

##### Laboratorio: Funciones y Procedimientos en C#

##### COMPETENCIAS

1. Comprender paradigmas de la programación orientada a objetos
2. Escribir funciones y procedimientos en lenguaje de programación C#
3. Declarar clases e instanciar objetos en Visual Studio con C#

**EQUIPOS, MATERIALES, PROGRAMAS Y RECURSOS**

* PC Personal.
* Sistema operativo Windows 7.
* Material disponible desde Tecsup Virtual.
* Software Visual Studio 2017

##### SEGURIDAD

* Colocar las mochilas en el gabinete al final del salón para evitar caídas en caso de sismo.
* No ingresar con bebidas ni comidas.
* Apagar los equipos y los monitores al culminar la sesión.

**INTRODUCCIÓN**

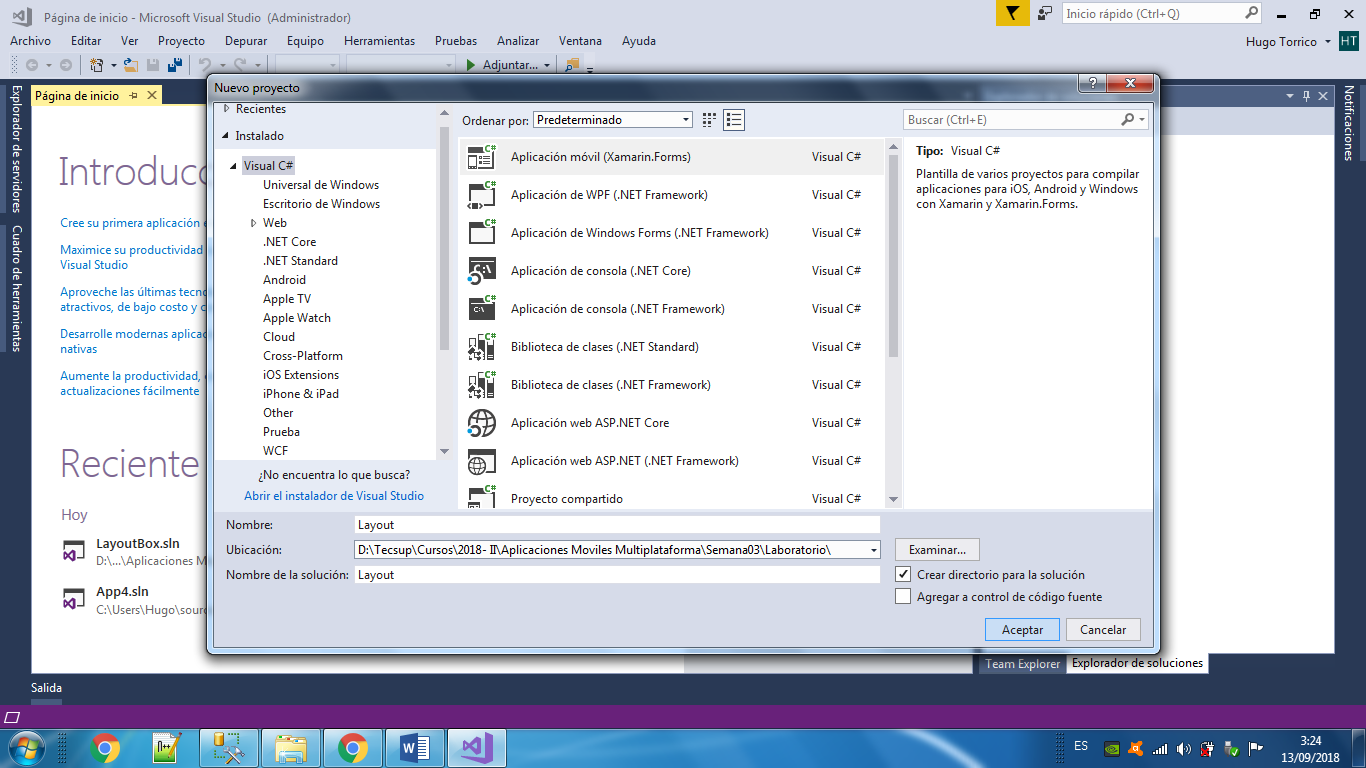
Existen diferentes lenguajes de programación orientado a objetos para escribir funciones y procedimientos pero el C# se acomoda a la necesidad del curso, porque nos va permitir realizar aplicaciones multiplataforma en móviles con XAMARIN y VISUAL STUDIO 2017

**PREPARACIÓN**

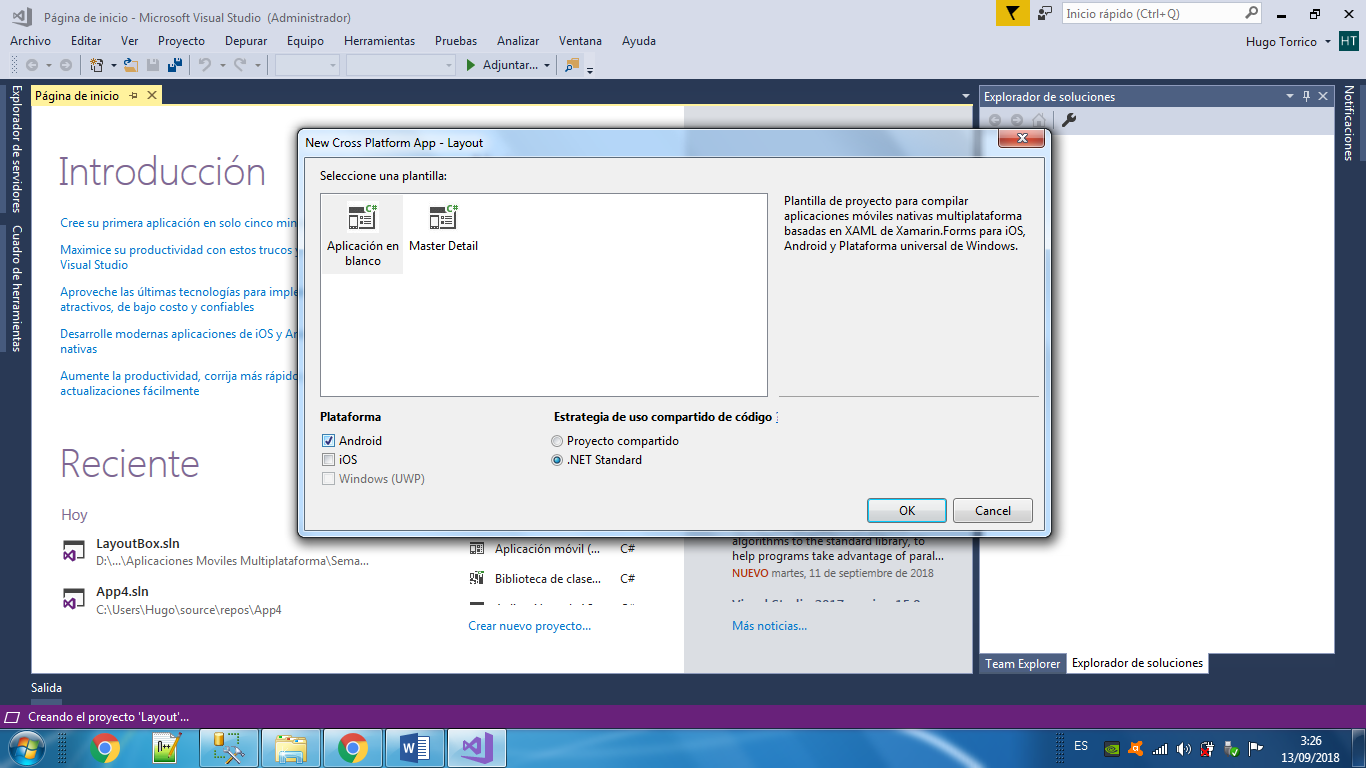
El Alumno debe revisar previamente el material del curso en Tecsup Virtual y revisar su texto.

**REVISION DEL CASO PRÁCTICO**

1. Crear proyecto de Visual Studio de tipo Xamarin.Forms de Nombre Ejercicio02



1. Seleccionar aplicación de tipo proyecto compartido.



**Main Page**

1. Agregar el XAML en MainPage

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

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

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

xmlns:local="clr-namespace:Ejercicio02"

x:Class="Ejercicio02.MainPage">

<StackLayout>

<Button x:Name="Item1" Text="Navigation"/>

<Button x:Name="Item2" Text="TabbedPage1"/>

<Button x:Name="Item3" Text="CarouselPage"/>

<Button x:Name="Item7" Text="CarouselPageCode"/>

<Button x:Name="Item5" Text="ModalPages"/>

<Button x:Name="Item6" Text="DisplayAlert"/>

</StackLayout>

</ContentPage>

1. Agregar el c# en MainPage

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

namespace Ejercicio02

{

public partial class MainPage : ContentPage

{

public MainPage()

{

InitializeComponent();

Item1.Clicked += async (sender, e) =>

{

await Navigation.PushAsync(new Page1());

};

Item2.Clicked += async (sender, e) =>

{

await Navigation.PushAsync(new TabbedPage1());

};

Item3.Clicked += async (sender, e) =>

{

await Navigation.PushAsync(new CarouselPage1());

};

Item7.Clicked += async (sender, e) =>

{

await Navigation.PushAsync(new CarouselPageCode());

};

Item5.Clicked += async (sender, e) =>

{

//Call Popup

await Navigation.PushModalAsync(new ModalPage1());

};

Item6.Clicked += async (sender, e) =>

{

//Call Popup

var answer = await DisplayAlert("Question?", "Would you like to play a game", "Yes", "No");

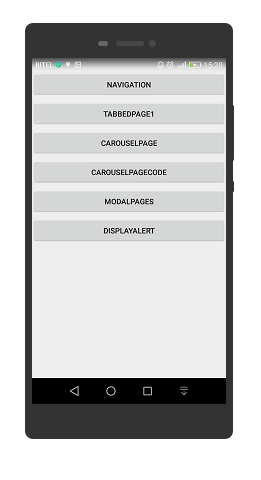
Debug.WriteLine("Answer: " + answer);

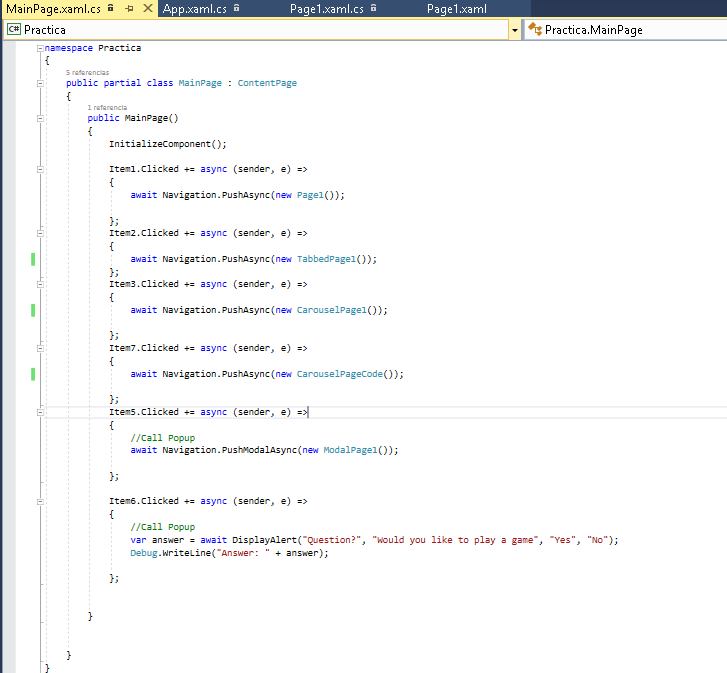
};

}

}

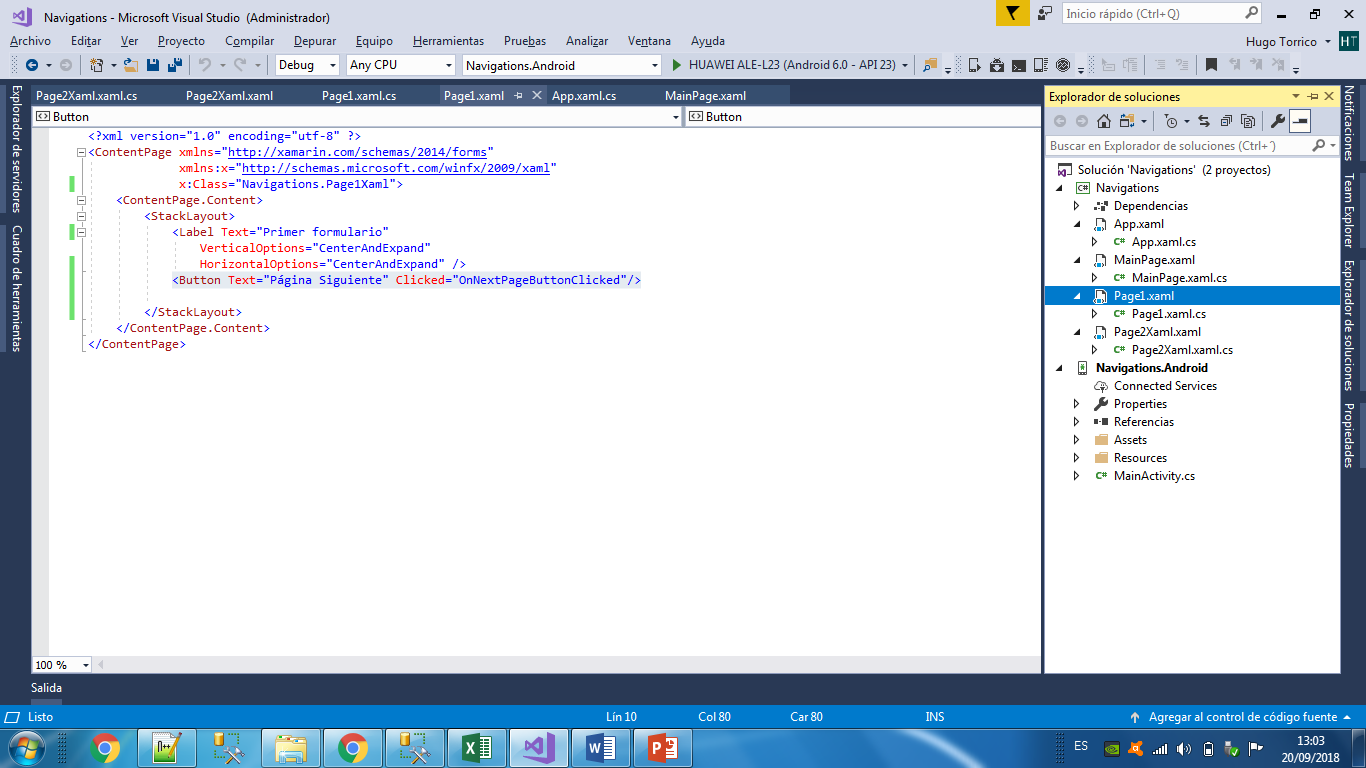
}





**Hierarchical Navigation**

1. Agregar 2 página de contenido de nombre “Page1” y “Page2”



1. Agregar el XML en Page1

<?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="Ejercicio02.Page1">

<ContentPage.Content>

<StackLayout>

<Label Text="First form"

VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand" />

<Button Text="Next Page" Clicked="OnNextPageButtonClicked"/>

<Button Text="Close" x:Name="btnClose"/>

</StackLayout>

</ContentPage.Content>

</ContentPage>

1. Agregar el C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Ejercicio02

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Page1 : ContentPage

{

public Page1 ()

{

InitializeComponent ();

btnClose.Clicked += async (sender, e) =>

{

await Navigation.PopAsync();

};

}

async void OnNextPageButtonClicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new Page2());

}

}

}

1. Agregar el XML en Page2

<?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="Ejercicio02.Page2">

<ContentPage.Content>

<StackLayout>

<Label Text="Page 2"

VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand" />

<Button Text="Previous Page" Clicked="OnPreviousPageButtonClicked"/>

</StackLayout>

</ContentPage.Content>

</ContentPage>

1. Agregar el C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Ejercicio02

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Page2 : ContentPage

{

public Page2 ()

{

InitializeComponent ();

}

async void OnPreviousPageButtonClicked(object sender, EventArgs e)

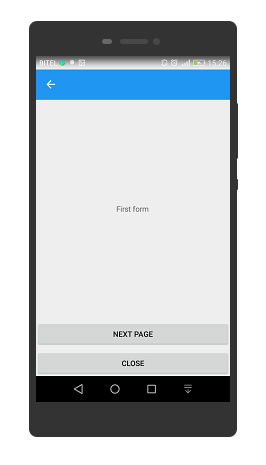
{

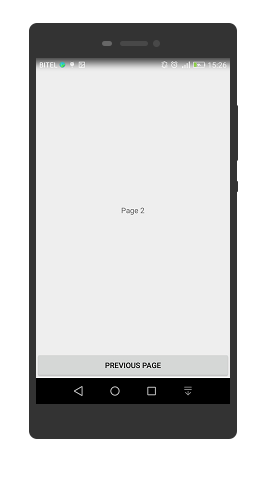
await Navigation.PopAsync();

}

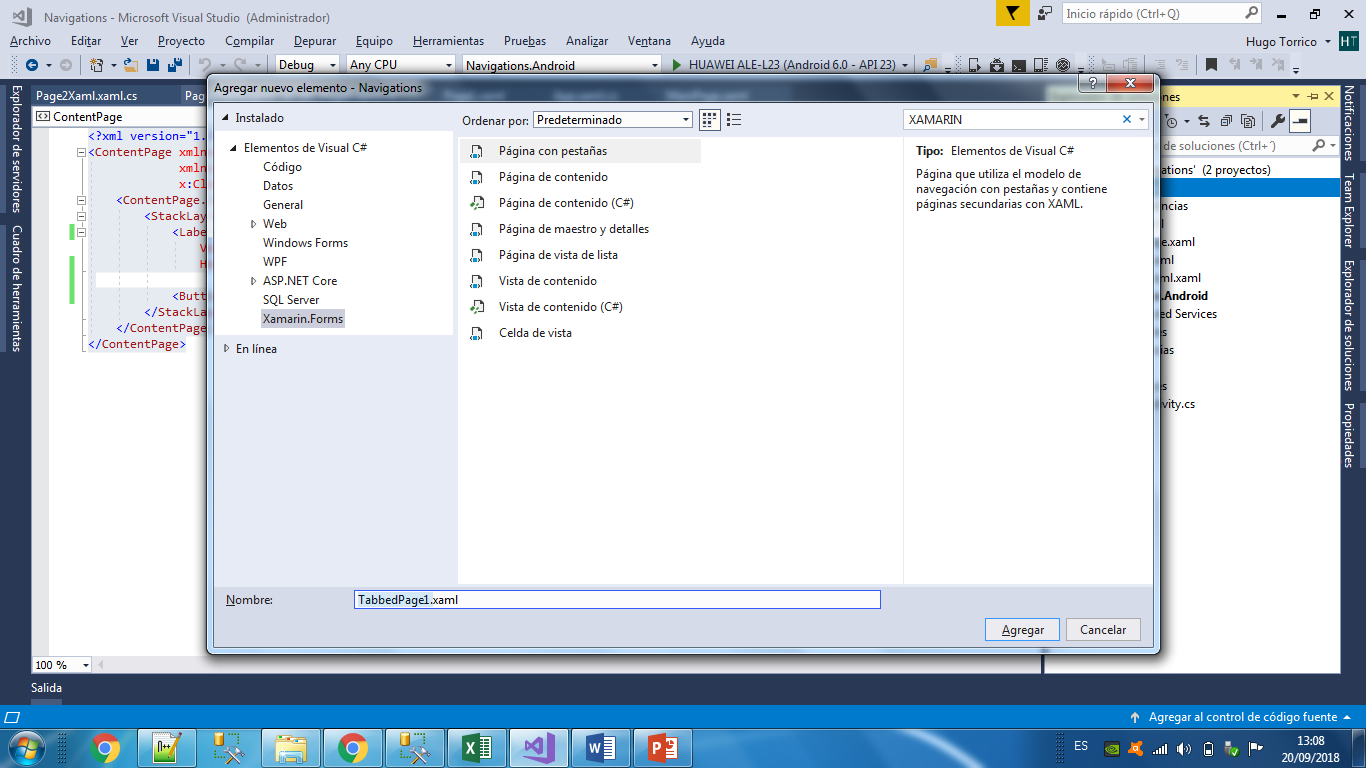
}

}



**Tabbed Page**

1. Agregar página de contenido de nombre “TabbedPage1.”



1. Agregar el XML en TabbedPage1

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

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

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

x:Class="Ejercicio02.TabbedPage1">

<ContentPage Title="Tab 1" >

<StackLayout>

<Button Text="Close" x:Name="btnClose"/>

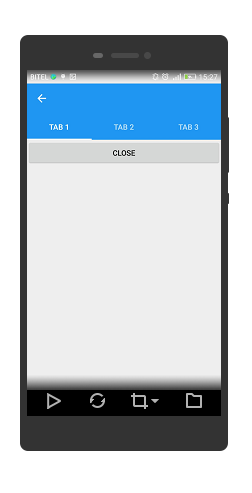
</StackLayout>

</ContentPage>

<ContentPage Title="Tab 2" />

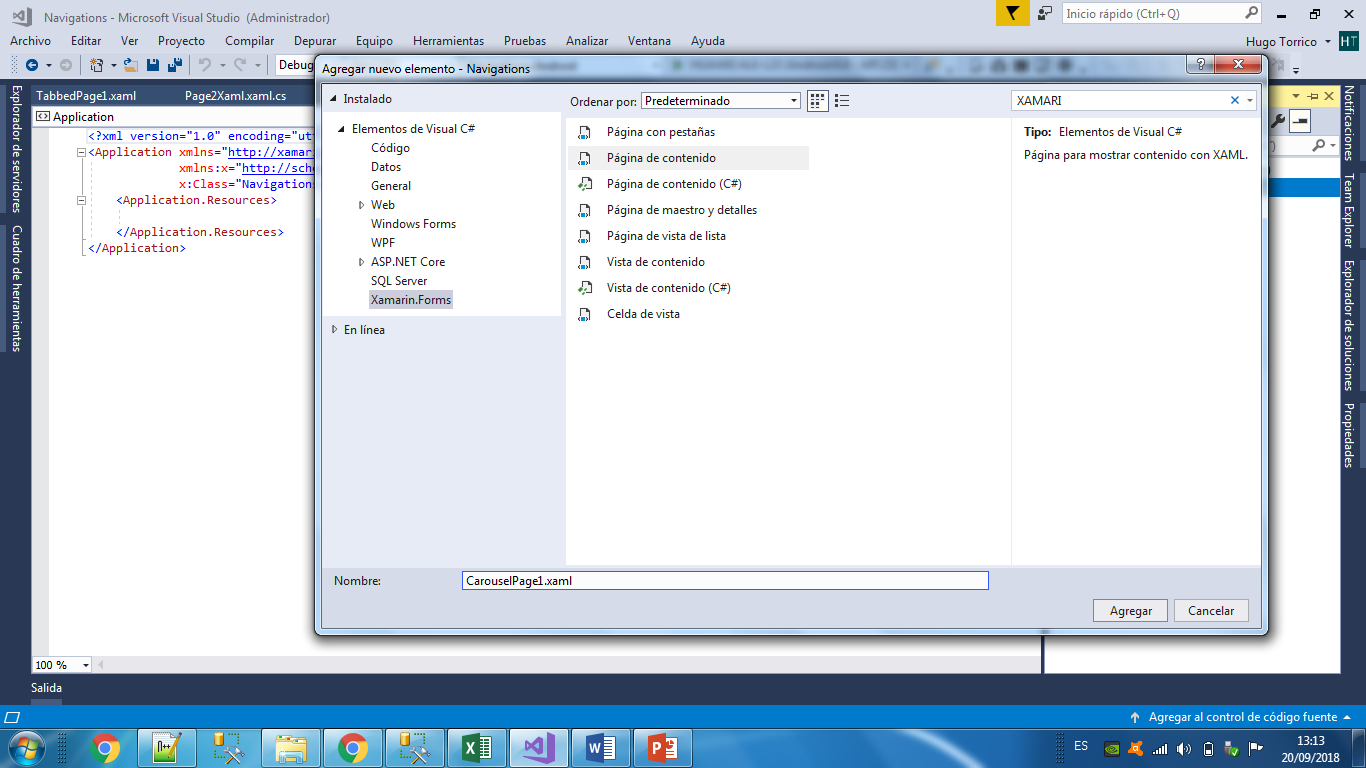
<ContentPage Title="Tab 3" />

</TabbedPage>



**Carousel Page**

1. Agregar página de contenido de nombre “CarouselPage1”



1. Agregar el XAML en CarouselPage1

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

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

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

x:Class="Ejercicio02.CarouselPage1">

<ContentPage>

<ContentPage.Padding>

<OnPlatform x:TypeArguments="Thickness">

<On Platform="iOS, Android" Value="0,40,0,0" />

</OnPlatform>

</ContentPage.Padding>

<StackLayout>

<Label Text="Red" FontSize="Medium" HorizontalOptions="Center" />

<BoxView Color="Red" WidthRequest="200" HeightRequest="200" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />

</StackLayout>

</ContentPage>

<ContentPage>

<StackLayout>

<Label Text="gree" FontSize="Medium" HorizontalOptions="Center" />

<BoxView Color="Green" WidthRequest="200" HeightRequest="200" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />

</StackLayout>

</ContentPage>

<ContentPage>

<StackLayout>

<Label Text="blue" FontSize="Medium" HorizontalOptions="Center" />

<BoxView Color="Blue" WidthRequest="200" HeightRequest="200" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" />

</StackLayout>

</ContentPage>

<ContentPage>

<Button Text="Close" x:Name="btnClose"/>

</ContentPage>

</CarouselPage>

1. Agregar el C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Ejercicio02

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class CarouselPage1 : CarouselPage

{

public CarouselPage1 ()

{

InitializeComponent ();

btnClose.Clicked += async (sender, e) =>

{

await Navigation.PopAsync();

};

}

}

}

1. Agregar página de contenido de nombre “CarouselPageCode”
2. Agregar XAML en CarouselPageCode

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

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

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

x:Class="Ejercicio02.CarouselPageCode">

<ContentPage>

<StackLayout>

<Label Text="Welcome to Xamarin.Forms!"

VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand" />

</StackLayout>

</ContentPage>

</CarouselPage>

1. Agregar Código C# en CarouselPageCode

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Ejercicio02

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class CarouselPageCode : CarouselPage

{

public CarouselPageCode ()

{

InitializeComponent ();

Thickness padding;

switch (Device.RuntimePlatform)

{

case Device.iOS:

case Device.Android:

padding = new Thickness(0, 40, 0, 0);

break;

default:

padding = new Thickness();

break;

}

var redContentPage = new ContentPage

{

Padding = padding,

Content = new StackLayout

{

Children = {

new Label {

Text = "Red",

FontSize = Device.GetNamedSize (NamedSize.Medium, typeof(Label)),

HorizontalOptions = LayoutOptions.Center

},

new BoxView {

Color = Color.Red,

WidthRequest = 200,

HeightRequest = 200,

HorizontalOptions = LayoutOptions.Center,

VerticalOptions = LayoutOptions.CenterAndExpand

}

}

}

};

var greenContentPage = new ContentPage

{

Padding = padding,

Content = new StackLayout

{

Children = {

new Label {

Text = "green",

FontSize = Device.GetNamedSize (NamedSize.Medium, typeof(Label)),

HorizontalOptions = LayoutOptions.Center

},

new BoxView {

Color = Color.Green,

WidthRequest = 200,

HeightRequest = 200,

HorizontalOptions = LayoutOptions.Center,

VerticalOptions = LayoutOptions.CenterAndExpand

}

}

}

};

var blueContentPage = new ContentPage

{

Padding = padding,

Content = new StackLayout

{

Children = {

new Label {

Text = "blue",

FontSize = Device.GetNamedSize (NamedSize.Medium, typeof(Label)),

HorizontalOptions = LayoutOptions.Center

},

new BoxView {

Color = Color.Blue,

WidthRequest = 200,

HeightRequest = 200,

HorizontalOptions = LayoutOptions.Center,

VerticalOptions = LayoutOptions.CenterAndExpand

}

}

}

};

Children.Add(redContentPage);

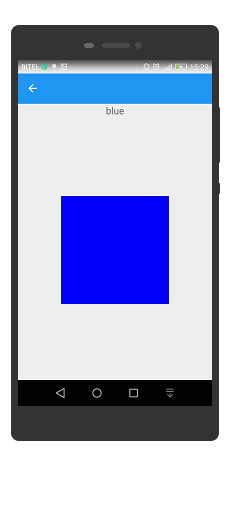
Children.Add(greenContentPage);

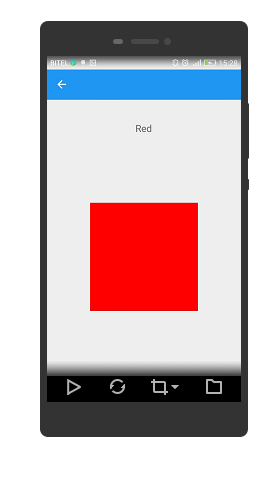
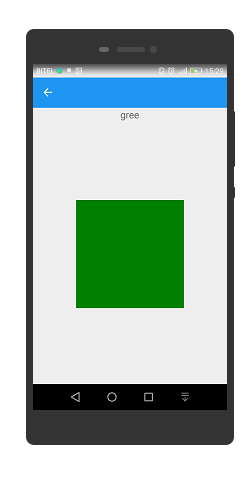
Children.Add(blueContentPage);

}

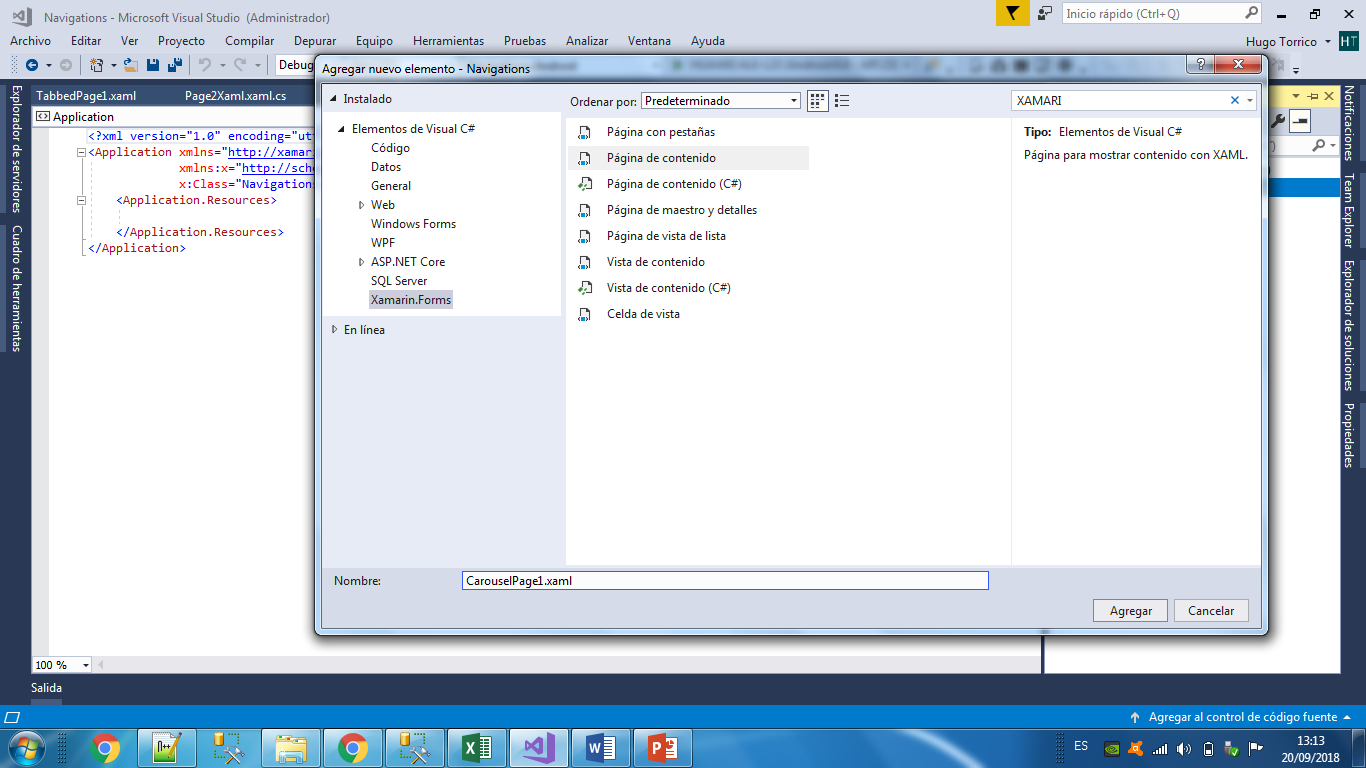
}

}



**Modal Pages**

1. Agregar página de contenido de nombre “ModalPage1”



1. Agregar el XAML en ModalPage1

<?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="Ejercicio02.ModalPage1">

<ContentPage.Content>

<StackLayout>

<Label Text="Primer formulario"

VerticalOptions="CenterAndExpand"

HorizontalOptions="CenterAndExpand" />

<Button Text="Close Popup" x:Name="btnClosePopup"/>

</StackLayout>

</ContentPage.Content>

</ContentPage>

1. Agregar el C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Ejercicio02

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class ModalPage1 : ContentPage

{

public ModalPage1 ()

{

InitializeComponent ();

btnClosePopup.Clicked += async (sender, e) =>

{

await Navigation.PopModalAsync();

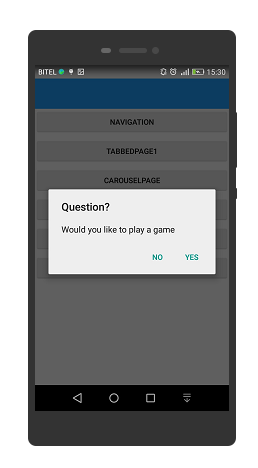
};

}

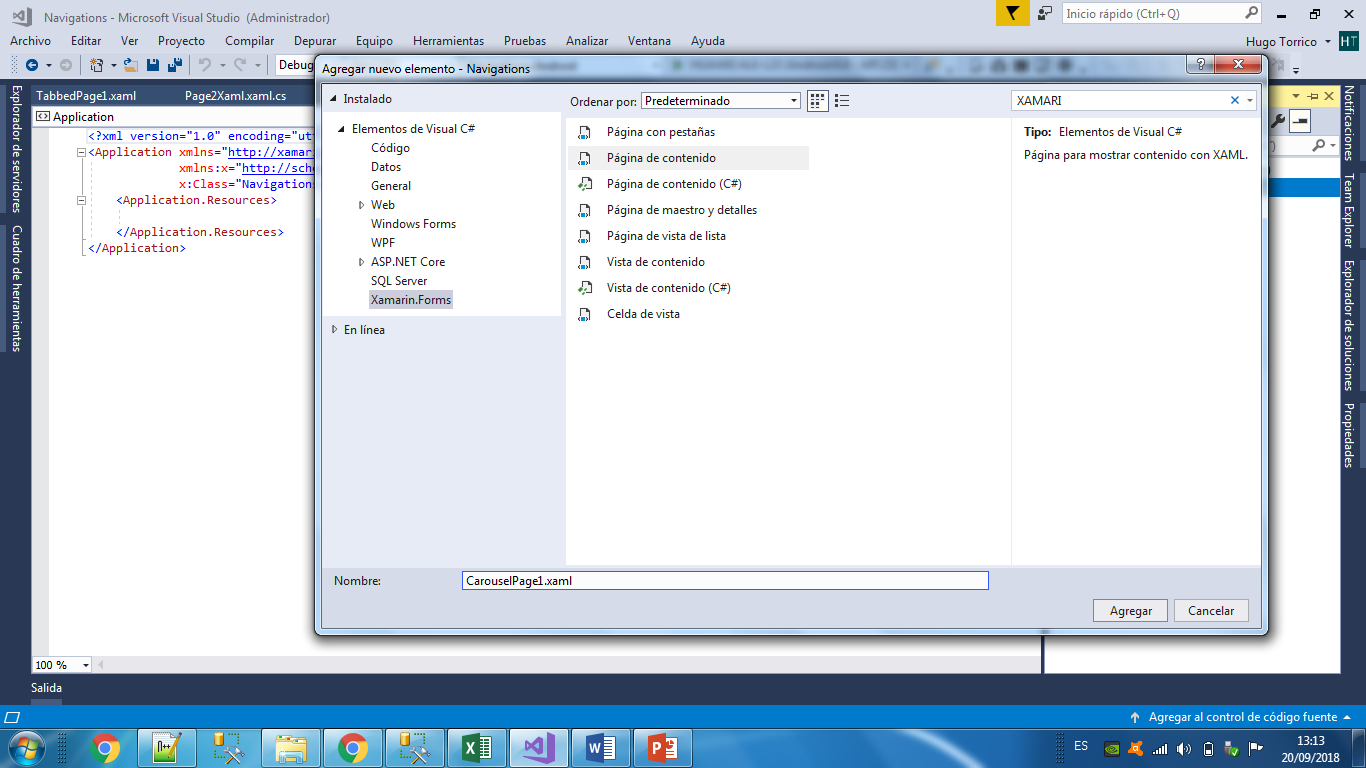
}

}

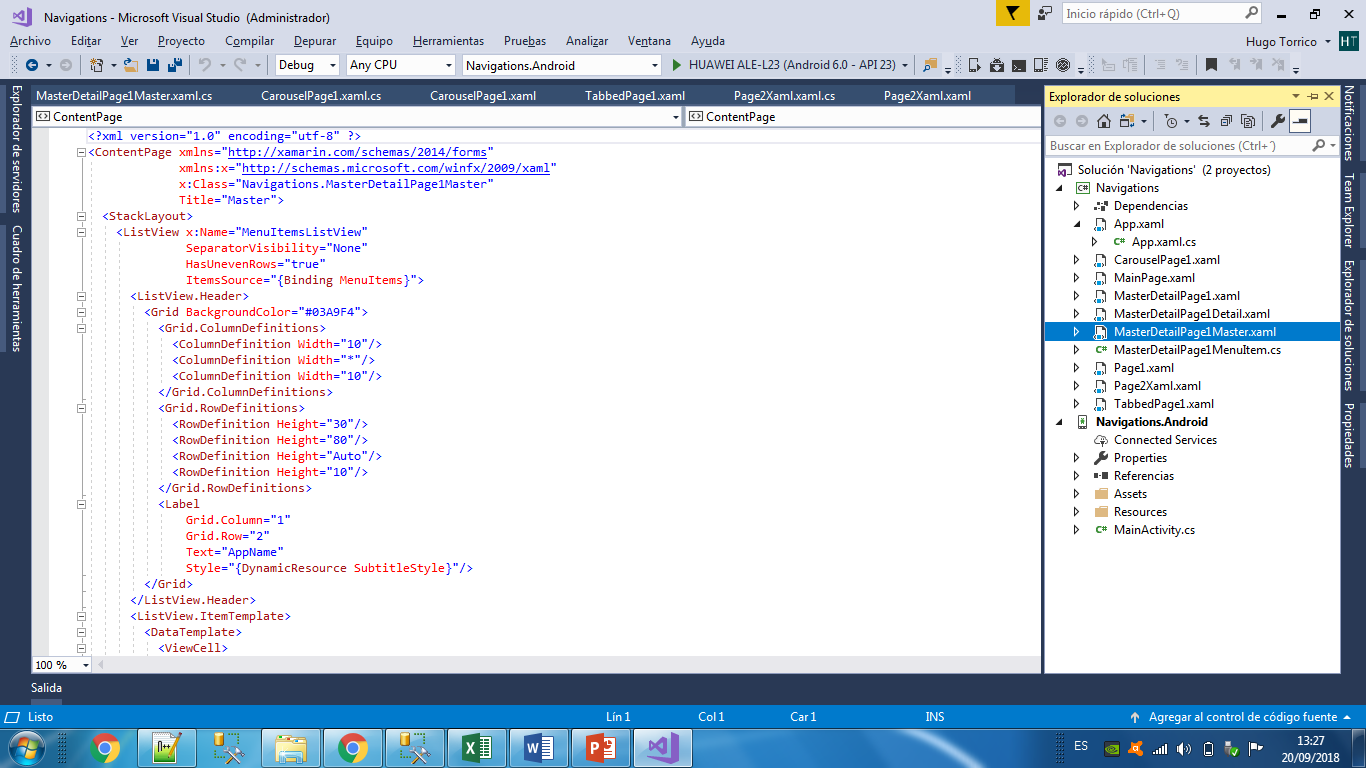


**Master-Detail Page**

1. Agregar página de contenido de nombre “MasterDetailPage1 ”



1. Se debe crear las siguienes páginas



Link Github:

<https://github.com/Camavilca/Aplicaciones-Multiplataforma/tree/master/CLASES/clase04>