Desarrollo Móvil con Xamarin

3 – XAML en Xamarin.Forms

Contenido

- ¿Por qué usar XAML?
- Manejo de eventos en XAML
- Markup extensions
- Valores para plataformas específicas en XAML

¿Por qué usar XAML?

- **XAML** (e**X**tensible **A**pplication **M**arkup **L**anguage) es un lenguaje creado por Microsoft para crear interfaces de usuario (UI).

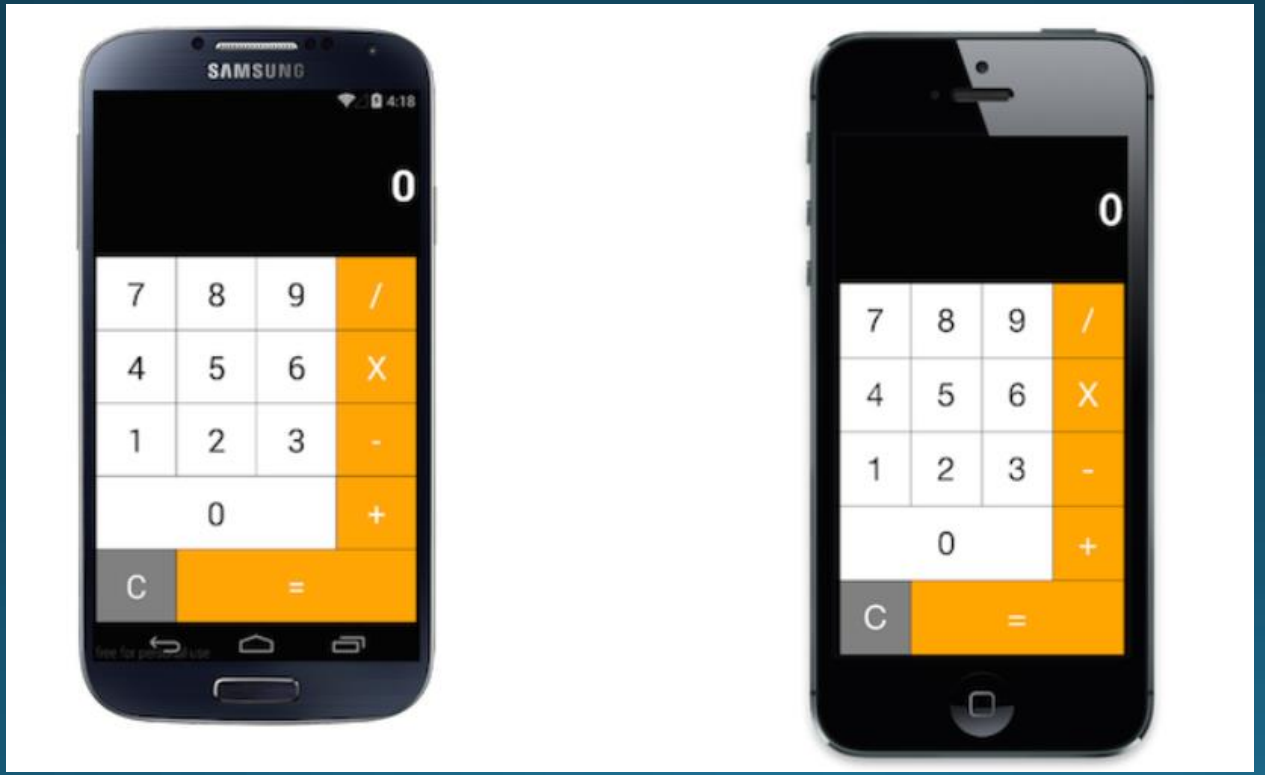
- En el ejemplo de Phoneword, la UI de la aplicación (un label, una caja de texto y dos botones) estaba definida en C#. En ese ejercicio, tanto la UI como su comportamiento estaban definidos en C#.

- **XAML** permite que la UI esté definida en su propio archivo, de manera que nuestro código C# solo defina comportamiento.

```
public MainPage()
{
    prompt          = new Label ();
    phoneNumberText = new Entry ();
    translateButton = new Button();
    callButton      = new Button();

    var panel = new StackLayout();
    panel.Children.Add(prompt);
    panel.Children.Add(phoneNumberText);
    panel.Children.Add(translateButton);
    panel.Children.Add(callButton);

    this.Content = panel;
}
```

```
<ContentPage>

    <ContentPage.Content>

        <StackLayout >
            <Label  />
            <Entry  />
            <Button />
            <Button />
        </StackLayout>

    </ContentPage.Content>

</ContentPage>
```

- Ejercicio #1 – Calculadora en XAML



Manejo de eventos en XAML

- Para acceder a los elementos definidos en XAML a través del codebehind, hay que asignarles un nombre.
- Esto se logra con la propiedad **x:Name** de XAML.
- La propiedad agrega un atributo privado a la clase parcial (codebehind) que podemos utilizar para interactuar con el elemento.

- También se pueden definir eventos como atributos de los controles:

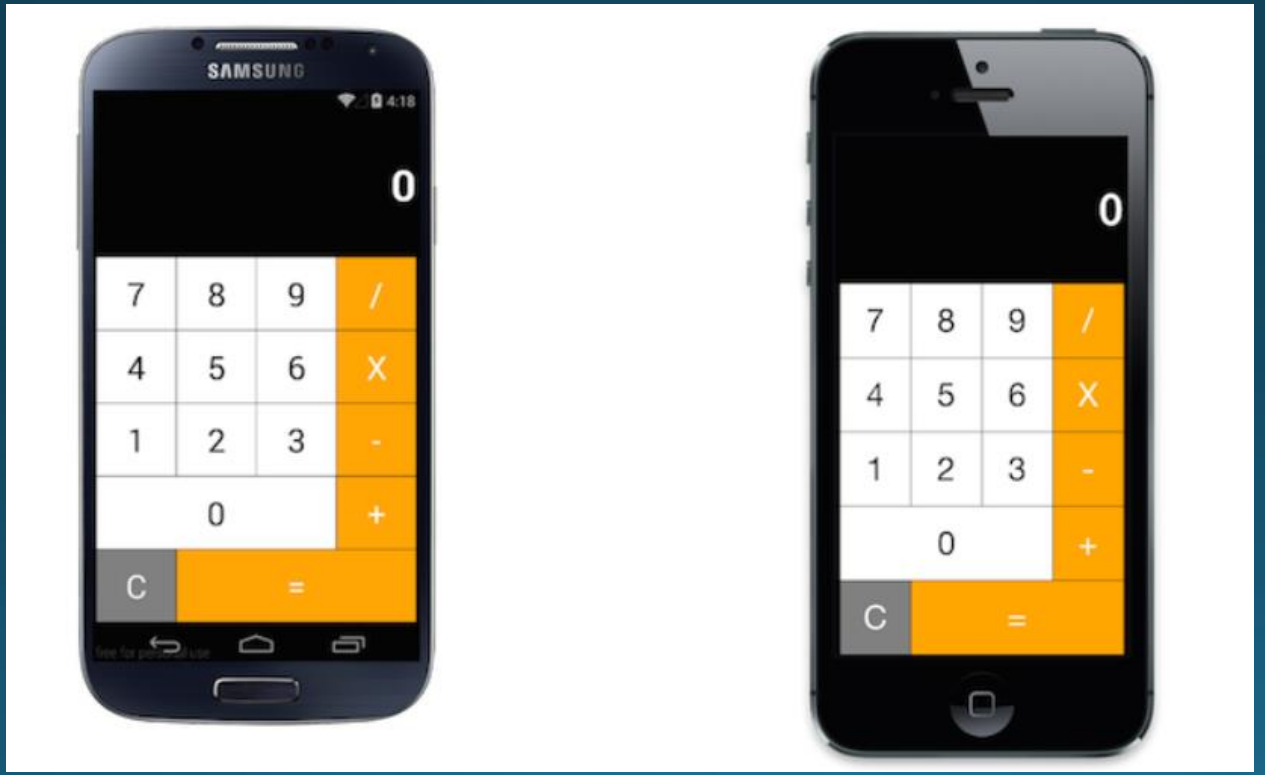
MainPage.xaml

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

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

MainPage.xaml.cs

- Ejercicio #1 – Calculadora en XAML (eventos)



Markup extensions

- Una gran parte de la especificación que definamos en XAML se sabrá de antemano, en **tiempo de compilación**, al definir valores literales en las propiedades de los controles.

- Sin embargo, habrá situaciones en las que el valor de alguna propiedad sólo se pueda conocer en **tiempo de ejecución** (como resultado de alguna operación o requerimiento desde el servidor, por ejemplo).

- Para este propósito existen las extensiones de marcado (**markup extensions**).

- Ejercicio #1 – Calculadora en XAML (markup extension)