**SERVICIO SERIES XML**

* + Vamos a realizar una aplicación en la que leeremos un Servicio Api CRUD sobre una base de datos SQL Server.
  + Utilizaremos una aplicación Xamarin para ir representando los datos e ir comprendiendo cada una de las características de la comunicación de componentes con Xamarin.

CREATE TABLE SERIES(

IDSERIE int NOT NULL PRIMARY KEY,

NOMBRESERIE nvarchar(160) NULL,

IMAGEN nvarchar(250) NULL,

ANYO int NULL)

GO

INSERT INTO SERIES

VALUES (1, 'Juego de tronos'

, 'https://cadenaser00.epimg.net/ser/imagenes/2019/05/23/television/1558591913\_020782\_1558595107\_noticia\_normal.jpg'

, 2011)

INSERT INTO SERIES

VALUES (2, 'The Mandalorian',

'https://images-na.ssl-images-amazon.com/images/I/71bBDuXUvOL.jpg'

, 2019)

INSERT INTO SERIES

VALUES (3, 'Stranger Things',

'https://e.rpp-noticias.io/xlarge/2020/02/14/145014\_901131.png'

, 2016)

INSERT INTO SERIES

VALUES (4, 'Narcos',

'https://pbs.twimg.com/media/CyrQ14AXEAArE-m.jpg'

, 2015)

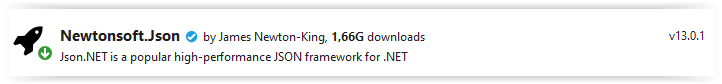
INSERT INTO SERIES

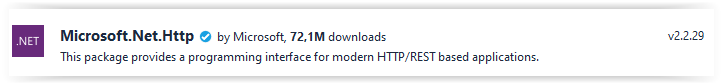
VALUES (5, 'The Boys',

'https://as01.epimg.net/epik/imagenes/2020/09/14/portada/1600093038\_124653\_1600093183\_noticia\_normal\_recorte1.jpg'

, 2019)

Agregamos los siguientes Nuget





Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Creamos una carpeta **Models**, yuna nueva clase llamada **Serie**.

**SERIE**

**public** **class** Serie

{

    [JsonProperty("idSerie")]

**public** **int** IdSerie { **get**; **set**; }

    [JsonProperty("nombreSerie")]

**public** **string** NombreSerie { **get**; **set**; }

    [JsonProperty("imagen")]

**public** **string** Imagen { **get**; **set**; }

    [JsonProperty("anyo")]

**public** **int** Anyo { **get**; **set**; }

}

Creamos una nueva carpeta llamada **Services** y una nueva clase llamada **ServiceSeries**

**SERVICESERIES**

**public** **class** ServiceSeries

{

**private** **string** UrlApi;

**public** ServiceSeries()

    {

        this**.**UrlApi **=** "https://apiseriesxamarin.azurewebsites.net/";

    }

**private** **async** Task<T> **CallApiAsync**<T>(String request)

    {

**using** (HttpClient client **=** **new** HttpClient())

        {

            client**.**BaseAddress **=** **new** Uri(this**.**UrlApi);

            client**.**DefaultRequestHeaders**.**Accept**.**Clear();

            client**.**DefaultRequestHeaders**.**Accept**.**Add

                (**new** MediaTypeWithQualityHeaderValue("application/json"));

            HttpResponseMessage response **=**

**await** client**.**GetAsync(request);

**if** (response**.**IsSuccessStatusCode)

            {

                String json **=**

**await** response**.**Content**.**ReadAsStringAsync();

                T data **=**

                    JsonConvert**.**DeserializeObject<T>(json);

**return** data;

            }

**else**

            {

**return** **default**(T);

            }

        }

    }

**public** **async** Task<List<Serie>> **GetSeriesAsync**()

    {

**string** request **=** "/api/series";

        List<Serie> series **=**

**await** this**.**CallApiAsync<List<Serie>>(request);

**return** series;

    }

}

Creamos una carpeta llamada **Base** y una clase llamada **ViewModelBase**

**VIEWMODELBASE**

**public** **class** ViewModelBase : INotifyPropertyChanged

{

**public** **event** PropertyChangedEventHandler PropertyChanged;

**public** **void** **OnPropertyChanged**(String propertyName)

    {

        PropertyChanged**?.**Invoke(**this**

            , **new** PropertyChangedEventArgs(propertyName));

    }

}

Creamos una carpeta llamada **ViewModels** y una clase llamada **SeriesViewModel**

**SERIESVIEWMODEL**

**public** **class** SeriesViewModel: ViewModelBase

{

**private** ServiceSeries service;

**private** ObservableCollection<Serie> \_Series;

**public** ObservableCollection<Serie> Series

    {

**get** { **return** this**.**\_Series; }

**set** {

            this**.**\_Series **=** value;

            OnPropertyChanged("Series");

        }

    }

**public** **async** Task **LoadSeriesAsync**()

    {

        List<Serie> series **=**

**await** this**.**service**.**GetSeriesAsync();

        this**.**Series **=** **new** ObservableCollection<Serie>(series);

        //this.Series = new ObservableCollection<Serie>();

    }

**public** SeriesViewModel(ServiceSeries service)

    {

**this.**service **=** service;

        Task**.**Run(**async** () **=>**

        {

**await** this**.**LoadSeriesAsync();

        });

    }

}

A continuación, debemos incluir una clase para resolver las dependencias, para ello necesitamos un contenedor que nos devuelva las clases

Sobre la carpeta **Services**, creamos una nueva clase llamada **ServiceIoC**

**SERVICEIOC**

**public** **class** ServiceIoC

{

**private** IContainer container;

**public** ServiceIoC()

    {

        this**.**RegisterDependencies();

    }

**private** **void** **RegisterDependencies**()

    {

        ContainerBuilder builder **=** **new** ContainerBuilder();

        builder**.**RegisterType<ServiceSeries>();

        builder**.**RegisterType<SeriesViewModel>();

        this**.**container **=** builder**.**Build();

    }

**public** SeriesViewModel SeriesViewModel

    {

**get**

        {

**return** this**.**container**.**Resolve<SeriesViewModel>();

        }

    }

}

Damos de alta nuestro servicio IoC dentro de **App.xaml.cs**

**APP.XAML.CS**

**private** **static** ServiceIoC \_ServiceLocator;

**public** **static** ServiceIoC ServiceLocator

{

**get**

    {

**return** \_ServiceLocator **=** \_ServiceLocator **??** **new** ServiceIoC();

    }

}

Ahora es el momento de hacer la parte Visual. Creamos una carpeta llamada **Views** y una nueva **ContentPage** llamada **SeriesView.xaml**

**SERIESVIEW.XAML**

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

             x:Class="App1.Views.SeriesView"

             xmlns:local="clr-namespace:App1"

             BindingContext="{Binding Source={x:Static local:App.ServiceLocator}

                , Path=SeriesViewModel}"

             x:Name="PageSeriesView">

    <ContentPage.Content>

        <StackLayout>

            <ListView ItemsSource="{Binding Series}">

                <ListView.ItemTemplate>

                    <DataTemplate>

                        <ViewCell>

                            <StackLayout>

                                <Label FontSize="Large"

                                   Text="{Binding NombreSerie}"/>

                            </StackLayout>

                        </ViewCell>

                    </DataTemplate>

                </ListView.ItemTemplate>

            </ListView>

        </StackLayout>

    </ContentPage.Content>

</ContentPage>

Ahora vamos a realizar una vista **Detalles** de la serie e incluir una acción en la página principal.

Lo primero será escribir un método en la clase **ServiceSeries**

**SERVICESERIES**

**public** **async** Task<Serie> **FindSerieAsync**(**int** id)

{

**string** request **=** "/api/series/" **+** id;

    Serie serie **=**

**await** this**.**CallApiAsync<Serie>(request);

**return** serie;

}

Creamos un método **Command** para realizar una acción dentro del ViewModel de series.

**SERIESVIEWMODEL**

**public** Command ShowSerieDetails

{

**get**

    {

**return** **new** Command(() **=>**

        {

        });

    }

}

A continuación, debemos enlazar el objeto Serie con la página XAML dentro del ListView.

**SERIESVIEW.XAML**

<ListView ItemsSource="{Binding Series}">

                <ListView.ItemTemplate>

                    <DataTemplate>

                        <ViewCell>

                            <StackLayout>

                                <Label FontSize="Large"

                       Text="{Binding NombreSerie}"/>

                                <Button Text="Detalles"

                            HorizontalOptions="End"

                            CommandParameter="{Binding .}"

                            Command="{Binding BindingContext.ShowSerieDetails

                        , Source={x:Reference PageSeriesView}}"/>

                            </StackLayout>

                        </ViewCell>

                    </DataTemplate>

                </ListView.ItemTemplate>

Y vamos a poner un mensaje Alert dentro del ViewModel para visualizar si recibimos el objeto.

**SERIESVIEWMODEL**

**public** Command ShowSerieDetails

{

**get**

    {

**return** **new** Command(**async** (objeto) **=>**

        {

            Serie serie **=** objeto **as** Serie;

**await** Application**.**Current**.**MainPage**.**DisplayAlert

            ("Alert", "Year " **+** serie**.**Anyo, "OK");

        });

    }

}

Una vez que nos hemos comunicado, debemos crearnos un ViewModel para los detalles de la serie.

Sobre **ViewModels** creamos una nueva clase llamada **SerieDetailsViewModel**

**SERIEDETAILSVIEWMODEL**

**public** **class** SerieDetailsViewModel: ViewModelBase

{

**private** ServiceSeries service;

**private** Serie \_Serie;

**public** Serie Serie

    {

**get** { **return** this**.**\_Serie; }

**set**

        {

            this**.**\_Serie **=** value;

            OnPropertyChanged("Serie");

        }

    }

**public** SerieDetailsViewModel(ServiceSeries service)

    {

**this.**service **=** service;

**if** (this**.**Serie **==** **null**)

        {

            this**.**Serie **=** **new** Serie();

        }

    }

}

Declaramos el nuevo ViewModel dentro de **IoC**

**SERVICEIOC**

**private** **void** **RegisterDependencies**()

{

    ContainerBuilder builder **=** **new** ContainerBuilder();

    builder**.**RegisterType<ServiceSeries>();

    builder**.**RegisterType<SeriesViewModel>();

    builder**.**RegisterType<SerieDetailsViewModel>();

    this**.**container **=** builder**.**Build();

}

**public** SerieDetailsViewModel SerieDetailsViewModel

{

**get**

    {

**return** this**.**container**.**Resolve<SerieDetailsViewModel>();

    }

}

Creamos una nueva página para el diseño.

Sobre **Views,** creamos una nueva vista llamada **SerieDetails.xaml**

**SERIEDETAILS.XAML**

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

             x:Class="App1.Views.SerieDetails">

    <ContentPage.Content>

        <StackLayout>

            <Label Text="{Binding Serie.NombreSerie}"

                   FontSize="Large"

                   TextColor="Blue"

                VerticalOptions="CenterAndExpand"

                HorizontalOptions="CenterAndExpand" />

            <Label Text="{Binding Serie.Anyo}" TextColor="Red"/>

            <Image Source="{Binding Serie.Imagen}"/>

        </StackLayout>

    </ContentPage.Content>

</ContentPage>

Y podremos modificar el código de nuestro **SeriesViewModel** para mostrar la página de detalles.

**SERIESVIEWMODEL**

**public** Command ShowSerieDetails

{

**get**

    {

**return** **new** Command(**async** (objeto) **=>**

        {

            Serie serie **=** objeto **as** Serie;

            SerieDetailsViewModel viewmodel **=**

            App**.**ServiceLocator**.**SerieDetailsViewModel;

            viewmodel**.**Serie **=** serie;

            SerieDetails view **=**

**new** SerieDetails();

            view**.**BindingContext **=** viewmodel;

**await** Application**.**Current**.**MainPage**.**Navigation

**.**PushModalAsync(view);

        });

    }

}