientoDTO dto = new TipoMovimientoDTO();
            TipoMovimiento tmEncontrado = Repo.FindById(id);

            if (tmEncontrado != null) {
                dto.Id = tmEncontrado.Id;
                dto.Nombre = tmEncontrado.Nombre;
            } else {
                throw new RegistroNoExisteException("El tipo de movimiento no existe");
            }

            return dto;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorIdUsuario.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorIdUsuario : ICUBuscarPorId<Usuario> {
        public IRepositoryUsuarios Repo { get; set; }
        public CUBuscarPorIdUsuario(IRepositoryUsuarios repo) {
            Repo = repo;
        }
        public Usuario BuscarPorId(int id) {
            return Repo.FindById(id);
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUBuscarPorRazonSocial.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using LogicaAplicacion.InterfacesCasosUso;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DataTransferObjects;

namespace LogicaAplicacion.CasosUso {
    public class CUBuscarPorRazonSocial : ICUBuscarPorRazonSocial {
        public IRepositoryClientes Repo { get; set; }
        public CUBuscarPorRazonSocial(IRepositoryClientes repo) {
            Repo = repo;
        }
        public List<ClienteDTO> BuscarPorRazonSocial(string nombre) {
            List<Cliente> clientes = Repo.BuscarPorRazonSocial(nombre);
            List<ClienteDTO> clientesDTO = MapperClientes.ToListDto(clientes);
            return clientesDTO;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUCantidadDePaginas.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUCantidadDePaginas : ICUCantidadDePaginas {
        public IRepositoryMovimientosStock Repo { get; set; }
        public IRepositoryParametros RepoParametros { get; set; }
        public CUCantidadDePaginas(IRepositoryMovimientosStock repo, IRepositoryParametros repoParametros) {
            Repo = repo;
            RepoParametros = repoParametros;
        }

        public double ObtenerCantidadDePaginas(int idArticulo, string tipoMovimiento) {
            // obtener el límite de records por página establecido en Parametros
            int limitePorPagina = (int)RepoParametros.ObtenerLimitePorPagina();

            // obtener cantidad de records que devolvio la consulta
            int totalQuery = Repo.CantidadDeMovimientos(idArticulo, tipoMovimiento);

            return Math.Ceiling( (double) totalQuery / (double) limitePorPagina);
        }

        public double ObtenerCantidadDePaginas(string desde, string hasta) {
            // obtener el límite de records por página establecido en Parametros
            int limitePorPagina = (int)RepoParametros.ObtenerLimitePorPagina();

            // convertir a fechas
        }
    }
}

```

```

        DateTime fechaDesde = DateTime.Parse(desde);
        DateTime fechaHasta = DateTime.Parse(hasta);

        // obtener cantidad de records que devolvio la consulta
        int totalQuery = Repo.CantidadDeMovimientos(fechaDesde, fechaHasta);

        return Math.Ceiling((double)totalQuery / (double)limitePorPagina);
    }
}

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoArticulos.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoArticulos : ICUListado<Articulo> {
        public IRepositoryArticulos Repo { get; set; }

        public CUListadoArticulos(IRepositoryArticulos repo) {
            Repo = repo;
        }

        public List<Articulo> ObtenerListado() {
            return Repo.GetAll();
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoClientes.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoClientes : ICUListado<ClienteDTO> {
        public IRepositoryClientes Repo { get; set; }

        public CUListadoClientes(IRepositoryClientes repo) {
            Repo = repo;
        }

        public List<ClienteDTO> ObtenerListado() {
            List<Cliente> clientes = Repo.GetAll();
            List<ClienteDTO> clientesDTO = MapperClientes.ToListDto(clientes);
            return clientesDTO;
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoMovimientosStock.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoMovimientosStock : ICUListado<MovimientoStockIndexDTO> {
        public IRepositoryMovimientosStock Repo { get; set; }
        public CUListadoMovimientosStock(IRepositoryMovimientosStock repo) {
            Repo = repo;
        }
        public List<MovimientoStockIndexDTO> ObtenerListado() {
            List<MovimientoStockIndexDTO> dtos = new List<MovimientoStockIndexDTO>();
            List<MovimientoStock> tmEncontrados = Repo.GetAll();

            if (tmEncontrados.Count > 0) {
                dtos = tmEncontrados.Select(ms => new MovimientoStockIndexDTO() {
                    Id = ms.Id,
                    Fecha = ms.Fecha,
                    Cantidad = ms.Cantidad,
                    ArticuloNombre = ms.Articulo.Nombre,
                    EmailUsuario = ms.Usuario.Email,
                    TipoMovimientoNombre = ms.TipoMovimiento.Nombre,
                })
                .ToList();
            }
            return dtos;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoPedidos.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoPedidos : ICUListado<Pedido> {
        public IRepositoryPedidos Repo { get; set; }

        public CUListadoPedidos(IRepositoryPedidos repo) {
            Repo = repo;
        }
        public List<Pedido> ObtenerListado() {
            return Repo.GetAll();
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoPedidosPendientes.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoPedidosPendientes : ICUListado<PedidoNoEntregadoDTO> {
        public IRepositoryPedidos Repo { get; set; }

        public CUListadoPedidosPendientes(IRepositoryPedidos repo) {
            Repo = repo;
        }
        public List<PedidoNoEntregadoDTO> ObtenerListado() {
            List<PedidoNoEntregadoDTO> dtos = new List<PedidoNoEntregadoDTO>();
            List<Pedido> pedidosEncontrados = Repo.ListarPedidosPendientes();
            if (pedidosEncontrados.Count > 0) {
                dtos = MapperPedidos.ToListDto(pedidosEncontrados);
            }
            return dtos;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoTiposMovimientos.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoTiposMovimientos : ICUListado<TipoMovimientoDTO> {
        public IRepositoryTiposMovimientos Repo { get; set; }

        public CUListadoTiposMovimientos(IRepositoryTiposMovimientos repo) {
            Repo = repo;
        }
        public List<TipoMovimientoDTO> ObtenerListado() {
            List<TipoMovimientoDTO> dtos = new List<TipoMovimientoDTO>();
            List<TipoMovimiento> tmEncontrados = Repo.GetAll();
            if (tmEncontrados.Count > 0) {

                dtos = tmEncontrados.Select(t => new TipoMovimientoDTO() {
                    Id = t.Id,
                    Nombre = t.Nombre
                })
                .ToList();
            }
            return dtos;
        }
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUListadoUsuarios.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUListadoUsuarios : ICUListado<Usuario> {
        public IRepositoryUsuarios Repo { get; set; }

        public CUListadoUsuarios(IRepositoryUsuarios repo) {
            Repo = repo;
        }
        public List<Usuario> ObtenerListado() {
            return Repo.GetAll();
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUModificarArticulo.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUModificarArticulo : ICUModificar<Articulo> {
        public IRepositoryArticulos Repo { get; set; }

        public CUModificarArticulo(IRepositoryArticulos repo) {
            Repo = repo;
        }
        public void Modificar(Articulo obj) {
            Repo.Update(obj);
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUModificarTipoMovimiento.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System.Security.Cryptography;

namespace LogicaAplicacion.CasosUso {
    public class CUModificarTipoMovimiento : ICUModificar<TipoMovimientoDTO> {
        public IRepositoryTiposMovimientos Repo { get; set; }
        public IRepositoryMovimientosStock RepoMovStock { get; set; }
}

```

```

public CUModificarTipoMovimiento(IRepositoryTiposMovimientos repo, IRepositoryMovimientosStock repoMovStock) {
    Repo = repo;
    RepoMovStock = repoMovStock;
}

public void Modificar(TipoMovimientoDTO obj) {
    TipoMovimiento aModificar = Repo.FindById(obj.Id);

    //chequear que el tipo no este siendo usado en ningun movimiento
    if (!RepoMovStock.TieneTipoMovimiento(aModificar.Id)) {
        aModificar.Nombre = obj.Nombre;

        Repo.Update(aModificar);
    } else {
        throw new DatosInvalidosException("El tipo de movimiento no puede ser modificado porque tiene un Movimiento de Stock asociado.");
    }
}

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUModificarUsuario.cs
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUModificarUsuario : ICUModificar<Usuario> {
        public IRepositoryUsuarios Repo { get; set; }

        public CUModificarUsuario(IRepositoryUsuarios repo) {
            Repo = repo;
        }

        public void Modificar(Usuario obj) {
            Repo.Update(obj);
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUOrdenarArticulosAsc.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUOrdenarArticulosAsc : ICUOrdenarArticulosAsc {
        public IRepositoryArticulos Repo { get; set; }
}

```

```

public CUOrdenarArticulosAsc(IRepositorioArticulos repo) {
    Repo = repo;
}

// Listado con todos los artículos ordenados alfabéticamente en forma ascendente
public List<ArticuloDTO> OrdenarPorNombreAsc() {
    List<ArticuloDTO> dtos = new List<ArticuloDTO>();
    List<Articulo> articulosOrdenados = Repo.OrdenarArticulosAsc();
    if (articulosOrdenados.Count > 0) {
        dtos = MapperArticulos.ToListDto(articulosOrdenados);
    }
    return dtos;
}
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUOrdenarPedidosAnuladosDesc.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {
    public class CUOrdenarPedidosAnuladosDesc : ICUOrdenarPedidosAnuladosDesc {
        public IRepositorioPedidos Repo { get; set; }
        public CUOrdenarPedidosAnuladosDesc(IRepositorioPedidos repo) {
            Repo = repo;
        }
        public List<PedidoNoEntregadoDTO> OrdenarPorFechaDesc() {
            List<PedidoNoEntregadoDTO> dtos = new List<PedidoNoEntregadoDTO>();
            List<Pedido> pedidosOrdenados = Repo.OrdenarPedidosAnuladosPorFechaDesc();
            if (pedidosOrdenados.Count > 0) {
                dtos = MapperPedidos.ToListDto(pedidosOrdenados);
            }
            return dtos;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\CasosUso\CUResumenMovimiento.s.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices.X86;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.CasosUso {

```

```

public class CUResumenMovimientos : ICUResumenMovimientos {
    public IRepositoryMovimientosStock Repo { get; set; }
    public CUResumenMovimientos(IRepositoryMovimientosStock repo) {
        Repo = repo;
    }
    public List<MovimientoCantidadPorAnioYTipoDTO> ObtenerResumen() {
        List<MovimientoCantidadPorAnioYTipoDTO> dtos = new List<MovimientoCantidadPorAnioYTipoDTO>();
        string msSerialized = Repo.ObtenerCantidadPorAnioYTipo();
        List<MovimientoCantidadPorAnioYTipoDTO> msEncontrados =
            JsonConvert.DeserializeObject<List<MovimientoCantidadPorAnioYTipoDTO>>(msSerialized);

        if (msEncontrados.Count > 0) {
            dtos = msEncontrados.Select(ms => new MovimientoCantidadPorAnioYTipoDTO() {
                Anio = ms.Anio,
                Cantidad = ms.Cantidad,
                TipoMovimiento = ms.TipoMovimiento
            })
            .ToList();
        } else {
            throw new RegistroNoExisteException("No existen registros");
        }

        return dtos;
    }
}
// https://www.gerbenvanadrichem.com/software-development/understanding-the-linq-nested-grouping-example/
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUAlta.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUAlta<T> {
        void Alta(T obj);
    }
}
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUAnularPedido.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUAnularPedido {
        void Anular(int id);
    }
}
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUAutenticarUsuario.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;

```

```

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

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUAutenticarUsuario {
        UsuarioDTO Autenticar(string email, string contraseña);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBaja.cs

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

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBaja<T> {
        void Baja(int id);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBuscarClie
ntesPorMontoPedido.cs

using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBuscarCientesPorMontoPedido {
        List<ClienteDTO> BuscarPorMontoPedido(decimal monto);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBuscarPor
ArticuloYTipoMovimiento.cs

using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBuscarPorArticuloYTipoMovimiento {
        List<MovimientoStockIndexDTO> BuscarMovimientosPorArticuloYTipo(int idArticulo, string tipoMovimiento, int pagina);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBuscarPor
FechaMovimiento.cs

using DataTransferObjects;
using System;
using System.Collections.Generic;
using System.Linq;

```

```

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

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBuscarPorFechaMovimiento {
        List<ArticuloDTO> BuscarPorFecha(DateTime desde, DateTime hasta, int pagina);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBuscarPor
FechaPedido.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBuscarPorFechaPedido {
        List<PedidoNoEntregadoDTO> BuscarPorFechaPedido(DateOnly fecha);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBuscarPor
d.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBuscarPorId<T> {
        T BuscarPorId(int id);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUBuscarPor
RazonSocial.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUBuscarPorRazonSocial {
        List<ClienteDTO> BuscarPorRazonSocial(string nombre);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUCantidadD
ePaginas.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUCantidadDePaginas {
        public double ObtenerCantidadDePaginas(int idArticulo, string tipoMovimiento);
        public double ObtenerCantidadDePaginas(string desde, string hasta);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUListado.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUListado<T> {
        List<T> ObtenerListado();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUModificar.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUModificar<T> {
        void Modificar(T obj);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUOrdenarArticulosAsc.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUOrdenarArticulosAsc {
        List<ArticuloDTO> OrdenarPorNombreAsc();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUOrdenarPedidosAnuladosDesc.cs
using DataTransferObjects;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUOrdenarPedidosAnuladosDesc {
        List<PedidoNoEntregadoDTO> OrdenarPorFechaDesc();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\InterfacesCasosUso\ICUResumen
Movimientos.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaAplicacion.InterfacesCasosUso {
    public interface ICUResumenMovimientos {
        public List<MovimientoCantidadPorAnioYTipoDTO> ObtenerResumen();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\obj\Debug\net8.0\.NETCoreApp,Ver
sion=v8.0.AssemblyAttributes.cs
// <autogenerated />
using System;
using System.Reflection;
[assembly: global::System.Runtime.Versioning.TargetFrameworkAttribute(".NETCoreApp,Version=v8.0", FrameworkDisplayName = ".NET 8.0")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\obj\Debug\net8.0\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("LogicaAplicacion")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+16d075e4bdc1af1f44ce24eefe9cb9e2b59eac12")]
[assembly: System.Reflection.AssemblyProductAttribute("LogicaAplicacion")]
[assembly: System.Reflection.AssemblyTitleAttribute("LogicaAplicacion")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\obj\Debug\net8.0\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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\obj\Release\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\obj\Release\net8.0\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("LogicaAplicacion")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+f9761fd17d5acfdc29824f17f329c90bf5a7c8da")]
[assembly: System.Reflection.AssemblyProductAttribute("LogicaAplicacion")]
[assembly: System.Reflection.AssemblyTitleAttribute("LogicaAplicacion")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaAplicacion\obj\Release\net8.0\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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Migrations\20240620173649_init.cs
using System;
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace LogicaDatos.Migrations
{
    /// <inheritdoc />
    public partial class init : Migration
    {
        /// <inheritdoc />

```

```

protected override void Up(MigrationBuilder migrationBuilder)
{
    migrationBuilder.CreateTable(
        name: "Articulos",
        columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Nombre = table.Column<string>(type: "nvarchar(200)", maxLength: 200, nullable: false),
        Descripcion = table.Column<string>(type: "nvarchar(max)", nullable: false),
        CodigoProveedor = table.Column<string>(type: "nvarchar(13)", maxLength: 13, nullable: false),
        Precio = table.Column<int>(type: "int", nullable: false),
        Stock = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Articulos", x => x.Id);
    });
}

migrationBuilder.CreateTable(
    name: "Clientes",
    columns: table => new
{
    Id = table.Column<int>(type: "int", nullable: false)
        .Annotation("SqlServer:Identity", "1, 1"),
    RazonSocial = table.Column<string>(type: "nvarchar(max)", nullable: true),
    Rut = table.Column<string>(type: "nvarchar(12)", maxLength: 12, nullable: true),
    Direccion_Calle_Valor = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Direccion_NumeroPuerta_Valor = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Direccion_Ciudad_Valor = table.Column<string>(type: "nvarchar(max)", nullable: false),
    DistanciaHastaDeposito = table.Column<int>(type: "int", nullable: true)
},
constraints: table =>
{
    table.PrimaryKey("PK_Clientes", x => x.Id);
});

migrationBuilder.CreateTable(
    name: "Parametros",
    columns: table => new
{
    Id = table.Column<int>(type: "int", nullable: false)
        .Annotation("SqlServer:Identity", "1, 1"),
    Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
    Valor = table.Column<decimal>(type: "decimal(18,2)", nullable: false)
},
constraints: table =>
{
    table.PrimaryKey("PK_Parametros", x => x.Id);
});

migrationBuilder.CreateTable(
    name: "TiposDeMovimientos",
    columns: table => new
{
    Id = table.Column<int>(type: "int", nullable: false)
        .Annotation("SqlServer:Identity", "1, 1"),
    Nombre = table.Column<string>(type: "nvarchar(450)", nullable: false),
    TipoAccion = table.Column<string>(type: "nvarchar(24)", nullable: false)
},
constraints: table =>

```

```

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

migrationBuilder.CreateTable(
    name: "Usuarios",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Email = table.Column<string>(type: "nvarchar(450)", nullable: false),
        Nombre = table.Column<string>(type: "nvarchar(max)", nullable: false),
        Apellido = table.Column<string>(type: "nvarchar(max)", nullable: false),
        Contraseña = table.Column<string>(type: "nvarchar(max)", nullable: false),
        ContraseniaEncriptada = table.Column<string>(type: "nvarchar(max)", nullable: true),
        Tipo = table.Column<string>(type: "nvarchar(24)", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Usuarios", x => x.Id);
    });
}

migrationBuilder.CreateTable(
    name: "Pedidos",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Fecha = table.Column<DateOnly>(type: "date", nullable: false),
        FechaEntrega = table.Column<DateOnly>(type: "date", nullable: false),
        ClientId = table.Column<int>(type: "int", nullable: false),
        Total = table.Column<decimal>(type: "decimal(18,2)", nullable: false),
        Iva = table.Column<decimal>(type: "decimal(18,2)", nullable: false),
        Recargo = table.Column<decimal>(type: "decimal(18,2)", nullable: false),
        Estado = table.Column<string>(type: "nvarchar(max)", nullable: false),
        Discriminator = table.Column<string>(type: "nvarchar(13)", maxLength: 13, nullable: false),
        PlazoEstipulado = table.Column<int>(type: "int", nullable: true),
        PedidoExpress_PlazoEstipulado = table.Column<int>(type: "int", nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Pedidos", x => x.Id);
        table.ForeignKey(
            name: "FK_Pedidos_Clientes_ClientId",
            column: x => x.ClientId,
            principalTable: "Clientes",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });
}

migrationBuilder.CreateTable(
    name: "MovimientosDeStock",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Fecha = table.Column<DateTime>(type: "datetime2", nullable: false),
        Articuloid = table.Column<int>(type: "int", nullable: false),
        Usuarioid = table.Column<int>(type: "int", nullable: false),
        TipMovimientoId = table.Column<int>(type: "int", nullable: false),
        Cantidad = table.Column<int>(type: "int", nullable: false)
    });

```

```

},
constraints: table =>
{
  table.PrimaryKey("PK_MovimientosDeStock", x => x.Id);
  table.ForeignKey(
    name: "FK_MovimientosDeStock_Articulos_Articuloid",
    column: x => x.Articuloid,
    principalTable: "Articulos",
    principalColumn: "Id",
    onDelete: ReferentialAction.Cascade);
  table.ForeignKey(
    name: "FK_MovimientosDeStock_TiposDeMovimientos_TipoMovimientoId",
    column: x => x.TipoMovimientoId,
    principalTable: "TiposDeMovimientos",
    principalColumn: "Id",
    onDelete: ReferentialAction.Cascade);
  table.ForeignKey(
    name: "FK_MovimientosDeStock_Usuarios_Usuarioid",
    column: x => x.Usuarioid,
    principalTable: "Usuarios",
    principalColumn: "Id",
    onDelete: ReferentialAction.Cascade);
});
};

migrationBuilder.CreateTable(
  name: "Lineas",
  columns: table => new
{
  Id = table.Column<int>(type: "int", nullable: false)
    .Annotation("SqlServer:Identity", "1, 1"),
  Articuloid = table.Column<int>(type: "int", nullable: false),
  UnidadesSolicitadas = table.Column<int>(type: "int", nullable: false),
  PreciodUnitario = table.Column<int>(type: "int", nullable: false),
  Pedidoid = table.Column<int>(type: "int", nullable: true)
},
constraints: table =>
{
  table.PrimaryKey("PK_Lineas", x => x.Id);
  table.ForeignKey(
    name: "FK_Lineas_Articulos_Articuloid",
    column: x => x.Articuloid,
    principalTable: "Articulos",
    principalColumn: "Id",
    onDelete: ReferentialAction.Cascade);
  table.ForeignKey(
    name: "FK_Lineas_Pedidos_Pedidoid",
    column: x => x.Pedidoid,
    principalTable: "Pedidos",
    principalColumn: "Id");
});
};

migrationBuilder.CreateIndex(
  name: "IX_Articulos_CodigoProveedor",
  table: "Articulos",
  column: "CodigoProveedor",
  unique: true);

migrationBuilder.CreateIndex(
  name: "IX_Articulos_Nombre",
  table: "Articulos",
  column: "Nombre",

```

```

        unique: true);

migrationBuilder.CreateIndex(
    name: "IX_Clientes_Rut",
    table: "Clientes",
    column: "Rut",
    unique: true,
    filter: "[Rut] IS NOT NULL");

migrationBuilder.CreateIndex(
    name: "IX_Lineas_Articuloid",
    table: "Lineas",
    column: "Articuloid");

migrationBuilder.CreateIndex(
    name: "IX_Lineas_Pedidoid",
    table: "Lineas",
    column: "Pedidoid");

migrationBuilder.CreateIndex(
    name: "IX_MovimientosDeStock_Articuloid",
    table: "MovimientosDeStock",
    column: "Articuloid");

migrationBuilder.CreateIndex(
    name: "IX_MovimientosDeStock_TipoMovimientoid",
    table: "MovimientosDeStock",
    column: "TipoMovimientoid");

migrationBuilder.CreateIndex(
    name: "IX_MovimientosDeStock_Usuarioid",
    table: "MovimientosDeStock",
    column: "Usuarioid");

migrationBuilder.CreateIndex(
    name: "IX_Pedidos_Clienteid",
    table: "Pedidos",
    column: "Clienteid");

migrationBuilder.CreateIndex(
    name: "IX_TiposDeMovimientos_Nombre",
    table: "TiposDeMovimientos",
    column: "Nombre",
    unique: true);

migrationBuilder.CreateIndex(
    name: "IX_Usuarios_Email",
    table: "Usuarios",
    column: "Email",
    unique: true);
}

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

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

```

```

migrationBuilder.DropTable(
    name: "Parametros");

migrationBuilder.DropTable(
    name: "Pedidos");

migrationBuilder.DropTable(
    name: "Articulos");

migrationBuilder.DropTable(
    name: "TiposDeMovimientos");

migrationBuilder.DropTable(
    name: "Usuarios");

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

}

}

//// File:
C:\Users\caro0\Documents\ORT\TP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Migrations\20240620173649_init.Design
er.cs
// <auto-generated />
using System;
using LogicaDatos.Repositorios;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

#nullable disable

namespace LogicaDatos.Migrations
{
    [DbContext(typeof(ObligatorioContext))]
    [Migration("20240620173649_init")]
    partial class init
    {
        /// <inheritdoc />
        protected override void BuildTargetModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "8.0.6")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

           SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

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

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

                b.Property<string>("CodigoProveedor")
                    .IsRequired()
                    .HasMaxLength(13)
            });
        }
    }
}

```

```

    .HasColumnType("nvarchar(13)");

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

    b.Property<string>("Nombre")
        .IsRequired()
        .HasMaxLength(200)
        .HasColumnType("nvarchar(200)");

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

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

    b.HasKey("Id");

    b.HasIndex("CodigoProveedor")
        .IsUnique();

    b.HasIndex("Nombre")
        .IsUnique();

    b.ToTable("Articulos", (string)null);
});

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

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

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

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

    b.Property<string>("Rut")
        .HasMaxLength(12)
        .HasColumnType("nvarchar(12)");

    b.HasKey("Id");

    b.HasIndex("Rut")
        .IsUnique()
        .HasFilter("[Rut] IS NOT NULL");

    b.ToTable("Clientes", (string)null);
});

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

```

```

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

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

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

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

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

b.HasKey("Id");

b.HasIndex("Articuloid");

b.HasIndex("Pedidolid");

b.ToTable("Lineas", (string)null);
});

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

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

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

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

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

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

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

    b.HasKey("Id");

    b.HasIndex("Articuloid");

    b.HasIndex("TipMovimentold");

    b.HasIndex("Usuariolid");

    b.ToTable("MovimientosDeStock", (string)null);
});

modelBuilder.Entity("LogicaNegocio.Dominio.Parametro", b =>
{
    b.Property<int>"Id"
        .ValueGeneratedOnAdd()

```

```

    .HasColumnType("int");

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

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

b.Property<decimal>("Valor")
    .HasColumnType("decimal(18,2)");

b.HasKey("Id");

    b.ToTable("Parametros", (string)null);
});

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

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

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

    b.Property<string>("Discriminator")
        .IsRequired()
        .HasMaxLength(13)
        .HasColumnType("nvarchar(13)");

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

    b.Property<DateOnly>("Fecha")
        .HasColumnType("date");

    b.Property<DateOnly>("FechaEntrega")
        .HasColumnType("date");

    b.Property<decimal>("Iva")
        .HasColumnType("decimal(18,2)");

    b.Property<decimal>("Recargo")
        .HasColumnType("decimal(18,2)");

    b.Property<decimal>("Total")
        .HasColumnType("decimal(18,2)");

    b.HasKey("Id");

    b.HasIndex("ClientId");

    b.ToTable("Pedidos", (string)null);

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

    b.UseTphMappingStrategy();
});

```

```

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

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

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

    b.Property<string>"TipoAccion")
        .IsRequired()
        .HasColumnType("nvarchar(24)");

    b.HasKey("Id");

    b.HasIndex("Nombre")
        .IsUnique();

    b.ToTable("TiposDeMovimientos", (string)null);
});

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

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

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

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

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

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

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

    b.Property<string>"Tipo")
        .IsRequired()
        .HasColumnType("nvarchar(24)");

    b.HasKey("Id");

    b.HasIndex("Email")
        .IsUnique();
}

```

```

        b.ToTable("Usuarios", (string)null);
    });

modelBuilder.Entity("LogicaNegocio.Dominio.PedidoComun", b =>
{
    b.HasBaseType("LogicaNegocio.Dominio.Pedido");

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

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

modelBuilder.Entity("LogicaNegocio.Dominio.PedidoExpress", b =>
{
    b.HasBaseType("LogicaNegocio.Dominio.Pedido");

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

    b.ToTable("Pedidos", t =>
    {
        t.Property("PlazoEstipulado")
            .HasColumnName("PedidoExpress_PlazoEstipulado");
    });

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

modelBuilder.Entity("LogicaNegocio.Dominio.Cliente", b =>
{
    b.OwnsOne("LogicaNegocio.ValueObjects.Direccion", "Direccion", b1 =>
    {
        b1.Property<int>"Clienteid"
            .HasColumnType("int");

        b1.HasKey("Clienteid");

        b1.ToTable("Clientes");

        b1.WithOwner()
            .HasForeignKey("Clienteid");
    });

    b1.OwnsOne("LogicaNegocio.ValueObjects.Calle", "Calle", b2 =>
    {
        b2.Property<int>"DireccionId"
            .HasColumnType("int");

        b2.Property<string>"Valor"
            .IsRequired()
            .HasColumnType("nvarchar(max)");

        b2.HasKey("DireccionId");

        b2.ToTable("Clientes");

        b2.WithOwner()
            .HasForeignKey("DireccionId");
    });
});

b1.OwnsOne("LogicaNegocio.ValueObjects.Ciudad", "Ciudad", b2 =>

```

```

{
    b2.Property<int>"DireccionId"
        .HasColumnType("int");

    b2.Property<string>"Valor"
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b2.HasKey("DireccionId");

    b2.ToTable("Clientes");

    b2.WithOwner()
        .HasForeignKey("DireccionId");
});

b1.OwnsOne("LogicaNegocio.ValueObjects.NumeroPuerta", "NumeroPuerta", b2 =>
{
    b2.Property<int>"DireccionId"
        .HasColumnType("int");

    b2.Property<string>"Valor"
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b2.HasKey("DireccionId");

    b2.ToTable("Clientes");

    b2.WithOwner()
        .HasForeignKey("DireccionId");
});

b1.Navigation("Calle")
    .IsRequired();

b1.Navigation("Ciudad")
    .IsRequired();

b1.Navigation("NumeroPuerta")
    .IsRequired();
});

b.Navigation("Direccion")
    .IsRequired();
});

modelBuilder.Entity("LogicaNegocio.Dominio.Linea", b =>
{
    b.HasOne("LogicaNegocio.Dominio.Articulo", "Articulo")
        .WithMany()
        .HasForeignKey("Articuloid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("LogicaNegocio.Dominio.Pedido", null)
        .WithMany("Lineas")
        .HasForeignKey("Pedidoid");

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

```

```

modelBuilder.Entity("LogicaNegocio.Dominio.MovimientoStock", b =>
{
    b.HasOne("LogicaNegocio.Dominio.Articulo", "Articulo")
        .WithMany()
        .HasForeignKey("Articuloid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("LogicaNegocio.Dominio.TipoMovimiento", "TipoMovimiento")
        .WithMany()
        .HasForeignKey("Tipomovimentoid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

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

    b.Navigation("Articulo");

    b.Navigation("Tipomovimiento");

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

modelBuilder.Entity("LogicaNegocio.Dominio.Pedido", b =>
{
    b.HasOne("LogicaNegocio.Dominio.Cliente", "Cliente")
        .WithMany()
        .HasForeignKey("Clienteid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

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

modelBuilder.Entity("LogicaNegocio.Dominio.Pedido", b =>
{
    b.Navigation("Lineas");
});

#pragma warning restore 612, 618
}

}

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Migrations\ObligatorioContextModelSnapshot.cs
// <auto-generated />
using System;
using LogicaDatos.Repositorios;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

#nullable disable

namespace LogicaDatos.Migrations

```

```

{
    [DbContext(typeof(ObligatorioContext))]
    partial class ObligatorioContextModelSnapshot : ModelSnapshot
    {
        protected override void BuildModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "8.0.6")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

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

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

                b.Property<string>"CodigoProveedor"
                    .IsRequired()
                    .HasMaxLength(13)
                    .HasColumnType("nvarchar(13)");

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

                b.Property<string>"Nombre"
                    .IsRequired()
                    .HasMaxLength(200)
                    .HasColumnType("nvarchar(200)");

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

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

                b.HasKey("Id");

                b.HasIndex("CodigoProveedor")
                    .IsUnique();

                b.HasIndex("Nombre")
                    .IsUnique();

                b.ToTable("Articulos", (string)null);
            });

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

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

                b.Property<int?>"DistanciaHastaDeposito"

```

```

    .HasColumnType("int");

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

    b.Property<string>"("Rut")
        .HasMaxLength(12)
        .HasColumnType("nvarchar(12)");

    b.HasKey("Id");

    b.HasIndex("Rut")
        .IsUnique()
        .HasFilter("[Rut] IS NOT NULL");

    b.ToTable("Clientes", (string)null);
});

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

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

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

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

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

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

    b.HasKey("Id");

    b.HasIndex("Articuloid");
    b.HasIndex("Pedidold");

    b.ToTable("Lineas", (string)null);
});

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

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

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

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

```

```

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

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

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

        b.HasKey("Id");

        b.HasIndex("ArticuloId");

        b.HasIndex("TipoMovimientoId");

        b.HasIndex("UsuarioId");

        b.ToTable("MovimientosDeStock", (string)null);
    });

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

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

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

    b.Property<decimal>"("Valor"
        .HasColumnType("decimal(18,2)");

    b.HasKey("Id");

    b.ToTable("Parametros", (string)null);
});

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

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

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

    b.Property<string>"("Discriminator"
        .IsRequired()
        .HasMaxLength(13)
        .HasColumnType("nvarchar(13)");

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

    b.Property<DateOnly>"("Fecha")

```

```

    .HasColumnType("date");

    b.Property<DateOnly>"FechaEntrega"
        .HasColumnType("date");

    b.Property<decimal>"Iva"
        .HasColumnType("decimal(18,2)");

    b.Property<decimal>"Recargo"
        .HasColumnType("decimal(18,2)");

    b.Property<decimal>"Total"
        .HasColumnType("decimal(18,2)");

    b.HasKey("Id");

    b.HasIndex("Clienteld");

    b.ToTable("Pedidos", (string)null);

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

    b.UseTphMappingStrategy();
});

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

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

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

    b.Property<string>"TipoAccion"
        .IsRequired()
        .HasColumnType("nvarchar(24)");

    b.HasKey("Id");

    b.HasIndex("Nombre")
        .IsUnique();

    b.ToTable("TiposDeMovimientos", (string)null);
};

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

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

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

```

```

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

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

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

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

        b.Property<string>("Tipo")
            .IsRequired()
            .HasColumnType("nvarchar(24)");

        b.HasKey("Id");

        b.HasIndex("Email")
            .IsUnique();

        b.ToTable("Usuarios", (string)null);
    });

modelBuilder.Entity("LogicaNegocio.Dominio.PedidoComun", b =>
{
    b.HasBaseType("LogicaNegocio.Dominio.Pedido");

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

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

modelBuilder.Entity("LogicaNegocio.Dominio.PedidoExpress", b =>
{
    b.HasBaseType("LogicaNegocio.Dominio.Pedido");

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

    b.ToTable("Pedidos", t =>
    {
        t.Property("PlazoEstipulado")
            .HasColumnName("PedidoExpress_PlazoEstipulado");
    });

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

modelBuilder.Entity("LogicaNegocio.Dominio.Cliente", b =>
{
    b.OwnsOne("LogicaNegocio.ValueObjects.Direccion", "Direccion", b1 =>
    {
        b1.Property<int>("ClientelId")
            .HasColumnType("int");

        b1.HasKey("ClientelId");
    });
});

```

```

b1.ToTable("Clientes");

b1.WithOwner()
    .HasForeignKey("ClienteId");

b1.OwnsOne("LogicaNegocio.ValueObjects.Calle", "Calle", b2 =>
{
    b2.Property<int>"DireccionId")
        .HasColumnType("int");

    b2.Property<string>"Valor")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b2.HasKey("DireccionId");

    b2.ToTable("Clientes");

    b2.WithOwner()
        .HasForeignKey("DireccionId");
});

b1.OwnsOne("LogicaNegocio.ValueObjects.Ciudad", "Ciudad", b2 =>
{
    b2.Property<int>"DireccionId")
        .HasColumnType("int");

    b2.Property<string>"Valor")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b2.HasKey("DireccionId");

    b2.ToTable("Clientes");

    b2.WithOwner()
        .HasForeignKey("DireccionId");
});

b1.OwnsOne("LogicaNegocio.ValueObjects.NumeroPuerta", "NumeroPuerta", b2 =>
{
    b2.Property<int>"DireccionId")
        .HasColumnType("int");

    b2.Property<string>"Valor")
        .IsRequired()
        .HasColumnType("nvarchar(max)");

    b2.HasKey("DireccionId");

    b2.ToTable("Clientes");

    b2.WithOwner()
        .HasForeignKey("DireccionId");
});

b1.Navigation("Calle")
    .IsRequired();

b1.Navigation("Ciudad")

```

```

        .IsRequired();

    b1.Navigation("NumeroPuerta")
        .IsRequired();
    });

    b.Navigation("Direccion")
        .IsRequired();
});

modelBuilder.Entity("LogicaNegocio.Dominio.Linea", b =>
{
    b.HasOne("LogicaNegocio.Dominio.Articulo", "Articulo")
        .WithMany()
        .HasForeignKey("Articuloid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("LogicaNegocio.Dominio.Pedido", null)
        .WithMany("Lineas")
        .HasForeignKey("Pedidoid");

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

modelBuilder.Entity("LogicaNegocio.Dominio.MovimientoStock", b =>
{
    b.HasOne("LogicaNegocio.Dominio.Articulo", "Articulo")
        .WithMany()
        .HasForeignKey("Articuloid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

    b.HasOne("LogicaNegocio.Dominio.TipoMovimiento", "TipoMovimiento")
        .WithMany()
        .HasForeignKey("Tipomovimentoid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

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

    b.Navigation("Articulo");

    b.Navigation("TipoMovimiento");

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

modelBuilder.Entity("LogicaNegocio.Dominio.Pedido", b =>
{
    b.HasOne("LogicaNegocio.Dominio.Cliente", "Cliente")
        .WithMany()
        .HasForeignKey("Clienteid")
        .OnDelete(DeleteBehavior.Cascade)
        .IsRequired();

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

```

```

    });

modelBuilder.Entity("LogicaNegocio.Dominio.Pedido", b =>
{
    b.Navigation("Lineas");
});

#pragma warning restore 612, 618
}

}
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\obj\Debug\net8.0\LogicaDatos.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("LogicaDatos")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+16d075e4bdc1af1f44ce24eefe9cb9e2b59eac12")]
[assembly: System.Reflection.AssemblyProductAttribute("LogicaDatos")]
[assembly: System.Reflection.AssemblyTitleAttribute("LogicaDatos")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\obj\Debug\net8.0\LogicaDatos.GlobalUs
ings.g.cs
// <autogenerated/>
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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\obj\Release\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")]

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\obj\Release\net8.0\LogicaDatos.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("LogicaDatos")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+f9761fd17d5acfdc29824f17f329c90bf5a7c8da")]
[assembly: System.Reflection.AssemblyProductAttribute("LogicaDatos")]
[assembly: System.Reflection.AssemblyTitleAttribute("LogicaDatos")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\obj\Release\net8.0\LogicaDatos.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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositories\ObligatorioContext.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.ValueObjects;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace LogicaDatos.Repositories {
    public class ObligatorioContext : DbContext {
        public DbSet<Usuario> Usuarios { get; set; }
        public DbSet<Cliente> Clientes { get; set; }
        public DbSet<Articulo> Articulos { get; set; }
        public DbSet<Pedido> Pedidos { get; set; }
        public DbSet<PedidoExpress> PedidoExpress { get; set; }
        public DbSet<PedidoComun> PedidoComun { get; set; }
        public DbSet<Linea> Lineas { get; set; }
        public DbSet<Parametro> Parametros { get; set; }
        public DbSet<TipoMovimiento> TiposMovimientos { get; set; }
    }
}

```

```

public DbSet<MovimientoStock> MovimientosStock { get; set; }

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

protected override void OnModelCreating(ModelBuilder modelBuilder) {
    base.OnModelCreating(modelBuilder);
    modelBuilder.Entity<Usuario>().ToTable("Usuarios");
    modelBuilder.Entity<Articulo>().ToTable("Articulos");
    modelBuilder.Entity<Cliente>().ToTable("Clientes");
    modelBuilder.Entity<Pedido>().ToTable("Pedidos");
    modelBuilder.Entity<Linea>().ToTable("Lineas");
    modelBuilder.Entity<Parametro>().ToTable("Parametros");
    modelBuilder.Entity<TipoMovimiento>().ToTable("TiposDeMovimientos");
    modelBuilder.Entity<MovimientoStock>().ToTable("MovimientosDeStock");

    modelBuilder.Entity<Cliente>().OwnsOne(
        c => c.Direccion, d => {
            d.OwnsOne(c => c.Calle);
            d.OwnsOne(n => n.NumeroPuerta);
            d.OwnsOne(ci => ci.Ciudad);
        });
}
}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) {
    base.OnConfiguring(optionsBuilder);
    //optionsBuilder.UseSqlServer(
    //    "Data Source=localhost\\SQLEXPRESS; Initial Catalog=ObligatorioP3; Integrated Security=SSPI;TrustServerCertificate=True"
    //);
}
}

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioArticulos.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioArticulos : IRepositoryArticulos {
        public ObligatorioContext Contexto { get; set; }

        public RepositorioArticulos(ObligatorioContext ctx) {
            Contexto = ctx;
        }
        public void Create(Articulo obj) {
            obj.EsValido();
            if (!Contexto.Articulos.Any(a => a.Nombre == obj.Nombre) && !Contexto.Articulos.Any(a => a.CodigoProveedor == obj.CodigoProveedor)) {
                Contexto.Articulos.Add(obj);
                Contexto.SaveChanges();
            } else {
                throw new DuplicadoException("Ya existe un artículo con ese nombre y código de proveedor.");
            }
        }
    }
}

```

```

    }

    public void Delete(int id) {
        Articulo aBorrar = FindById(id);
        if (aBorrar != null) {
            Contexto.Articulos.Remove(aBorrar);
            Contexto.SaveChanges();
        } else {
            throw new RegistroNoExisteException("El articulo no existe");
        }
    }

    public Articulo FindById(int id) {
        return Contexto.Articulos.Find(id);
    }

    public List<Articulo> GetAll() {
        return Contexto.Articulos.ToList();
    }

    public void Update(Articulo obj) {
        obj.EsValido();
        if (!Contexto.Articulos.Any(a => a.Nombre == obj.Nombre) && !Contexto.Articulos.Any(a => a.CodigoProveedor == obj.CodigoProveedor)) {
            Contexto.Articulos.Update(obj);
            Contexto.SaveChanges();
        } else {
            throw new DuplicadoException("Ya existe un artículo con ese nombre y código de proveedor.");
        }
    }

    // Listado con todos los artículos ordenados alfabéticamente en forma ascendente
    public List<Articulo> OrdenarArticulosAsc() {
        return Contexto.Articulos.OrderBy(a => a.Nombre).ToList();
    }
}

//// File:  

C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioClientes.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioClientes : IRepositoryClientes {
        public ObligatorioContext Contexto { get; set; }
        public RepositorioClientes(ObligatorioContext ctx) {
            Contexto = ctx;
        }
        public List<Cliente> BuscarPorRazonSocial(string nombre) {
            return Contexto.Clientes.Where(c => c.RazonSocial.Contains(nombre)).ToList();
        }

        public void Create(Cliente obj) {
            obj.EsValido();
            Contexto.Clientes.Add(obj);
        }
    }
}

```

```

        Contexto.SaveChanges();
    }

    public void Delete(int id) {
        Cliente aBorrar = FindById(id);
        if (aBorrar != null) {
            Contexto.Clientes.Remove(aBorrar);
            Contexto.SaveChanges();
        } else {
            throw new RegistroNoExisteException("El cliente no existe");
        }
    }

    public Cliente FindById(int id) {
        return Contexto.Clientes.Find(id);
    }

    public List<Cliente> GetAll() {
        return Contexto.Clientes.ToList();
    }

    public void Update(Cliente obj) {
        obj.EsValido();
        Contexto.Clientes.Update(obj);
        Contexto.SaveChanges();
    }
}

//// File:  

C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioLineas.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioLineas : IRepositoryLineas {
        public ObligatorioContext Contexto { get; set; }

        public RepositorioLineas(ObligatorioContext ctx) {
            Contexto = ctx;
        }

        public void Create(Linea obj) {
            //obj.EsValido();
            Contexto.Lineas.Add(obj);
            Contexto.SaveChanges();
        }

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

        public Linea FindById(int id) {
            throw new NotImplementedException();
        }

        public List<Linea> GetAll() {

```

```

        throw new NotImplementedException();
    }

    public void Update(Linea obj) {
        throw new NotImplementedException();
    }
}

}

//// File:
C:\Users\caro0\Documents\ORTP3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioMovimientosStock.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection.Metadata.Ecma335;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioMovimientosStock : IRepositoryMovimientosStock {
        public ObligatorioContext Contexto { get; private set; }

        public RepositorioMovimientosStock(ObligatorioContext ctx) {
            Contexto = ctx;
        }

        public void Create(MovimientoStock obj) {
            obj.EsValido();
            Contexto.Entry(obj.Articulo).State = EntityState.Unchanged;
            Contexto.Entry(obj.TipoMovimiento).State = EntityState.Unchanged;
            Contexto.MovimientosStock.Add(obj);
            Contexto.SaveChanges();
        }

        public MovimientoStock FindById(int id) {
            return Contexto.MovimientosStock.Find(id);
        }

        public List<MovimientoStock> GetAll() {
            return Contexto.MovimientosStock
                .Include(ms => ms.Articulo)
                .Include(ms => ms.TipoMovimiento)
                .Include(ms => ms.Usuario)
                .ToList();
        }

        public List<MovimientoStock> BuscarMovimientosPorArticuloYTipo(int idArticulo, string tipoMovimiento, int pagina, int limitePorPagina) {
            return Contexto.MovimientosStock
                .Where(ms => ms.Articulo.Id == idArticulo
                    && ms.TipoMovimiento.Nombre == tipoMovimiento)
                .Include(ms => ms.TipoMovimiento)
                .Include(ms => ms.Articulo)
                .Include(ms => ms.Usuario)
                .OrderByDescending(ms => ms.Fecha)
                .ThenBy(ms => ms.Cantidad)
                .Skip((pagina - 1) * limitePorPagina)
                .Take(limitePorPagina)
        }
    }
}

```

```

        .ToList();
    }

    public List<Articulo> BuscarArticulosConMovimientosPorFecha(DateTime desde, DateTime hasta, int pagina, int limitePorPagina) {
        return Contexto.MovimientosStock
            .Where(ms => ms.Fecha >= desde && ms.Fecha <= hasta)
            .Select(ms => ms.Articulo)
            .Skip((pagina - 1) * limitePorPagina)
            .Take(limitePorPagina)
            .ToList();
    }

    public string ObtenerCantidadPorAnioYTipo() {
        List<MovimientoCantidadPorAnioYTipoDTO> consulta = new List<MovimientoCantidadPorAnioYTipoDTO>();

        consulta = Contexto.MovimientosStock
            .GroupBy(ms => new { Anio = ms.Fecha.Year, ms.TipoMovimiento.Nombre })
            .Select(grupo => new MovimientoCantidadPorAnioYTipoDTO {
                Anio = grupo.Key.Anio,
                TipoMovimiento = grupo.Key.Nombre,
                Cantidad = grupo.Sum(g => g.Cantidad)
            })
            .ToList();

        return JsonConvert.SerializeObject(consulta);
    }

    public int CantidadDeMovimientos(int idArticulo, string tipoMovimiento) {
        return Contexto.MovimientosStock
            .Where(ms => ms.Articulo.Id == idArticulo
                && ms.TipoMovimiento.Nombre == tipoMovimiento)
            .Count();
    }

    public int CantidadDeMovimientos(DateTime desde, DateTime hasta) {
        return Contexto.MovimientosStock
            .Where(ms => ms.Fecha >= desde && ms.Fecha <= hasta)
            .Count();
    }

    public bool TieneTipoMovimiento(int idTipoMovimiento) {
        return Contexto.MovimientosStock.Any(ms => ms.TipoMovimiento.Id == idTipoMovimiento);
    }
}

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioParametros.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioParametros : IRepositoryParametros {
        public ObligatorioContext Contexto { get; set; }
        public RepositorioParametros(ObligatorioContext ctx) {
            Contexto = ctx;
        }
    }
}

```

```

}

public decimal ObtenerIVA() {
    return Contexto.Parametros.Where(p => p.Nombre == "IVA").OrderByDescending(p => p.Id).First().Valor;
}

public decimal ObtenerLimitePorPagina() {
    return Contexto.Parametros.Where(p => p.Nombre == "LimitePorPagina").OrderByDescending(p => p.Id).First().Valor;
}

public decimal ObtenerLimiteTopeDeMovimientos() {
    return Contexto.Parametros.Where(p => p.Nombre == "TopeDeMovimientos").OrderByDescending(p => p.Id).First().Valor;
}

}

//// File:  

C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioPedidos.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace LogicaDatos.Repositorios {
    public class RepositorioPedidos : IRepositoryPedidos {
        public ObligatorioContext Contexto { get; set; }

        public RepositorioPedidos(ObligatorioContext ctx) {
            Contexto = ctx;
        }

        public void Create(Pedido obj) {
            obj.EsValido();
            Contexto.Entry(obj.Cliente).State = EntityState.Unchanged;
            Contexto.Pedidos.Add(obj);
            Contexto.SaveChanges();
        }

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

        public Pedido FindById(int id) {
            return Contexto.Pedidos.Include(p => p.Lineas).FirstOrDefault(p => p.Id == id);
        }

        public List<Pedido> GetAll() {
            return Contexto.Pedidos.ToList();
        }

        public void Update(Pedido obj) {
            obj.EsValido();
            //Contexto.Entry(obj.Cliente).State = EntityState.Unchanged;
            Contexto.Pedidos.Update(obj);
            Contexto.SaveChanges();
        }
    }
}

```

```

public void AnularPedido(int id) {
    Pedido aAnular = FindById(id);
    if (aAnular != null) {
        aAnular.Estado = "Anulado";
        Contexto.Pedidos.Update(aAnular);
        Contexto.SaveChanges();
    } else {
        throw new RegistroNoExisteException("El pedido no existe");
    }
}

public List<Pedido> BuscarPorFechaDeEmision(DateOnly fecha) {
    return Contexto.Pedidos.Where(p => p.Fecha == fecha && p.Estado == "Pendiente").Include(p => p.Cliente).ToList();
}

public List<Pedido> OrdenarPedidosAnuladosPorFechaDesc() {
    return Contexto.Pedidos.Where(p => p.Estado == "Anulado").Include(p => p.Cliente).OrderByDescending(p => p.Fecha).ToList();
}

public List<Pedido> ListarPedidosPendientes() {
    return Contexto.Pedidos.Where(p => p.Estado == "Pendiente").Include(p => p.Cliente).ToList();
}

public List<Cliente> BuscarClientes(decimal monto) {
    return Contexto.Pedidos.Where(p => p.Total >= monto).Select(p => p.Cliente).Distinct().ToList();
}

}

// https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/implementing-inheritance-with-the-entity-framework-in-an-asp-net-mvc-application
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioTiposMovimientos.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioTiposMovimientos : IRepositorioTiposMovimientos {
        public ObligatorioContext Contexto { get; private set; }

        public RepositorioTiposMovimientos(ObligatorioContext ctx) {
            Contexto = ctx;
        }

        public void Create(TipoMovimiento obj) {
            if (!Contexto.TiposMovimientos.Any(t => t.Nombre.ToLower() == obj.Nombre.ToLower())) {
                Contexto.TiposMovimientos.Add(obj);
                Contexto.SaveChanges();
            } else {
                throw new DuplicadoException("Ya existe un tipo de movimiento con ese nombre.");
            }
        }
    }
}

```

```

public void Delete(int id) {
    TipoMovimiento aBorrar = FindById(id);
    if (aBorrar != null) {
        Contexto.TiposMovimientos.Remove(aBorrar);
        Contexto.SaveChanges();
    } else {
        throw new RegistroNoExisteException("El tipo de movimiento no existe");
    }
}

public TipoMovimiento FindById(int id) {
    return Contexto.TiposMovimientos.Find(id);
}

public List<TipoMovimiento> GetAll() {
    return Contexto.TiposMovimientos.ToList();
}

public void Update(TipoMovimiento obj) {
    obj.EsValido();
    TipoMovimiento tm = Contexto.TiposMovimientos.Where(t => t.Nombre.ToLower() == obj.Nombre.ToLower()).SingleOrDefault();

    if (tm != null) {
        if(tm.Id == obj.Id) {
            throw new DuplicadoException("Ya existe un tipo de movimiento con ese nombre.");
        } else {
            Contexto.Entry(tm).State = EntityState.Detached;
        }
    }

    Contexto.TiposMovimientos.Update(obj);
    Contexto.SaveChanges();
}
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaDatos\Repositorios\RepositorioUsuarios.cs
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaDatos.Repositorios {
    public class RepositorioUsuarios : IRepositoryUsuarios {
        public ObligatorioContext Contexto { get; set; }
        public RepositorioUsuarios(ObligatorioContext ctx) {
            Contexto = ctx;
        }
        public void Create(Usuario obj) {
            obj.EsValido();
            if (!Contexto.Usuarios.Any(u => u.Email.ToLower() == obj.Email.ToLower())) {
                Contexto.Usuarios.Add(obj);
                Contexto.SaveChanges();
            } else {
                throw new DuplicadoException("Ya existe un usuario registrado con ese email.");
            }
        }
    }
}

```

```

}

public void Delete(int id) {
    Usuario aBorrar = FindById(id);
    if (aBorrar != null) {
        Contexto.Usuarios.Remove(aBorrar);
        Contexto.SaveChanges();
    } else {
        throw new RegistroNoExisteException("El usuario no existe");
    }
}

public void Update(Usuario obj) {
    obj.EsValido();
    if (!Contexto.Usuarios.Any(u => u.Email.ToLower() == obj.Email.ToLower())) {
        Contexto.Usuarios.Update(obj);
        Contexto.SaveChanges();
    } else {
        throw new DuplicadoException("Ya existe un usuario registrado con ese email.");
    }
}

public Usuario BuscarPorEmail(string email) {
    return Contexto.Usuarios.Where(u => u.Email.ToLower() == email.ToLower()).SingleOrDefault();
}

public Usuario BuscarPorEmail(string email, string password) {
    return Contexto.Usuarios.Where(u => u.Email.ToLower() == email.ToLower() && u.Contraseña == password).SingleOrDefault();
}

public List<Usuario> GetAll() {
    return Contexto.Usuarios.ToList();
}

public Usuario FindById(int id) {
    return Contexto.Usuarios.Find(id);
}

}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\Articulo.cs
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesDominio;
using System.ComponentModel.DataAnnotations;
using Microsoft.EntityFrameworkCore;
using System.ComponentModel.DataAnnotations.Schema;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;

namespace LogicaNegocio.Dominio {
    [Index(nameof(Nombre), IsUnique = true)]
    [Index(nameof(CodigoProveedor), IsUnique = true)]
    public class Articulo : IValidar {
        public int Id { get; set; }
        [Required(ErrorMessage = "El nombre es requerido")]
        [StringLength(200, MinimumLength = 10, ErrorMessage = "El nombre debe tener entre 10 y 200 caracteres")]
        public string Nombre { get; set; } // no vacío y único. Min 10 char max 200 char
        [MinLength(5, ErrorMessage = "La descripción debe tener un mínimo de 5 caracteres")]
}

```

```

public string Descripcion { get; set; } // largo minimo de 5 caracteres
[Display(Name = "Código de Proveedor")]
[Required(ErrorMessage = "El código de proveedor es requerido")]
[StringLength(13, MinimumLength = 13, ErrorMessage = "El código de proveedor debe tener 13 dígitos")]
public stringCodigoProveedor { get; set; } // 13 digitos, no vacio
public int Precio { get; set; }
public int Stock { get; set; }

public void EsValido() {
    if (Nombre.Length < 10 || Nombre.Length > 200 || string.IsNullOrEmpty(Nombre)) {
        throw new DatosInvalidosException("El nombre debe tener entre 10 y 200 caracteres");
    }

    if (CodigoProveedor.Length != 13 || string.IsNullOrEmpty(CodigoProveedor)) {
        throw new DatosInvalidosException("El código de proveedor es requerido y debe tener 13 dígitos.");
    }

    if (Descripcion.Length < 5) {
        throw new DatosInvalidosException("La descripción debe tener un mínimo de 5 caracteres.");
    }

    if (Precio < 0 || Stock < 0) {
        throw new DatosInvalidosException("Precio y Stock no pueden ser negativos.");
    }
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\Cliente.cs
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesDominio;
using LogicaNegocio.ValueObjects;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;

namespace LogicaNegocio.Dominio {
    [Index(nameof(Rut), IsUnique = true)]
    public class Cliente : IValidar {
        [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
        [Display(Name = "Razon Social")]
        public string? RazonSocial { get; set; }
        [StringLength(12, MinimumLength = 12)]
        public string? Rut { get; set; } // 12 digitos
        public Direccion Direccion { get; set; } // calle, numero, ciudad
        public int? DistanciaHastaDeposito { get; set; }
        public void EsValido() {
            if (Rut.Length != 12) {
                throw new DatosInvalidosException("El RUT debe tener 12 dígitos.");
            }
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\Linea.cs
using LogicaNegocio.InterfacesDominio;
using System;

```

```

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

namespace LogicaNegocio.Dominio {
    public class Linea {
        public int Id { get; set; }
        public Articulo Articulo { get; set; }
        public int UnidadesSolicitadas { get; set; }
        public int PreciodUnitario { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\MovimientoStock.cs
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesDominio;

namespace LogicaNegocio.Dominio {
    public class MovimientoStock : IValidar {
        private DateTime _fecha = DateTime.Now;
        public int Id { get; set; }
        public DateTime Fecha { get { return _fecha; } set { _fecha = value; } }
        public Articulo Articulo { get; set; }
        public Usuario Usuario { get; set; }
        public TipoMovimiento TipoMovimiento { get; set; }
        public int Cantidad { get; set; }

        public void EsValido() {
            if (Cantidad <= 0) {
                throw new DatosInvalidosException("La cantidad no puede ser negativa o 0.");
            }
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\Parametro.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Dominio {
    public class Parametro {
        public int Id { get; set; }
        public string Nombre { get; set; }
        public decimal Valor { get; set; }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\Pedido.cs
using LogicaNegocio.InterfacesDominio;
using LogicaNegocio.Excepciones;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Dominio {
    public abstract class Pedido : IValidar {
        private DateOnly _fecha = DateOnly.FromDateTime(DateTime.Now);
        public static decimal _iva;
}

```

```

public int Id { get; set; }
public DateOnly Fecha { get { return _fecha; } set { _fecha = value; } }
public DateOnly FechaEntrega { get; set; }
public Cliente Cliente { get; set; }
public List<Linea> Lineas { get; set; }
public decimal Total { get; set; }
public decimal Iva { get { return _iva; } set { _iva = value; } }
public decimal Recargo { get; set; }
public string Estado { get; set; } // Pendiente, Entregado, Anulado

public virtual void EsValido() {
    if (Lineas.Count == 0) {
        throw new DatosInvalidosException("El pedido debe contener al menos un artículo");
    }
}

public abstract void CalcularRecargo(decimal a, decimal b);
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\PedidoComun.cs
using LogicaNegocio.InterfacesDominio;
using LogicaNegocio.Excepciones;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Dominio {
    public class PedidoComun : Pedido, IValidar {
        private int _plazoEstipulado;

        // > 7 días
        public int PlazoEstipulado {
            get { return _plazoEstipulado = FechaEntrega.DayNumber - Fecha.DayNumber; }
            set { _plazoEstipulado = value; }
        }

        public override void EsValido() {
            if (PlazoEstipulado < 7) {
                throw new DatosInvalidosException("El plazo de entrega para un pedido común no puede ser menor a 7 días");
            }
        }

        // si Cliente.DistanciaHastaDeposito > 100 ? 5% : 0
        public override void CalcularRecargo(decimal a, decimal b) {
            Recargo = Cliente != null && Cliente.DistanciaHastaDeposito > 100 ? a : b;
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\PedidoExpress.cs
using LogicaNegocio.InterfacesDominio;
using LogicaNegocio.Excepciones;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Dominio {

```

```

public class PedidoExpress : Pedido, IValidar {
    private int _plazoEstipulado;

    // <= 5 días
    public int PlazoEstipulado {
        get { return _plazoEstipulado = FechaEntrega.DayNumber - Fecha.DayNumber; }
        set { _plazoEstipulado = value; }
    }

    public override void EsValido() {
        if (PlazoEstipulado > 5) {
            throw new DatosInvalidosException("El plazo de entrega para un pedido express no puede ser mayor a 5 días");
        }
    }

    // si plazo = 1 ? 15% : 10%
    public override void CalcularRecargo(decimal a, decimal b) {
        Recargo = PlazoEstipulado == 1 ? a : b;
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\TipoMovimiento.cs
using LogicaNegocio.Excepciones;
using LogicaNegocio.InterfacesDominio;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Dominio {
    [Index(nameof(Nombre), IsUnique = true)]
    public class TipoMovimiento : IValidar {
        public int Id { get; set; }
        public string Nombre { get; set; }
        [Column(TypeName = "nvarchar(24)")]
        public Tipo TipoAccion { get; set; }

        public void EsValido() {
            if (string.IsNullOrEmpty(Nombre)) {
                throw new DatosInvalidosException("El nombre es requerido");
            }
        }
    }
}

public enum Tipo {
    Reducion,
    Aumento
}
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Dominio\Usuario.cs
using LogicaNegocio.InterfacesDominio;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Text.RegularExpressions;
using Microsoft.EntityFrameworkCore;
using LogicaNegocio.Excepciones;

namespace LogicaNegocio.Dominio {
    [Index(nameof(Email), IsUnique = true)]
    public class Usuario : IValidar {
        [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
        [EmailAddress]
        public string Email { get; set; } // unico, aplicar formateado
        [RegularExpression(@"^([A-Za-z]+[A-Za-z'-]*[A-Za-z]+)$")]
        public string Nombre { get; set; }
        [RegularExpression(@"^([A-Za-z]+[A-Za-z'-]*[A-Za-z]+)$")]
        public string Apellido { get; set; }
        [MinLength(6)]
        [RegularExpression(@"^(?=.*[A-Z])(?=.*[a-z])(?=.*[!.,$])[6,]$")]
        public string Contraseña { get; set; } // min 6 char, mayuscula, minuscula, char especial (punto, punto y coma, coma, signo de admiración de cierre)
        public string? ContraseniaEncriptada { get; set; }
        [Column(TypeName = "nvarchar(24)")]
        public TipoUsuario Tipo { get; set; }

        public void EsValido() {
            if(!ValidarContraseña(Contraseña)) {
                throw new DatosInvalidosException("La contraseña debe tener un mínimo de 6 caracteres, al menos una mayuscula y una minuscula y un caracter especial (punto, coma, punto y coma o signo de admiración)");
            }
        }
    }

    private bool ValidarContraseña(String contraseña) {
        var tieneNumero = new Regex(@"[0-9]+");
        var tieneMayuscula = new Regex(@"[A-Z]+");
        var tieneMinuscula = new Regex(@"[a-z]+");
        var tieneMinSeisChar = new Regex(@"[6,]");
        var tieneSimbolos = new Regex(@"[!.,?]+");

        return tieneNumero.IsMatch(contraseña) && tieneMayuscula.IsMatch(contraseña) && tieneMinuscula.IsMatch(contraseña) &&
        tieneMinSeisChar.IsMatch(contraseña) && tieneSimbolos.IsMatch(contraseña);
    }
}

public enum TipoUsuario {
    Administrador,
    Encargado,
    Estandar
}
// https://learn.microsoft.com/en-us/ef/core/modeling/value-conversions?tabs=data-annotations#pre-defined-conversions
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Excepciones\DatosInvalidosException.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Excepciones {
    public class DatosInvalidosException : Exception {
        public DatosInvalidosException() {

```

```

}

public DatosInvalidosException(string mensaje) : base(mensaje) {

}

public DatosInvalidosException(string mensaje, Exception e) : base(mensaje, e) {

}

}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Excepciones\DuplicadoException.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Excepciones {
    public class DuplicadoException : Exception {
        public DuplicadoException() {

        }

        public DuplicadoException(string mensaje) : base(mensaje) {

        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Excepciones\NoStockException.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Excepciones {
    public class NoStockException : Exception {
        public NoStockException() {

        }

        public NoStockException(string mensaje) : base(mensaje) {

        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\Excepciones\RegistroNoExisteException.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.Excepciones {
    public class RegistroNoExisteException : Exception {
        public RegistroNoExisteException() {

```

```

    }

    public RegistroNoExisteException(string mensaje) : base(mensaje) {
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesDominio\IValidar.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesDominio {
    public interface IValidar {
        void EsValido();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorio.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorio<T> {
        void Create(T obj);
        void Delete(int id);
        void Update(T obj);
        List<T> GetAll();
        T FindById(int id);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioArticulos.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioArticulos : IRepositorio<Articulo> {
        List<Articulo> OrdenarArticulosAsc();
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioClientes.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioClientes : IRepositorio<Cliente> {
        List<Cliente> BuscarPorRazonSocial(string nombre);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioLineas.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioLineas : IRepositorio<Linea>{
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioMovimientosStock.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioMovimientosStock {
        List<MovimientoStock> GetAll();
        void Create(MovimientoStock obj);
        MovimientoStock FindById(int id);
        List<MovimientoStock> BuscarMovimientosPorArticuloYTipo(int idArticulo, string tipoMovimiento, int pagina, int limitePorPagina);
        string ObtenerCantidadPorAnioYTipo();
        List<Articulo> BuscarArticulosConMovimientosPorFecha(DateTime desde, DateTime hasta,int pagina, int limitePorPagina);
        public int CantidadDeMovimientos(int idArticulo, string tipoMovimiento);
        public int CantidadDeMovimientos(DateTime desde, DateTime hasta);
        public bool TieneTipoMovimiento(int idTipoMovimiento);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioParametros.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioParametros {
        public decimal ObtenerIva();
        public decimal ObtenerLimitePorPagina();
        public decimal ObtenerLimiteTopeDeMovimientos();
    }
}

```

```

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioPedidos.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioPedidos : IRepositorio<Pedido>{
        void AnularPedido(int id);
        List<Pedido> BuscarPorFechaDeEmision(DateOnly fecha);
        List<Pedido> OrdenarPedidosAnuladosPorFechaDesc();
        List<Pedido> ListarPedidosPendientes();
        List<Cliente> BuscarClientes(decimal monto);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioTiposMovimientos.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioTiposMovimientos : IRepositorio<TipoMovimiento> {
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\InterfacesRepositorios\IRepositorioUsuarios.cs
using LogicaNegocio.Dominio;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.InterfacesRepositorios {
    public interface IRepositorioUsuarios : IRepositorio<Usuario> {
        Usuario BuscarPorEmail(string email);
        Usuario BuscarPorEmail(string email, string password);
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\obj\Debug\net8.0\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("LogicaNegocio")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+16d075e4bdc1af1f44ce24eefe9cb9e2b59eac12")]
[assembly: System.Reflection.AssemblyProductAttribute("LogicaNegocio")]
[assembly: System.Reflection.AssemblyTitleAttribute("LogicaNegocio")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\obj\Debug\net8.0\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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\obj\Release\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\obj\Release\net8.0\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("LogicaNegocio")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+f9761fd17d5acfdc29824f17f329c90bf5a7c8da")]
[assembly: System.Reflection.AssemblyProductAttribute("LogicaNegocio")]
[assembly: System.Reflection.AssemblyTitleAttribute("LogicaNegocio")]

```

```

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

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\obj\Release\net8.0\LogicaNegocio.GI
obalUsing.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;
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\ValueObjects\Calle.cs
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.ValueObjects {
    // [Owned]
    public class Calle {
        public int DireccionId { get; set; }
        public string Valor { get; init; }
        private Calle() {

        }

        public Calle(string valor, int direccionId) {
            Valor = valor;
            Validar();
            DireccionId = direccionId;
        }

        private void Validar() {
            if (string.IsNullOrEmpty(Valor))
                throw new Exception("El nombre de la calle es obligatorio");
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\ValueObjects\Ciudad.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.ValueObjects {
    public class Ciudad {
        public int DireccionId { get; set; }
        public string Valor { get; init; }
        private Ciudad() {

        }

        public Ciudad(string valor, int direccionId) {
            Valor = valor;
        }
    }
}

```

```

        Validar();
        DireccionId = direccionId;
    }

    private void Validar() {
        if (string.IsNullOrEmpty(Valor))
            throw new Exception("El nombre de la ciudad es obligatorio");
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\ValueObjects\Direccion.cs
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.ValueObjects {
    [Owned]
    public class Direccion {
        public int ClientId { get; set; }
        public Calle Calle { get; set; }
        public NumeroPuerta NumeroPuerta { get; set; }
        public Ciudad Ciudad { get; set; }

        private Direccion() {

        }

        public Direccion(int clientId, Calle calle, NumeroPuerta nroPuerta, Ciudad ciudad) {
            Calle = calle;
            NumeroPuerta = nroPuerta;
            Ciudad = ciudad;
            ClientId = clientId;
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\LogicaNegocio\ValueObjects\NumeroPuerta.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LogicaNegocio.ValueObjects {
    public class NumeroPuerta {
        public int DireccionId { get; set; }
        public string Valor { get; init; }

        private NumeroPuerta() {

        }

        public NumeroPuerta(string valor, int direccionId) {
            Valor = valor;
            Validar();
            DireccionId = direccionId;
        }
    }
}

```

```

private void Validar() {
    if (string.IsNullOrEmpty(Valor))
        throw new Exception("El numero de puerta es obligatorio");
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Program.cs
using DataTransferObjects;
using LogicaAplicacion.CasosUso;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaDatos.Repositorios;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddSession(options => {
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});

// Add services to the container.
builder.Services.AddControllersWithViews();

builder.Services.AddScoped<ICUAlta<Articulo>, CUAltaArticulo>();
builder.Services.AddScoped<ICUBaja<Articulo>, CUBajaArticulo>();
builder.Services.AddScoped<ICUListado<Articulo>, CUListadoArticulos>();
builder.Services.AddScoped<ICUModificar<Articulo>, CUModificarArticulo>();
builder.Services.AddScoped<ICUBuscarPorId<Articulo>, CUBuscarPorIdArticulo>();

builder.Services.AddScoped<ICUAlta<Usuario>, CUAltaUsuario>();
builder.Services.AddScoped<ICUBaja<Usuario>, CUBajaUsuario>();
builder.Services.AddScoped<ICUListado<Usuario>, CUListadoUsuarios>();
builder.Services.AddScoped<ICUModificar<Usuario>, CUModificarUsuario>();
builder.Services.AddScoped<ICUBuscarPorId<Usuario>, CUBuscarPorIdUsuario>();
builder.Services.AddScoped<ICUAutenticarUsuario, CUAutenticarUsuario>();

builder.Services.AddScoped<ICUListado<ClienteDTO>, CUListadoClientes>();
builder.Services.AddScoped<ICUBuscarPorRazonSocial, CUBuscarPorRazonSocial>();
builder.Services.AddScoped<ICUBuscarClientesPorMontoPedido, CUBuscarClientesPorMontoPedido>();

builder.Services.AddScoped<ICUAlta<PedidoDTO>, CUAltaPedido>();
builder.Services.AddScoped<ICUBuscarPorFechaPedido, CUBuscarPorFechaPedido>();
builder.Services.AddScoped<ICUAnularPedido, CUAnularPedido>();
builder.Services.AddScoped<ICUBuscarPorId<PedidoDTO>, CUBuscarPorIdPedido>();
builder.Services.AddScoped<ICUModificar<PedidoDTO>, CUAgregarArticuloEnPedido>();
builder.Services.AddScoped<ICUListado<Pedido>, CUListadoPedidos>();
builder.Services.AddScoped<ICUListado<PedidoNoEntregadoDTO>, CUListadoPedidosPendientes>();

builder.Services.AddScoped<IRepositoryUsuarios, RepositoryUsuarios>();
builder.Services.AddScoped<IRepositoryClientes, RepositoryClientes>();
builder.Services.AddScoped<IRepositoryArticulos, RepositoryArticulos>();
builder.Services.AddScoped<IRepositoryPedidos, RepositoryPedidos>();
builder.Services.AddScoped<IRepositoryLineas, RepositoryLineas>();
builder.Services.AddScoped<IRepositoryParametros, RepositoryParametros>();

string conStr = builder.Configuration.GetConnectionString("Caro-Zenbook");
builder.Services.AddDbContext<ObligatorioContext>(options => options.UseSqlServer(conStr));

```

```

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment()) {
    app.UseExceptionHandler("/Home/Error");
}
}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseSession();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();
//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Controllers\ArticulosController.cs
using LogicaAplicacion.CasosUso;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using Microsoft.AspNetCore.Mvc;

namespace Obligatorio.Controllers {
    public class ArticulosController : Controller {
        public ICUAlta<Articulo> CUAlta { get; set; }
        public ICUBaja<Articulo> CUBaja { get; set; }
        public ICUListado<Articulo> CUListado { get; set; }
        public ICUModificar<Articulo> CUModificar { get; set; }
        public ICUBuscarPorId<Articulo> CUBuscarPorIdArticulo { get; set; }

        public ArticulosController(ICUAlta<Articulo> cuAlta, ICUListado<Articulo> cuListado, ICUBuscarPorId<Articulo> cuBuscarPorIdArticulo,
ICUModificar<Articulo> cuModificar, ICUBaja<Articulo> cuBaja) {
            CUAlta = cuAlta;
            CUBaja = cuBaja;
            CUListado = cuListado;
            CUModificar = cuModificar;
            CUBuscarPorIdArticulo = cuBuscarPorIdArticulo;
        }

        public IActionResult Index() {
            return View(CUListado.ObtenerListado());
        }

        //-----
        //----- CREATE -----
        //-----

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

        [HttpPost]
        public ActionResult Create(Articulo nuevo) {
            try {
                CUAlta.Alta(nuevo);
                return RedirectToAction("Index", "Articulos");
            } catch (DatosInvalidosException e) {

```

```

        ViewBag.ErrorMsg = e.Message;
    } catch (DuplicadoException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }

    return View();
}

//-----
//----- UPDATE -----
//-----

public ActionResult Edit(int id) {
    Articulo a = CUBuscarPorIdArticulo.BuscarPorId(id);
    return View(a);
}

[HttpPost]
public ActionResult Edit(int id, Articulo a) {
    try {
        CUModificar.Modificar(a);
        return RedirectToAction("Index", "Articulos");
    } catch (DatosInvalidosException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (DuplicadoException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }

    return View();
}

//-----
//----- DELETE -----
//-----

public ActionResult Delete(int id) {
    Articulo a = CUBuscarPorIdArticulo.BuscarPorId(id);
    return View(a);
}

[HttpPost]
public ActionResult Delete(int id, Articulo a) {
    try {
        CUBaja.Baja(id);
        return RedirectToAction("Index", "Articulos");
    } catch (RegistroNoExisteException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }
    return View();
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Controllers\ClientesController.cs
using Microsoft.AspNetCore.Mvc;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;

```

```

using DataTransferObjects;

namespace Obligatorio.Controllers {
    public class ClientesController : Controller {
        public ICUListado<ClienteDTO> CUListado { get; set; }
        public ICUBuscarPorRazonSocial CUBuscarPorRazonSocial { get; set; }
        public ICUBuscarClientesPorMontoPedido CUBuscarClientesPorMontoPedido { get; set; }

        public ClientesController(ICUListado<ClienteDTO> cuListado, ICUBuscarPorRazonSocial cuBuscarPorRazonSocial,
ICUBuscarClientesPorMontoPedido cuBuscarClientesPorMontoPedido)
        {
            CUListado = cuListado;
            CUBuscarPorRazonSocial = cuBuscarPorRazonSocial;
            CUBuscarClientesPorMontoPedido = cuBuscarClientesPorMontoPedido;
        }

        public IActionResult Index() {
            return View(CUListado.ObtenerListado());
        }

        public ActionResult BuscarClientes() {
            return View(CUListado.ObtenerListado());
        }

        [HttpPost]
        public ActionResult BuscarClientes(string razonSocial) {
            List<ClienteDTO> clientes = CUBuscarPorRazonSocial.BuscarPorRazonSocial(razonSocial);
            if (clientes.Count == 0) ViewBag.ErrorMsg = "No existen registros";
            return View(clientes);
        }

        public ActionResult BuscarPorMonto() {
            return View(CUListado.ObtenerListado());
        }

        [HttpPost]
        public ActionResult BuscarPorMonto(decimal monto) {
            try {
                List<ClienteDTO> clientes = CUBuscarClientesPorMontoPedido.BuscarPorMontoPedido(monto);
                return View(clientes);
            } catch (RegistroNoExisteException e) {
                ViewBag.ErrorMsg = e.Message;
            } catch (Exception e) {
                ViewBag.ErrorMsg = e.Message;
            }
            return View();
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Controllers\HomeController.cs
using Obligatorio.Models;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using System;
using System.Collections.Generic;
using System.Diagnostics;

namespace Obligatorio.Controllers {
    public class HomeController : Controller {

```

```

public IActionResult Index() {
    return View();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Error() {
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
}
}

/// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Controllers\PedidosController.cs
using DataTransferObjects;
using LogicaAplicacion.CasosUso;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using Microsoft.AspNetCore.Mvc;
using Obligatorio.Filters;
using Obligatorio.Models;

namespace Obligatorio.Controllers {
    public class PedidosController : Controller {
        public ICUListado<Articulo> CUListadoArticulos { get; set; }
        public ICUListado<ClienteDTO> CUListadoClientes { get; set; }
        public ICUAlta<PedidoDTO> CUAlta { get; set; }
        public ICUBuscarPorFechaPedido CUBuscarPorFechaPedido { get; set; }
        public ICUAnularPedido CUAnularPedido { get; set; }
        public ICUBuscarPorId<PedidoDTO> CUBuscarPorIdPedido { get; set; }
        public ICUListado<PedidoNoEntregadoDTO> CUListadoPedidosPendientes { get; set; }
        public ICUModificar<PedidoDTO> CUAgregarArticulo { get; set; }

        public PedidosController(
            ICUListado<Articulo> cuListadoArticulos,
            ICUListado<ClienteDTO> cuListadoClientes,
            ICUAlta<PedidoDTO> cuAlta,
            ICUBuscarPorFechaPedido cuBuscarPorFechaPedido,
            ICUAnularPedido cuAnularPedido,
            ICUBuscarPorId<PedidoDTO> cuBuscarPorIdPedido,
            ICUListado<PedidoNoEntregadoDTO> cuListadoPedidosPendientes,
            ICUModificar<PedidoDTO> cuAgregarArticulo
        ) {
            CUListadoArticulos = cuListadoArticulos;
            CUListadoClientes = cuListadoClientes;
            CUAlta = cuAlta;
            CUBuscarPorFechaPedido = cuBuscarPorFechaPedido;
            CUAnularPedido = cuAnularPedido;
            CUBuscarPorIdPedido = cuBuscarPorIdPedido;
            CUListadoPedidosPendientes = cuListadoPedidosPendientes;
            CUAgregarArticulo = cuAgregarArticulo;
        }

        [Privado(TipoUsuarios = "Administrador")]
        public IActionResult Index() {
            return View(CUListadoPedidosPendientes.ObtenerListado());
        }

        //-----
        //----- CREATE -----
        //-----
    }
}

```

```

[Privado(TipoUsuarios = "Administrador")]
public ActionResult Create() {
    AltaPedidoViewModel vm = new AltaPedidoViewModel() {
        Articulos = CUListadoArticulos.ObtenerListado(),
        Clientes = CUListadoClientes.ObtenerListado()
    };
    return View(vm);
}

```

```

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult Create(AltaPedidoViewModel vm) {
    try {
        PedidoDTO dto = new PedidoDTO();
        dto.IdCliente = vm.IdCliente;
        dto.TipoPedido = vm.TipoPedido;
        dto.FechaEntrega = vm.FechaEntrega;
        dto.IdArticulo = vm.IdArticulo;
        dto.Cantidad = vm.Cantidad;
        dto.Estado = "Pendiente";

        CUAlta.Alta(dto);

        return RedirectToAction("Index", "Pedidos");
    } catch (DatosInvalidosException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (RegistroNoExisteException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (NoStockException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }

    vm.Articulos = CUListadoArticulos.ObtenerListado();
    vm.Clientes = CUListadoClientes.ObtenerListado();
    return View(vm);
}

```

```

//-----
//----- ANULAR -----
//-----

[Privado(TipoUsuarios = "Administrador")]
public ActionResult AnularPedidos(int id) {
    PedidoDTO p = CUBuscarPorIdPedido.BuscarPorId(id);
    return View(p);
}

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult AnularPedidos(int id, PedidoDTO p) {
    try {
        CUAnularPedido.Anular(id);
        return RedirectToAction("Index", "Pedidos");
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }
    return View();
}

//-----

```

```

//----- BUSCAR -----
//-----
[Privado(TipoUsuarios = "Administrador")]
public ActionResult BuscarPedidos() {
    return View();
}

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult BuscarPedidos(string fecha) {
    DateOnly fechaABuscar = DateOnly.Parse(fecha);
    List<PedidoNoEntregadoDTO> pedidos = CUBuscarPorFechaPedido.BuscarPorFechaPedido(fechaABuscar);
    if (pedidos.Count == 0) ViewBag.ErrorMsg = "No existen registros";
    return View(pedidos);
}

//-----
//----- AGREGAR ARTICULO -----
//-----
[Privado(TipoUsuarios = "Administrador")]
public ActionResult AgregarArticulo(int id) {
    PedidoDTO p = CUBuscarPorIdPedido.BuscarPorId(id);
    ViewBag.Articulos = CUListadoArticulos.ObtenerListado();
    return View(p);
}

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult AgregarArticulo(int id, PedidoDTO p) {
    try {
        CUAgregarArticulo.Modificar(p);
        return RedirectToAction("Index", "Pedidos");
    } catch (RegistroNoExisteException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (NoStockException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.ToString();
    }

    ViewBag.Articulos = CUListadoArticulos.ObtenerListado();
    return View(p);
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Controllers\UsuariosController.cs
using LogicaAplicacion.CasosUso;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Http;
using Obligatorio.Models;
using LogicaNegocio.Excepciones;
using Obligatorio.Filters;
using DataTransferObjects;
using Newtonsoft.Json;

namespace Obligatorio.Controllers {
    public class UsuariosController : Controller {
        public ICUAlta<Usuario> CUAAlta { get; set; }
        public ICUBaja<Usuario> CUBaja { get; set; }
    }
}

```

```

public ICUListado<Usuario> CUListado { get; set; }
public ICUModificar<Usuario> CUModificar { get; set; }
public ICUAutenticarUsuario CUAutenticarUsuario { get; set; }
public ICUBuscarPorId<Usuario> CUBuscarPorIdUsuario { get; set; }

public UsuariosController(ICUAutenticarUsuario cuAutenticarUsuario, ICUAlta<Usuario> cuAlta, ICUListado<Usuario> cuListado,
ICUModificar<Usuario> cuModificar, ICUBuscarPorId<Usuario> cuBuscarPorIdUsuario, ICUBaja<Usuario> cuBaja) {
    CUAutenticarUsuario = cuAutenticarUsuario;
    CUAlta = cuAlta;
    CUListado = cuListado;
    CUModificar = cuModificar;
    CUBuscarPorIdUsuario = cuBuscarPorIdUsuario;
    CUBaja = cuBaja;
}

[Privado(TipoUsuarios = "Administrador")]
public IActionResult Index() {
    return View(CUListado.ObtenerListado());
}

//-----
//----- INGRESAR -----
//-----

public IActionResult Ingresar() {
    return View();
}

[HttpPost]
public IActionResult Ingresar(IngresarViewModel model) {
    if (!ModelState.IsValid) {
        ViewBag.Error = "Email o Contraseña no pueden ser vacíos";
        return View();
    }

    UsuarioDTO usuarioDTO = CUAutenticarUsuario.Autenticar(model.Email, model.Password);

    if (usuarioDTO != null) {
        HttpContext.Session.SetString("Email", model.Email);
        HttpContext.Session.SetString("TIPOUSUARIO", usuarioDTO.Tipo);
        return RedirectToAction("Index", "Home");
    }

    HttpContext.Session.Clear();

    ViewBag.Error = "Las credenciales no son correctas.";
    return View();
}

public IActionResult CerrarSesion() {
    HttpContext.Session.Clear();
    return RedirectToAction("Index", "Home");
}

//-----
//----- CREATE -----
//-----

[Privado(TipoUsuarios = "Administrador")]
public ActionResult Create() {
    return View();
}

```

```

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult Create(Usuario nuevo) {
    try {
        CUAlta.Alta(nuevo);
        return RedirectToAction("Index", "Usuarios");
    } catch (DatosInvalidosException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (DuplicadoException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }

    return View();
}

//-----
//----- UPDATE -----
//-----

[Privado(TipoUsuarios = "Administrador")]
public ActionResult Edit(int id) {
    Usuario u = CUBuscarPorIdUsuario.BuscarPorId(id);
    return View(u);
}

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult Edit(int id, Usuario u) {
    try {
        CUModificar.Modificar(u);
        return RedirectToAction("Index", "Usuarios");
    } catch (DatosInvalidosException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (DuplicadoException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {
        ViewBag.ErrorMsg = e.Message;
    }

    return View();
}

//-----
//----- DELETE -----
//-----

[Privado(TipoUsuarios = "Administrador")]
public ActionResult Delete(int id) {
    Usuario u = CUBuscarPorIdUsuario.BuscarPorId(id);
    return View(u);
}

[Privado(TipoUsuarios = "Administrador")]
[HttpPost]
public ActionResult Delete(int id, Usuario u) {
    try {
        CUBaja.Baja(id);
        return RedirectToAction("Index", "Usuarios");
    } catch (RegistroNoExisteException e) {
        ViewBag.ErrorMsg = e.Message;
    } catch (Exception e) {

```

```

        ViewBag.ErrorMsg = e.Message;
    }
    return View();
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Filters\PrivadoAttribute.cs
using System;
using System.Linq;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Filters;

namespace Obligatorio.Filters {
    public class PrivadoAttribute : Attribute, IActionFilter {
        public string TipoUsuarios { get; set; }

        public void OnActionExecuted(ActionExecutedContext context) {
        }

        public void OnActionExecuting(ActionExecutingContext context) {
            bool usuarioAutenticado = !string.IsNullOrEmpty(context.HttpContext.Session.GetString("TIPOUSUARIO"));
            if (!usuarioAutenticado) {
                context.Result = new RedirectResult("/usuarios/ingresar");
            } else {
                if (!string.IsNullOrEmpty(TipoUsuarios)) {
                    if (!TipoUsuarios.Split(",").Contains(context.HttpContext.Session.GetString("TIPOUSUARIO")))
                        context.Result = new RedirectResult("/home/error");
                }
            }
        }
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Models\AltaArticuloViewModel.cs
namespace Obligatorio.Models {
    public class AltaArticuloViewModel {
    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Models\AltaPedidoViewModel.cs
using DataTransferObjects;
using LogicaNegocio.Dominio;

namespace Obligatorio.Models {
    public class AltaPedidoViewModel {
        public List<ClienteDTO>? Clientes { get; set; }
        public int IdCliente { get; set; }
        public List<Articulo>? Articulos { get; set; }
        public int IdArticulo { get; set; }
        public DateOnly FechaEntrega { get; set; }
        public string TipoPedido { get; set; }
        public int Cantidad { get; set; }

    }
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Models\ErrorViewModel.cs
namespace Obligatorio.Models {
    public class ErrorViewModel {
        public string RequestId { get; set; }
        public string Mensaje { get; set; }
    }
}
```

```

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

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Models\IngresarViewModel.cs
using System.ComponentModel.DataAnnotations;

namespace Obligatorio.Models {
    public class IngresarViewModel {
        [Required]
        [EmailAddress]
        public string Email { get; set; }

        [Required]
        public string Password { get; set; }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Debug\net8.0\Obligatorio.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("Obligatorio")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Debug")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+16d075e4bcd1af1f44ce24eefe9cb9e2b59eac12")]
[assembly: System.Reflection.AssemblyProductAttribute("Obligatorio")]
[assembly: System.Reflection.AssemblyTitleAttribute("Obligatorio")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Debug\net8.0\Obligatorio.GlobalUsing.sg.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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Debug\net8.0\Obligatorio.RazorAssem
blyInfo.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("Microsoft.AspNetCore.Mvc.ApplicationParts.ConsolidatedAssembly
ApplicationPartFact" +
"ory, Microsoft.AspNetCore.Mvc.Razor")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Release\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")]
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Release\net8.0\Obligatorio.AssemblyI
nfo.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("Obligatorio")]
[assembly: System.Reflection.AssemblyConfigurationAttribute("Release")]
[assembly: System.Reflection.AssemblyFileVersionAttribute("1.0.0.0")]
[assembly: System.Reflection.AssemblyInformationalVersionAttribute("1.0.0+f9761fd17d5acfdc29824f17f329c90bf5a7c8da")]
[assembly: System.Reflection.AssemblyProductAttribute("Obligatorio")]
[assembly: System.Reflection.AssemblyTitleAttribute("Obligatorio")]
[assembly: System.Reflection.AssemblyVersionAttribute("1.0.0.0")]

```

```

// Generado por la clase WriteCodeFragment de MSBuild.

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Release\net8.0\Obligatorio.GlobalUsin
gs.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;
//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\obj\Release\net8.0\Obligatorio.RazorAsse
mablyInfo.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("Microsoft.AspNetCore.Mvc.ApplicationParts.ConsolidatedAssembly
ApplicationPartFact" +
"ory, Microsoft.AspNetCore.Mvc.Razor")]

// Generado por la clase WriteCodeFragment de MSBuild.

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Views\Shared\Error.cshtml.cs
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using System.Diagnostics;

namespace Obligatorio.Views.Shared
{
    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
    [IgnoreAntiforgeryToken]
    public class ErrorModel : PageModel
    {
        public string? RequestId { get; set; }

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

        private readonly ILogger<ErrorModel> _logger;

```

```

public ErrorModel(ILogger<ErrorModel> logger)
{
    _logger = logger;
}

public void OnGet()
{
    RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier;
}
}

}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\Obligatorio\Views\Shared\Privacy.cshtml.cs
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;

namespace Obligatorio.Views.Shared
{
    public class PrivacyModel : PageModel
    {
        private readonly ILogger<PrivacyModel> _logger;

        public PrivacyModel(ILogger<PrivacyModel> logger)
        {
            _logger = logger;
        }

        public void OnGet()
        {
        }
    }
}

}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Program.cs
using DataTransferObjects;
using LogicaAplicacion.CasosUso;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaDatos.Repositorios;
using LogicaNegocio.Dominio;
using LogicaNegocio.InterfacesRepositorios;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.EntityFrameworkCore;
using Microsoft.IdentityModel.Tokens;

namespace WebAPI
{
    public class Program {
        public static void Main(string[] args) {
            var builder = WebApplication.CreateBuilder(args);

            // Add services to the container.
            builder.Services.AddScoped<ICUOrdenarArticulosAsc, CUOrdenarArticulosAsc>();
            builder.Services.AddScoped<ICUOrdenarPedidosAnuladosDesc, CUOrdenarPedidosAnuladosDesc>();

            builder.Services.AddScoped<ICUAlta<TipoMovimientoDTO>, CUAltaTipoMovimiento>();
            builder.Services.AddScoped<ICUBaja<TipoMovimientoDTO>, CUBajaTipoMovimiento>();
            builder.Services.AddScoped<ICUListado<TipoMovimientoDTO>, CUListadoTiposMovimientos>();
            builder.Services.AddScoped<ICUModificar<TipoMovimientoDTO>, CUModificarTipoMovimiento>();
            builder.Services.AddScoped<ICUBuscarPorId<TipoMovimientoDTO>, CUBuscarPorIdTipoMovimiento>();
        }
    }
}

```

```

builder.Services.AddScoped<ICUAlta<MovimientoStockDTO>, CUAltaMovimientoStock>();
builder.Services.AddScoped<CUListado<MovimientoStockIndexDTO>, CUListadoMovimientosStock>();
builder.Services.AddScoped<CUBuscarPorId<MovimientoStockDTO>, CUBuscarPorIdMovimientoStock>();
builder.Services.AddScoped<CUBuscarPorFechaMovimiento, CUBuscarPorFechaMovimiento>();
builder.Services.AddScoped<CUBuscarPorArticuloYTipoMovimiento, CUBuscarPorArticuloYTipoMovimiento>();
builder.Services.AddScoped<CUResumenMovimientos, CUResumenMovimientos>();
builder.Services.AddScoped<ICUCantidadDePaginas, CUCantidadDePaginas>();

builder.Services.AddScoped<ICUAutenticarUsuario, CUAutenticarUsuario>();

builder.Services.AddScoped<IRepositoryArticulos, RepositoryArticulos>();
builder.Services.AddScoped<IRepositoryPedidos, RepositoryPedidos>();
builder.Services.AddScoped<IRepositoryUsuarios, RepositoryUsuarios>();
builder.Services.AddScoped<IRepositoryTiposMovimientos, RepositoryTiposMovimientos>();
builder.Services.AddScoped<IRepositoryMovimientosStock, RepositoryMovimientosStock>();
builder.Services.AddScoped<IRepositoryParametros, RepositoryParametros>();

string conStr = builder.Configuration.GetConnectionString("Caro-Zenbook");
builder.Services.AddDbContext<ObligatorioContext>(options => options.UseSqlServer(conStr));

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

// Configuraciòn autenticaciòn
var claveSecreta = "ZWRpw6fDo28gZW0gY29tcHV0YWRvcme=";
builder.Services.AddAuthentication(aut => {
    aut.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
    aut.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
})
.AddJwtBearer(aut => {
    aut.RequireHttpsMetadata = false;
    aut.SaveToken = true;
    aut.TokenValidationParameters = new TokenValidationParameters {
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(System.Text.Encoding.ASCII.GetBytes(claveSecreta)),
        ValidateIssuer = false,
        ValidateAudience = false
    };
});
};

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment()) {
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseAuthorization();

app.MapControllers();

app.Run();
}
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Controllers\ArticulosController.cs
using DataTransferObjects;

```

```

using LogicaAplicacion.InterfacesCasosUso;
using Microsoft.AspNetCore.Mvc;

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

namespace WebAPI.Controllers {
    [Route("api/[controller]")]
    [ApiController]
    public class ArticulosController : ControllerBase {
        public ICUOrdenarArticulosAsc CUOrdenarArticulosAsc { get; set; }
        public ArticulosController(ICUOrdenarArticulosAsc cuOrdenarArticulosAsc) {
            CUOrdenarArticulosAsc = cuOrdenarArticulosAsc;
        }

        // GET: api/<ArticulosController>
        [HttpGet]
        public IActionResult Get() {
            try {
                List<ArticuloDTO> articuloDTOs = CUOrdenarArticulosAsc.OrdenarPorNombreAsc();
                if (articuloDTOs.Any()) {
                    // 200 ok
                    return Ok(articuloDTOs);
                } else {
                    // 404 - Not Found
                    return NotFound("No existen artículos.");
                }
            } catch (Exception ex) {
                // 500 - Internal Server Error
                return StatusCode(500, "Ocurrió un error en el servidor.");
            }
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Controllers\MovimientosStockController.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Dominio;
using LogicaNegocio.Excepciones;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;

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

namespace WebAPI.Controllers {
    [Route("api/[controller]")]
    [ApiController]
    public class MovimientosStockController : ControllerBase {
        public ICUAlta<MovimientoStockDTO> CUAlta { get; set; }
        public ICUListado<MovimientoStockIndexDTO> CUListado { get; set; }
        public ICUBuscarPorId<MovimientoStockDTO> CUBuscarPorIdMS { get; set; }
        public ICUBuscarPorFechaMovimiento CUBuscarPorFecha { get; set; }
        public ICUBuscarPorArticuloYTipoMovimiento CUBuscarPorArticuloYTipo { get; set; }
        public ICUResumenMovimientos CUREsumenMovimientos { get; set; }
        public ICUCantidadDePaginas CUCantidadDePaginas { get; set; }

        public MovimientosStockController(
            ICUAlta<MovimientoStockDTO> cuAlta,
            ICUListado<MovimientoStockIndexDTO> cuListado,
            ICUBuscarPorId<MovimientoStockDTO> cuBuscarPorIdMS,

```

```

ICUBuscarPorFechaMovimiento cuBuscarPorFecha,
ICUBuscarPorArticuloYTipoMovimiento cuBuscarPorArticuloYTipo,
ICUResumenMovimientos cuResumenMovimientos,
ICUCantidadDePaginas cuCantidadPaginas
){
    CUAlta = cuAlta;
    CUListado = cuListado;
    CUBuscarPorIdMS = cuBuscarPorIdMS;
    CUBuscarPorFecha = cuBuscarPorFecha;
    CUBuscarPorArticuloYTipo = cuBuscarPorArticuloYTipo;
    CUResumenMovimientos = cuResumenMovimientos;
    CUCantidadDePaginas = cuCantidadPaginas;
}

// GET: api/<MovimientosStockController>
[HttpGet]
public IActionResult Get() {
    List<MovimientoStockIndexDTO> movimientosStock = CUListado.ObtenerListado();
    return Ok(movimientosStock);
}

// GET api/<MovimientosStockController>/5
[HttpGet("{id}", Name = "BuscarPorId")]
public IActionResult Get(int id) {
    if (id <= 0) return BadRequest("El id debe ser positivo");
    MovimientoStockDTO msDTO = CUBuscarPorIdMS.BuscarPorId(id);
    if (msDTO == null) return NotFound("El movimiento de stock no existe");
    return Ok(msDTO);
}

//-----
//----- CREATE -----
//-----

// POST api/<MovimientosStockController>
[HttpPost]
[Authorize(Roles = "Encargado")]
public IActionResult Post([FromBody] MovimientoStockDTO msDTO) {
    if (msDTO == null) return BadRequest("Faltan datos requeridos para el alta");
    if (msDTO.Articuloid <= 0) return BadRequest("El artículo es requerido para el alta");

    try {
        CUAlta.Alta(msDTO);
        return CreatedAtRoute("BuscarPorId", new { id = msDTO.Id }, msDTO);
    } catch (DatosInvalidosException e) {
        return BadRequest(e.Message);
    } catch (DuplicadoException e) {
        return BadRequest(e.Message);
    } catch (Exception e) {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}

//-----
//----- BUSCAR -----
//-----

[HttpGet("MovimientosPorFecha/{desde}/{hasta}/{page}")]
[Authorize(Roles = "Encargado")]
public IActionResult MovimientosPorFecha(string desde, string hasta, int page) {
    DateTime fechaDesde = DateTime.Parse(desde);

```

```

DateTime fechaHasta = DateTime.Parse(hasta);

if (desde == null || hasta == null) return BadRequest("Las fechas son requeridas.");
try {
    List<ArticuloDTO> articulos = CUBuscarPorFecha.BuscarPorFecha(fechaDesde, fechaHasta, page);
    if (articulos == null) return NotFound("No existen movimientos para las fechas seleccionadas.");
    return Ok(articulos);
} catch {
    return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
}
}

[HttpGet("MovimientosPorArticuloYTipo/{articuloId}/{tipoMovimiento}/{page}")]
[Authorize(Roles = "Encargado")]
public IActionResult MovimientosPorArticuloYTipo(int articuloId, string tipoMovimiento, int page) {
    if (articuloId <= 0 || String.IsNullOrEmpty(tipoMovimiento)) return BadRequest("Artículo y tipo de movimiento son requeridos.");
    try {
        List<MovimientoStockIndexDTO> movimientos = CUBuscarPorArticuloYTipo.BuscarMovimientosPorArticuloYTipo(articuloId, tipoMovimiento, page);
        if (movimientos == null) return NotFound("No existen movimientos para la combinacion de Artículo y tipo de movimiento seleccionados.");
        return Ok(movimientos);
    } catch {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}

//-----
//----- RESUMEN -----
//-----

[HttpGet("ResumenMovimientos")]
[Authorize(Roles = "Encargado")]
public IActionResult ResumenMovimientos() {
    try {
        List<MovimientoCantidadPorAnioYTipoDTO> movimientos = CUResumenMovimientos.ObtenerResumen();
        return Ok(movimientos);
    } catch (RegistroNoExisteException e) {
        return NotFound(e.Message);
    } catch {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}

//-----
//----- PAGINACIÓN -----
//-----

[HttpGet("CantidadDePaginas/{articuloId}/{tipoMovimiento}")]
public IActionResult CantidadDePaginas(int articuloId, string tipoMovimiento) {
    try {
        double cantidad = CUCantidadDePaginas.ObtenerCantidadDePaginas(articuloId, tipoMovimiento);
        return Ok(cantidad);
    } catch (Exception ex) {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}

//-----
//----- PAGINACIÓN -----
//-----

[HttpGet("CantidadDePaginasFechas/{articuloId}/{tipoMovimiento}")]
public IActionResult CantidadDePaginas(string articuloId, string tipoMovimiento) {
    try {

```

```

        double cantidad = CUCantidadDePaginas.ObtenerCantidadDePaginas(articuloid, tipoMovimiento);
        return Ok(cantidad);
    } catch (Exception ex) {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}
}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Controllers\PedidosController.cs
using DataTransferObjects;
using LogicaAplicacion.CasosUso;
using LogicaAplicacion.InterfacesCasosUso;
using Microsoft.AspNetCore.Mvc;

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

namespace WebAPI.Controllers {
    [Route("api/[controller]")]
    [ApiController]
    public class PedidosController : ControllerBase {
        public ICUOrdenarPedidosAnuladosDesc CUOrdenarPedidosAnuladosDesc { get; set; }

        public PedidosController(ICUOrdenarPedidosAnuladosDesc cuOrdenarPedidosAnuladosDesc) {
            CUOrdenarPedidosAnuladosDesc = cuOrdenarPedidosAnuladosDesc;
        }

        // GET: api/<PedidosController>
        [HttpGet]
        public IActionResult Get() {
            try {
                List<PedidoNoEntregadoDTO> pedidoDTOS = CUOrdenarPedidosAnuladosDesc.OrdenarPorFechaDesc();
                if(pedidoDTOS.Any()) {
                    // 200 ok
                    return Ok(pedidoDTOS);
                } else {
                    // 404 - Not Found
                    return NotFound("No existen pedidos.");
                }
            } catch (Exception ex) {
                // 500 - Internal Server Error
                return StatusCode(500, "Ocurrió un error en el servidor.");
            }
        }
    }
}

//// File:
C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Controllers\TiposMovimientosController.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Excepciones;
using Microsoft.AspNetCore.Mvc;

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

namespace WebAPI.Controllers {
    [Route("api/[controller]")]
    [ApiController]
    public class TiposMovimientosController : ControllerBase {
        public ICUAlta<TipoMovimientoDTO> CUAlta { get; set; }

```

```

public ICUBaja<TipoMovimientoDTO> CUBaja { get; set; }
public ICUListado<TipoMovimientoDTO> CUListado { get; set; }
public ICUModificar<TipoMovimientoDTO> CUModificar { get; set; }
public ICUBuscarPorId<TipoMovimientoDTO> CUBuscarPorIdTM { get; set; }

public TiposMovimientosController(
    ICUALta<TipoMovimientoDTO> cuAlta,
    ICUBaja<TipoMovimientoDTO> cuBaja,
    ICUListado<TipoMovimientoDTO> cuListado,
    ICUModificar<TipoMovimientoDTO> cuModificar,
    ICUBuscarPorId<TipoMovimientoDTO> cuBuscarPorIdTM
)
{
    CUAlta = cuAlta;
    CUBaja = cuBaja;
    CUListado = cuListado;
    CUModificar = cuModificar;
    CUBuscarPorIdTM = cuBuscarPorIdTM;
}

// GET: api/<TiposMovimientosController>
[HttpGet]
public IActionResult Get() {
    List<TipoMovimientoDTO> tm = CUListado.ObtenerListado();
    return Ok(tm);
}

// GET api/<TiposMovimientosController>/5
[HttpGet("{id}", Name = "BuscarPorIdTM")]
public IActionResult Get(int id) {
    if (id <= 0) return BadRequest("El id debe ser positivo");
    TipoMovimientoDTO tm = CUBuscarPorIdTM.BuscarPorId(id);
    if (tm == null) return NotFound("El tipo de movimiento no existe");
    return Ok(tm);
}

//-----
//----- CREATE -----
//-----

// POST api/<TiposMovimientosController>
[HttpPost]
public IActionResult Post([FromBody] TipoMovimientoDTO tmDTO) {
    if (tmDTO == null) return BadRequest("Faltan datos requeridos para el alta");

    try {
        CUAlta.Alta(tmDTO);
        return CreatedAtRoute("BuscarPorIdTM", new { id = tmDTO.Id }, tmDTO);
    } catch (DatosInvalidosException e) {
        return BadRequest(e.Message);
    } catch (DuplicadoException e) {
        return BadRequest(e.Message);
    } catch (Exception e) {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}

//-----
//----- UPDATE -----
//-----

```

```

// PUT api/<TiposMovimientosController>/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody] TipoMovimientoDTO tmDTO) {
    if (id <= 0) return BadRequest("El id debe ser positivo");
    if (tmDTO == null) return BadRequest("Faltan datos requeridos para la modificación");

    try {
        CUModificar.Modificar(tmDTO);
        return Ok(tmDTO);
    } catch (DatosInvalidosException e) {
        return BadRequest(e.Message);
    } catch (DuplicadoException e) {
        return BadRequest(e.Message);
    } catch (Exception e) {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}

//-----
//----- DELETE -----
//-----

// DELETE api/<TiposMovimientosController>/5
[HttpDelete("{id}")]
public IActionResult Delete(int id) {
    if (id <= 0) return BadRequest("El id debe ser positivo");

    try {
        TipoMovimientoDTO aBorrar = CUBuscarPorIdTM.BuscarPorId(id);
        if (aBorrar == null) return NotFound("El tipo de movimiento no existe");
        CUBaja.Baja(id);
        return NoContent();
    } catch (RegistroNoExisteException e) {
        return BadRequest(e.Message);
    } catch (DatosInvalidosException e) {
        return BadRequest(e.Message);
    } catch (Exception e) {
        return StatusCode(500, e.Message);
    }
}
}

}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Controllers\UsuariosController.cs
using DataTransferObjects;
using LogicaAplicacion.InterfacesCasosUso;
using LogicaNegocio.Excepciones;
using Microsoft.AspNetCore.Mvc;
using System.Runtime.InteropServices;
using WebAPI.Token;

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

namespace WebAPI.Controllers {
    [Route("api/[controller]")]
    [ApiController]
    public class UsuariosController : ControllerBase {

        public ICUAutenticarUsuario CUAutenticar { get; set; }

        public UsuariosController(ICUAutenticarUsuario cuAutenticar)
        {

```

```

        CUAutenticar = cuAutenticar;
    }

//-----
//----- INGRESAR -----
//-----

[HttpPost("Ingresar")]
public IActionResult Ingresar([FromBody] UsuarioDTO usuarioDTO) {
    if (usuarioDTO == null) return BadRequest("Datos incorrectos");

    try {
        usuarioDTO = CUAutenticar.Autenticar(usuarioDTO.Email, usuarioDTO.Password);
        if (usuarioDTO == null) return NotFound("El usuario no existe");

        UsuarioAutenticadoDTO usuarioAutenticado = new UsuarioAutenticadoDTO();
        usuarioAutenticado.Tipo = usuarioDTO.Tipo;
        usuarioAutenticado.Email = usuarioDTO.Email;
        usuarioAutenticado.Token = TokenHandler.GenerarToken(usuarioDTO);

        return Ok(usuarioAutenticado);
    } catch (DatosInvalidosException e) {
        return BadRequest(e.Message);
    } catch {
        return StatusCode(500, "Ocurrió un error inesperado en el servidor. Reintente más tarde.");
    }
}
}

}

//// File: C:\Users\caro0\Documents\ORT\P3\Obligatorio\Obligatorio2\ObligatorioP3_2\Obligatorio_WebAPI\WebAPI\Token\TokenHandler.cs
using DataTransferObjects;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;

namespace WebAPI.Token {
    public class TokenHandler {
        public static string GenerarToken(UsuarioDTO usu) {

            JwtSecurityTokenHandler tokenHandler = new JwtSecurityTokenHandler();

            //clave secreta, generalmente se incluye en el archivo de configuración
            //Debe ser un vector de bytes
            byte[] clave = Encoding.ASCII.GetBytes("ZWRpw6fDo28gZW0gY29tcHV0YWWRvcmE=");

            //Se incluye un claim para el email
            SecurityTokenDescriptor tokenDescriptor = new SecurityTokenDescriptor {
                Subject = new ClaimsIdentity(new Claim[] {
                    new Claim(ClaimTypes.Email, usu.Email),
                    new Claim(ClaimTypes.Role, usu.Tipo)
                }),
                Expires = DateTime.UtcNow.AddMonths(1),
                SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(clave),
                    SecurityAlgorithms.HmacSha256Signature)
            };
            var token = tokenHandler.CreateToken(tokenDescriptor);

            return tokenHandler.WriteToken(token);
        }
    }
}

```