

##### 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**

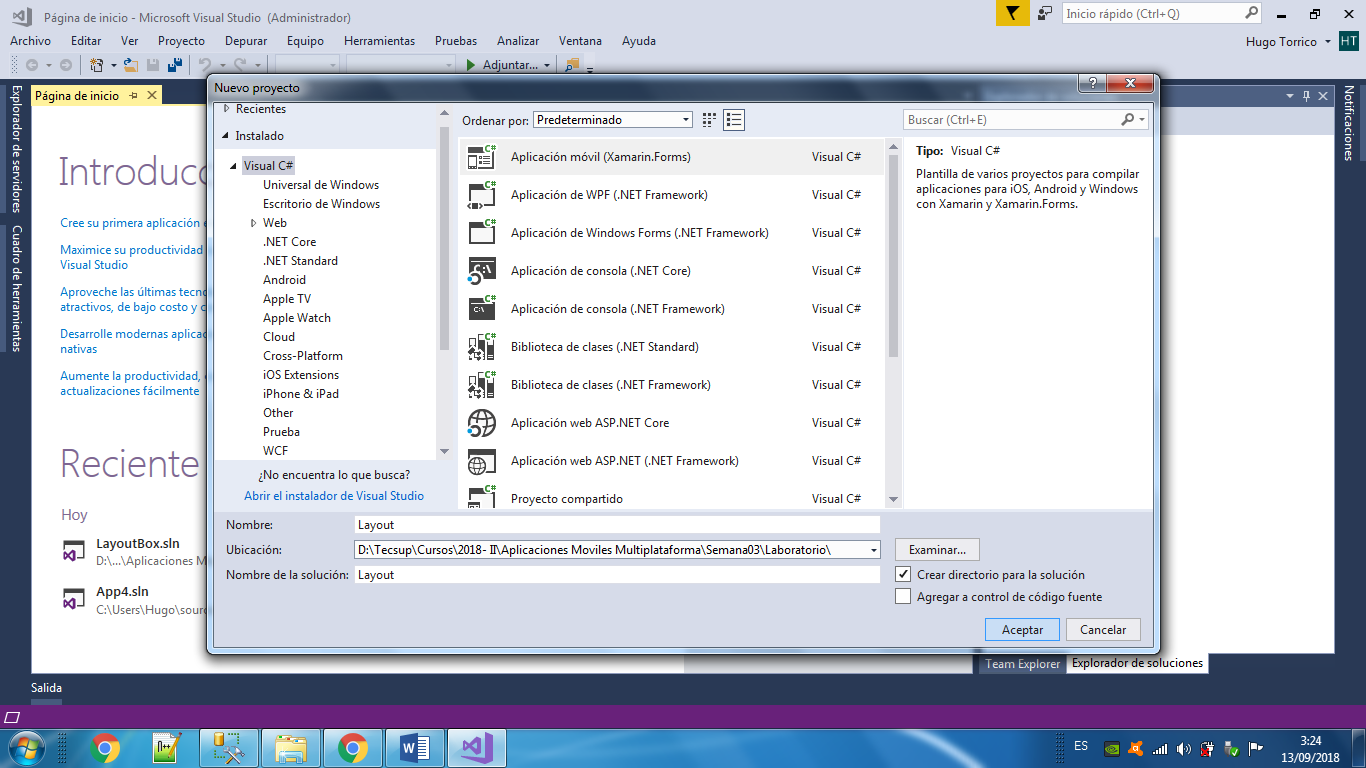
Existen diferentes lenguajes de programación orientado a objetos para escribir funciones y procedimientos pero el C# se acomoda a la necesidad del curso, porque nos va permitir realizar aplicaciones multiplataforma en móviles con XAMARIN y VISUAL STUDIO 2017

**PREPARACIÓN**

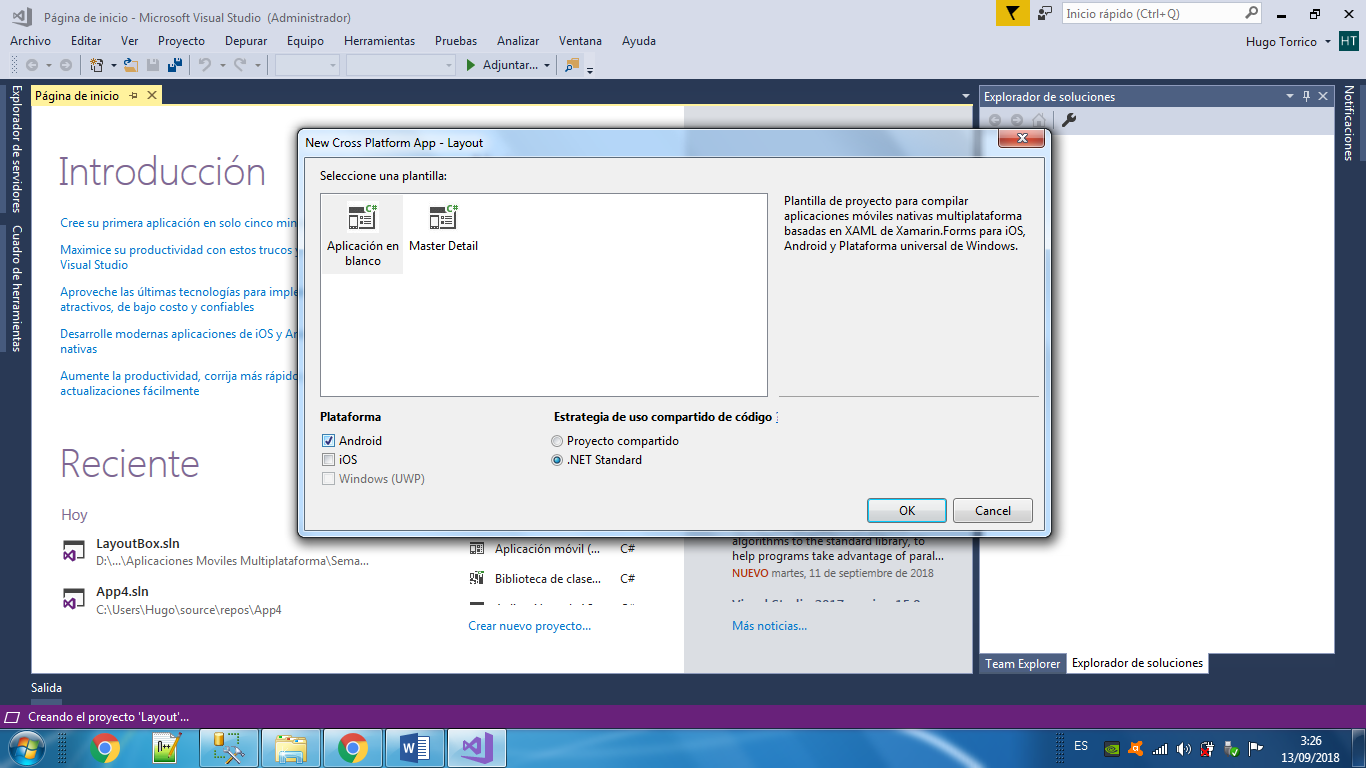
El Alumno debe revisar previamente el material del curso en Tecsup Virtual y revisar su texto.

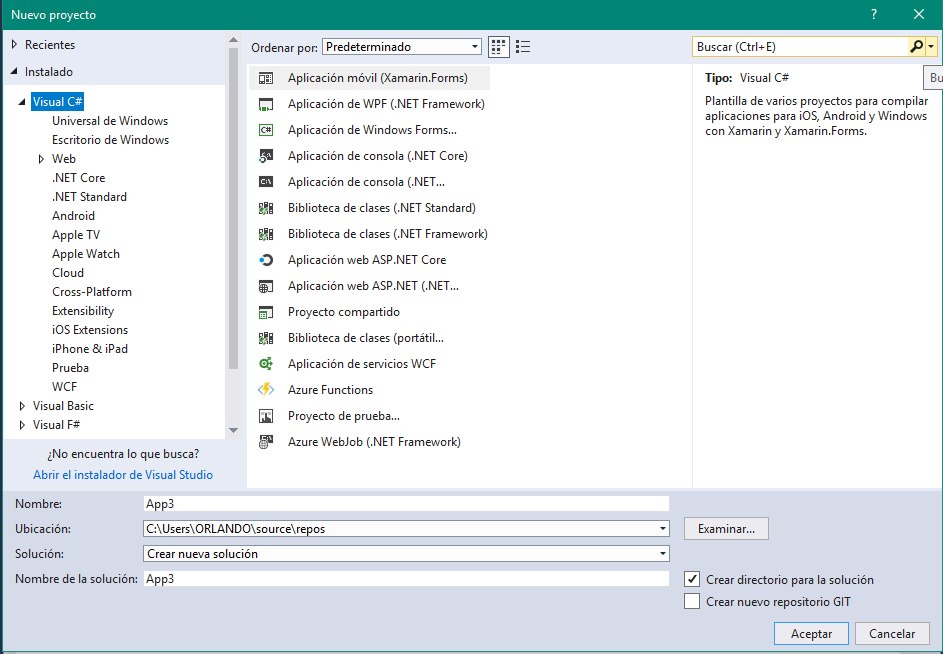
**REVISION DEL CASO PRÁCTICO**

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



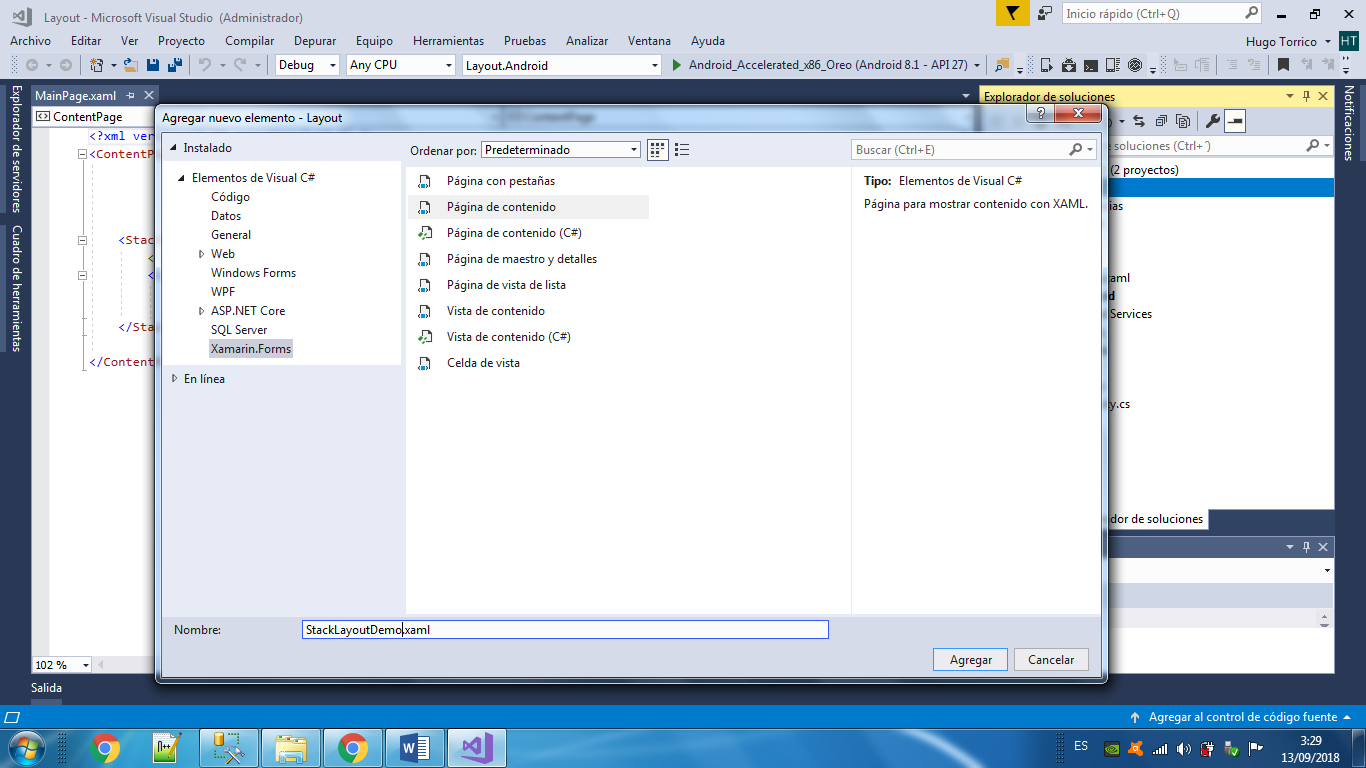
1. Seleccionar aplicación de tipo proyecto compartido.





**StockLayout**

1. Agregar página de contenido de nombre “StackLayoutDemo”



1. Agregar el siguiente código:

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="LayoutSamples.StackLayoutDemo"

Title="StackLayout Demo">

<ContentPage.Content>

<StackLayout Spacing="10" x:Name="layout">

<Button Text="StackLayout" VerticalOptions="Start"

HorizontalOptions="FillAndExpand" />

<BoxView Color="Yellow" VerticalOptions="FillAndExpand"

HorizontalOptions="FillAndExpand" />

<BoxView Color="Green" VerticalOptions="FillAndExpand"

HorizontalOptions="FillAndExpand" />

<BoxView HeightRequest="75" Color="Blue" VerticalOptions="End"

HorizontalOptions="FillAndExpand" />

</StackLayout>

</ContentPage.Content>

</ContentPage>

In C#:

public class StackLayoutCode : ContentPage

{

public StackLayoutCode ()

{

var layout = new StackLayout ();

var button = new Button { Text = "StackLayout", VerticalOptions = LayoutOptions.Start,

HorizontalOptions = LayoutOptions.FillAndExpand };

var yellowBox = new BoxView { Color = Color.Yellow, VerticalOptions = LayoutOptions.FillAndExpand, HorizontalOptions = LayoutOptions.FillAndExpand };

var greenBox = new BoxView { Color = Color.Green, VerticalOptions = LayoutOptions.FillAndExpand, HorizontalOptions = LayoutOptions.FillAndExpand };

var blueBox = new BoxView { Color = Color.Blue, VerticalOptions = LayoutOptions.FillAndExpand,

HorizontalOptions = LayoutOptions.FillAndExpand, HeightRequest = 75 };

layout.Children.Add(button);

layout.Children.Add(yellowBox);

layout.Children.Add(greenBox);

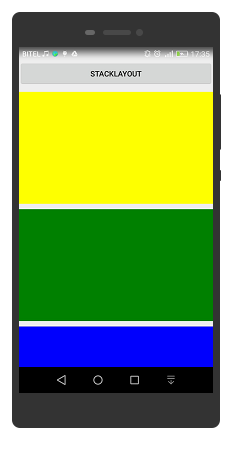
layout.Children.Add(blueBox);

layout.Spacing = 10;

Content = layout;

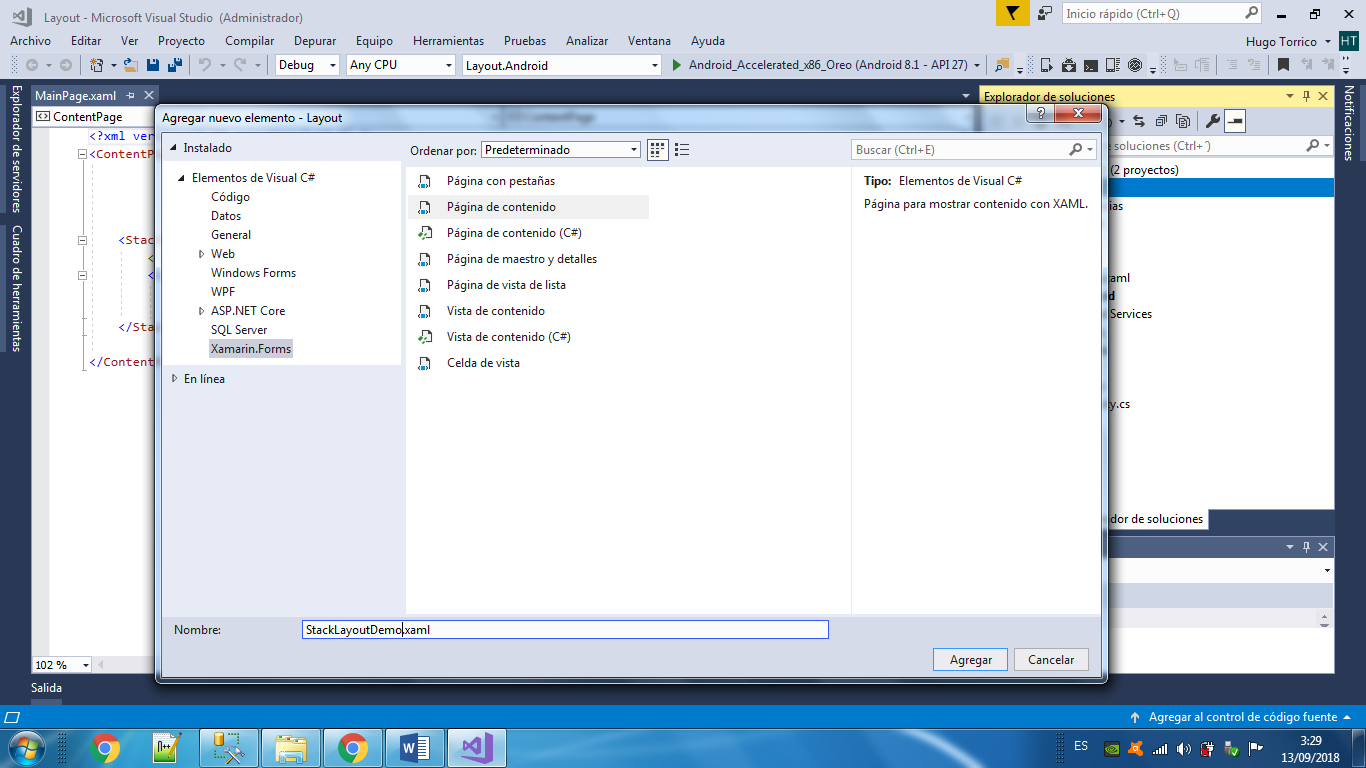
}

}



**AbsoluteLayout**

1. Agregar página de contenido de nombre “AbsoluteLayoutDemo”



1. Agregar el siguiente código:

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="LayoutSamples.AbsoluteLayoutExploration"

Title="Absolute Layout Exploration">

<ContentPage.Content>

<AbsoluteLayout>

<Label Text="I'm centered on iPhone 4 but no other device"

AbsoluteLayout.LayoutBounds="115,150,100,100" LineBreakMode="WordWrap" />

<Label Text="I'm bottom center on every device."

AbsoluteLayout.LayoutBounds=".5,1,.5,.1" AbsoluteLayout.LayoutFlags="All"

LineBreakMode="WordWrap" />

<BoxView Color="Olive" AbsoluteLayout.LayoutBounds="1,.5, 25, 100"

AbsoluteLayout.LayoutFlags="PositionProportional" />

<BoxView Color="Red" AbsoluteLayout.LayoutBounds="0,.5,25,100"

AbsoluteLayout.LayoutFlags="PositionProportional" />

<BoxView Color="Blue" AbsoluteLayout.LayoutBounds=".5,0,100,25"

AbsoluteLayout.LayoutFlags="PositionProportional" />

</AbsoluteLayout>

</ContentPage.Content>

</ContentPage>

C#

public class AbsoluteLayoutExplorationCode : ContentPage

{

public AbsoluteLayoutExplorationCode ()

{

Title = "Absolute Layout Exploration - Code";

var layout = new AbsoluteLayout();

var centerLabel = new Label {

Text = "I'm centered on iPhone 4 but no other device.",

LineBreakMode = LineBreakMode.WordWrap};

AbsoluteLayout.SetLayoutBounds (centerLabel, new Rectangle (115, 159, 100, 100));

// No need to set layout flags, absolute positioning is the default

var bottomLabel = new Label { Text = "I'm bottom center on every device.", LineBreakMode = LineBreakMode.WordWrap };

AbsoluteLayout.SetLayoutBounds (bottomLabel, new Rectangle (.5, 1, .5, .1));

AbsoluteLayout.SetLayoutFlags (bottomLabel, AbsoluteLayoutFlags.All);

var rightBox = new BoxView{ Color = Color.Olive };

AbsoluteLayout.SetLayoutBounds (rightBox, new Rectangle (1, .5, 25, 100));

AbsoluteLayout.SetLayoutFlags (rightBox, AbsoluteLayoutFlags.PositionProportional);

var leftBox = new BoxView{ Color = Color.Red };

AbsoluteLayout.SetLayoutBounds (leftBox, new Rectangle (0, .5, 25, 100));

AbsoluteLayout.SetLayoutFlags (leftBox, AbsoluteLayoutFlags.PositionProportional);

var topBox = new BoxView{ Color = Color.Blue };

AbsoluteLayout.SetLayoutBounds (topBox, new Rectangle (.5, 0, 100, 25));

AbsoluteLayout.SetLayoutFlags (topBox, AbsoluteLayoutFlags.PositionProportional);

layout.Children.Add (bottomLabel);

layout.Children.Add (centerLabel);

layout.Children.Add (rightBox);

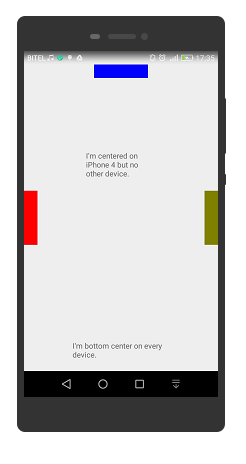
layout.Children.Add (leftBox);

layout.Children.Add (topBox);

Content = layout;

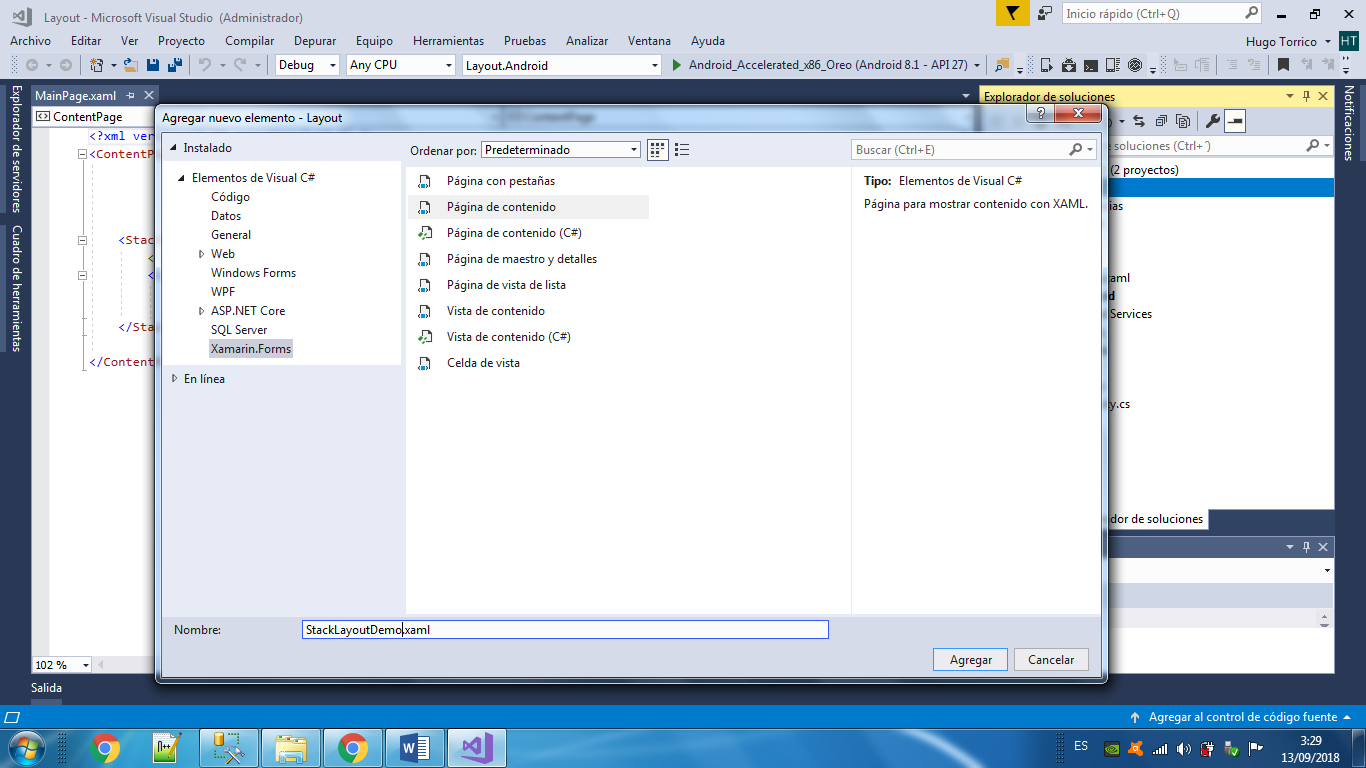
}

}



**RelativeLayout**

1. Agregar página de contenido de nombre “RelativeLayoutDemo”



1. Agregar el siguiente código:

XAML

<BoxView Color="Green"

WidthRequest="50"

HeightRequest="50"

RelativeLayout.XConstraint =

"{ConstraintExpression Type=RelativeToParent,

Property=Width,

Factor=0.5,

Constant=-100}"

RelativeLayout.YConstraint =

"{ConstraintExpression Type=RelativeToParent,

Property=Height,

Factor=0.5,

Constant=-100}" />

C#

layout.Children.Add(box, Constraint.RelativeToParent((parent) =>

{

return (.5 \* parent.Width) - 100;

}),

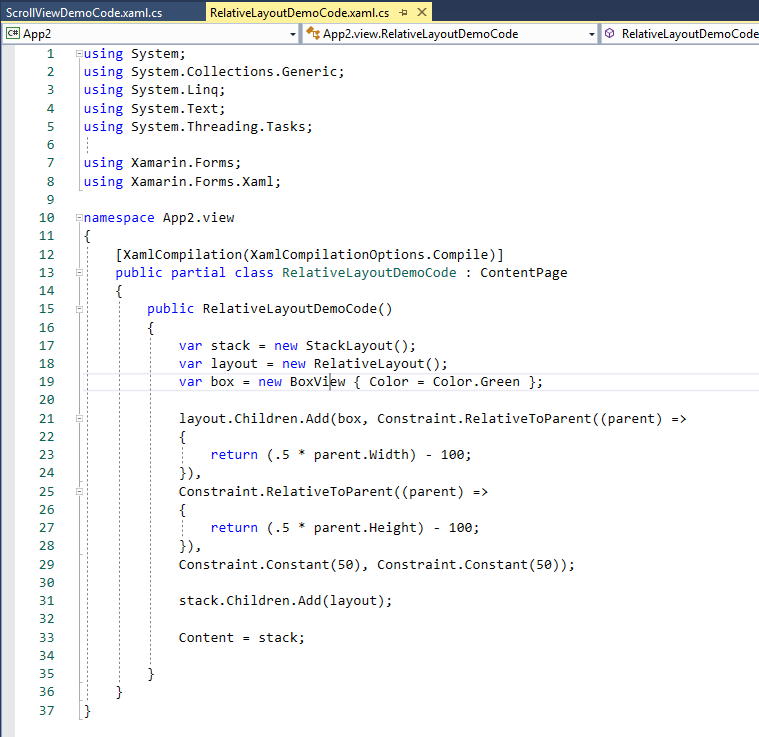
Constraint.RelativeToParent((parent) =>

{

return (.5 \* parent.Height) - 100;

}),

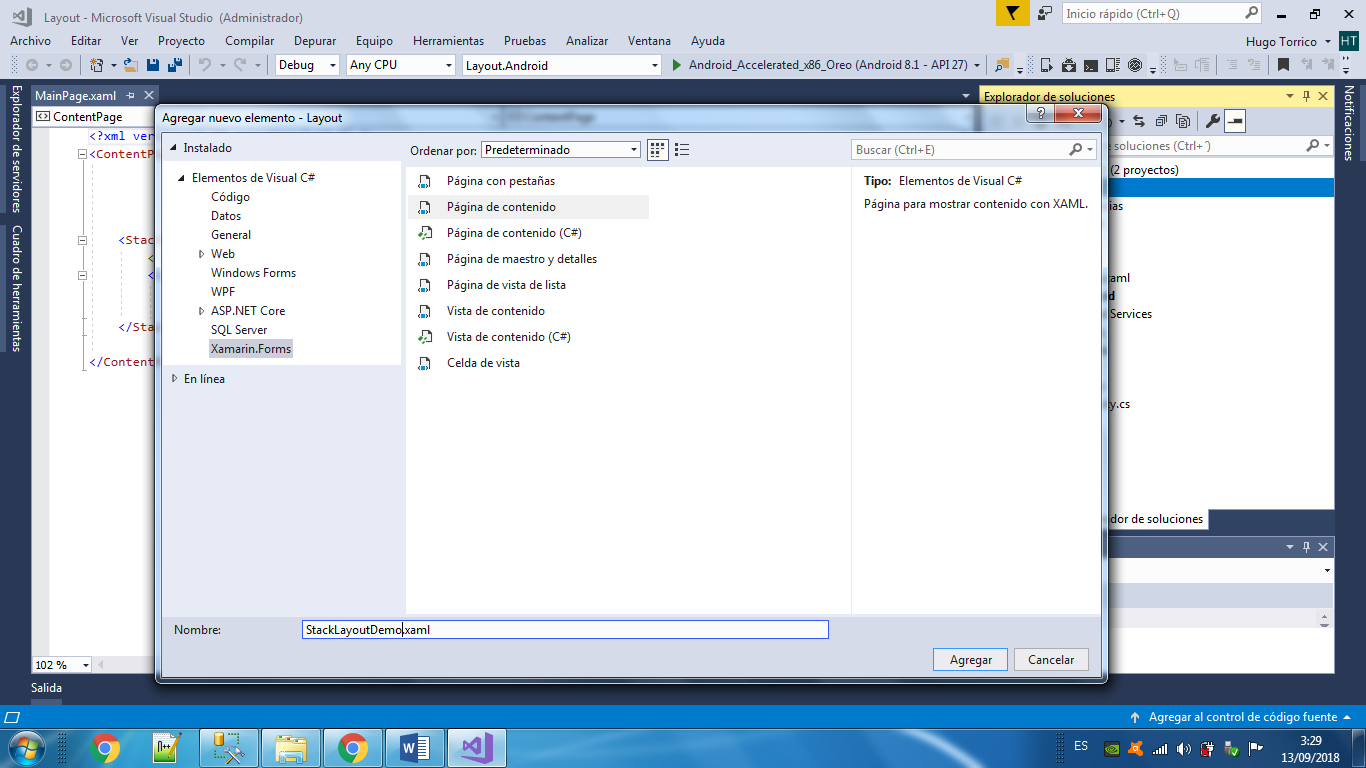
Constraint.Constant(50), Constraint.Constant(50));





**Grid**

1. Agregar página de contenido de nombre “GridDemo”



1. Agregar el siguiente código:

XAML

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="2\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="200" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

</Grid>

C#

var grid = new Grid();

grid.RowDefinitions.Add (new RowDefinition { Height = new GridLength(2, GridUnitType.Star) });

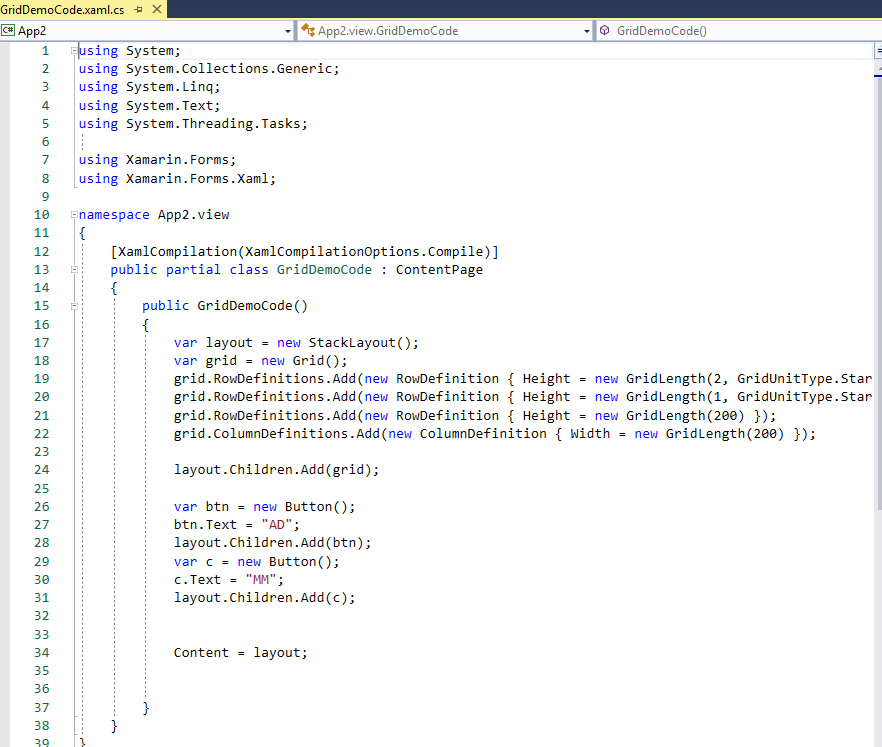
grid.RowDefinitions.Add (new RowDefinition { Height = new GridLength (1, GridUnitType.Star) });

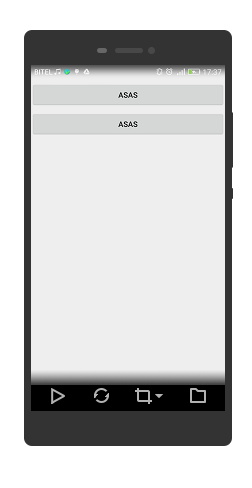
grid.RowDefinitions.Add (new RowDefinition { Height = new GridLength(200)});

grid.ColumnDefinitions.Add (new ColumnDefinition{ Width = new GridLength (200) });

Codigo:

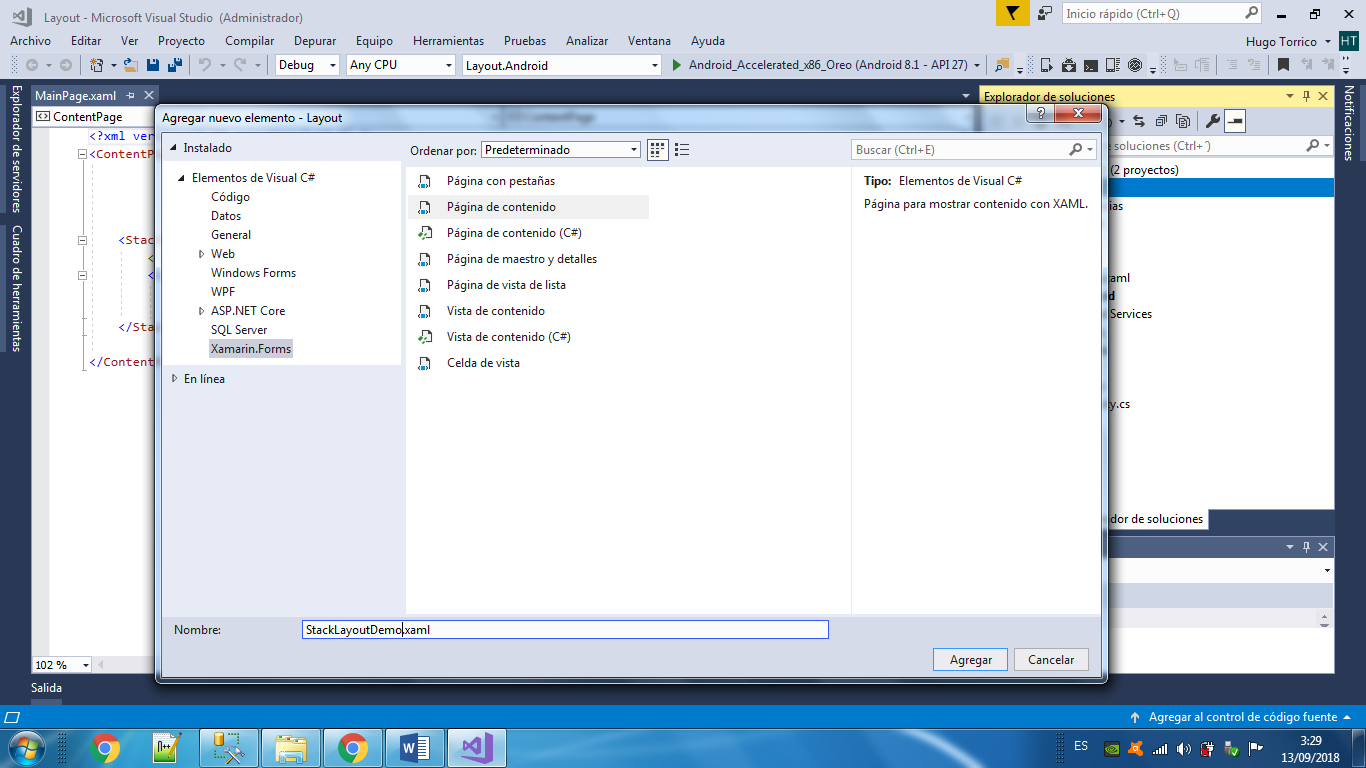






**ScrollView**

1. Agregar página de contenido de nombre “ScrollViewDemo”



1. Agregar el siguiente código:

XAML

<ContentPage.Content>

<ScrollView>

<StackLayout>

<BoxView BackgroundColor="Red" HeightRequest="600" WidthRequest="150" />

<Entry />

</StackLayout>

</ScrollView>

</ContentPage.Content>

C#

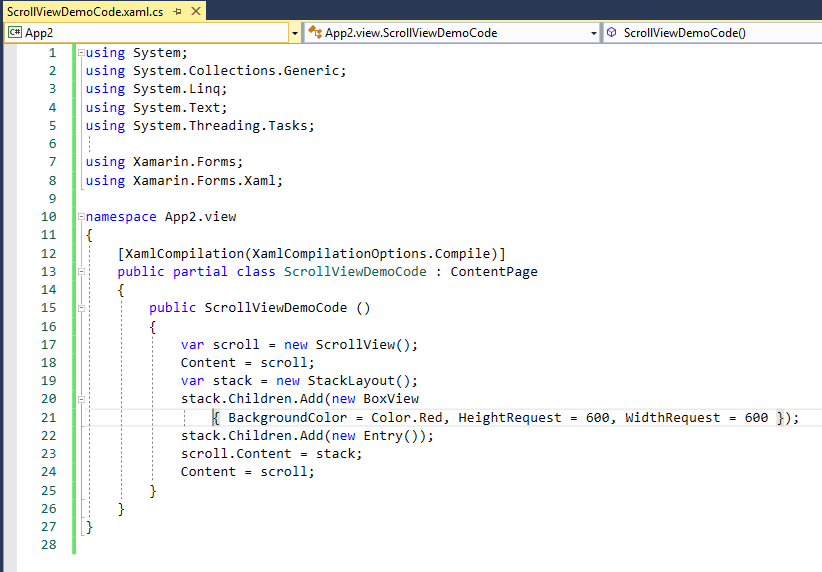
var scroll = new ScrollView();

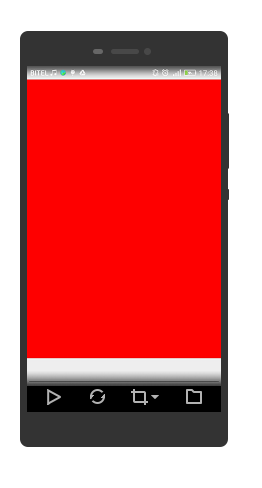
Content = scroll;

var stack = new StackLayout();

stack.Children.Add(new BoxView { BackgroundColor = Color.Red, HeightRequest = 600, WidthRequest = 600 });

stack.Children.Add(new Entry());





**Menu Principal**

using System.Threading.Tasks;

using App2.view;

using Xamarin.Forms;

namespace App2

{

public partial class MainPage : ContentPage

{

public MainPage()

{

InitializeComponent();

}

async private void ejem01(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new StackLayoutDemo());

}

async private void ejem02(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new AbsoluteLayoutDemo());

}

async private void ejem03(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new RelativeLayoutDemo());

}

async private void ejem04(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new GridDemo());

}

async private void ejem05(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new ScrollViewDemo());

}

async private void ejem06(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new GridDemoCode());

}

async private void ejem07(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new RelativeLayoutDemoCode());

}

async private void ejem08(object sender, EventArgs e)

{

await Navigation.PushModalAsync(new ScrollViewDemoCode());

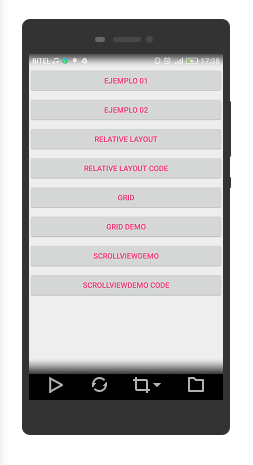
}

}

}







Link de Cogido Fuente:

<https://github.com/Camavilca/Aplicaciones-Multiplataforma/tree/master/CLASES/clase03>