

MOBILE
TECH
WEEK



Xamarin



Pablo Piovano



@ppiova



ppiova

Integrante de CUBITS y CABL

Fan of Microsoft Technologies. Xamarin Forms / Blazor

Microsoft Azure enthusiast (AZ-900 Certificate Profesional)

Open Source Contributor

Participo en varios grupos de tecnologías

¿Por qué aprender desarrollo Mobile?

El mundo se volvió Mobile

Cada vez se utilizan más los dispositivos

Video llamadas

Trabajo cotidiano

Para estudiar

Redes Sociales

Games!!!



Desarrollo Nativo

¿Qué debemos saber?



- Mac OS X
- Xcode
- Objective-C
- Swift
- IOS SDK
- Apple Tools



- OS
- Android Studio
- Java
- Kotlin
- Android SDK
- SDK Tools



- Windows OS
- Visual Studio
- C#
- F#
- .NET Framework
- UWP

Mobile Complexity

On Mobile, Quality is Hard



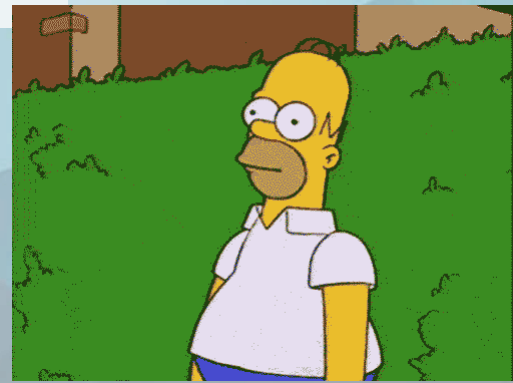
5 OS versions
20 devices
20 languages
35 locales
6 screen sizes



9 OS versions
6 Screen configurations
27 screen sizes
39 languages

57 locales
15 Manufacturers
Over **19,000** distinct devices

Una gran fragmentación de Sistemas Operativos,
Devices, Screens y Fabricantes



Historia

Miguel de Icaza @migueldeicaza.

Creador del proyecto MONO.



Objetivo era deployar apps .NET en ambientes Linux.

Lo compro NOVEL (quedo abandonado ese proyecto).

Miguel recuperó la licencia y funda Xamarin (similar idea en móvil).

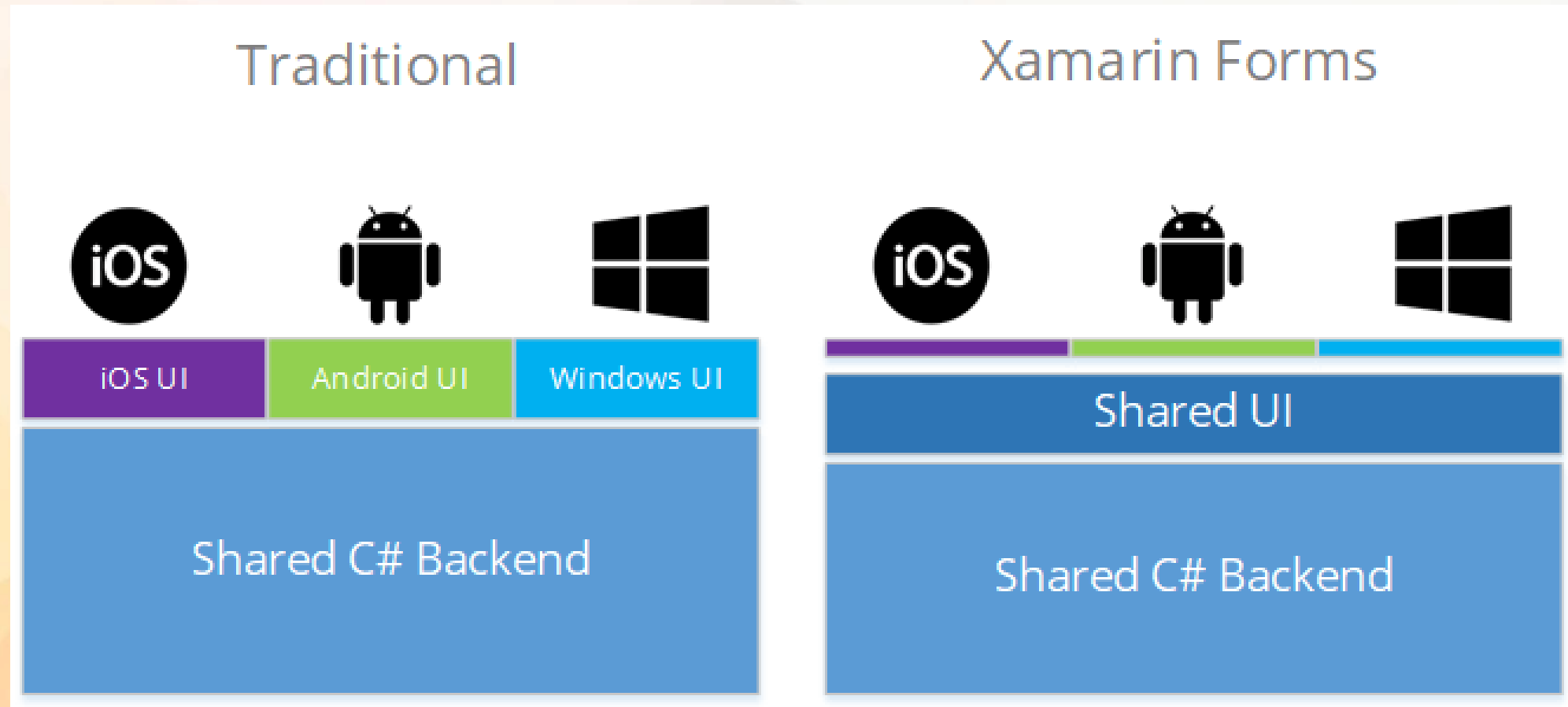
En el 2016 lo compra Microsoft. (sin licencia y proyecto OpenSource)

En estos últimos años se generó una gran Comunidad (**Mostrar Links**)

Microsoft dio un gran enfoque a Xamarin Forms!!!



Tipos de desarrollo en Xamarin



Xamarin es una plataforma de desarrollo de aplicaciones móviles para construir aplicaciones nativas de iOS, Android y Windows desde una base de código C#/.NET común.

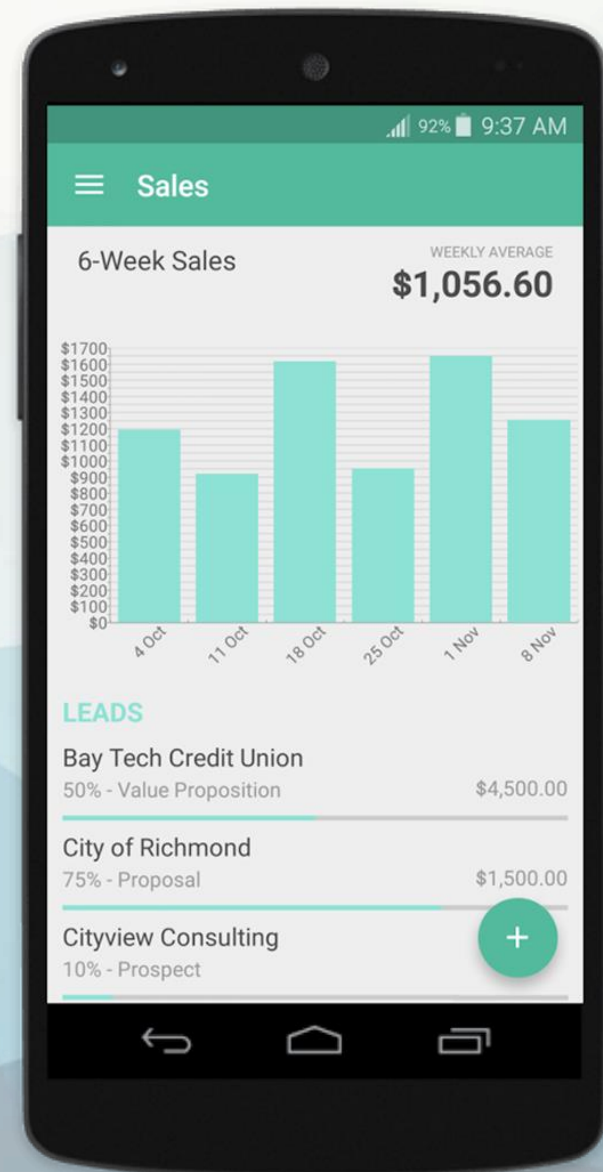
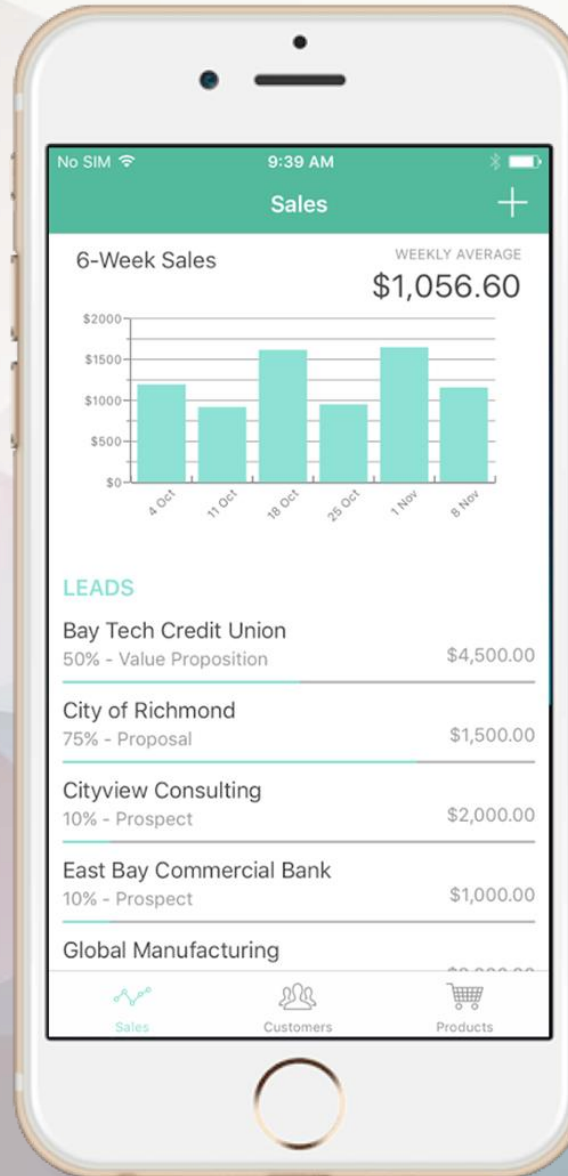
Xamarin Forms es un UI framework que nos permite crear interfaces nativas que pueden compartirse entre todas las plataformas (iOS, Android & Windows).

Xamarin Forms UI

Hacemos las vistas “UNA SOLA VEZ”, pero la forma en como se va a mostrar va a depender de cada plataforma (Android o iOS)

El UI en Xamarin Forms se puede definir de dos formas, por XAML o por code behind (C#).

Con XAML se puede utilizar para separar la capa de presentación de la lógica en nuestras aplicaciones.



Rompiendo mitos

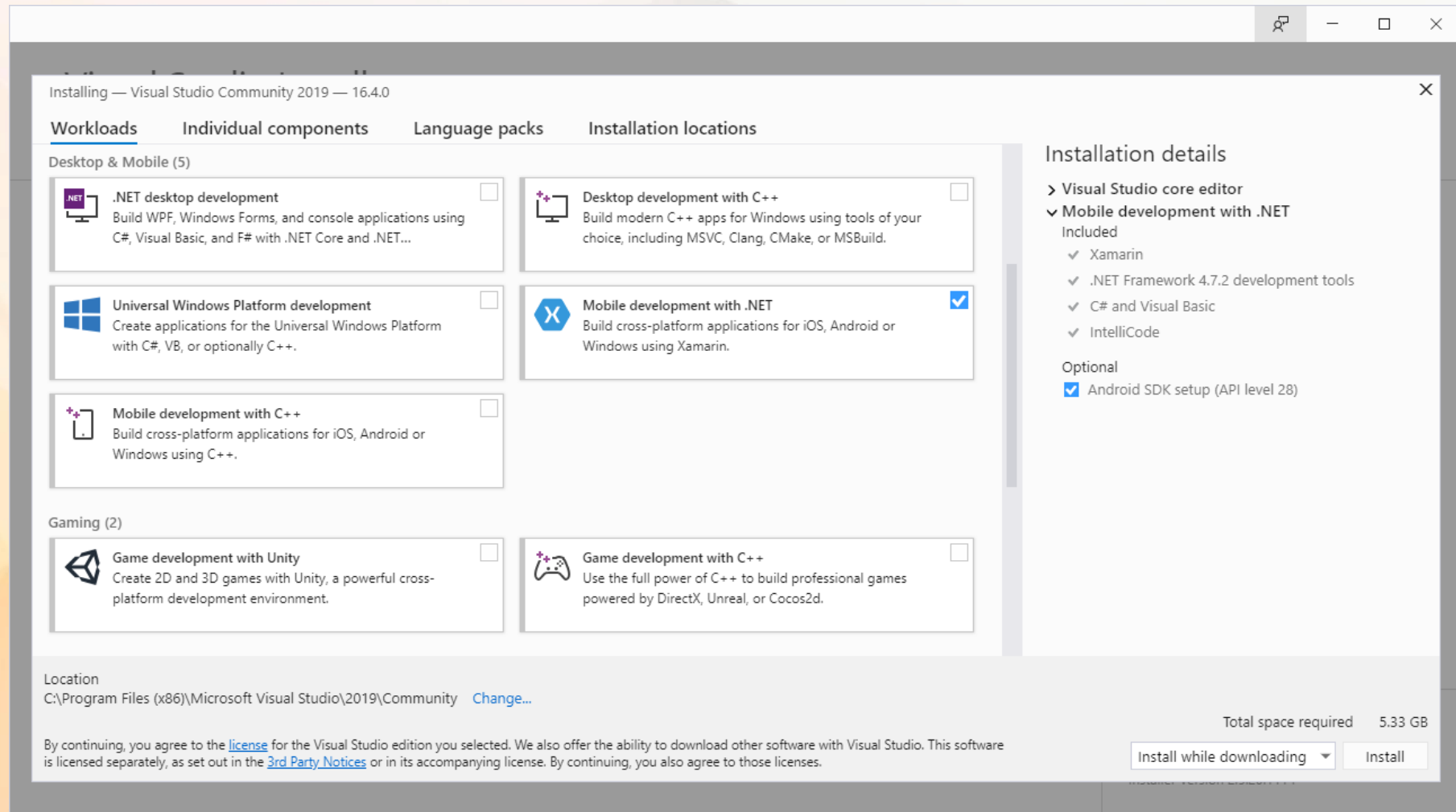


- No hay que pagar Licencias
- No es solamente para hacer aplicaciones empresariales
- Es de simple instalación
- La performance es muy buena y la fueron mejorando
- Funciona muy bien, ejemplo FishAngler (<https://www.fishangler.com/>)
- Microsoft no le agregó costos: no esta orientado cobrar licencia por los desarrollos, sino quiere que mas developers utilicen su tecnología y desarrollen en la misma. Tiene una Filosofia muy distinta de hace varios años atrás.

Ventajas

- Necesito 1 solo IDE - Visual Studio
- No hay costo de licencia
- Utiliza C# como lenguaje
- Puedo compartir vistas entre las plataformas
- La Curva de aprendizaje es rápida
- Puedo crear componentes reutilizables
- Hay una comunidad muy grande y sobretodo muy activa, siempre aparecen nuevos controles o features para facilitarnos el desarrollo
- Es de código abierto (Repository Git)

Instalación Xamarin



XAML

XAML (Extensible Application Markup Language) es un lenguaje de marcado basado en XML que se usa para la creación de interfaces de usuario en Xamarin.Forms, sin necesidad de utilizar código.

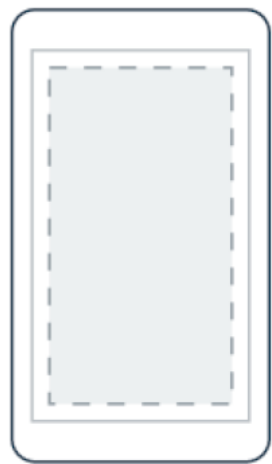
XAML

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="XamlSamples.HelloXamlPage"
              Title="Hello XAML Page">
  <ContentPage.Content>

    <Label Text="Hello, XAML!"
           VerticalOptions="Center"
           HorizontalTextAlignment="Center"
           Rotation="-15"
           IsVisible="true"
           FontSize="Large"
           FontAttributes="Bold"
           TextColor="Blue" />

  </ContentPage.Content>
</ContentPage>
```

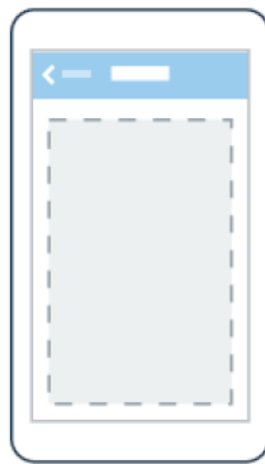
Pages



ContentPage



MasterDetailPage



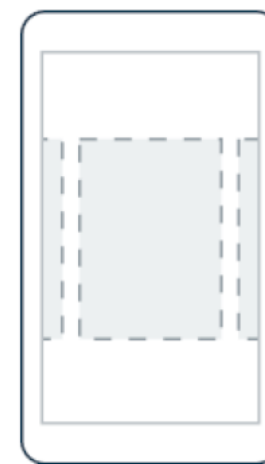
NavigationPage



TabbedPage



TemplatedPage



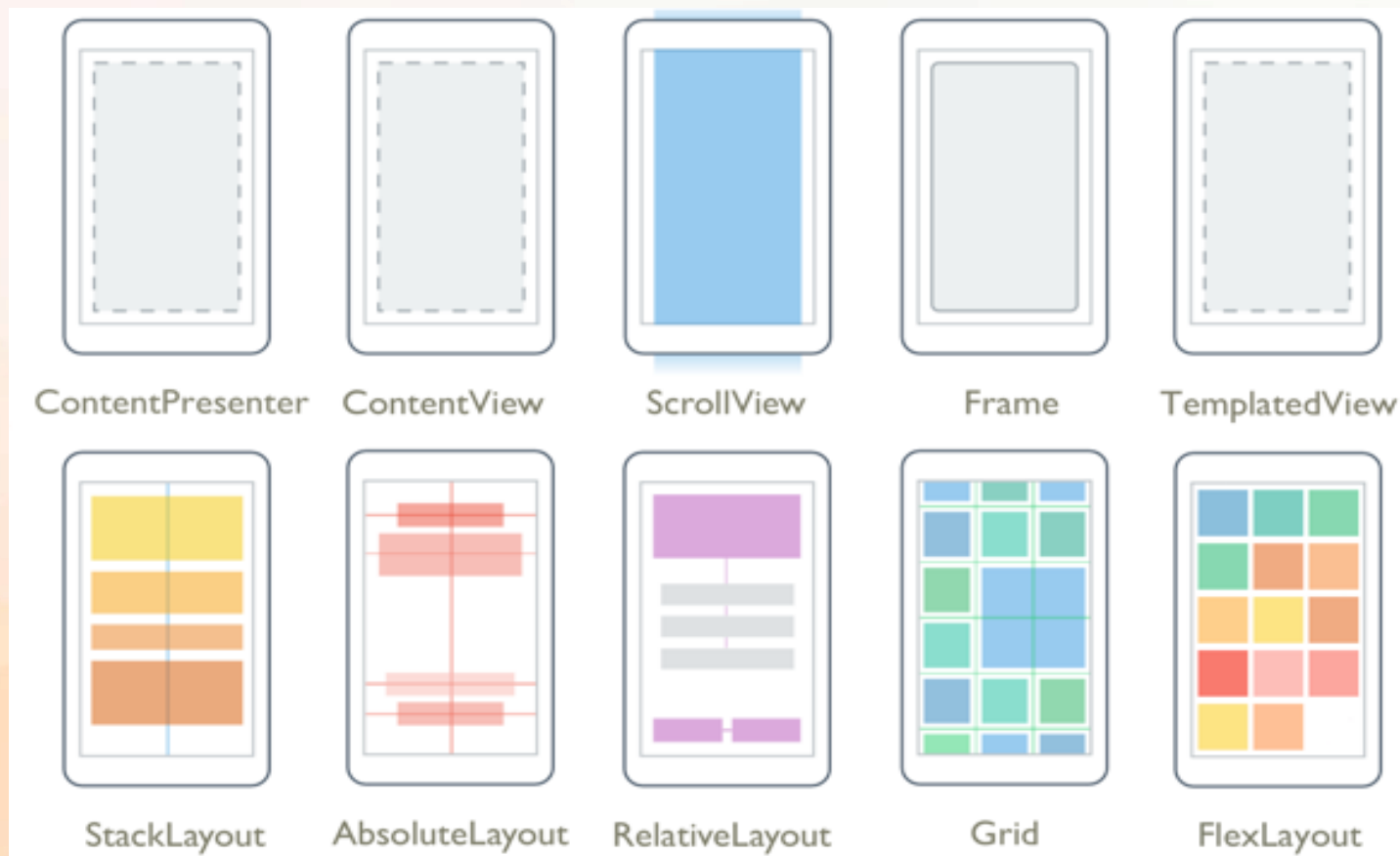
CarouselPage

Las páginas son elementos contenedores que representan una pantalla de la aplicación.

Xamarin.Forms.Page representa un **ViewController** en iOS, un **Page** en UWP, en Android cada página se comporta como un **Activity**, pero no lo son.

<https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/controls/pages>

Layouts



Los layouts son elementos **contenedores** de otros layouts o vistas (controles), son especialmente necesarios ya que las páginas sólo pueden contener un elemento hijo.

Los utilizaremos para establecer la posición y alineación de los elementos que contienen.

<https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/controls/layouts>

Controles

Las vistas son objetos de la interfaz de usuario como etiquetas, botones y controles deslizantes que se conocen comúnmente como *controles* o *widgets* en otros entornos de programación gráfica.

```
<Label Text="Hello, Xamarin.Forms!"  
  FontSize="Large"  
  FontAttributes="Italic"  
  HorizontalTextAlignment="Center" />
```

```
<Entry Keyboard="Email"  
  Placeholder="Enter email address"  
  VerticalOptions="CenterAndExpand" />
```

```
<Button Text="Click Me!"  
  Font="Large"  
  BorderWidth="1"  
  HorizontalOptions="Center"  
  VerticalOptions="CenterAndExpand"  
  Clicked="OnButtonClicked" />
```

```
<ListView ItemsSource="{Binding Monkeys}"  
  ItemTemplate="{StaticResource MonkeyTemplate}" />
```

Y muchos más!!!

<https://docs.microsoft.com/en-us/xamarin/xamarin-forms/xaml/xaml-controls>



Baboon

Africa & Asia



Capuchin Monkey

Central & South America

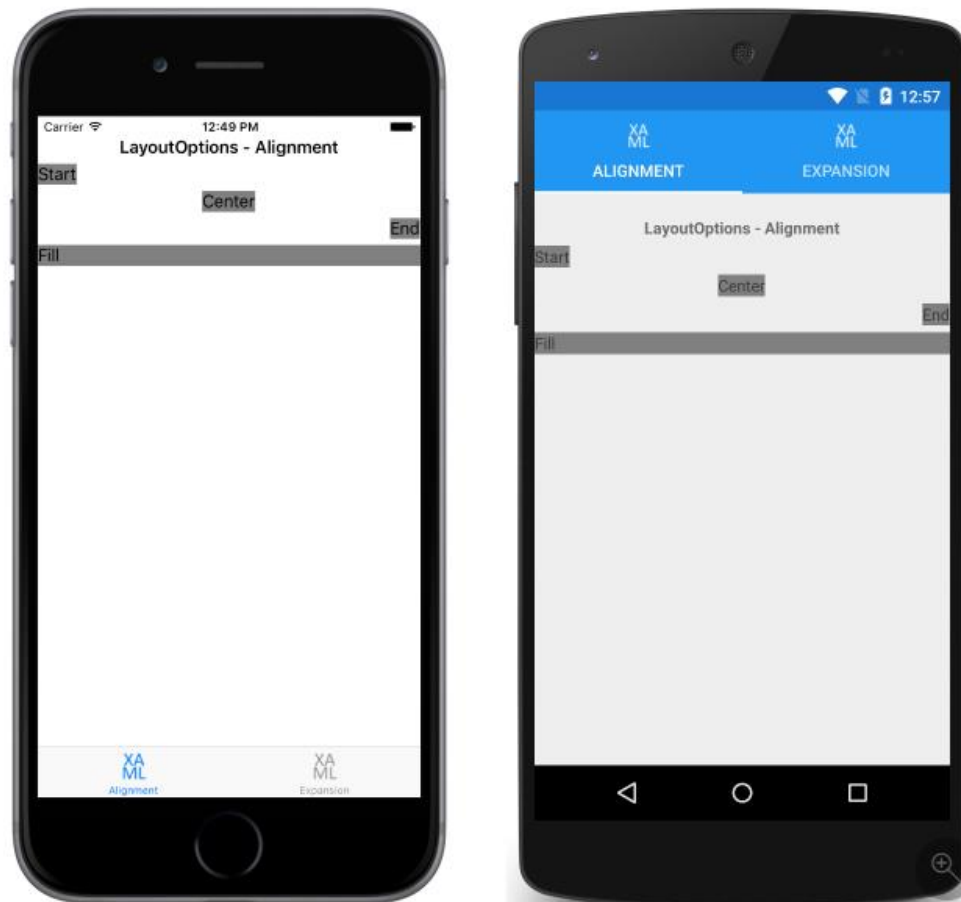


Blue Monkey

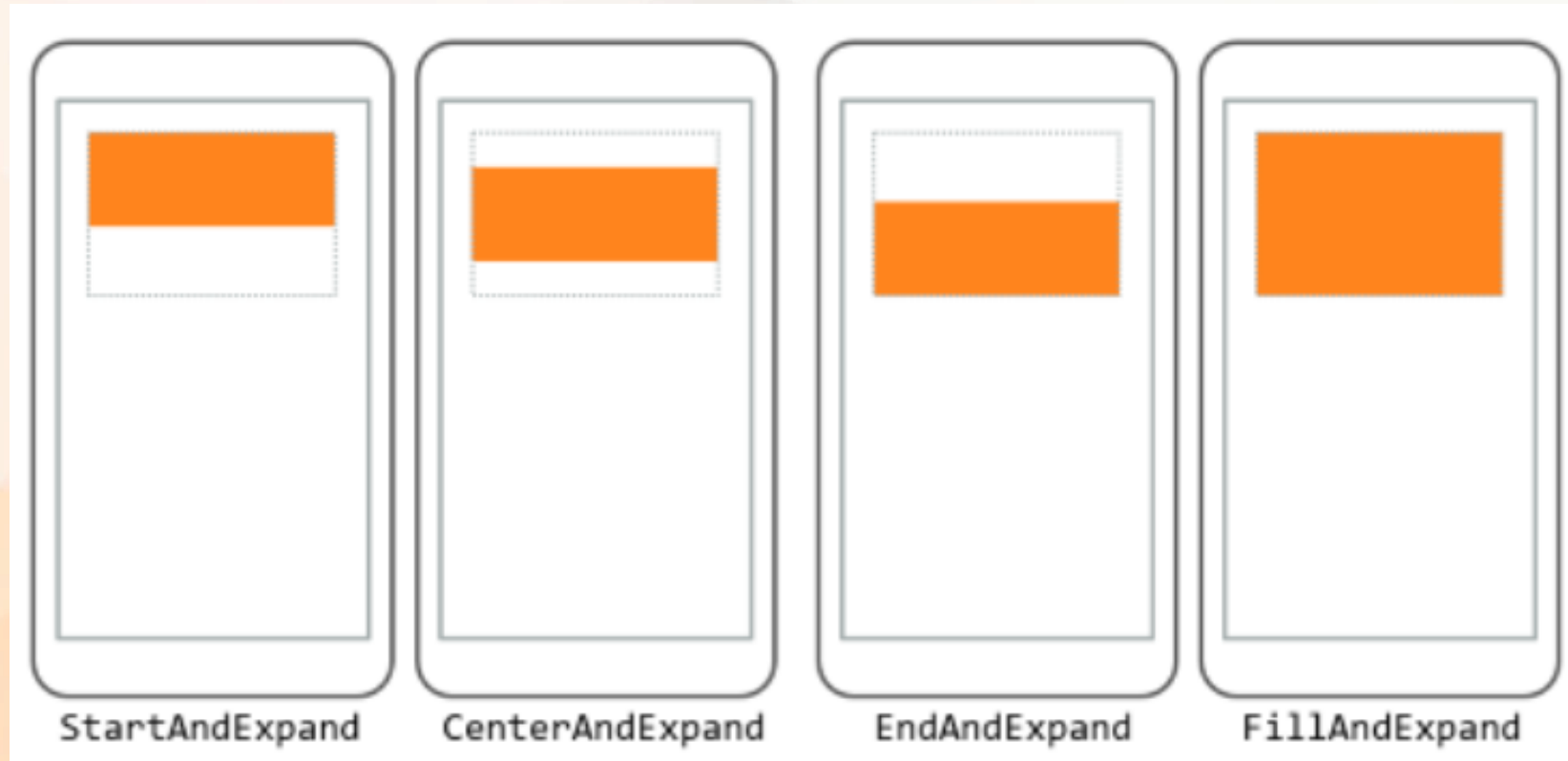
Central and East Africa

Alineación

```
<StackLayout Margin="0,20,0,0">  
    ...  
    <Label Text="Start" BackgroundColor="Gray" HorizontalOptions="Start" />  
    <Label Text="Center" BackgroundColor="Gray" HorizontalOptions="Center" />  
    <Label Text="End" BackgroundColor="Gray" HorizontalOptions="End" />  
    <Label Text="Fill" BackgroundColor="Gray" HorizontalOptions="Fill" />  
</StackLayout>
```



Expandir



Para los StackLayouts va a ser mas común usar FillAndExpand para que expandan lo máximo que se pueda.
Y StartAndExpand, CenterAndExpand y EndAndExpand son más utilizados para los controles que van a estar como hijos del StackLayouts.

Links de Interés para aprender

Kym Phillpotts - <https://kymphillpotts.com/>

Rendy - <https://www.xamboy.com/>

Xamarin Guy - https://www.youtube.com/channel/UCBKLE3rLU0CJebTBJ_N4ajQ/videos

Gerald Versluis - <https://ifversluis.dev/>

Leomaris Reyes - <https://askxammy.com/>

DevCrux Technologies - <https://www.youtube.com/channel/UCVwwehJC74qBhMhwFWJkwEQ/videos>

James Montemagno - <https://montemagno.com/>

Javier Suárez - <https://javiersuarezruiz.wordpress.com/>

Angel Garcia- <http://talkwithangel.com/>

MUCHAS GRACIAS!!!

