

Introducción a Xamarin Picando Retro Games GO



Xamarinuy



XamarinUY

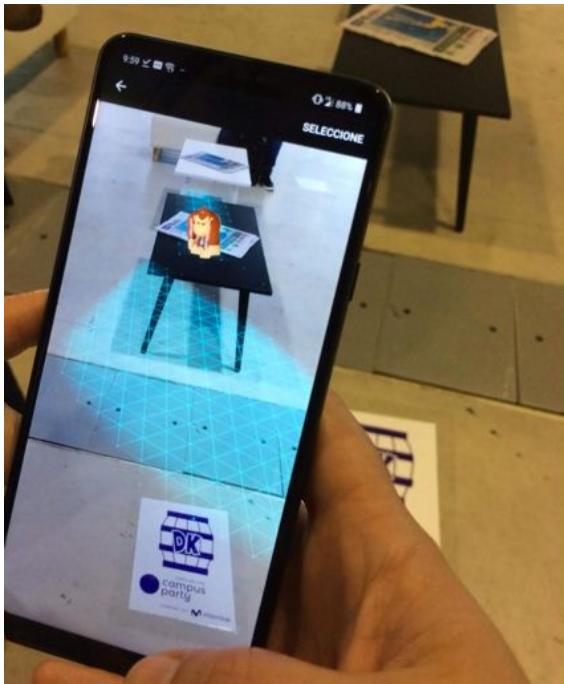


2 años de monos

2 años de monos

- meetup.com/XamarinUY
- 350+ monkey devs
- 2 años
- 9 meetups
 - temas -> +∞
 - toques, asados, networking, mucha 🎉
- github.com/xamarinuy
- @xamarinUY #XamarinUY
- Plan 2019
 - Niveles (estadísticas enero de 2019 vía meetup.com)
 - 16.7% avanzado
 - 30.5% medio
 - 52.8% principiante
 - Tutores: avanzados ayudando a medios y medios ayudando a principiantes

Retro Games GO!



¡DESCARGATE LA APP!

RG GO!

GET IT ON
Google Play

Download on the
App Store

GO!

??? #1

Donkey Kong #2

1981

Es un juego de máquina recreativa creado por Nintendo en el año 1981. Es un primitivo juego del género plataformas que se centra en controlar al personaje sobre una serie de plataformas mientras

CAPTURAR

VER PERSONAJE 3D

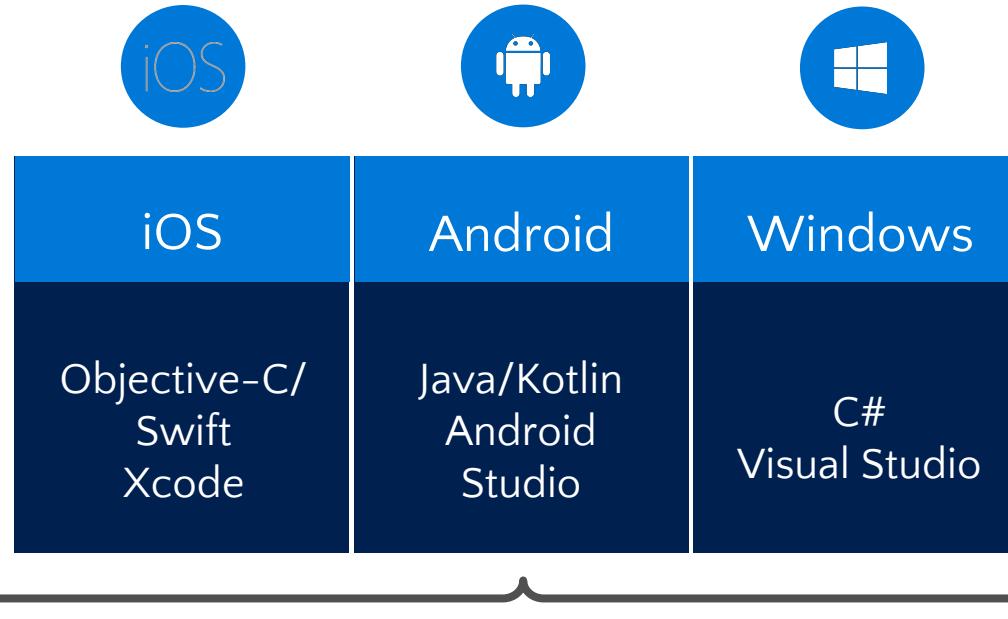
Retro Games GO!

- #retroGamesGO
- <https://apple.co/2HAJp3I>
- <http://bit.ly/2THspzs>
- <https://github.com/XamarinUY/retro-games-go>
- en 35 días
- 169 commits y contando...



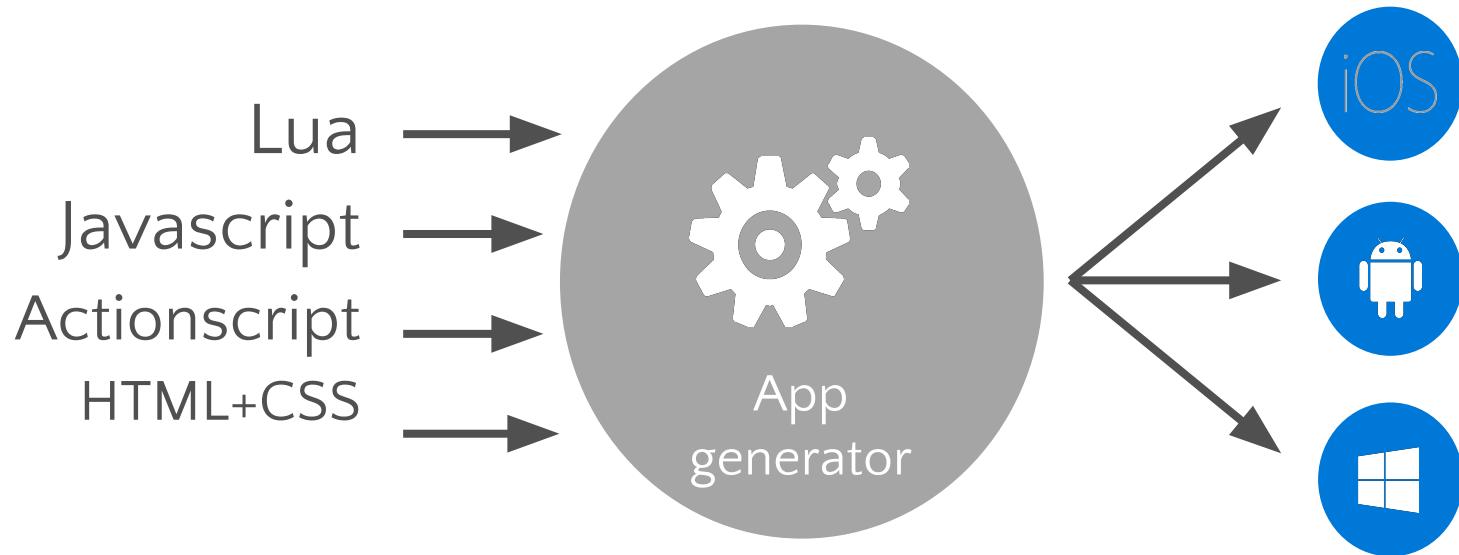
Introducción a Xamarin

Silo approach



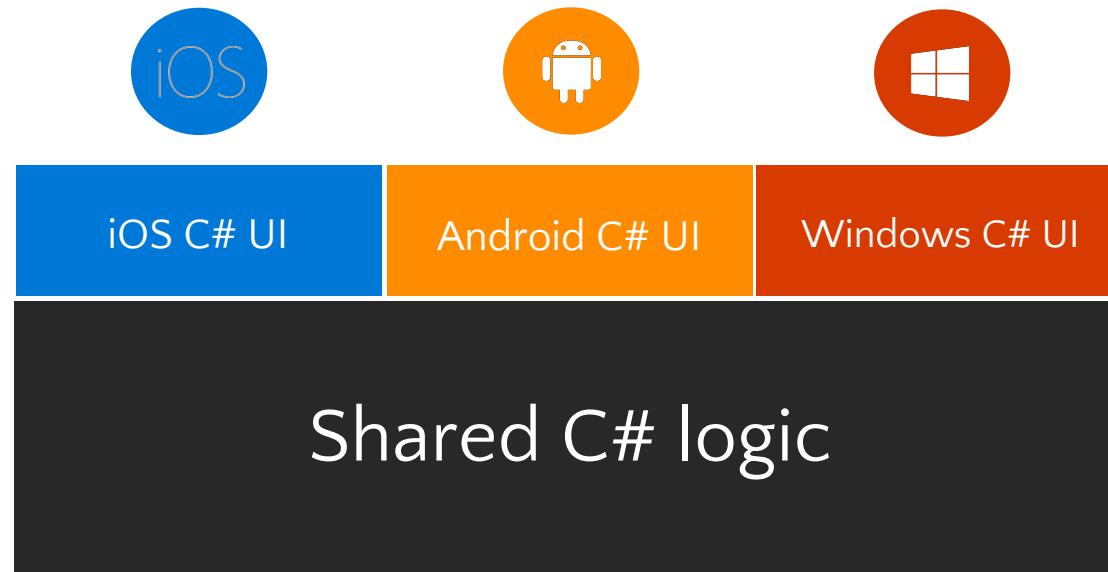
No shared code · Many languages and development environments · Multiple teams

Write once, run anywhere



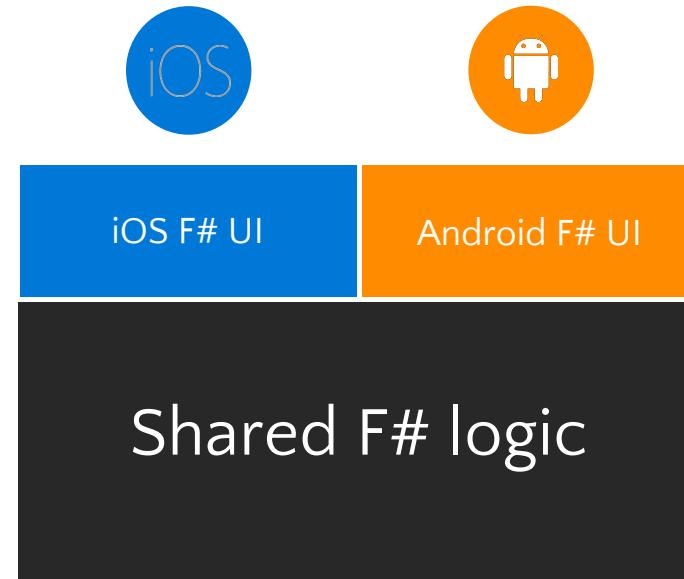
Limited native API access · Slow performance · Poor user experience

Xamarin's unique approach

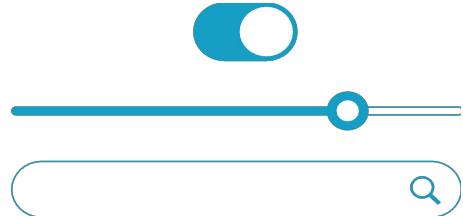


Shared C# codebase • 100% native API access • High performance

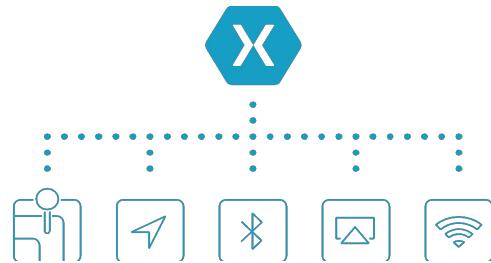
Xamarin's unique approach



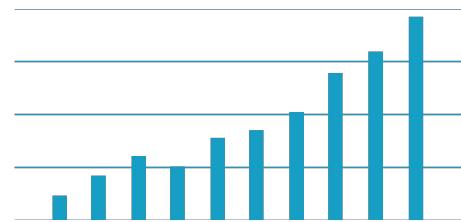
Shared F# codebase • 100% native API access • High performance



Native User Interfaces



Native API Access



Native Performance

iOS—100% API coverage

MapKit	UIKit	iBeacon	CoreGraphics	CoreMotion
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel



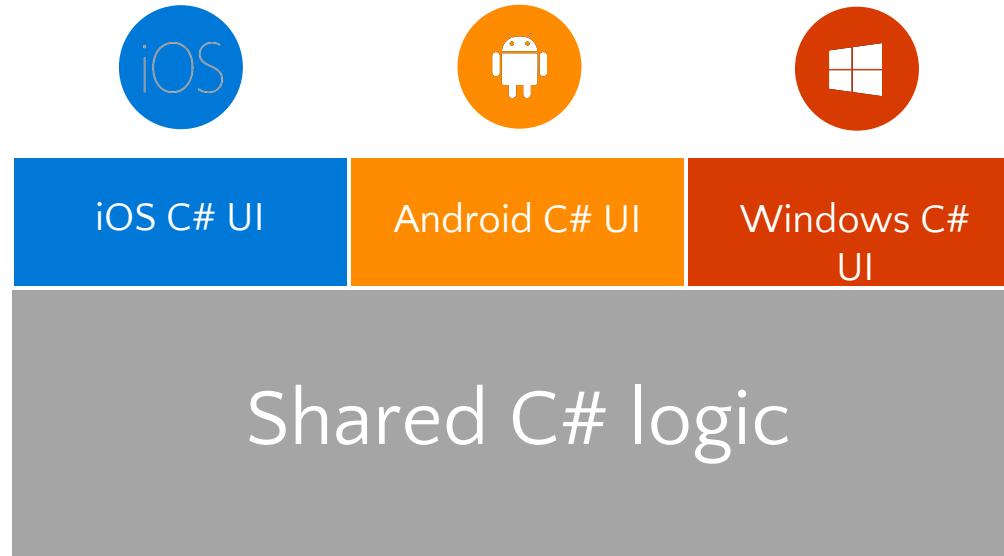
C#

Android—100% API coverage

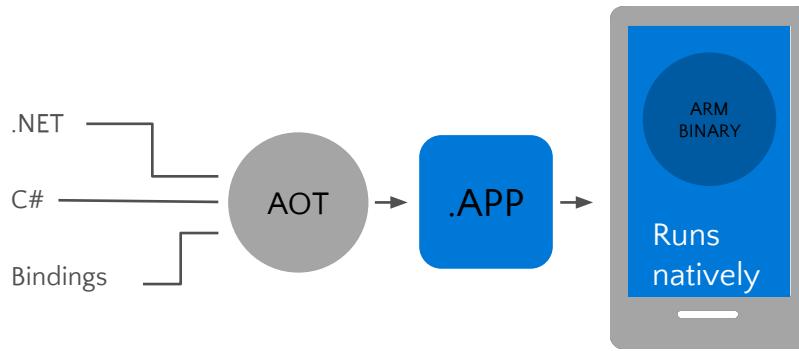
Text-to-speech	ActionBar	Printing Framework	Renderscript	NFC
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

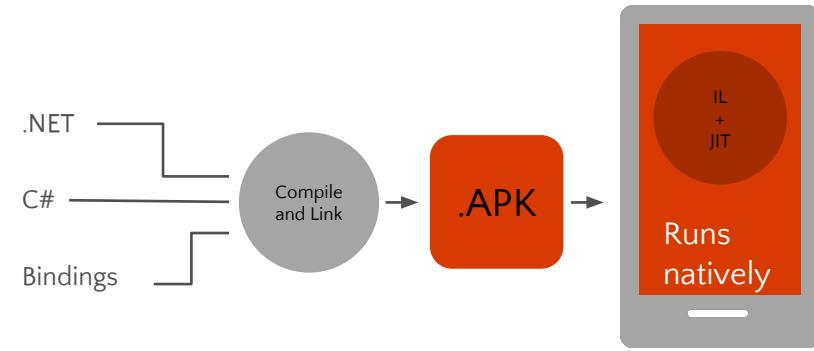
Xamarin app



Native performance



Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.



Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

✓ Always up-to-date

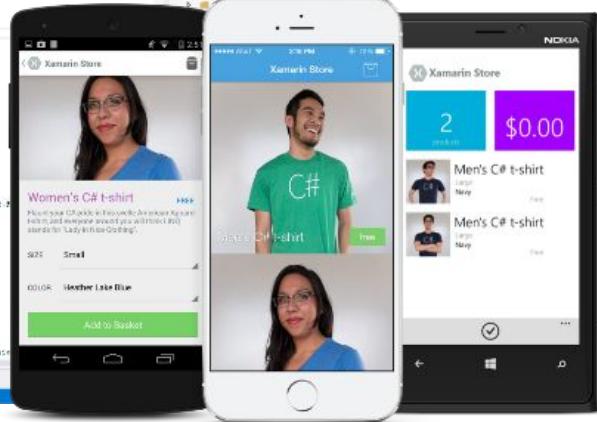
Same-day support:

- iOS 5
- iOS 6
- iOS 7
- iOS 8
- iOS 9
- iOS 10
- iOS 11

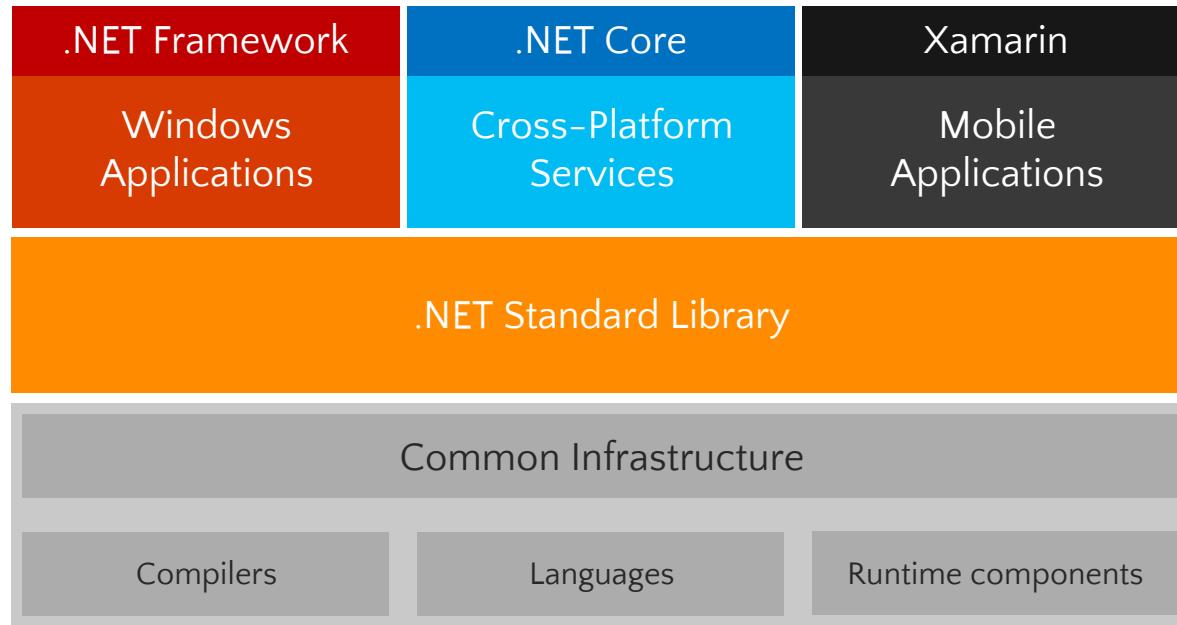
Full support for:

- Apple Watch
- Apple TV
- Android Wear
- Amazon Fire TV
- Google Glass
- and much more

Anything you can do in Objective-C, Swift, or Java can be done in C# with Xamarin



.NET Standard



.NET goes to every single client platform



Android
C#



macOS C#

iOS

iOS C#

tvOS

tvOS
C#



Linux C#



Windows
C#



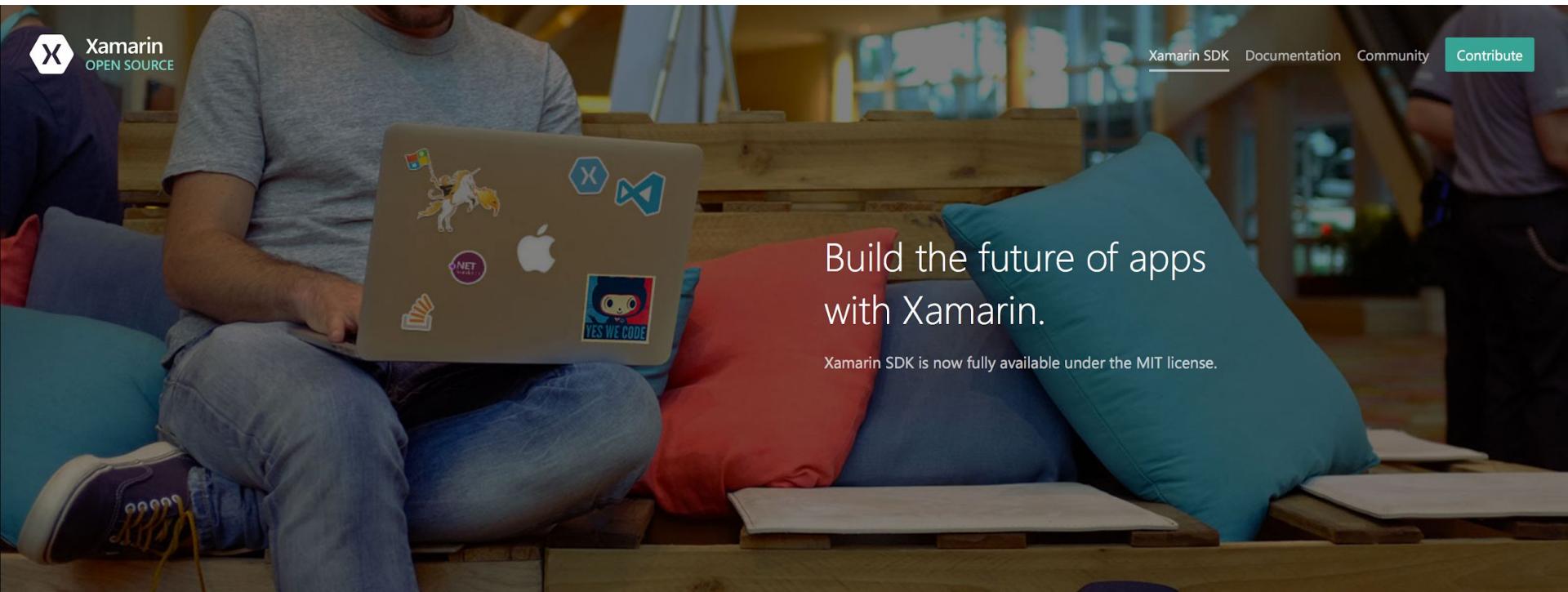
Web

Shared C# Logic

Y más

- Visual Studio
- App Center
 - Continuously build
 - Test
 - Release
 - Monitor
- Xamarin Docs
- Xamarin University

Open Source—open.xamarin.com



Xamarin
OPEN SOURCE

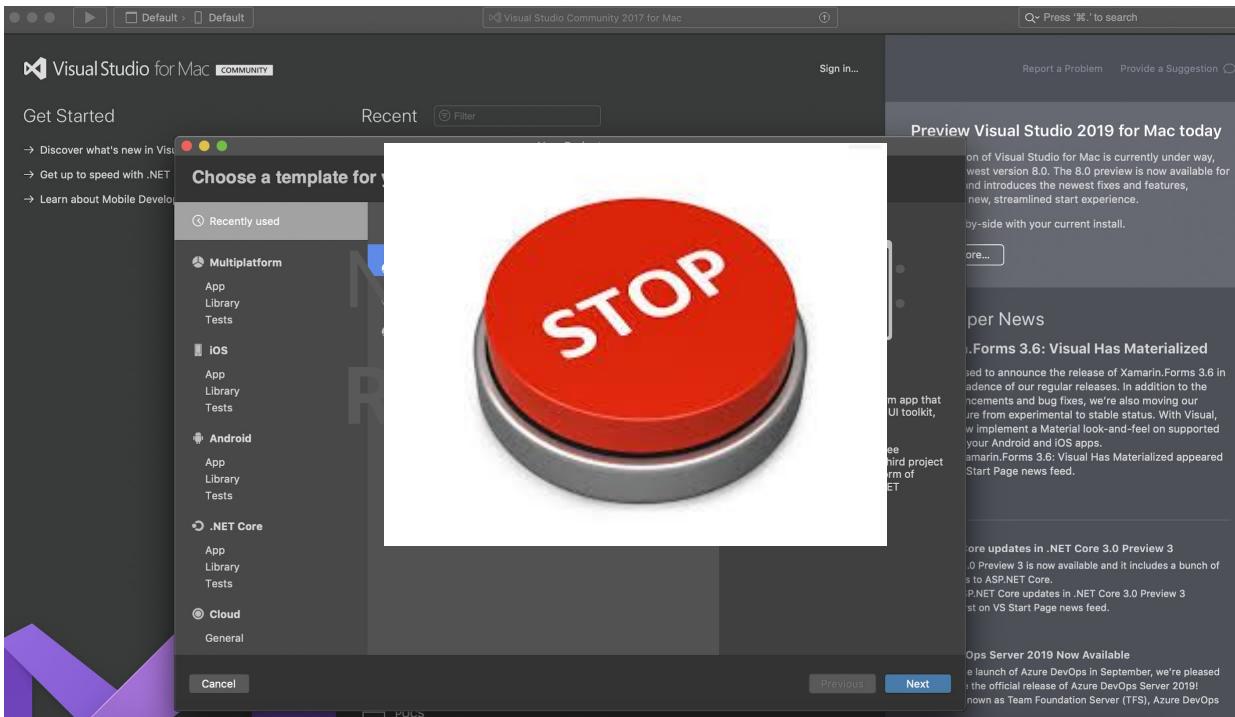
[Xamarin SDK](#) [Documentation](#) [Community](#) [Contribute](#)

Build the future of apps
with Xamarin.

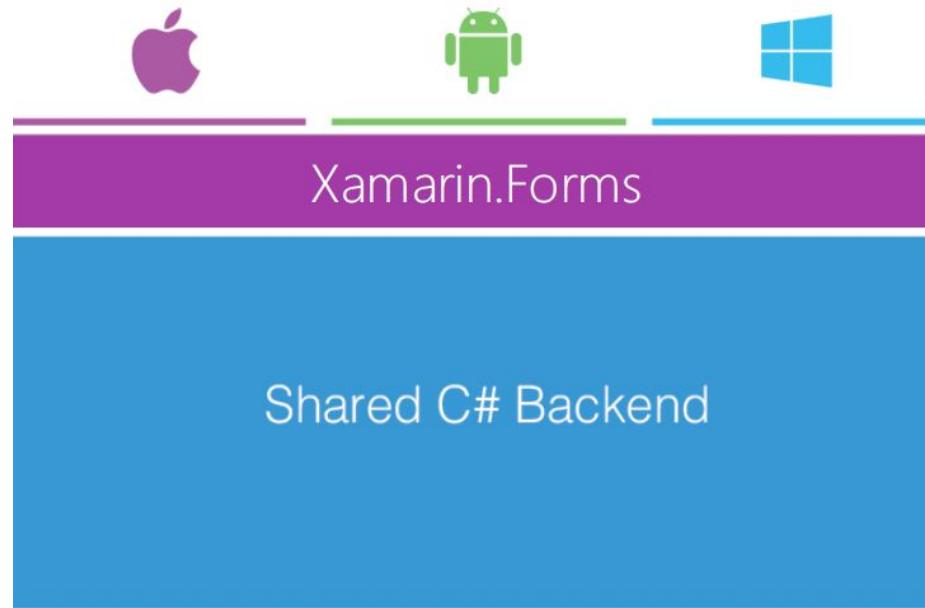
Xamarin SDK is now fully available under the MIT license.

Introducción a Xamarin Forms

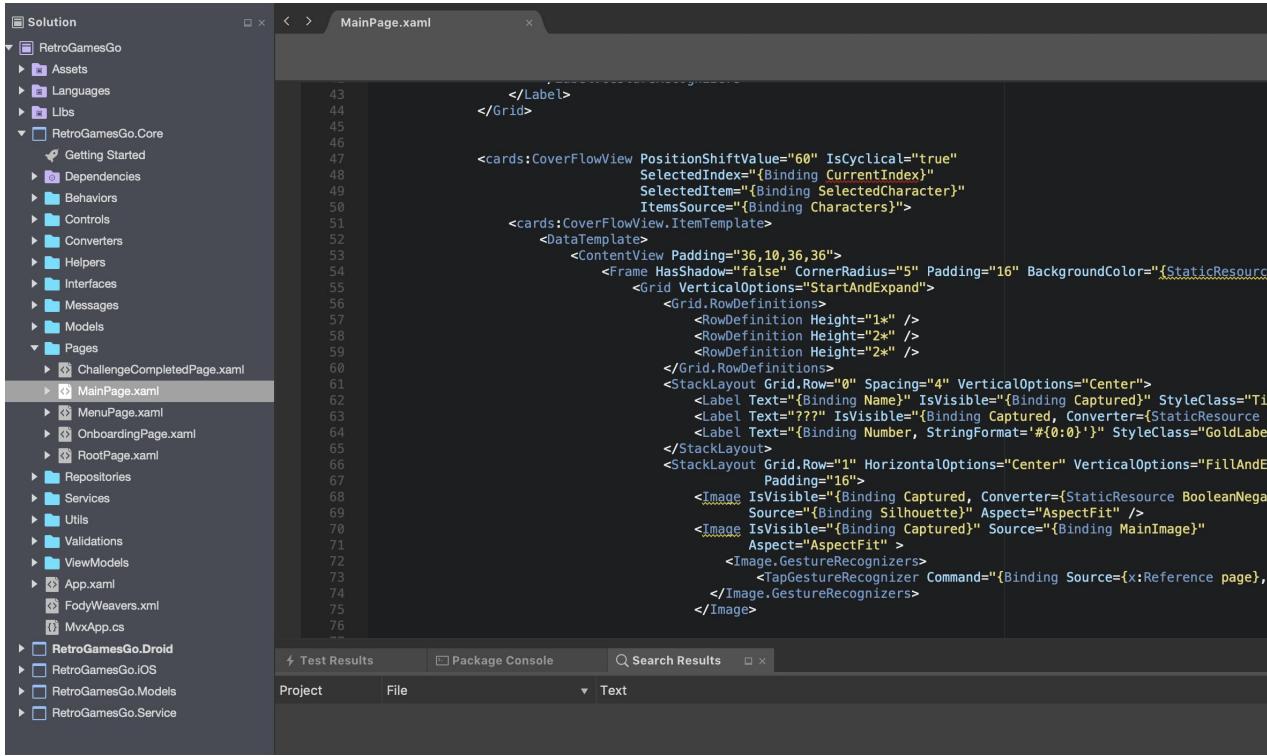
Creamos Proyecto Xamarin Forms



Xamarin Forms app



Xamarin Forms



The screenshot shows a Visual Studio interface with a Xamarin Forms project. The Solution Explorer on the left lists various files and folders, including Assets, Languages, Libs, RetroGamesGo.Core (which contains Getting Started, Dependencies, Behaviors, Controls, Converters, Helpers, Interfaces, Messages, Models, and Pages), and other platform-specific projects like RetroGamesGo.Droid, RetroGamesGo.iOS, RetroGamesGo.Models, and RetroGamesGo.Service. The main window displays the XAML code for MainPage.xaml, which includes a CoverFlowView control with ItemTemplate and DataTemplate sections. The code uses Binding and Converter logic to manage item visibility and styling. At the bottom, there are tabs for Test Results, Package Console, and Search Results, along with a navigation bar for Project, File, and Text.

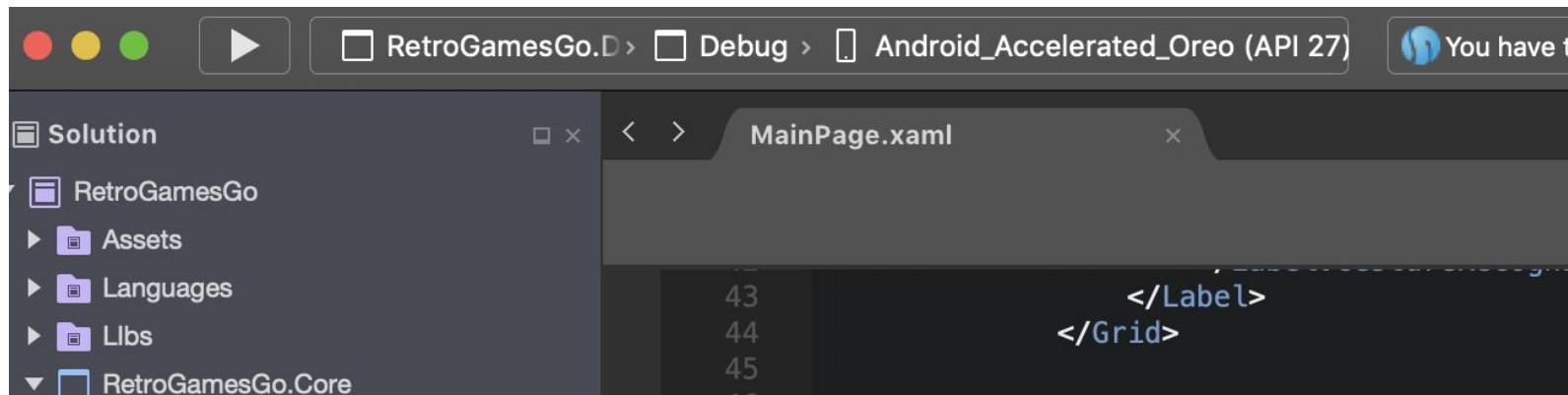
```
</Label>
</Grid>

<cards:CoverflowView PositionShiftValue="60" IsCyclical="true"
    SelectedIndex="{Binding CurrentIndex}"
    SelectedItem="{Binding SelectedCharacter}"
    ItemsSource="{Binding Characters}">
    <cards:CoverflowView.ItemTemplate>
        <DataTemplate>
            <ContentView Padding="36,10,36,36">
                <Frame HasShadow="false" CornerRadius="5" Padding="16" BackgroundColor="{StaticResource ColorBackground}" Grid VerticalOptions="StartAndExpand">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="1*" />
                        <RowDefinition Height="2*" />
                        <RowDefinition Height="2*" />
                    </Grid.RowDefinitions>
                    <StackLayout Grid.Row="0" Spacing="4" VerticalOptions="Center">
                        <Label Text="{Binding Name}" IsVisible="{Binding Captured}" StyleClass="TitleLabel" />
                        <Label Text="??" IsVisible="{Binding Captured, Converter={StaticResource BooleanNegationConverter}}"/>
                        <Label Text="{Binding Number, StringFormat="#{0:0}"}" StyleClass="GoldLabel" />
                    </StackLayout>
                    <StackLayout Grid.Row="1" HorizontalOptions="Center" VerticalOptions="FillAndExpand" Padding="16">
                        <Image IsVisible="{Binding Captured, Converter={StaticResource BooleanNegationConverter}}" Source="{Binding Silhouette}" Aspect="AspectFit" />
                        <Image IsVisible="{Binding Captured}" Source="{Binding MainImage}" Aspect="AspectFit" />
                        <Image.GestureRecognizers>
                            <TapGestureRecognizer Command="{Binding Source={x:Reference page}, Path=TapCommand}" />
                        </Image.GestureRecognizers>
                    </StackLayout>
                </Grid>
            </ContentView>
        </DataTemplate>
    </cards:CoverflowView.ItemTemplate>
</cards:CoverflowView>
```

Xamarin Forms - Ventajas

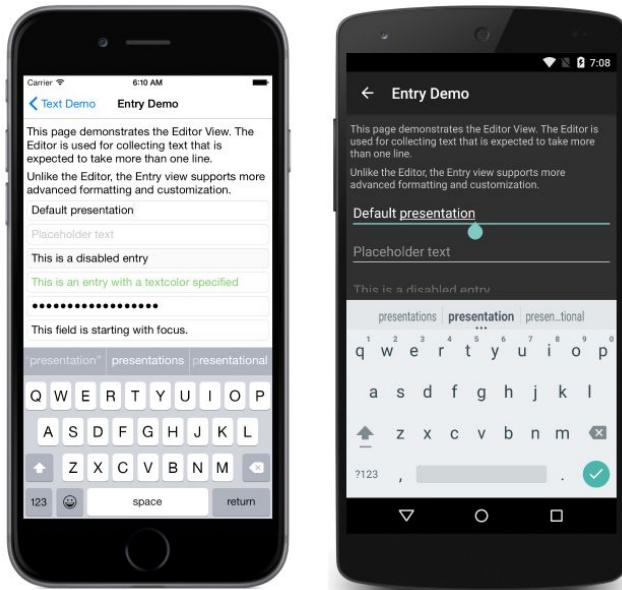
- Reutilización de vistas (xaml) - 80%
- Look and feel nativo
- 100% de acceso APIs nativas (camara, gps, nfc)
- Cada vez más librerías de la comunidad
- Mejor time to market

Xamarin Forms - Deploy de App

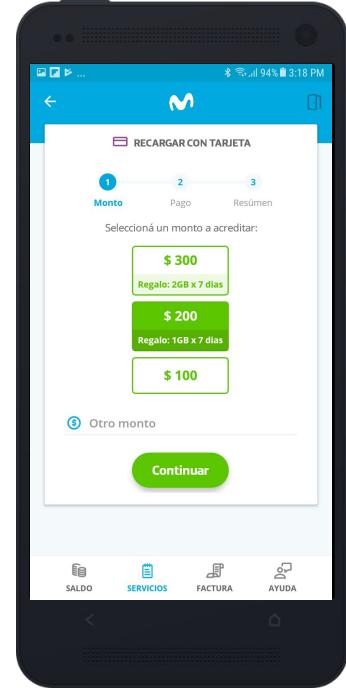
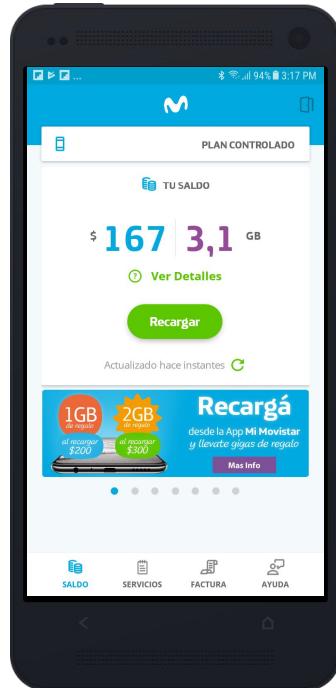


Diseño de la App

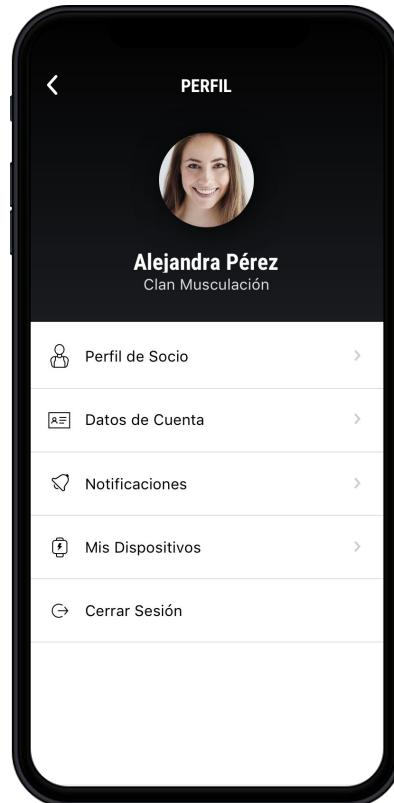
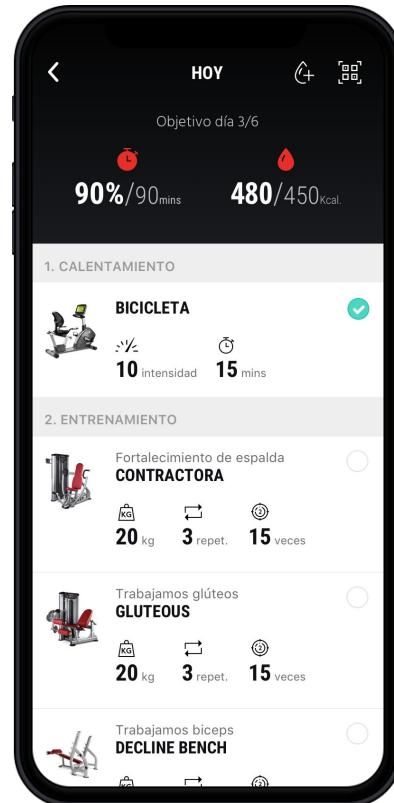
La creatividad es el límite



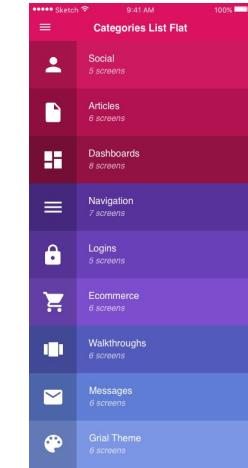
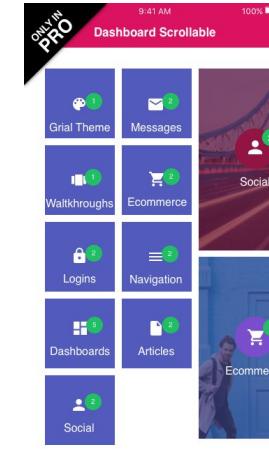
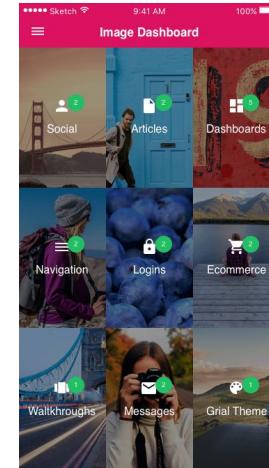
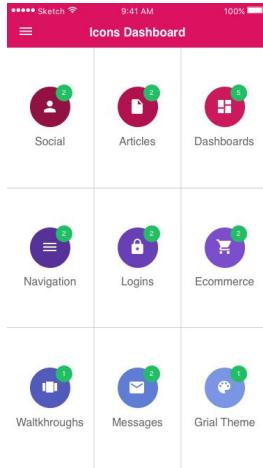
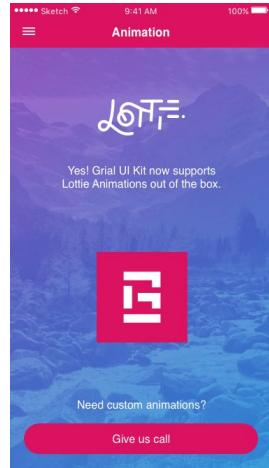
La creatividad es el límite



La creatividad es el límite



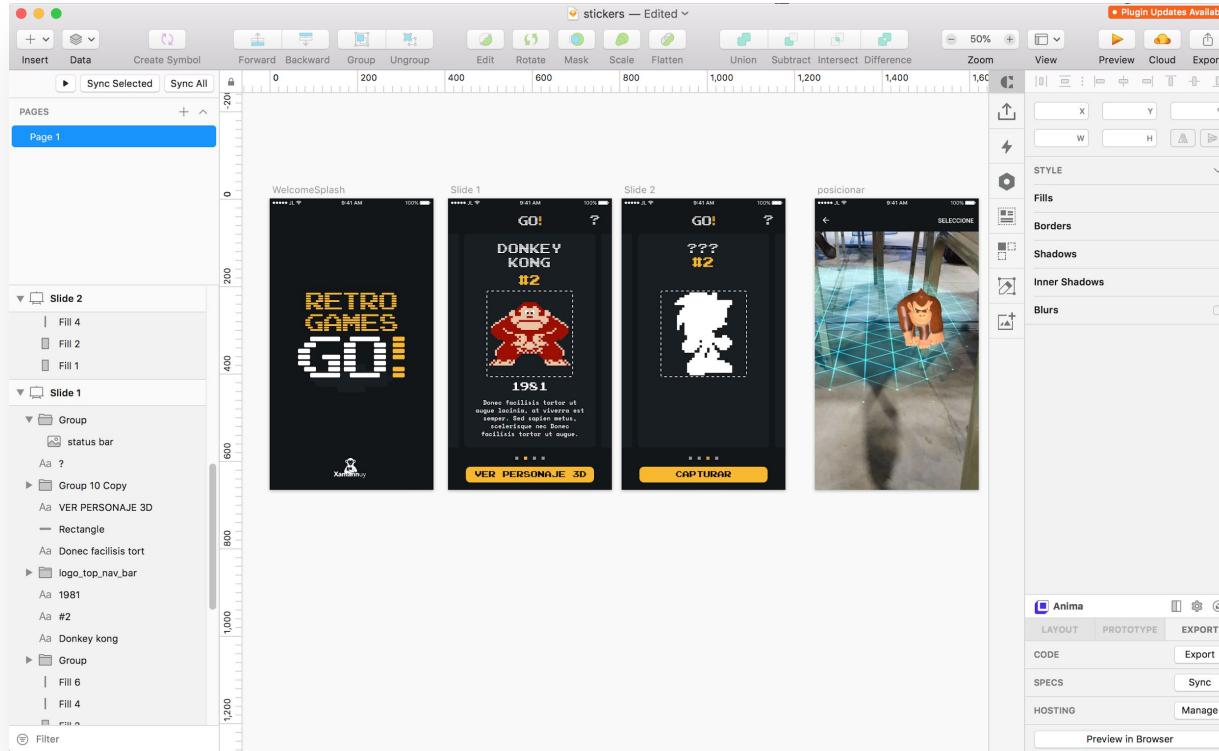
La creatividad es el límite



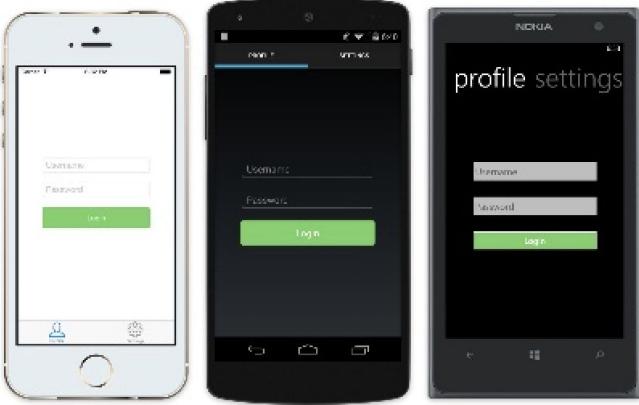
<https://grialkit.com>

La creatividad es el límite



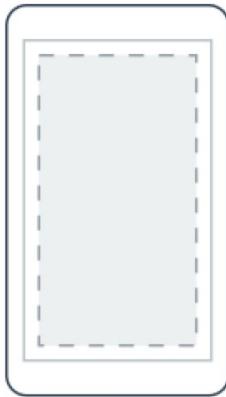


XAML eXtensible Application Markup Language



```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MyApp.MainPage">
    <TabbedPage.Children>
        <ContentPage Title="Profile" Icon="Profile.png">
            <StackLayout Spacing="20" Padding="20"
                         VerticalOptions="Center">
                <Entry Placeholder="Username"
                      Text="{Binding Username}"/>
                <Entry Placeholder="Password"
                      Text="{Binding Password}"
                      IsPassword="true"/>
                <Button Text="Login" TextColor="White"
                       BackgroundColor="#77D0E5"
                       Command="{Binding LoginCommand}"/>
            </StackLayout>
        </ContentPage>
        <ContentPage Title="Settings" Icon="Settings.png">
            <!-- Settings -->
        </ContentPage>
    </TabbedPage.Children>
```

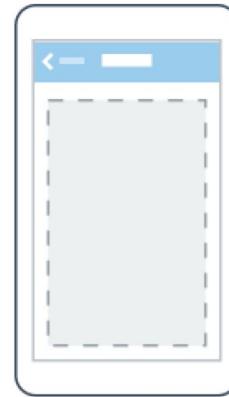
Pages



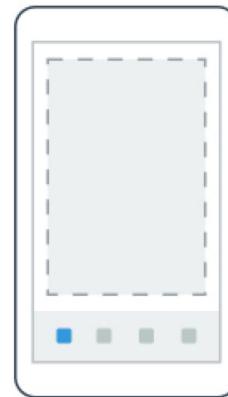
ContentPage



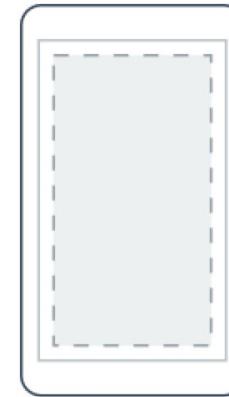
MasterDetailPage



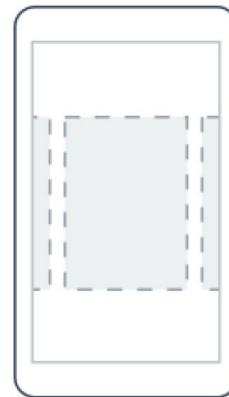
NavigationPage



TabbedPage

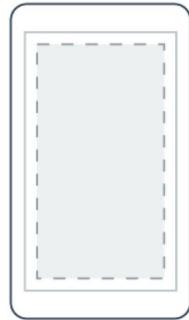


TemplatedPage



CarouselPage

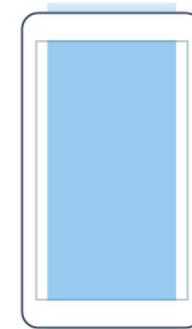
Layouts



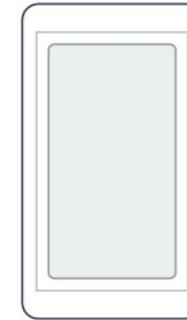
ContentPresenter



ContentView



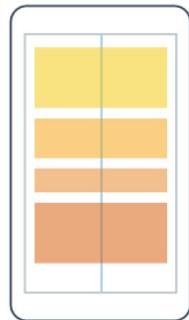
ScrollView



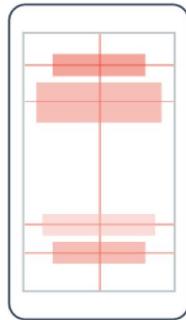
Frame



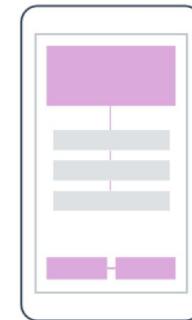
TemplatedView



StackLayout



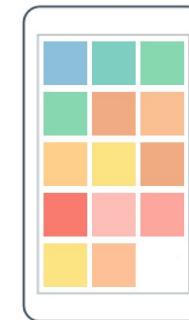
AbsoluteLayout



RelativeLayout



Grid



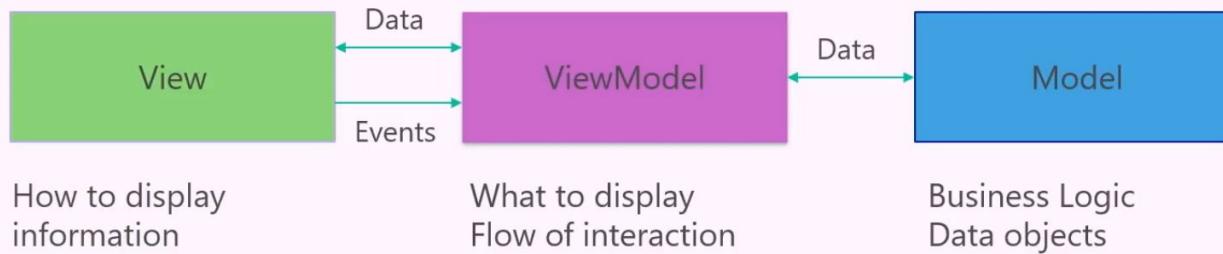
FlexLayout

Controls

ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	

MVVM & Bindings

MVVM Design pattern



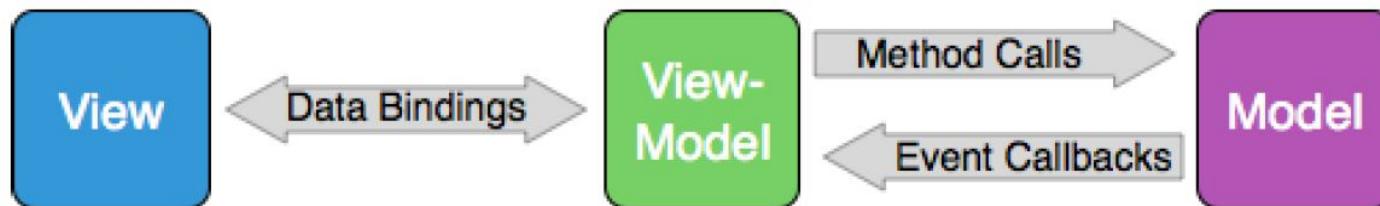
mvvm

benefits

- Isolate business logic
- Unit Test
- The app UI can be redesigned

Data Bindings

On Xamarin.Forms => Data bindings:



Data Bindings

Data-binding source (viewModel)

```
using MySeries.Model;
using System.Collections.Generic;
using System.ComponentModel;

namespace MySeries.ViewModel
{
    public class SeriesViewModel : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;

        protected void OnPropertyChanged(string propertyName)
        {
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
        }

        private List<Series> _series;
        public List<Series> Series
        {
            get { return _series; }
            set
            {
                _series = value;
                OnPropertyChanged("Series");
            }
        }

        public SeriesViewModel()
        {
            LoadSeries();
        }
    }
}
```

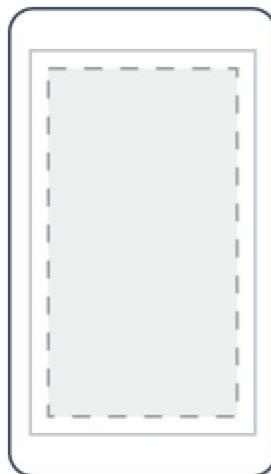
Data-binding target (xaml)

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="MySeries.View.MySeriesPage"
              Title="My Series">

    <StackLayout>
        <ListView ItemsSource="{Binding Series}">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <TextCell Text="{Binding Name}" Detail="{Binding Description}" />
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>
</ContentPage>
```

Navigation

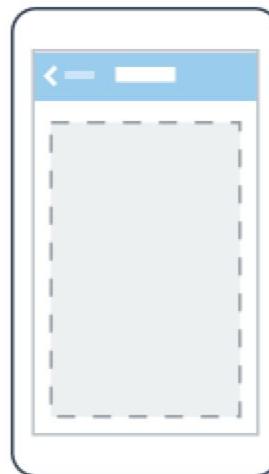
Tipos de navegación



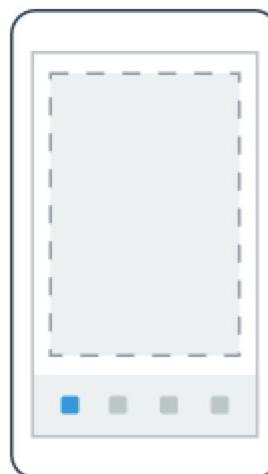
ContentPage



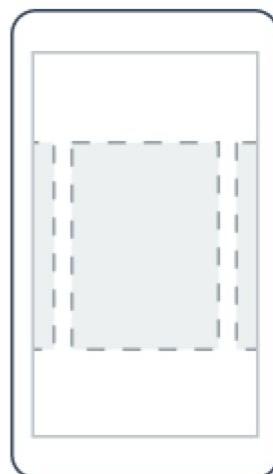
MasterDetailPage



NavigationPage



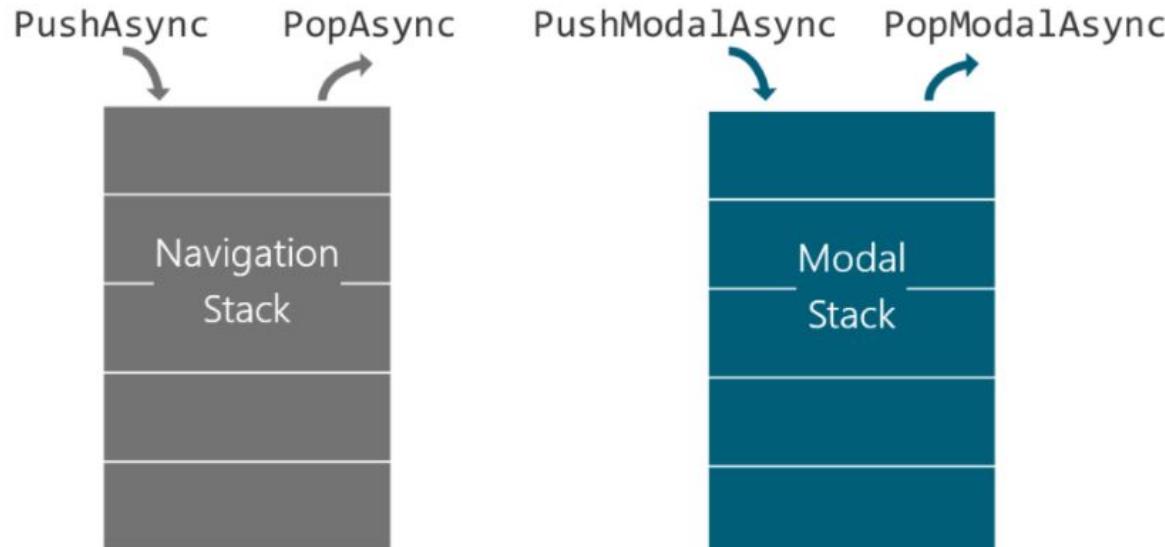
TabbedPage



CarouselPage

NavigationPage

- Stack de páginas
- LIFO (Last In - First Out)
- Customizable



¿Cómo navego?

Para navegar hacia una nueva página debemos hacer **Push**.



Para volver a la página anterior simplemente hacemos **Pop**.



lets . code (it) ;

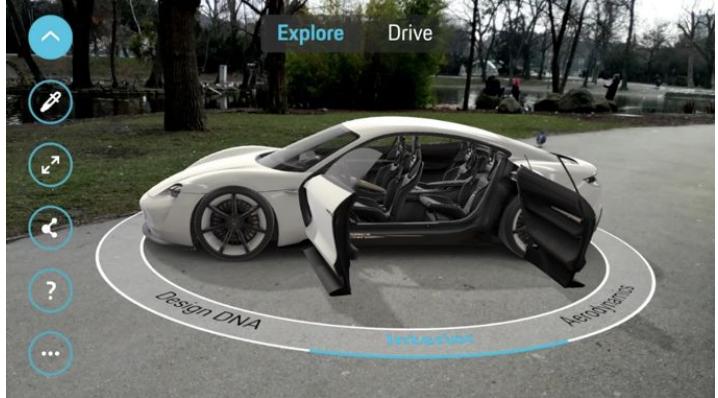
AR en Xamarin

¿Qué es realidad aumentada?

Una **experiencia interactiva** en donde elementos del mundo real se complementan con información digital o elementos virtuales a través de un dispositivo tecnológico.

Busca incorporar elementos digitales a la **percepción** que tiene la persona del mundo real.

Ejemplos



AR en la industria



Soporte nativo



ARKit



ARCore

¿Qué ofrecen?

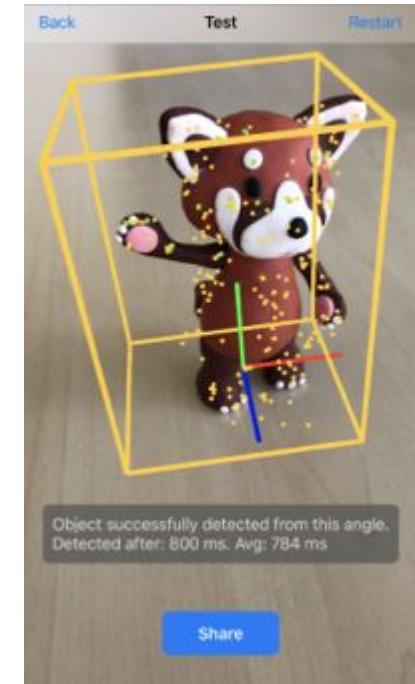
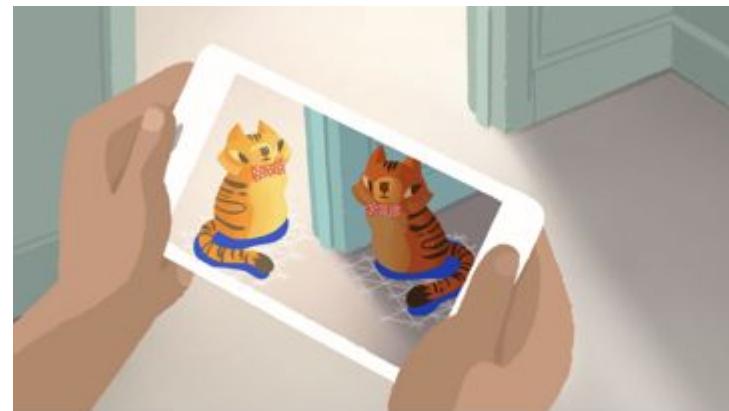
Conceptos:

- Tracking
- Scene understanding
- Rendering

Objetos:

- Session
- Plane
- Anchor

¿Qué puedo hacer?



Android

ARCore

- Android +7.0
- Últimos modelos de dispositivos.

[Dispositivos soportados](#)

iOS

ARKit

- ARKit 1.5 -> iOS 11
- ARKit 2 -> iOS 12
- + iPhone 6

La triste realidad...

- Hay pobre documentación.
- Herramientas básicas (ARCore).
- Los demos de referencia son básicos.

¿Por qué utilizar Xamarin?

ARKit

- Xamarin mapea ambos SDK en C#.
- Compartir el código y lógica de negocios.
- Desarrollar una única UI (Forms).

Desafío

Retro Games GO!



Xamarinuy

