



XAML en Xamarin.Forms

Ricardo Vasquez Sierra

www.ricardovasquez.postach.io



Microsoft
CERTIFIED
Professional

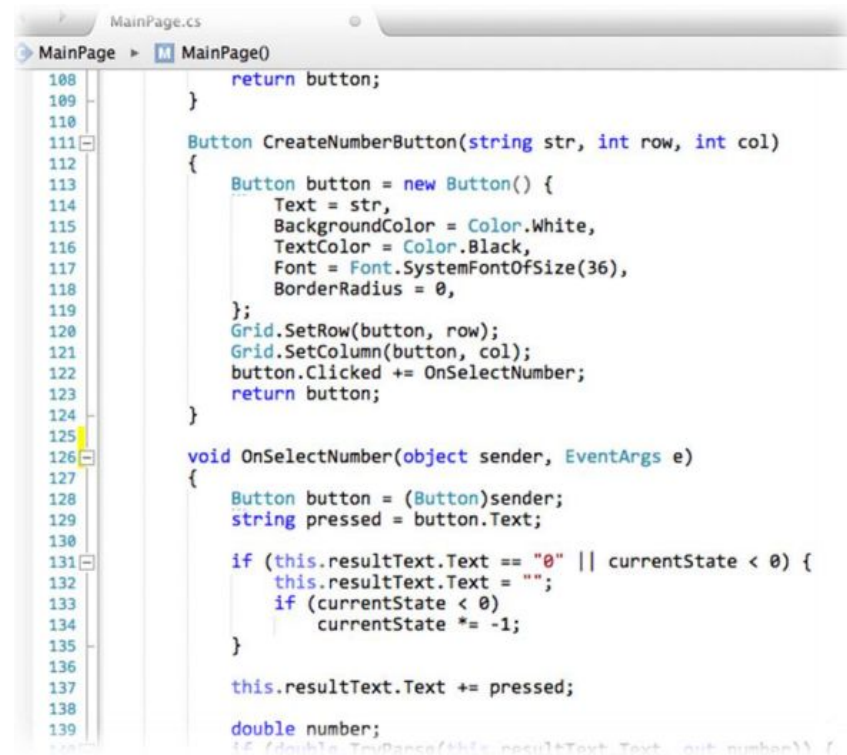
Agenda

- ❖ ¿Porque usar XAML?
- ❖ XAML 101
- ❖ Usando XAML con Xamarin.Forms



Creando Páginas en Código

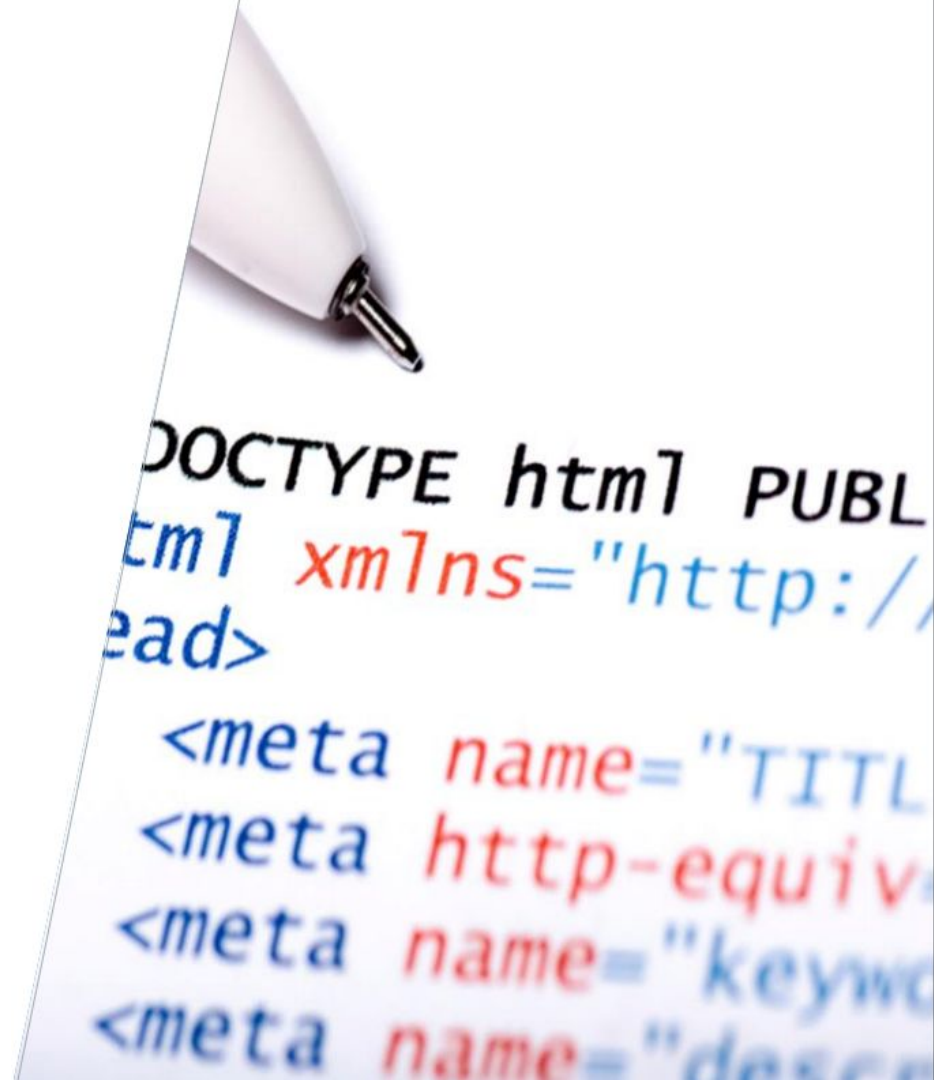
- ❖ Parte del code behind tiende a crear y configurar UI
- ❖ Colocar UI y comportamiento de la misma en un solo archivo puede originar código difícil de mantener/leer.
- ❖ Limita el trabajar en conjunto con diseñadores y que el desarrollador es forzado a hacer todo.



```
108         return button;
109     }
110
111     Button CreateNumberButton(string str, int row, int col)
112     {
113         Button button = new Button() {
114             Text = str,
115             BackgroundColor = Color.White,
116             TextColor = Color.Black,
117             Font = Font.SystemFontOfSize(36),
118             BorderRadius = 0,
119         };
120         Grid.SetRow(button, row);
121         Grid.SetColumn(button, col);
122         button.Clicked += OnSelectNumber;
123         return button;
124     }
125
126     void OnSelectNumber(object sender, EventArgs e)
127     {
128         Button button = (Button)sender;
129         string pressed = button.Text;
130
131         if (this.resultText.Text == "0" || currentState < 0) {
132             this.resultText.Text = "";
133             if (currentState < 0)
134                 currentState *= -1;
135         }
136
137         this.resultText.Text += pressed;
138
139         double number;
```

Lenguajes de Marcado

- ❖ Los lenguajes de marcado son excelentes para definir interfaces porque:
 - Fáciles de leer
 - Pueden ser extendidos



XAML(Extensible Application Markup Language)

- ❖ Creado por Microsoft para describir UI y permite separar la UI de su comportamiento.

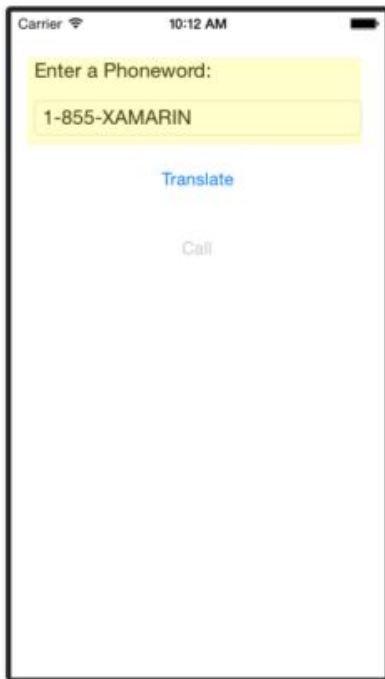


Xamarin Forms + XAML
= Sweetness!

Añadiendo una Página XAML



ContentPage para ocupar toda la pantalla



ContentView se utiliza a manera de control reutilizable

Tipos de Páginas en Xamarin.Forms



ContentPage



MasterDetailPage



NavigationPage



TabbedPage

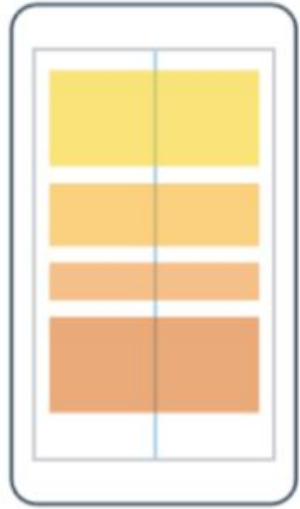


TemplatedPage

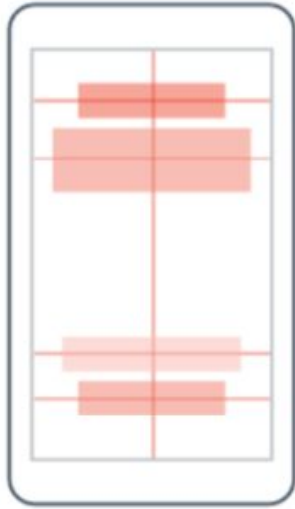


CarouselPage

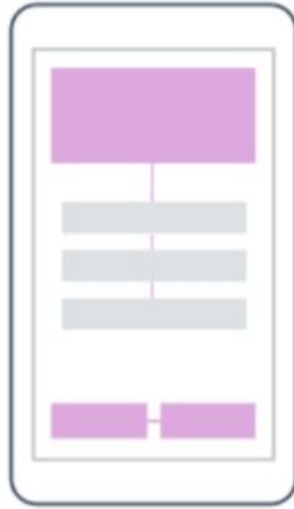
Tipos de Layouts en Xamarin.Forms



StackLayout



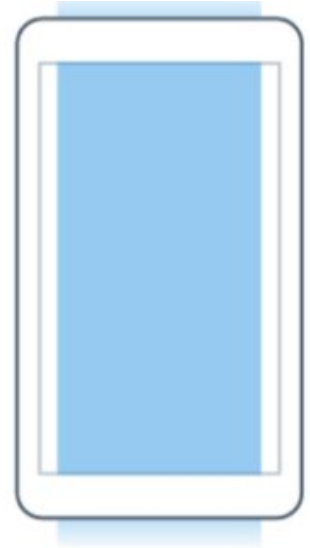
AbsoluteLayout



RelativeLayout



Grid



ScrollView

Demostración


Creando layouts en XAML



Tipos Personalizados

- ❖ XAML puede crear cualquier objeto público incluso si tiene constructores parametrizados. Solo hay que especificar dónde encontrar el tipo.

Must supply the namespace, and *possibly* the assembly, the type is defined in



```
<scg:List x:TypeArguments="x:String"
  xmlns:scg="clr-namespace:System.Collections.Generic;assembly=microsoft.win32.Task.Data"
  <x:String>One</x:String>
  <x:String>Two</x:String>
  <x:String>Three</x:String>
</scg:List>
```

Eventos en XAML

- ❖ Los manejadores de eventos deben de ser declarados en el archivo code behind respetando siempre la firma utilizada en el archivo XAML para evitar errores de compilación.

```
<Entry Placeholder="Number" TextChanged="OnTextChanged" />
```

```
public partial class MainPage : ContentPage
{
    ...
    void OnTextChanged(object sender, TextChangedEventArgs e) {
        ...
    }
}
```

Optimizando XAML (XAMLC)

- ❖ Permite que el código XAML sea compilado en vez de interpretado que tiene muchos beneficios entre los que destacan:
 - Detectar errores en tiempo de compilación en vez de en tiempo de ejecución.
 - Reduce el tamaño de la aplicación al no empaquetar los archivos xaml dentro de bundles.
 - Mejora el tiempo de carga de las pantallas ya que los elementos son instanciados de manera más rápida y eficiente.

```
using Xamarin.Forms.Xaml;

...

[XamlCompilation (XamlCompilationOptions.Compile)]
public class MainPage : ContentPage
{
    ...
}
```

```
using Xamarin.Forms.Xaml;

...

[assembly: XamlCompilation (XamlCompilationOptions.Compile)]
namespace App
{
    ...
}
```

Valores de Dispositivo

XAML permite especificar valores específicos dependiendo del OS actual a través de la propiedad **Device.OnPlatform** que puede ser utilizado desde XAML/C#.

x:TypeArguments used for generic instantiation



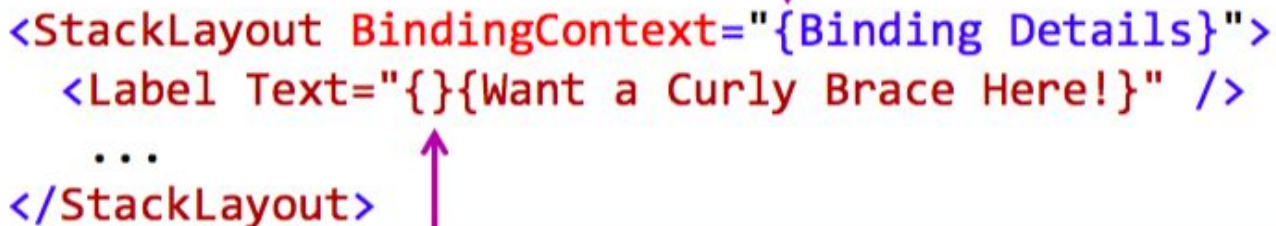
```
<StackLayout Spacing="10">
  <StackLayout.Padding>
    <OnPlatform x:TypeArguments="Thickness"
      iOS="0,20,0,0" Android="0" WinPhone="0" />
  </StackLayout.Padding>
  ...
</StackLayout>
```

can then supply different platform-specific value for property

Valores en tiempo de Ejecución

- ❖ XAML ofrece una manera de utilizar valores generados en tiempo de ejecución llamada **Extensiones de marcado**.

parser expects to find a class named **BindingExtension** that implements **IMarkupExtension** when it encounters the curly brace as the first character



```
<StackLayout BindingContext="{Binding Details}">  
    <Label Text="{}{Want a Curly Brace Here!}" />  
    ...  
</StackLayout>
```

literal curly braces need to be escaped properly to avoid a parser error

Utilizando propiedades Estáticas

- ❖ La extensión **x:Static** permite obtener el valor de miembros estáticos.

```
public static class Constants
{
    public static string Title = "Hello, Forms";
    public static Thickness Padding = new Thickness(5, Device.OnPlatform(20, 0, 0), 5, 0);
    public static Color TextColor = Color.Yellow;
}
```

```
<ContentPage ... Padding="{x:Static me:Constants.Padding}">
    <Label Text="{x:Static me:Constants.Title}"
           TextColor="{x:Static me:Constants.TextColor}" />
</ContentPage>
```

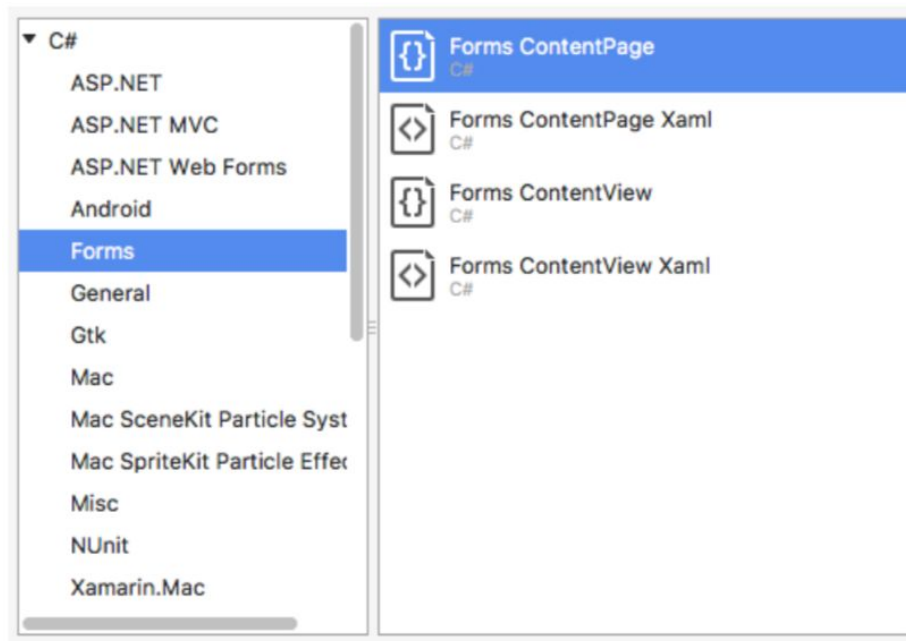
Otras Extensiones de Marcado

- ❖ Para usar recursos **{StaticResource}** y **{DynamicResource}**
- ❖ Añadir un valor null **{x:null}**
- ❖ Buscar un tipo determinado **{x:type}**
- ❖ Crear un arreglo **{x:Array}**

```
<ListView SelectedItem="{x:Null}">
  <ListView.ItemsSource>
    <x:Array Type="{x:type x:Int32}">
      <x:Int32>10</x:Int32>
      <x:Int32>20</x:Int32>
      <x:Int32>30</x:Int32>
    </x:Array>
  </ListView.ItemsSource>
</ListView>
```


Controles personalizados en XAML

- ❖ Similar al concepto de fragmentos en **Android** o **User Controls** en Windows. Permite reutilizar partes específicas de una UI donde se requiera.



Demostración

Creando controles XAML





Gracias! :)

Referencias:

Xamarin University



Microsoft
CERTIFIED
Professional