


---

# TAREA CLASE 12

---

## TAREAS CAPACITACIÓN C#

 **Ezequiel Remus**  
ezequielremus@gmail.com

### RESUMEN

Siguiendo con la aplicación web de Stock, lo que vamos a hacer ahora es modificar el Controller de Artículos, Depositos y Stock con la intención de poder hacer una tabla que nos muestre los datos respectivos en cada vista.

## 1. Mostrando Datos

### 1.1. Enunciado

Reutilizando los controladores de la tarea Anterior agregar que cada controlador muestre en su acción Index los datos del modelo que esta "Controlando".

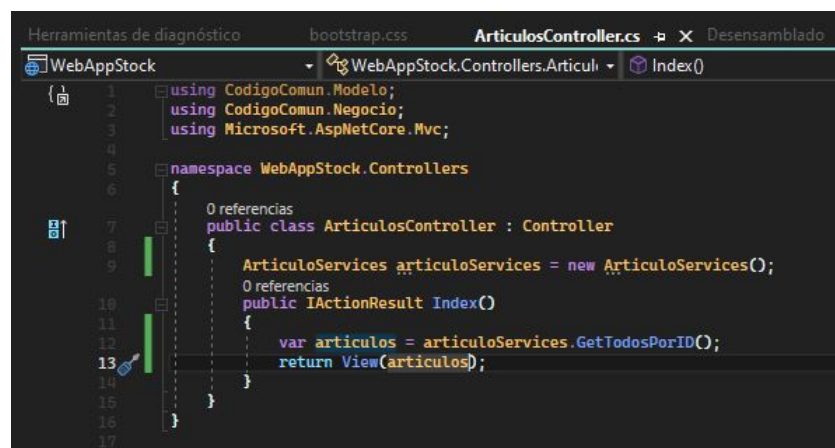
Por ejemplo, el controlador **DepositoController** me tiene que mostrar todos los depósitos que tengo en la base. Idem para *articulo* y *stock*

En el Stock debería mostrarme en la tabla el *nombre del articulo*, el *código del articulo* y en el *deposito* el *nombre*.

Enviar captura de pantalla de las 3 Views index donde se visualicen todos los datos.

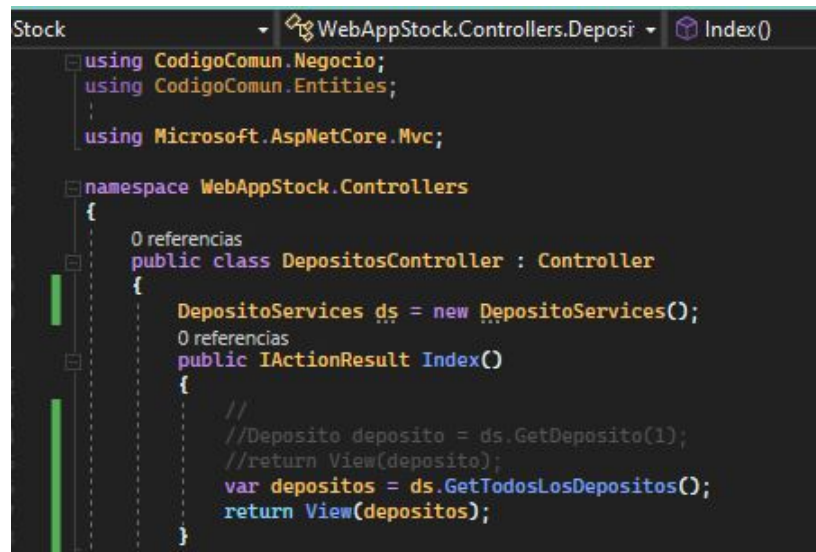
### 1.2. Solución

Lo que primero tenemos que hacer es modificar los controllers. En los tres casos hay que realizar cosas muy similares, es instanciar un servicio y a través de la acción retornar una lista de cada entidad.



```
Herramientas de diagnóstico bootstrap.css ArticulosController.cs X Desensamblado
WebAppStock WebAppStock.Controllers.Articulos Index()
1 using CodigoComun.Modelo;
2 using CodigoComun.Negocio;
3 using Microsoft.AspNetCore.Mvc;
4
5 namespace WebAppStock.Controllers
6 {
7     0 referencias
8     public class ArticulosController : Controller
9     {
10         ArticuloServices articuloServices = new ArticuloServices();
11         0 referencias
12         public IActionResult Index()
13         {
14             var articulos = articuloServices.GetTodosPorID();
15             return View(articulos);
16         }
17     }
```

Figura 1: Controller del Articulo

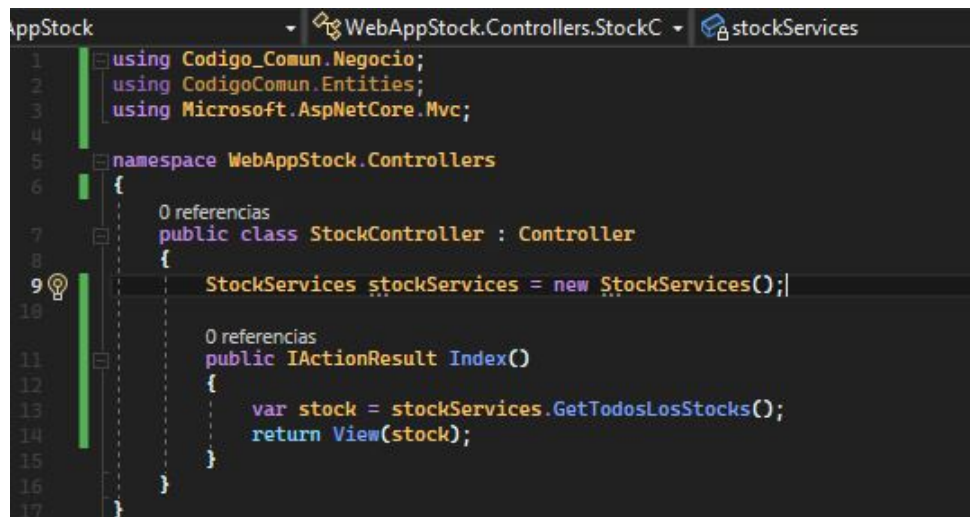


```
Stock
WebAppStock.Controllers.DepositosController
Index()

using CodigoComun.Negocio;
using CodigoComun.Entities;
using Microsoft.AspNetCore.Mvc;

namespace WebAppStock.Controllers
{
    0 referencias
    public class DepositosController : Controller
    {
        DepositoServices ds = new DepositoServices();
        0 referencias
        public IActionResult Index()
        {
            //
            //Deposito deposito = ds.GetDeposito(1);
            //return View(deposito);
            var depositos = ds.GetTodosLosDepositos();
            return View(depositos);
        }
    }
}
```

Figura 2: Controller del Deposito



```
AppStock
WebAppStock.Controllers.StockController
stockServices

1 using CodigoComun.Negocio;
2 using CodigoComun.Entities;
3 using Microsoft.AspNetCore.Mvc;
4
5 namespace WebAppStock.Controllers
6 {
7     0 referencias
8     public class StockController : Controller
9     {
10         StockServices stockServices = new StockServices();
11
12         0 referencias
13         public IActionResult Index()
14         {
15             var stock = stockServices.GetTodosLosStocks();
16             return View(stock);
17         }
18     }
19 }
20 }
```

Figura 3: Controller del Stock

Luego, debemos modificar las vistas. En este caso vamos a hacer una tabla basandonos en Bootstrap (ya que viene incorporada en el proyecto).

Para el articulo, solo usaremos el modelo. Recorriendo la lista vamos tomando cada campo e incorporandolo a la tabla

```

Index.cshtml X Index.cshtml Index.cshtml
1  @*
2  For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?linkid=391809
3  *@
4  @model Articulo;
5  @usingCodigoComun.Modelo;
6  @modelIEnumerable<Articulo>;
7  @{}
8  <div class="container">
9  <h1 class="display-1 text-dark">
10  Index <small class="text-white bg-primary bg-opacity-35">Articulos</small>
11  </h1>
12  <br/><br/>
13
14  <table class="table table-striped">
15  <thead class="text-white text-center">
16  <tr>
17  <th class="bg-dark" scope="col">Id</th>
18  <th class="bg-primary" scope="col">Nombre</th>
19  <th class="bg-dark" scope="col">Marca</th>
20  <th class="bg-primary" scope="col">Proveedor</th>
21  <th class="bg-dark" scope="col">MinimoStock</th>
22  <th class="bg-primary" scope="col">Codigo</th>
23  </tr>
24  </thead>
25  @foreach (var i in Model)
26  {
27  <tbody class="text-center">
28  <tr>
29  <th class="bg-light p-2 text-dark bg-opacity-75" scope="row">@i.Id</th>
30  <td class="bg-primary p-2 text-white bg-opacity-50">@i.Nombre</td>
31  <td class="bg-light p-2 text-dark bg-opacity-50">@i.Marca</td>
32  <td class="bg-primary p-2 text-white bg-opacity-50">@i.Proveedor</td>
33  <td class="bg-light p-2 text-dark bg-opacity-50">@i.MinimoStock</td>
34  <td class="bg-primary p-2 text-white bg-opacity-50">@i.Codigo</td>
35  </tr>
36  </tbody>
37  }
38  </table>
39  </div>

```

Figura 4: View del Articulo

Para el deposito, hacemos lo mismo que para articulo, pero con los datos respectivos de un Deposito.

```

4  @usingCodigoComun.Entities;
5  @modelIEnumerable<Deposito>;
6  @{}
7
8  <div class="container">
9  <h1 class="display-1 text-dark">
10  Index <small class="text-white bg-secondary bg-opacity-35">Depositos</small>
11  </h1>
12  <br/><br/>
13
14  <table class="table table-striped">
15  <thead class="text-white text-center">
16  <tr>
17  <th class="bg-dark" scope="col">Id</th>
18  <th class="bg-secondary" scope="col">Nombre</th>
19  <th class="bg-dark" scope="col">Dirección</th>
20  <th class="bg-secondary" scope="col">Capacidad</th>
21  </tr>
22  </thead>
23  @foreach (var i in Model)
24  {
25  <tbody class="text-center">
26  <tr>
27  <th class="bg-light p-2 text-dark bg-opacity-75" scope="row">@i.Id</th>
28  <td class="bg-secondary p-2 text-white bg-opacity-50">@i.Nombre</td>
29  <td class="bg-light p-2 text-dark bg-opacity-50">@i.Direccion</td>
30  <td class="bg-secondary p-2 text-white bg-opacity-50">@i.Capacidad</td>
31  </tr>
32  </tbody>
33  }
34  </table>
35  </div>

```

Figura 5: View del Deposito

Luego, en el caso de la vista de Stock, no solo debemos pasarle el modelo, sino que para poder mostrar los datos de los nombres del deposito, el articulo y el código del articulo debemos aprovechar que podemos escribir código C#. Entonces, debemos instanciar un servicio de un deposito y de un articulo e instanciar tambien un objeto de cada para poder obtener los articulos y depositos en cada stock.

El código final de la vista de Stock es la de la 1.2.

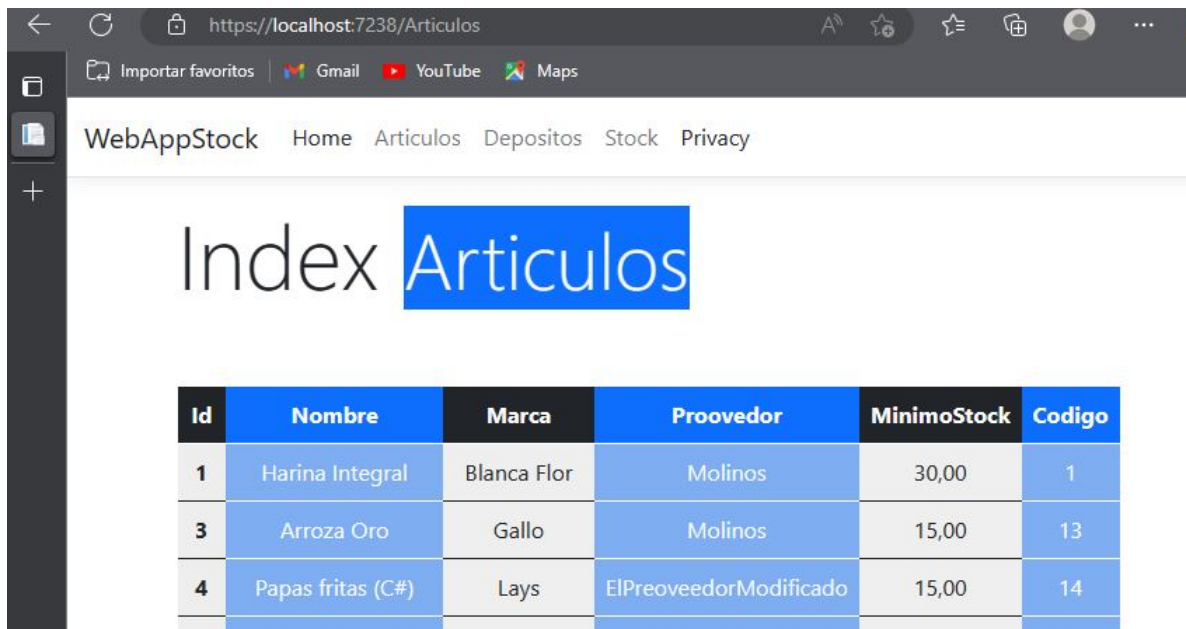
```
@usingCodigoComun.Modelo;
@usingCodigoComun.Entities;
@usingCodigoComun.Negocio;
@modelIEnumerable<Stock>;

@{
    DepositoServices DS = new DepositoServices();
    Deposito deposito = new Deposito();
    ArticuloServices AS = new ArticuloServices();
    Articulo articulo = new Articulo();
}

<div class="container">
    <h1 class="display-1 text-dark">
        Index <small class="text-white bg-success bg-opacity-35"> Stock </small>
    </h1>
    <br/><br/>
    <table class="table table-striped">
        <thead class="text-white text-center">
            <tr>
                <th class="bg-dark" scope="col">Id</th>
                <th class="bg-success" scope="col">Deposito</th>
                <th class="bg-dark" scope="col">Articulo</th>
                <th class="bg-success" scope="col">Codigo Articulo</th>
            </tr>
        </thead>
        <tbody class="text-center">
            @foreach (var i in Model)
            {
                deposito = DS.GetDeposito(Convert.ToInt32(i.IdDeposito));
                articulo = AS.GetPorID(Convert.ToInt32(i.IdArticulo));
                <tr>
                    <th class="bg-light p-2 text-dark bg-opacity-75" scope="row">@i.Id</th>
                    <td class="bg-success p-2 text-white bg-opacity-50">@deposito.Nombre</td>
                    <td class="bg-light p-2 text-dark bg-opacity-50">@articulo.Nombre</td>
                    <td class="bg-success p-2 text-white bg-opacity-50">@articulo.Codigo</td>
                </tr>
            }
        </tbody>
    </table>
</div>
```

Figura 6: View del Stock

Luego, corremos la página para poder visualizar cada una de las vistas



Id	Nombre	Marca	Proveedor	MinimoStock	Codigo
1	Harina Integral	Blanca Flor	Molinos	30,00	1
3	Arroza Oro	Gallo	Molinos	15,00	13
4	Papas fritas (C#)	Lays	ElPreoveedorModificado	15,00	14

Figura 7: Página de Artículo

Id	Nombre	Dirección	Capacidad
1	Depositos Mayores	Av Nazca 3500	4000,00
3	Super Deposito	Av Corrientes 4000	1000,00
5	DepoActualizadoC#	Av Monroe 3000	5000,00
6	DepoparaStock (C#)	Av Monroe 3500	4000,00
7	ModificoNombreDesdeApp	Blanco Encalada 3000	2000,00
11	ModificoServiceEntity	Entity 300	3000,00

Figura 8: Pagina de Deposito

Id	Deposito	Articulo	Codigo Articulo
1	Depositos Mayores	Harina Integral	1
3	DepoparaStock (C#)	ArticuloParaStock (C#)	17
6	ModificoNombreDesdeApp	Arroza Oro	13
7	Depositos Mayores	Harina Integral	1

© 2023 - WebAppStock - [Privacy](#)

Figura 9: Pagina de Stock

## **2. Referencias**

1. Link al github con el Código
2. Tablas en Bootstrap
3. Colores en Bootstrap