



Repaso

RAY - Eliseo Martín Mora



¿Qué hemos hecho?

- 1 Crear un proyecto.
- 2 Modificar el aspecto visual (xaml).
- 3 Modificar el comportamiento (xaml.cs).
- 4 Condicionales.
- 5 Usar elementos visuales desde el código.

1 Creación del proyecto


Create a new project


Recent project templates


A list of your recently accessed templates will be displayed here.


Xamarin

LanguagePlatformProject type

 **Mobile App (Xamarin.Forms)**
A multiproject template for building apps for iOS and Android with Xamarin and Xamarin.Forms.
C# Android iOS Windows Mobile

 **Android App (Xamarin)**
Project templates for creating Android phone and tablet apps with Xamarin.
C# Android Mobile

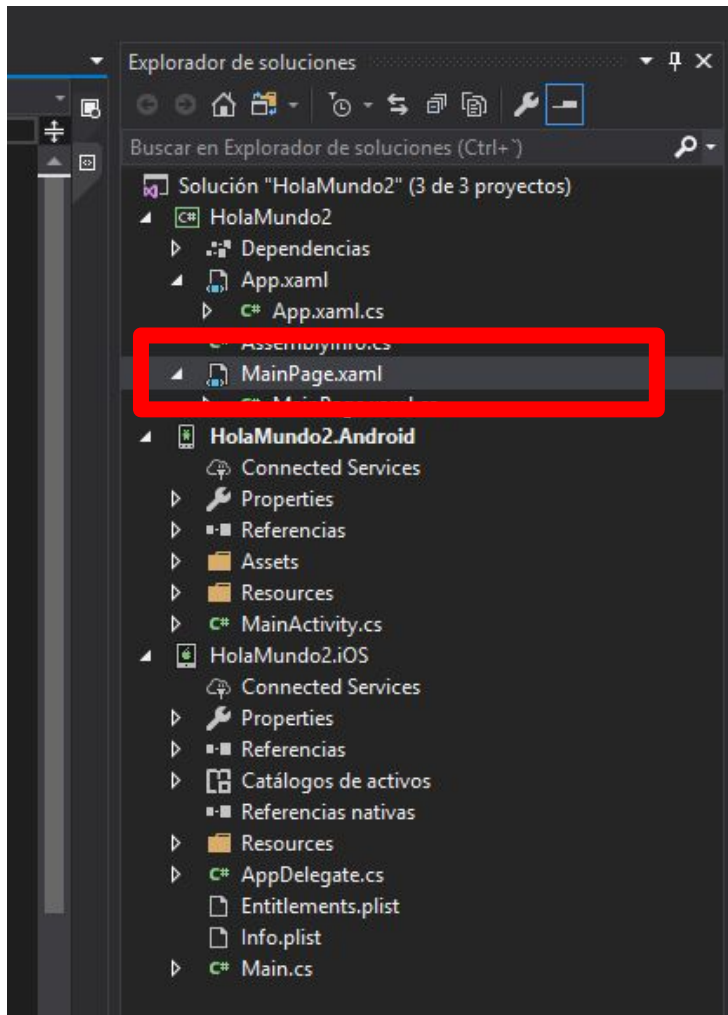
 **iOS App (Xamarin)**
Project templates for creating iOS apps for iPhone and iPad with Xamarin.
C# iOS Mobile

 **Android Wear App (Xamarin)**
A project for creating an Android Wear app with Xamarin.
C# Android Mobile

Back

Next

2 Modificar aspecto visual



Los ficheros acabados en .xaml serán los ficheros que contienen las pantallas, es decir si queremos añadir botones, textos, imágenes... irán en el fichero xaml que referencia a la pantalla.

```
<?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:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="HolaMundo2.MainPage">
  <StackLayout>
    <!-- Place new controls here -->
    <Label Text="Welcome to Xamarin.Forms!"
      HorizontalOptions="Center"
      VerticalOptions="CenterAndExpand" />
    <Button Text="Click Me" Clicked="Button_Clicked" />
  </StackLayout>
</ContentPage>
```

Indicamos que vamos a utilizar una página de contenido.

todos los parámetros suelen venir creados de manera automática.

Indica dónde acaba la página.

```
<?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:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="HolaMundo2.MainPage">

  <StackLayout>
    <!-- Place new controls here -->
    <Label Text="Welcome to Xamarin.Forms!"
      HorizontalOptions="Center"
      VerticalOptions="CenterAndExpand" />

    <Button Text="Click Me" Clicked="Button_Clicked" />
  </StackLayout>
</ContentPage>
```

StackLayout es la forma en la que estarán organizados los elementos que se encuentren entre esas dos etiquetas.

El StackLayout permite apilar elementos tanto en vertical como en horizontal.


```
<?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:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="HolaMundo2.MainPage">

  <StackLayout>
    <!-- Place new controls here -->
    <Label Text="Welcome to Xamarin.Forms!"
      HorizontalOptions="Center"
      VerticalOptions="CenterAndExpand" />

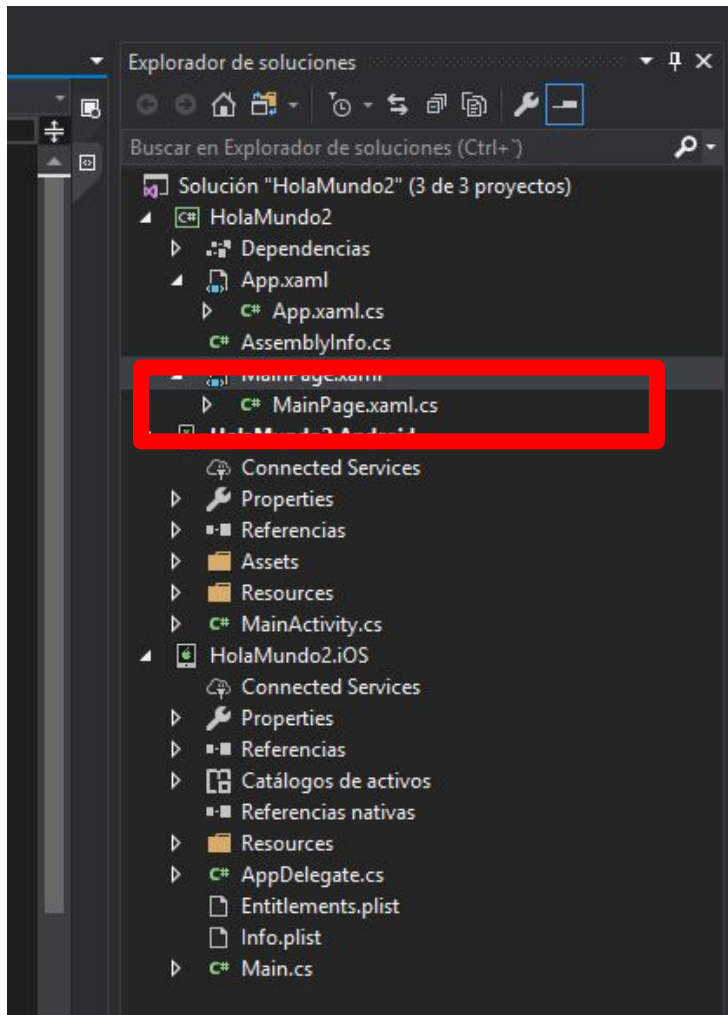
    <Button Text="Click Me" Clicked="Button_Clicked" />
  </StackLayout>
</ContentPage>
```

Label Text es el texto que se mostraba que ponía "Welcome to Xamarin.Forms!" vemos que tiene un atributo para centrarlo en horizontal y centrarlo en vertical.

```
<?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:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="HolaMundo2.MainPage">
  <StackLayout>
    <!-- Place new controls here -->
    <Label Text="Welcome to Xamarin.Forms!"
      HorizontalOptions="Center"
      VerticalOptions="CenterAndExpand" />
    <Button Text="Click Me" Clicked="Button_Clicked" />
  </StackLayout>
</ContentPage>
```

Este es el código que hemos añadido, se trata de un Botón que tenga el texto "Click Me" y que al ser clicado llama a la Función "Button_Clicked"

3 Modificar el comportamiento



Los ficheros acabados en .xaml.cs contienen el comportamiento, en este caso vamos a modificar el comportamiento de la pantalla inicial que se llama MainPage. El fichero a modificar será MainPage.xaml.cs

 MainPage()



MainPage.xaml.cs* X MainPage.xaml

C# HolaMundo

HolaMundo.MainPage

OnButtonClicked(object sender, EventArgs e)

```
5 using System.Text;
6 using System.Threading.Tasks;
7 using Xamarin.Forms;
8
9 namespace HolaMundo
10 {
11     // Learn more about making custom code visible in the
12     // by visiting https://aka.ms/xamarinforms-previewer
13     [DesignTimeVisible(false)]
14     public partial class MainPage : ContentPage
15     {
16         public MainPage()
17         {
18             InitializeComponent();
19         }
20
21         void OnButtonClicked(object sender, EventArgs e)
22         {
23         }
24     }
25 }
26
27
```

Creamos una función para modificar el comportamiento al pulsar el botón, el nombre de la función debe ser el mismo que pusimos en el .xaml

La función recibe por defecto unos parámetros que nos permitirán acceder por ejemplo al botón que ha llamado a la función que será en este caso llamado como **sender**


```
14 public partial class MainPage : ContentPage
15 {
16     1 referencia
17     public MainPage()
18     {
19         InitializeComponent();
20     }
21     int count = 0;
22
23     0 referencias
24     void OnButtonClicked(object sender,
25     {
26     }
27 }
28
29
```

Declaramos una variable de tipo **Integer** (entero) esto sirve para almacenar un número.

Queremos almacenar el número de veces que hacemos click en el botón


```
14 public partial class MainPage : ContentPage
15 {
16     1 referencia
17     public MainPage()
18     {
19         InitializeComponent();
20     }
21     int count = 0;
22
23     0 referencias
24     void OnButtonClicked(object sender,
25     {
26     }
27 }
28
29
```

Declaramos una variable de tipo **Integer** (entero) esto sirve para almacenar un número.

Queremos almacenar el número de veces que hacemos click en el botón

```
using System.Text;
6 using System.Threading.Tasks;
7 using Xamarin.Forms;
8
9 namespace HolaMundo
10 {
11     // Learn more about making custom code
12     // by visiting https://aka.ms/xamarinfor
13     [DesignTimeVisible(false)]
14     public partial class MainPage : ContentPage
15     {
16         public MainPage()
17         {
18             InitializeComponent();
19         }
20
21         int count = 0;
22
23         void OnButtonClicked(object sender, EventArgs e)
24         {
25             count++;
26             ((Button)sender).Text = $"Pulsado {count}";
27         }
28     }
29 }
30
31
```

Vamos a hacer que al pulsar el botón se aumente el contador de veces pulsado.

Podemos hacerlo de dos maneras

`count = count + 1;`

o simplificarlo con `count++`;

```
using System.Text;
6 using System.Threading.Tasks;
7 using Xamarin.Forms;
8
9 namespace HolaMundo
10 {
11     // Learn more about making custom code
12     // by visiting https://aka.ms/xamarininfo
13     [DesignTimeVisible(false)]
14     public partial class MainPage : ContentPage
15     {
16         public MainPage()
17         {
18             InitializeComponent();
19         }
20
21         int count = 0;
22
23         void OnButtonClicked(object sender, System.EventArgs e)
24         {
25             count++;
26             ((Button)sender).Text = $"Pulsado {count}";
27         }
28     }
29 }
30
31
```

Sabemos que sender es el objeto que ha llamado a la función y que es de tipo Button.

Así que le decimos que vamos a modificar el Texto del botón.

((Button)sender) nos permite acceder al botón y el botón tiene una propiedad llamada Text que contiene el texto vamos a cambiar el texto por el número de veces pulsado.

4 Condicionales



IF

If permite ejecutar el código si se cumple una determinada condición

```
if(condición)
```

```
{
```

```
// Hacer cosas
```

```
}
```



IF

```
if(edad < 18)
```

```
{
```

```
    Mostrar("Eres jovencito...");
```

```
}
```



IF

En este caso si la edad es menor que 18 años se mostrará el mensaje “Eres jovencito...”

```
if(edad < 18)
```

```
{
```

```
Mostrar(“Eres jovencito...”);
```

```
}
```



ELSE

En caso de que la condición no se cumpla podemos realizar otras acciones

```
if(condición)
```

```
{
```

```
// Hacer cosas
```

```
}
```

```
else
```

```
{
```

```
//Hacer cosas
```

```
}
```




ELSE

```
if(edad < 18)
```

```
{
```

```
    Mostrar("Eres jovencito...");
```

```
}
```

```
else
```

```
{
```

```
    Mostrar("Las urnas te llaman");
```

```
}
```



ELSE

En caso de ser mayor o igual que 18 años mostrará “Las urnas te llaman”

```
if(edad < 18)
```

```
{
```

```
Mostrar(“Eres jovencito...”);
```

```
}
```

```
else
```

```
{
```

```
Mostrar(“Las urnas te llaman”);
```

```
}
```



ELSE-IF

Puede suceder que busquemos
tener más condiciones añadidas

```
if(condición)
```

```
{
```

```
//Cosas
```

```
}
```

```
else if (condición2 && condición3)
```

```
{
```

```
//Otras cosas
```

```
}
```

```
else
```

```
{
```

```
//Cosas para el resto
```

```
}
```



ELSE-IF

Puede suceder que busquemos
tener más condiciones añadidas

```
if(edad < 18)
```

```
{
```

```
//Cosas de jóvenes
```

```
}
```

```
else if (edad >= 18 && edad <= 30)
```

```
{
```

```
//Cosas para edades entre 18 y 30
```

```
}
```

```
else
```

```
{
```

```
//Cosas para mayores de 30
```

```
}
```

5 Uso de elementos visuales desde el código.

Paso 1

El elemento visual que vamos a querer acceder necesita tener un nombre, a través del cual podrá ser identificado.

Esto se hace en el fichero 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"
  xmlns:d="http://xamarin.com/schemas/2014/forms/design"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="HolaMundo.MainPage">

  <StackLayout>
    <!-- Place new controls here -->
    <Label Text="Welcome to Xamarin.Forms!"
      HorizontalOptions="Center"
      VerticalOptions="CenterAndExpand" />

    <Button Text="Click me" Clicked="OnButtonClicked"/>
  </StackLayout>
</ContentPage>
```

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

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/xaml"
  xmlns:d="http://xamarin.com/schemas/2014/forms"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  x:Class="HolaMundo.MainPage">
```

Etiqueta que vamos a cambiar el texto al pulsar el botón

```
<StackLayout>
```

```
<!-- Place new controls here -->
```

```
<Label Text="Welcome to Xamarin.Forms!"
  HorizontalOptions="Center"
  VerticalOptions="CenterAndExpand" />
```

```
<Button Text="Click me" Clicked="OnButtonClicked"/>
```

```
</StackLayout>
```

```
</ContentPage>
```



```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3             xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml"
4             xmlns:d="http://xamarin.com/schemas/2014/forms"
5             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6             mc:Ignorable="d"
7             x:Class="HolaMundo.MainPage">
8
9     <StackLayout>
10         <!-- Place new controls here -->
11         <Label x:Name="mainLabel" Text="Welcome to Xamarin.Forms"
12               HorizontalOptions="Center"
13               VerticalOptions="CenterAndExpand" />
14
15         <Button Text="Click me" Clicked="OnButtonClicked" />
16     </StackLayout>
17
18 </ContentPage>
```

Le añadimos el parámetro `x:Name` a la etiqueta, de este modo podremos referirnos a la etiqueta desde el `xaml.cs` como `"mainLabel"` o el nombre que querramos

Paso 2

Vamos a modificar el contenido de la etiqueta según pulsamos el botón.

Esto será en el código xaml.cs

```
[DesignTimeVisible(false)]
```

5 referencias

```
public partial class MainPage
```

```
{
```

1 referencia

```
public MainPage()
```

```
{
```

```
InitializeComponent
```

```
}
```

```
int count = 0;
```

0 referencias

```
void OnButtonClicked(object sender, EventArgs e)
```

```
{
```

```
count++;
```

```
mainLabel.Text = $"El botón ha sido pulsado {count}";
```

```
((Button)sender).Text = $"Pulsado {count}";
```

```
}
```

```
}
```

Vamos a MainPage.xaml.cs y editamos la función que ejecuta el código al pulsar el botón.

Vamos a poner el número de veces que ha sido pulsado el botón en la etiqueta.

Para ello basta con usar el nombre de la etiqueta y su atributo Text

