

TAREAS CLASES 16 Y 17

TAREAS CAPACITACIÓN C#

 **Ezequiel Remus**
ezequielremus@gmail.com

RESUMEN

En este apunte se adjuntan las capturas correspondientes a las tareas de las clases 16 y 17

1. Enunciado General

1. Crear una solución de Xamarin Forms con el nombre AppMobileStock
2. Crear la pantalla de ArticuloPage y ArticuloABMPage
3. La pantalla de Articulo Page tiene que tener un botón que rediriga a ArticuloAbmPage
4. En el ArticuloABMPage agregar todos los campos para dar de alta un articulo
5. Crear la lógica para que se realice el alta del articulo utilizando el Api Rest de Stock.
6. IDem pero para depósitos.

Enviar Print del XAML de articulos y depósitos

Enviar Print del código del ApiArticuloServices y ApiDepositoServices

Los prints indicados se encuentran en las hojas siguientes

2. XAML Articulos

2.1. ArtículoPage

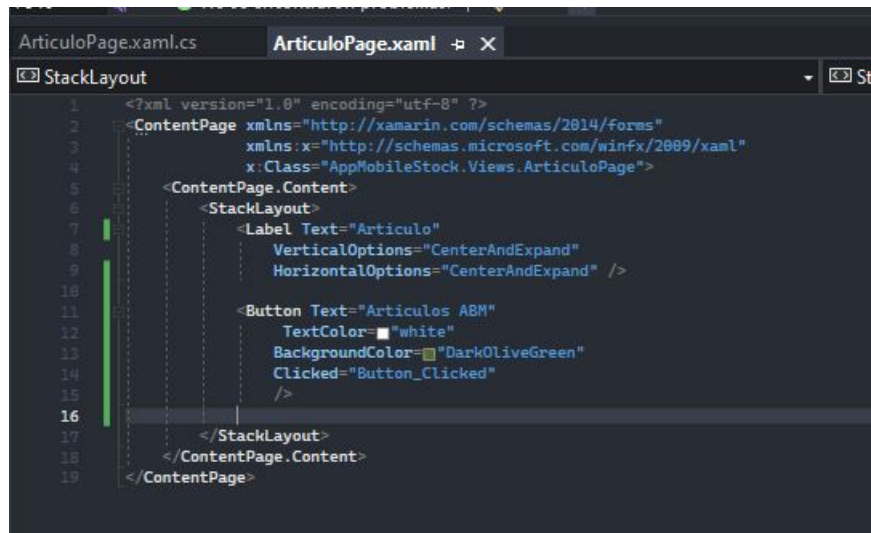


Figura 1: ArtículoPage.xaml

En cuanto a los xaml.cs nos quedan como sigue



Figura 2: ArtículoPage.xaml.cs

2.2. ArtículoABMPage

Del xaml del ArtículoABMPage.xaml, para que entre todo minimizamos las etiquetas. Lo importante es que esten las propiedades x:Name las cuales se pueden ver



Figura 3: ArtículoABMPage.xaml

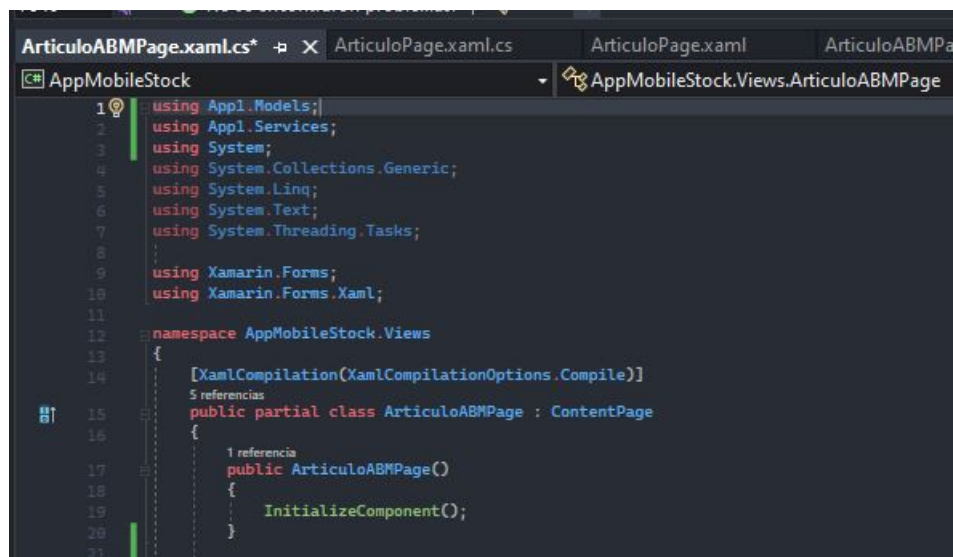


Figura 4: Parte 1: ArtículoABMPage.xaml.cs

```
21
22
0 referencias
23 private async void Button_Clicked(object sender, EventArgs e)
24 {
25     try
26     {
27         ArtículoDTO articuloDTOaAgregar = new ArtículoDTO();
28         articuloDTOaAgregar.Nombre = txtNombre.Text;
29         articuloDTOaAgregar.Marca = txtNombre.Text;
30         articuloDTOaAgregar.MinimoStock = Convert.ToDecimal(txtMinimoStock.Text);
31         articuloDTOaAgregar.Proveedor = txtProveedor.Text;
32         articuloDTOaAgregar.Precio = Convert.ToDecimal(txtPrecio.Text);
33         articuloDTOaAgregar.Codigo = txtCodigo.Text;
34         articuloDTOaAgregar.Mensaje = "-";
35         articuloDTOaAgregar.Origen = "AppMobile";
36
37         ApiArticulosServices apiArticulosServices = new ApiArticulosServices();
38         articuloDTOaAgregar = await apiArticulosServices.SendArticulo(articuloDTOaAgregar);
39
40         if (articuloDTOaAgregar.HuboError)
41         {
42             await DisplayAlert("Error", "Ocurrió un error " + articuloDTOaAgregar.Mensaje, "Ok");
43         }
44         else
45         {
46             await DisplayAlert("Confirmación", articuloDTOaAgregar.Mensaje, "Ok");
47         }
48     }
49     catch (Exception ex)
50     {
51     }
52
53     await DisplayAlert("Error", "Ocurrió un error " + ex.Message, "Ok");
54 }
55
```

Figura 5: Parte 2: ArticuloABMPage.xaml.cs

3. XAML Depositos

3.1. DepositoPage

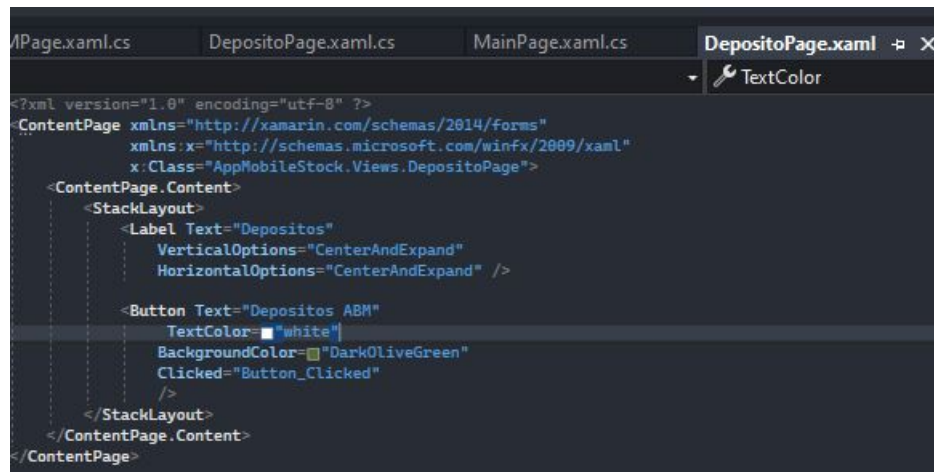


Figura 6: DepositoPage.xaml

En cuanto a los xaml.cs nos quedan como sigue

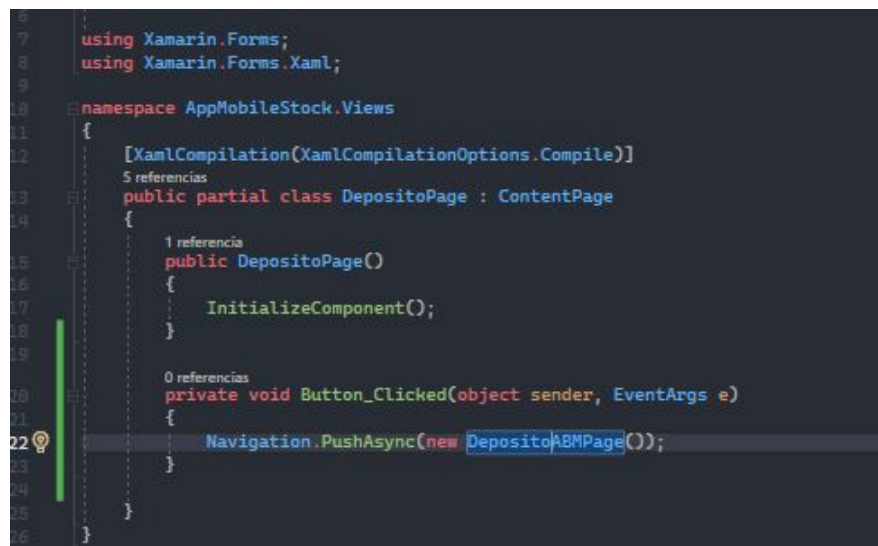


Figura 7: DepositoPage.xaml.cs

3.2. DepositosABMPage

Del xaml del DepositosABMPage.xaml, para que entre todo minimizamos las etiquetas. Lo importante es que esten las propiedades x:Name las cuales se pueden ver

```

21      <ColumnDefinition Width="*" />
22      <ColumnDefinition Width="Auto" />
23    </Grid.ColumnDefinitions>
24    <!--Titulo-->
25    <Label Grid.Column="1".../>
26
35    <!--Ingreso: Nombre-->
36    <Label Grid.Column="1".../>
37    <Entry x:Name="txtNombre".../>
43
51    <!--Ingreso: Capacidad-->
52    <Label Grid.Column="1".../>
53    <Entry x:Name="txtCapacidad".../>
59
67    <!--Ingreso: Dirección-->
68    <Label Grid.Column="1".../>
69    <Entry x:Name="txtDireccion".../>
74
82    <Button Grid.Column="1"
83            Grid.Row="4"
84            Text="Agregar Deposoito"
85            TextColor="#005245"
86            BackgroundColor="#96e5e2"
87            Clicked="Button_Clicked"
88            />
89
90  </Grid>
91

```

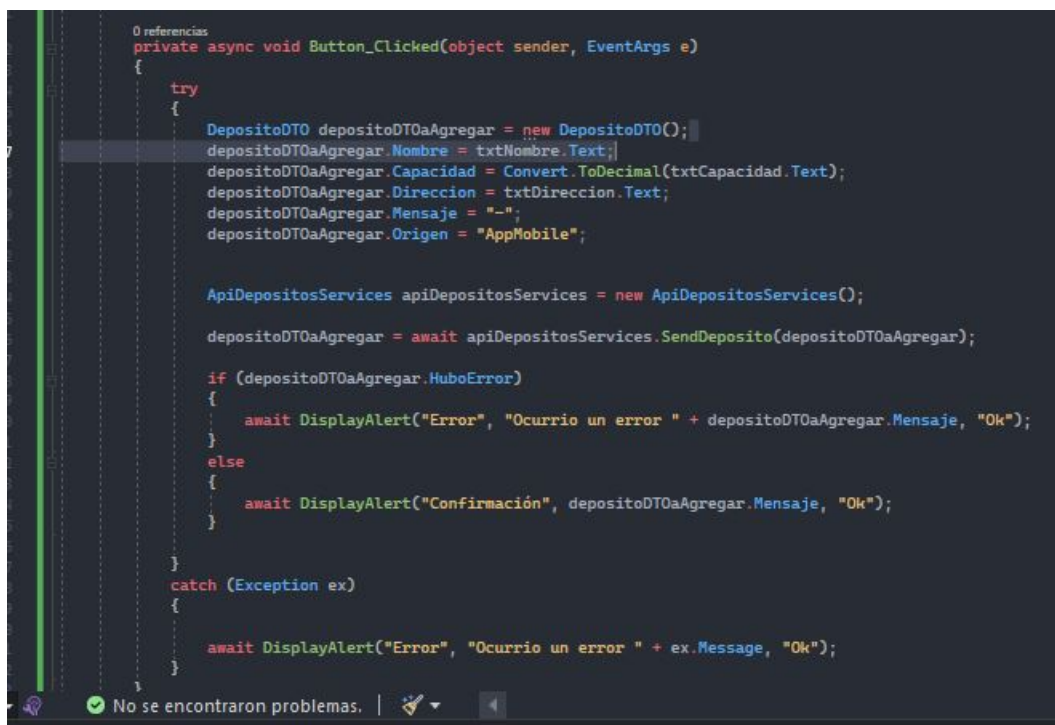
Figura 8: DepositosABMPage.xaml

```

DepositoABMPage.xaml.cs  DepositoPage.xaml.cs  MainPage.xaml.cs  Depo
AppMobileStock
1  using Appl.Models;
2  using Appl.Services;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  using Xamarin.Forms;
10 using Xamarin.Forms.Xaml;
11
12 namespace AppMobileStock.Views
13 {
14     [XamlCompilation(XamlCompilationOptions.Compile)]
15     5 referencias
16     public partial class DepositoABMPage : ContentPage
17     {
18         1 referencia
19         public DepositoABMPage()
20         {
21             InitializeComponent();
22         }
23     }

```

Figura 9: Parte 1: DepositosABMPage.xaml.cs



```
0 referencias
private async void Button_Clicked(object sender, EventArgs e)
{
    try
    {
        DepositoDTO depositoDTOaAgregar = new DepositoDTO();
        depositoDTOaAgregar.Nombre = txtNombre.Text;
        depositoDTOaAgregar.Capacidad = Convert.ToDecimal(txtCapacidad.Text);
        depositoDTOaAgregar.Direccion = txtDireccion.Text;
        depositoDTOaAgregar.Mensaje = "-";
        depositoDTOaAgregar.Origen = "AppMobile";

        ApiDepositosServices apiDepositosServices = new ApiDepositosServices();

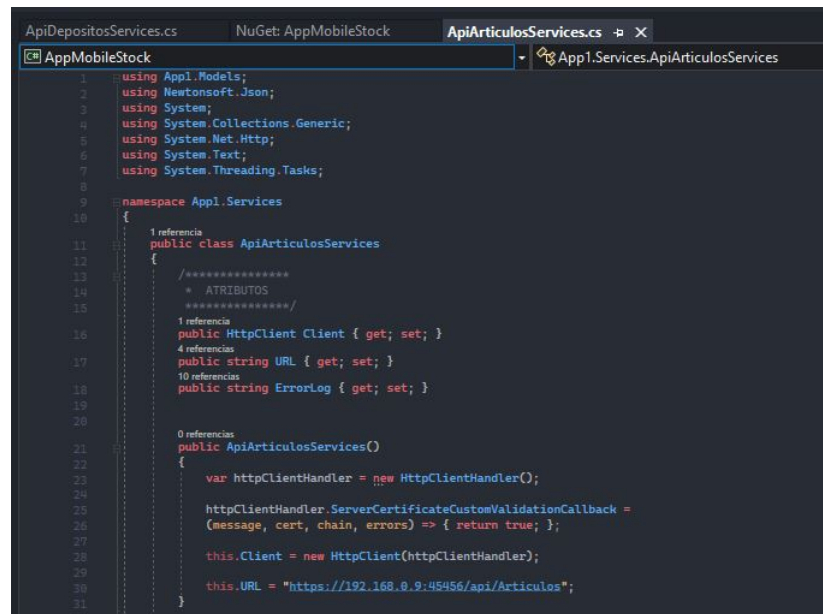
        depositoDTOaAgregar = await apiDepositosServices.SendDeposito(depositoDTOaAgregar);

        if (depositoDTOaAgregar.HuboError)
        {
            await DisplayAlert("Error", "Ocurrio un error " + depositoDTOaAgregar.Mensaje, "Ok");
        }
        else
        {
            await DisplayAlert("Confirmación", depositoDTOaAgregar.Mensaje, "Ok");
        }
    }
    catch (Exception ex)
    {
        await DisplayAlert("Error", "Ocurrio un error " + ex.Message, "Ok");
    }
}
```

✓ No se encontraron problemas.

Figura 10: Parte 2: DepositosABMPPage.xaml.cs

4. ApiArticuloServices

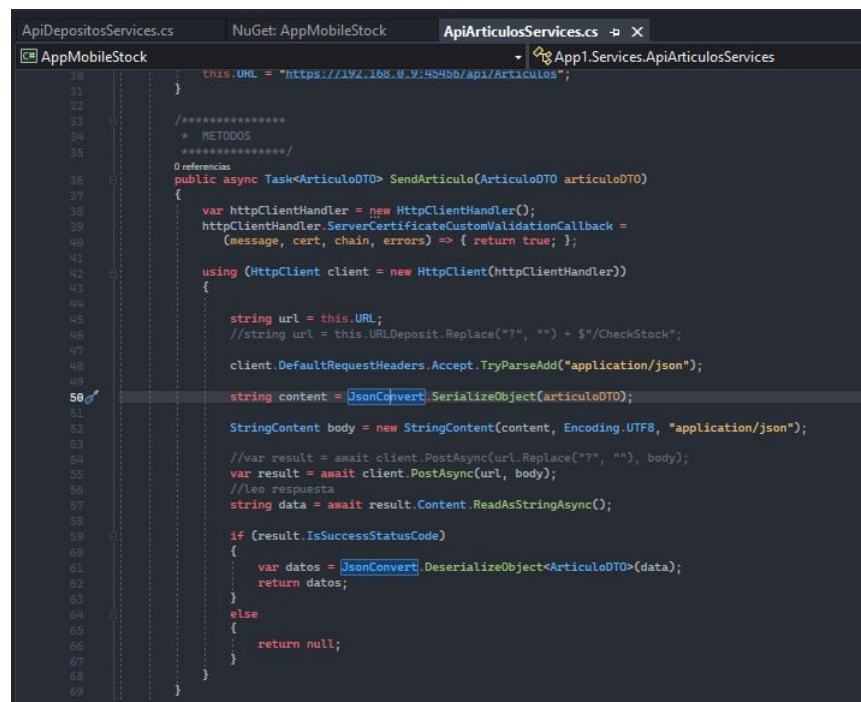


```

1  using App1.Models;
2  using Newtonsoft.Json;
3  using System;
4  using System.Collections.Generic;
5  using System.Net.Http;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace App1.Services
10 {
11     1 referencia
12     public class ApiArticulosServices
13     {
14         /* ATRIBUTOS
15         *****/
16         1 referencia
17         public HttpClient Client { get; set; }
18         4 referencias
19         public string URL { get; set; }
20         10 referencias
21         public string ErrorLog { get; set; }
22
23         0 referencias
24         public ApiArticulosServices()
25         {
26             var httpClientHandler = new HttpClientHandler();
27             httpClientHandler.ServerCertificateCustomValidationCallback =
28                 (message, cert, chain, errors) => { return true; };
29             this.Client = new HttpClient(httpClientHandler);
30             this.URL = "https://192.168.0.9:49456/api/Articulos";
31         }
32     }
33 }

```

Figura 11: ApiArticuloServices Parte 1



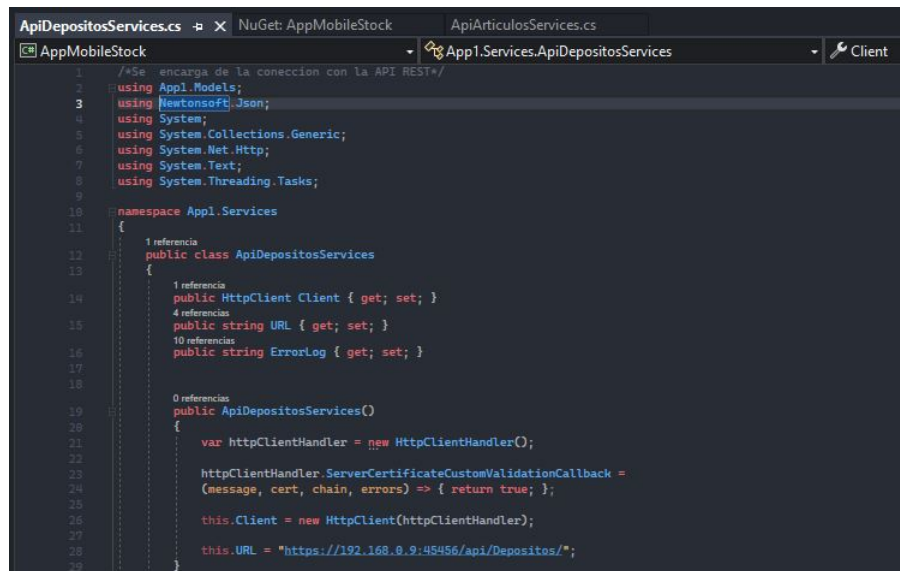
```

34     this.URL = "https://192.168.0.9:49456/api/Articulos";
35
36     /* METODOS
37     *****/
38     0 referencias
39     public async Task<ArticuloDTO> SendArticulo(ArticuloDTO articuloDTO)
40     {
41         var httpClientHandler = new HttpClientHandler();
42         httpClientHandler.ServerCertificateCustomValidationCallback =
43             (message, cert, chain, errors) => { return true; };
44         using (HttpClient client = new HttpClient(httpClientHandler))
45         {
46             string url = this.URL;
47             //string url = this.URLDeposit.Replace("?", "") + $"/CheckStock";
48             client.DefaultRequestHeaders.Accept.TryParseAdd("application/json");
49             string content = JsonConvert.SerializeObject(articuloDTO);
50             StringContent body = new StringContent(content, Encoding.UTF8, "application/json");
51             //var result = await client.PostAsync(url.Replace("?", ""), body);
52             var result = await client.PostAsync(url, body);
53             //Leo respuesta
54             string data = await result.Content.ReadAsStringAsync();
55
56             if (result.IsSuccessStatusCode)
57             {
58                 var datos = JsonConvert.DeserializeObject<ArticuloDTO>(data);
59                 return datos;
60             }
61             else
62             {
63                 return null;
64             }
65         }
66     }
67 }
68
69

```

Figura 12: ApiArticuloServices Parte 2

5. ApiDepositoServices

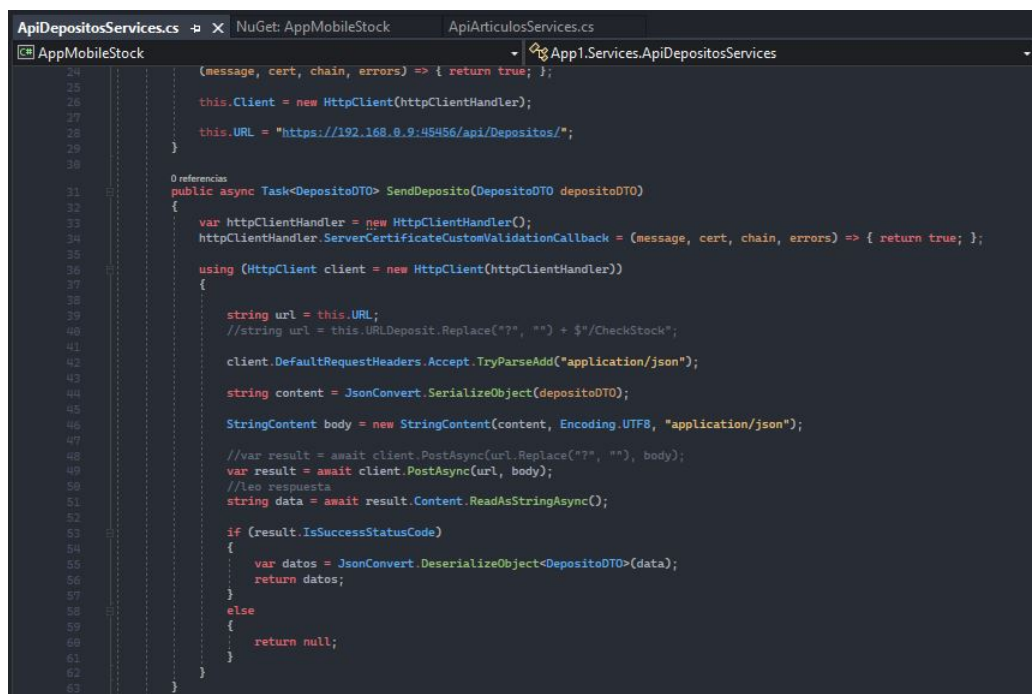


```

1  /*Se encarga de la conexión con la API REST*/
2  using Appl.Models;
3  using Newtonsoft.Json;
4  using System;
5  using System.Collections.Generic;
6  using System.Net.Http;
7  using System.Text;
8  using System.Threading.Tasks;
9
10 namespace Appl.Services
11 {
12     1 referencia
13     public class ApiDepositoServices
14     {
15         1 referencia
16         public HttpClient Client { get; set; }
17         4 referencias
18         public string URL { get; set; }
19         10 referencias
20         public string ErrorLog { get; set; }
21
22         0 referencias
23         public ApiDepositoServices()
24         {
25             var httpClientHandler = new HttpClientHandler();
26             httpClientHandler.ServerCertificateCustomValidationCallback =
27                 (message, cert, chain, errors) => { return true; };
28             this.Client = new HttpClient(httpClientHandler);
29             this.URL = "https://192.168.0.9:45456/api/Depositos/";
30         }
31     }

```

Figura 13: ApiDepositoServices Parte 1



```

24         (message, cert, chain, errors) => { return true; };
25
26         this.Client = new HttpClient(httpClientHandler);
27
28         this.URL = "https://192.168.0.9:45456/api/Depositos/";
29     }
30
31     0 referencias
32     public async Task<DepositoDTO> SendDeposito(DepositoDTO depositoDTO)
33     {
34         var httpClientHandler = new HttpClientHandler();
35         httpClientHandler.ServerCertificateCustomValidationCallback = (message, cert, chain, errors) => { return true; };
36
37         using (HttpClient client = new HttpClient(httpClientHandler))
38         {
39             string url = this.URL;
40             //string url = this.URL.Replace("?", "") + "/CheckStock";
41
42             client.DefaultRequestHeaders.Accept.TryParseAdd("application/json");
43
44             string content = JsonConvert.SerializeObject(depositoDTO);
45
46             StringContent body = new StringContent(content, Encoding.UTF8, "application/json");
47
48             //var result = await client.PostAsync(url.Replace("?", ""), body);
49             var result = await client.PostAsync(url, body);
50             //lee respuesta
51             string data = await result.Content.ReadAsStringAsync();
52
53             if (result.IsSuccessStatusCode)
54             {
55                 var datos = JsonConvert.DeserializeObject<DepositoDTO>(data);
56                 return datos;
57             }
58             else
59             {
60                 return null;
61             }
62         }
63     }

```

Figura 14: ApiDepositoServices Parte 2

6. Referencias

1. Link al Drive con el Código