



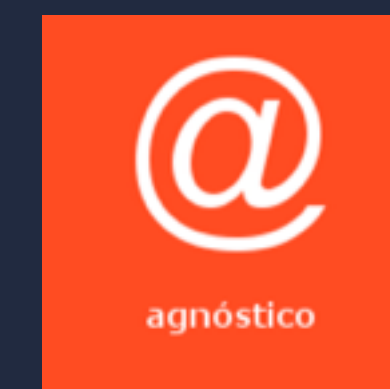
MASTERING XAML - XAMARIN FORMS

By Charlin Agramonte



Xamarin mobile developer

Charlin Agramonte
@Chard003





¿Que es XAML?



eXtended Application Markup Language



Estructura basica

```
<?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="MastetingXamlExamples.Pages.MyPage">
</ContentPage>
```



Syntaxis básica

```
var label = new Label () { Text="Hello 1"};  
var label2 = new Label () { Text = "Hello 2" };  
  
Content = new StackLayout {  
    Orientation = StackOrientation.Horizontal,  
    Children =  
    {  
        label,  
        label2  
    }  
};
```

```
<StackLayout Orientation="Horizontal">  
    <Label Text="Hello 1"/>  
    <Label Text="Hello 2"/>  
</StackLayout>
```




Syntaxis básica

```
<?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="MastetingXamlExamples.Pages.MyPage">
  <ContentPage.Content>
    <StackLayout>
      <Label />
      <Button/>
      <DatePicker/>
      <SearchBar/>
      <Slider />
      <Switch />
      <BoxView/>
      <Image/>
      <Picker/>
    </StackLayout>
  </ContentPage.Content>
</ContentPage>
```

```
<StackLayout VerticalOptions="Center"
              WidthRequest="200">
  <Label Text="I'm a text"
          TextColor="Blue"/>
</StackLayout>
```



Styles

Permite personalizar la apariencia de un control, así mismo como eliminar código repetitivo y propenso a errores.

Tipos

- Explicitos
- Implícitos
- Globales
- Herencia
- Dinámicos
- Dispositivos



Content Views

Su finalidad es la de servir como una clase base para vistas compuestas definidos por el usuario .



TRIGGERS

“Permite cambiar acciones declaradas en Xaml que cambian su apariencia en eventos o `propertyChanges`”



BEHAVIORS

“Permite agregar funcionalidad a los controles de interfaz de usuario sin tener que agregar una subclase.”



DATA BINDING

Permiten unir propiedades de dos objetos de modo que un cambio en uno causa un cambio en la otra. Esta es una herramienta muy valiosa.



Ventajas y desventajas

- Código más corto fácil y de leer
 - Puede ser escrito a mano, pero también se presta para ser utilizado en herramienta de diseños
 - Código usando Xaml es mas fácil de mantener y modificar
- No continue c# code dentro de el
 - No continue Loops para procesos repetitivos

QUIZAS NO SABIAS EN XAML



- XamlCompilation
- Usar StringFormat [https://msdn.microsoft.com/en-us/library/system.string.format\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string.format(v=vs.110).aspx)
- Salto de Linea
- Pasar elementos por el constructor <http://www.computermagic.gr/snippets/xamarin-forms/xaml/constructor-with-arguments/>
- Hacer emoticons <http://www.charbase.com/block/emoticons>
- Usar Margenes/paddings negativos = AbsoluteLayout



Resumen

- Utilizar Styles siempre, puede salvarte la vida luego
- EventHandlers en CodeBehind sustituyelo por DataBinding siempre y cuando sea posible
- Usa Triggers cuando quieras trabajar con eventos
- Usa Behaviors para modificar el comportamiento de un componente



REFERENCIAS

- XAML <https://developer.xamarin.com/guides/xamarin-forms/xaml/>
- Creating Mobile Apps with Xamarin.Forms <https://developer.xamarin.com/guides/xamarin-forms/creating-mobile-apps-xamarin-forms/>
- Xaml vs Code <https://adventuresinxamarinforms.com/2015/03/17/xaml-v-code/>
- Pasar elementos por el constructor <http://www.computermagic.gr/snippets/xamarin-forms/xaml/constructor-with-arguments/>
- Emoticons Referencia <http://www.charbase.com/block/emoticons>

Recomendado

Xamarin Forms Performance Tips

<http://kent-boogaart.com/blog/jason-smith's-xamarin-forms-performance-tips>



PREGUNTAS