



Curso de formación

Xamarin & Cross Platform App Development

Tragsa

27/02/2017



Sobre Bravent

Somos una Consultora IT
especializada en tecnologías
Microsoft.

Formada por un equipo de
profesionales de la tecnología y
que combina experiencia e
ilusión para ponerlo al servicio
de tu organización y, por
extensión, de tus usuarios.



Partners oficiales de Microsoft y Xamarin

¿Quiénes somos?

El valor añadido de **Bravent** se centra en su equipo de profesionales. Contamos con especialistas proactivos que trabajan de manera coordinada y eficaz para ofrecer resultados óptimos en cada una de las fases del proyecto.



Mario López Baratas

Mobile Technical Leader



@mariolobar



/mario-lopez-baratas



mlopez@bravent.net



Sobre Xamarin

Xamarin, la herramienta para desarrolladores de apps móviles, recientemente adquirida por Microsoft, ya está siendo utilizada por más de 15.000 empresas de todos los sectores del mercado para desarrollar increíbles aplicaciones móviles disminuyendo costes y tiempos de desarrollo





¿Cómo se creó?

Timeline

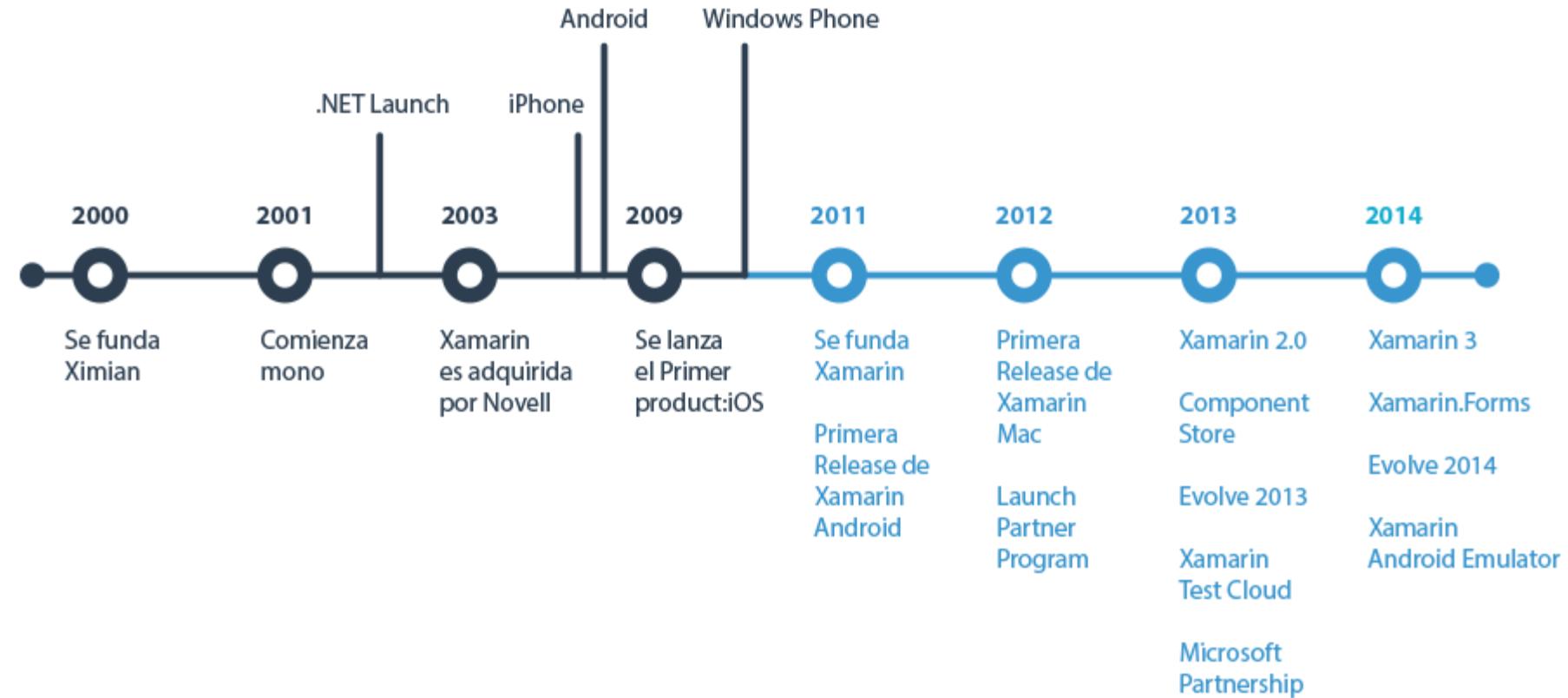
Xamarin surgió en el año 2011 de la mano de Miguel de Icaza y

Nat Friedman

Gracias al proyecto Mono pudieron ser capaces de traducir .Net a iOS y Android

Actualmente cuentan con más de 1.000.000 de descargas de la plataforma Xamarin Studio.

Cómo se creó Xamarin: historia





¿Qué ofrece Xamarin?

Xamarin surgió en el año 2011 de la mano de Miguel de Icaza y Nat Friedman

Gracias al proyecto Mono pudieron ser capaces de traducir .Net a iOS y Android

Actualmente cuentan con más de 1.000.000 de descargas de la plataforma Xamarin Studio.



Xamarin Platform

Permite crear apps en C# para todas las plataformas del mercado.



Xamarin Insights

Supervisa el estado de la app, conocer a los usuarios, observar problemas que puedan surgir, entender a la audiencia y descubrir tendencias para adaptarte a la demanda generada.



Xamarin Test Cloud

Testea automáticamente tu app en cientos de dispositivos móviles, para identificar los errores en tiempo real a través de pruebas automáticas sin tener que volver a compilar.



Xamarin University

Permite mantenerte al día a través de más de 70 clases tanto en línea como de manera presencial impartidas por expertos en Xamarin.

Xamarin **gratis para estudiantes**



Ventajas de usar Xamarin

Todas las ventajas de Xamarin la convierten en una potente herramienta de desarrollo multiplataforma.

Ventajas de usar Xamarin



Producto soportado y **oficial de Microsoft**



Usa un solo lenguaje de **programación C#**



Compartir código entre plataformas (**PLC**)



Mejora el "**time to market**"



Reduce los tiempos y costes de mantenimiento



Reduce el número de bugs al tener el código compartido entre plataformas

- ✓ **Una sola base de código** para las tres plataformas mayoritarias, haciendo más sencillo el desarrollo y el posterior mantenimiento.
- ✓ **Compartimos entre un 75% y un 90% del código**, dependiendo del caso particular de cada aplicación
- ✓ Al **generar aplicaciones totalmente nativas**, el rendimiento es igual al de una aplicación para cada plataforma.
- ✓ Xamarin ya de por sí es una herramienta que permite compartir código entre plataformas.
- ✓ Xamarin Forms permite ir un paso más adelante, **compartiendo la interfaz de usuario**, pero manteniendo los controles nativos.
- ✓ Xamarin.Forms compila a nativo para cada plataforma, por lo que la experiencia final de usuario es la misma que una aplicación nativa.



**KEEP
CALM
WE USE
MVVM**

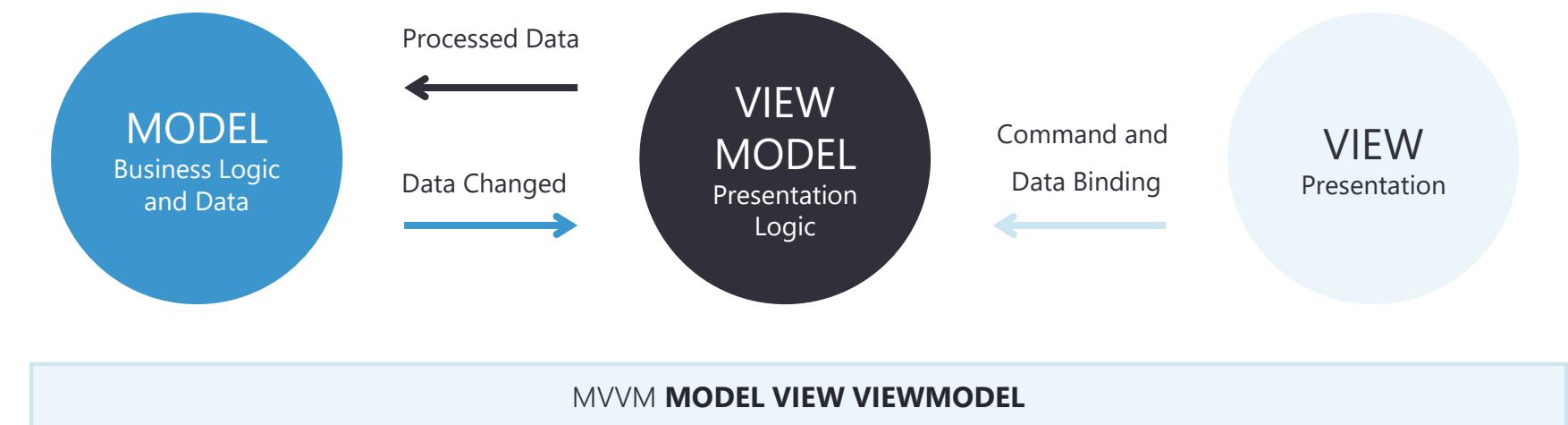


¿Qué es el patrón MVVM?

MVVM es una evolución del patrón MVC (Model/View/Controller), intenta facilitar el trabajo en paralelo entre alguien encargado de diseñar la interfaz de usuario y otra persona encargada de generar el código que la sustentará.

Partes del patrón

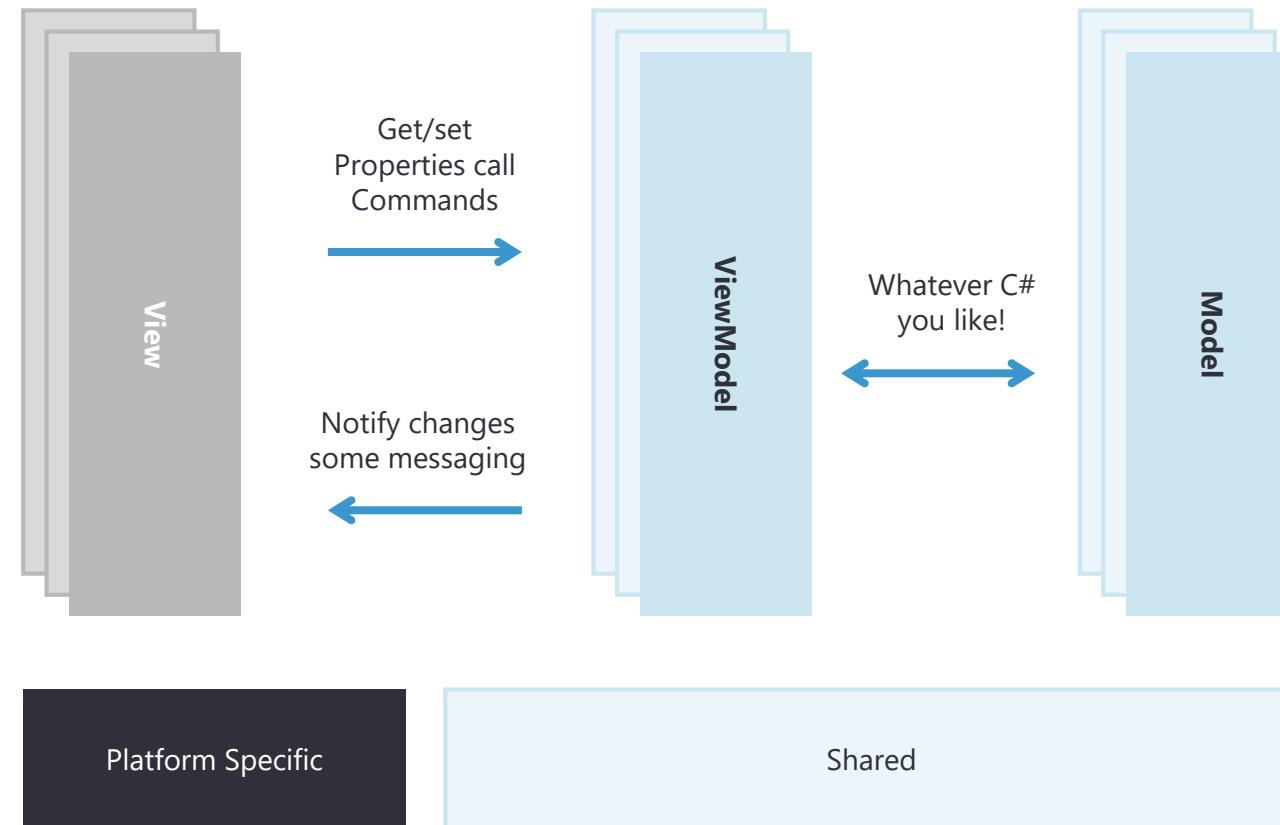
- ✓ La View se expresará en XAML, AXML o XIB (la interfaz de usuario, dependiendo de la plataforma).
- ✓ El ViewModel contendrá el lenguaje que usaremos para desarrollar la lógica de presentación (VB.NET, C# o C++).
- ✓ El Model será el encargado del acceso a datos y la lógica de negocio.





¿Qué es el patrón MVVM?

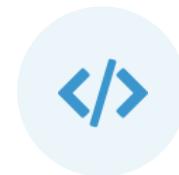
MVVM es una evolución del patrón MVC (Model/View/Controller), intenta facilitar el trabajo en paralelo entre alguien encargado de diseñar la interfaz de usuario y otra persona encargada de generar el código que la sustentará.



</>

¿Por qué MVVM?

[REDACTED]



Mayor facilidad para mantener,
extender y compartir el código



Más facilidad de colaborar



Testing



Más fácil de diseñar

```
</>
```

```
INotifyPropertyChanged
```

```
namespace System.ComponentModel
{
    /// 
    ///     Notifies clients that a property value has changed.
    public interface INotifyPropertyChanged
    {
        /// 
        ///     Occurs when a property value changes.
        event PropertyChangedEventHandler PropertyChanged;
    }
}
```



</>

Propiedad de un ViewModel

```
public class Employee: INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;
    private void RaisePropertyChanged(string propertyName)
    {
        if (PropertyChanged != null)
        {
            var eventArgs = new PropertyChangedEventArgs(propertyName);
            PropertyChanged(this, eventArgs);
        }
    }

    private string _FirstName;
    public string FirstName
    {
        get { return _FirstName; }
        set
        {
            _FirstName = value;
            RaisePropertyChanged("FirstName");
        }
    }
}
```

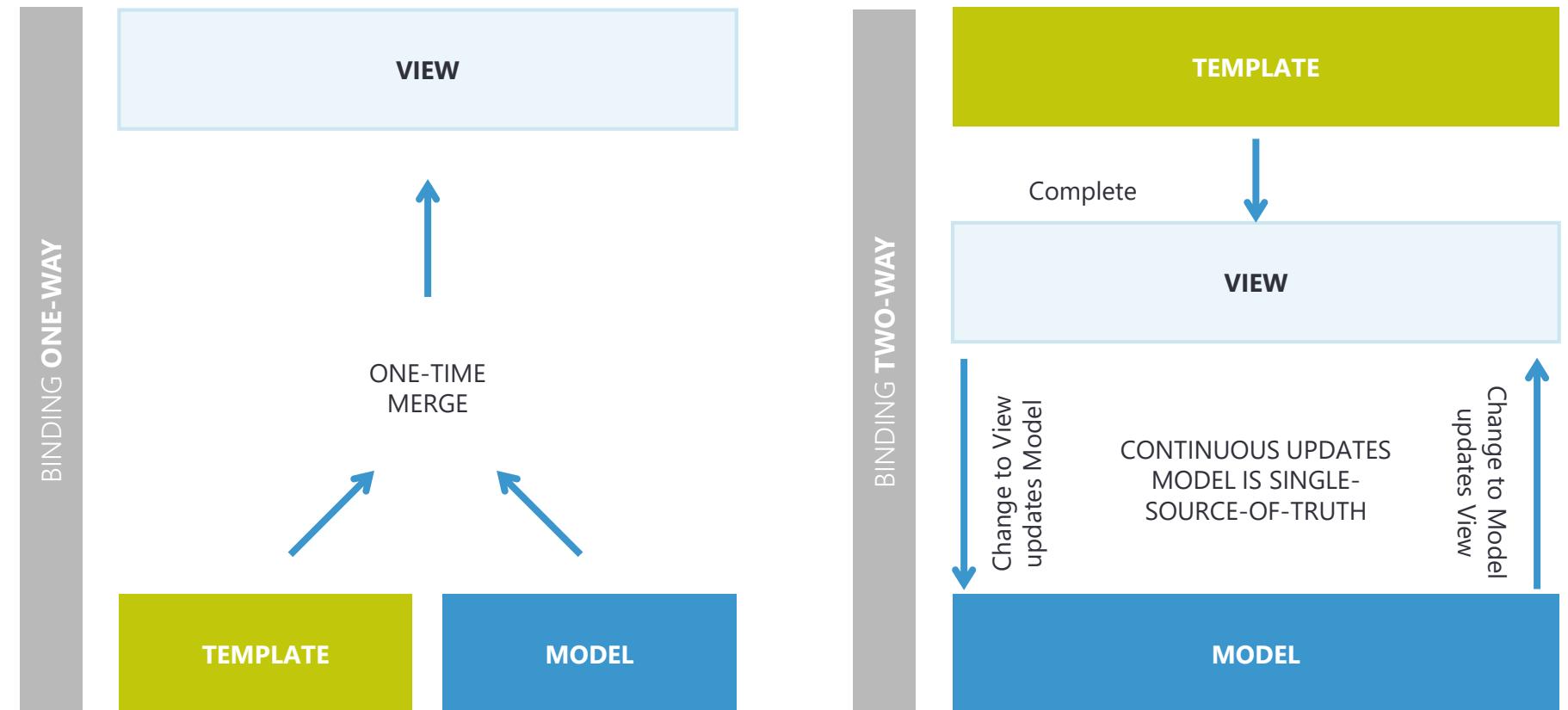


</>

Binding

El binding relaciona dos propiedades entre sí haciendo que se mantengan sincronizadas: si una cambia la otra también

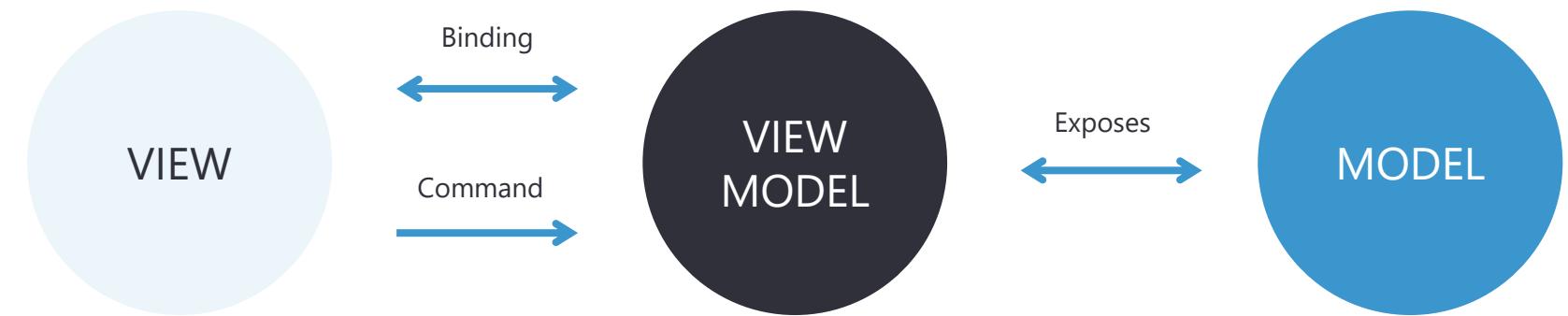
- ✓ El binding puede ser unidireccional o bidireccional. En caso de un binding bidireccional, cualquiera de los dos extremos del vínculo puede ser originador de las modificaciones. Esta característica será de gran utilidad porque provee una forma muy sencilla que la vista refleje automáticamente cambios en el modelo.



</>

Commands

Este patrón permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Para ello se encapsula la petición como un objeto, con lo que además facilita la parametrización de los métodos.



Demo MVVM

Ejemplo demo MVVM



DEMO 1

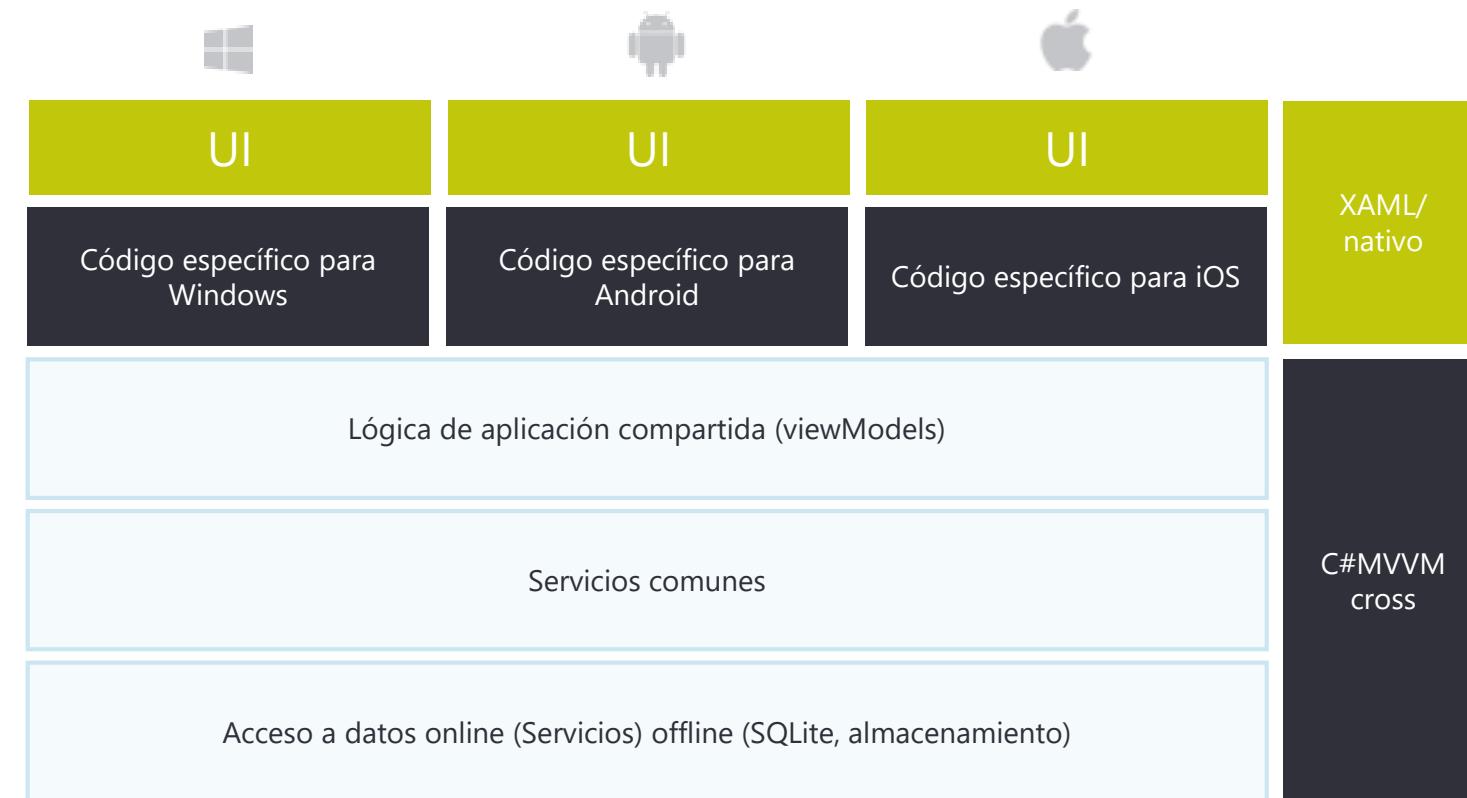
Ejemplo MVVM

A

Xamarin Classic

Cuando hablamos de **Xamarin Classic** nos estamos refiriendo fundamentalmente al desarrollo de las aplicaciones utilizando las APIs clásicas de cada plataforma

Arquitectura Xamarin Classic



M
Mvvm
M

¿Qué es MvvmCross?

MvvmCross nos permite compartir gran cantidad del código entre las distintas plataformas.

MvvmCross es un proyecto Open-Source creado por [Stuart Lodge](#), uno de los más grandes contribuidores al desarrollo multiplataforma usando Xamarin.

MvvmCross ha pasado de convertirse a un toolkit a un Framework completo para desarrollar aplicaciones móviles.

MvvmCross un viejo conocido que aguanta el tirón



Características

- ✓ Framework basado en la arquitectura MVVM.
- ✓ Dependency Injection implementado por defecto.
- ✓ Podemos modularizar la capa de negocio de la implementación gráfica de las apps.
- ✓ Framework extensible en el que nosotros podemos crear nuestros propios Plugins.



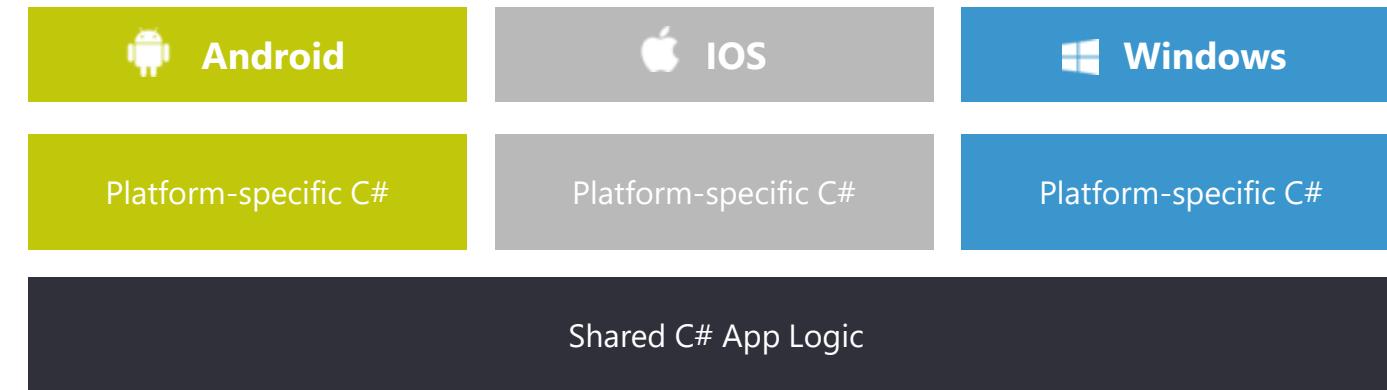
M
MvvM
M

¿Qué es MvvmCross?

MvvmCross nos permite compartir gran cantidad del código entre las distintas plataformas.

MvvmCross es un proyecto Open-Source creado por [Stuart Lodge](#), uno de los más grandes contribuidores al desarrollo multiplataforma usando Xamarin.

MvvmCross ha pasado de convertirse a un toolkit a un Framework completo para desarrollar aplicaciones móviles.



Links de ayuda

[Ninja Coder for MvvmCross and Xamarin Forms 2013](#)

[Página de GitHub](#)

[Wiki MvvmCross](#)

[Videos tutoriales](#)

[Plugins disponibles para utilizar en tus apps](#)



¿Qué es MvvmCross?

MvvmCross nos permite compartir gran cantidad del código entre las distintas plataformas.

MvvmCross es un proyecto Open-Source creado por [Stuart Lodge](#), uno de los más grandes contribuidores al desarrollo multiplataforma usando Xamarin.

MvvmCross ha pasado de convertirse a un toolkit a un Framework completo para desarrollar aplicaciones móviles.

No todo es perfecto

- ✓ Desde la salida de Xamarin.Forms ha tenido un pequeño bajón, pero aguanta.
- ✓ Carece de sobre-ingeniería y nos aleja de las implementaciones nativas.
- ✓ El desarrollo de apps lleva más tiempo que usando Xamarin.Forms.

¿Cuándo debemos usar MvvmCross?

- ✓ Cuando la app requiere cierta complejidad en diseño.
- ✓ Cuando los requerimientos de nuestra app requieren de animaciones complejas.
- ✓ Complejidad en las APIs requeridas en la app, en las que si existen soluciones para Xamarin.Android y Xamarin.iOS, pero aun no en Xamarin.Forms.

En el caso de que no cumplirse ninguna de estas tres opciones Xamarin.Forms será suficiente para desarrollar la app.



M
Mvvm
M

Estructura de un proyecto con Xamarin

Core (Lógica de
negocio de la aplicación)

Proyecto iOS

Proyecto Android

Proyecto Windows

Solution

- ▼  XamMvvmCrossMarvel (master)
 - ▶  XamMvvmCrossMarvel.Universal
 - ▶  XamMvvmCrossMarvel.Core
 - ▶  XamMvvmCrossMarvel.Droid
 - ▶  XamMvvmCrossMarvel.iOS



Demo MVVM

Ejemplo demo MVVMCross



DEMO 2

Marvel Characters MVVMCross





Sobre Xamarin 4.0



Sobre Xamarin 4.0



Xamarin Forms 2.0

- Compilación de XAML previa a la creación del paquete.
- Mejora de rendimiento en los listados.
- Soporte para Android Material Design.
- Soporte a Windows Universal Apps



Xamarin Test Recorder y Xamarin.UITest

- Compilación de XAML previa a la creación del paquete.
- Mejora de rendimiento en los listados.
- Soporte para Android Material Design.
- Soporte a Windows Universal Apps



Xamarin Insights ya no es beta

- Herramienta para grabar nuestras pruebas funcionales.
- Integrada con Xamarin Test Cloud

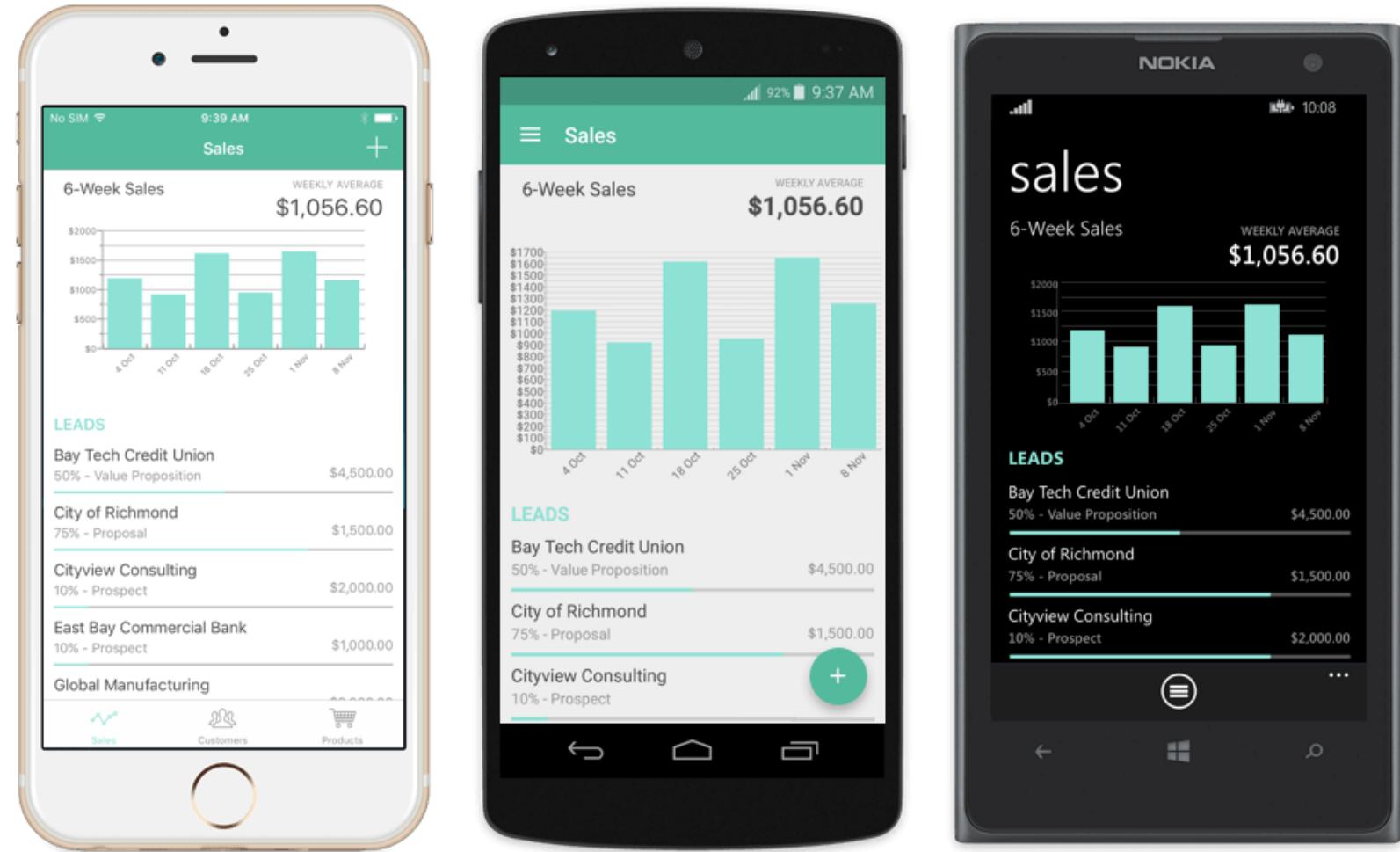


Xamarin Inspector Review

- Modificaciones de diseño de la app en tiempo real
- Ahorramos tiempo de compilación para cada cambio de diseño que queramos aplicar.



Xamarin Forms 2.0



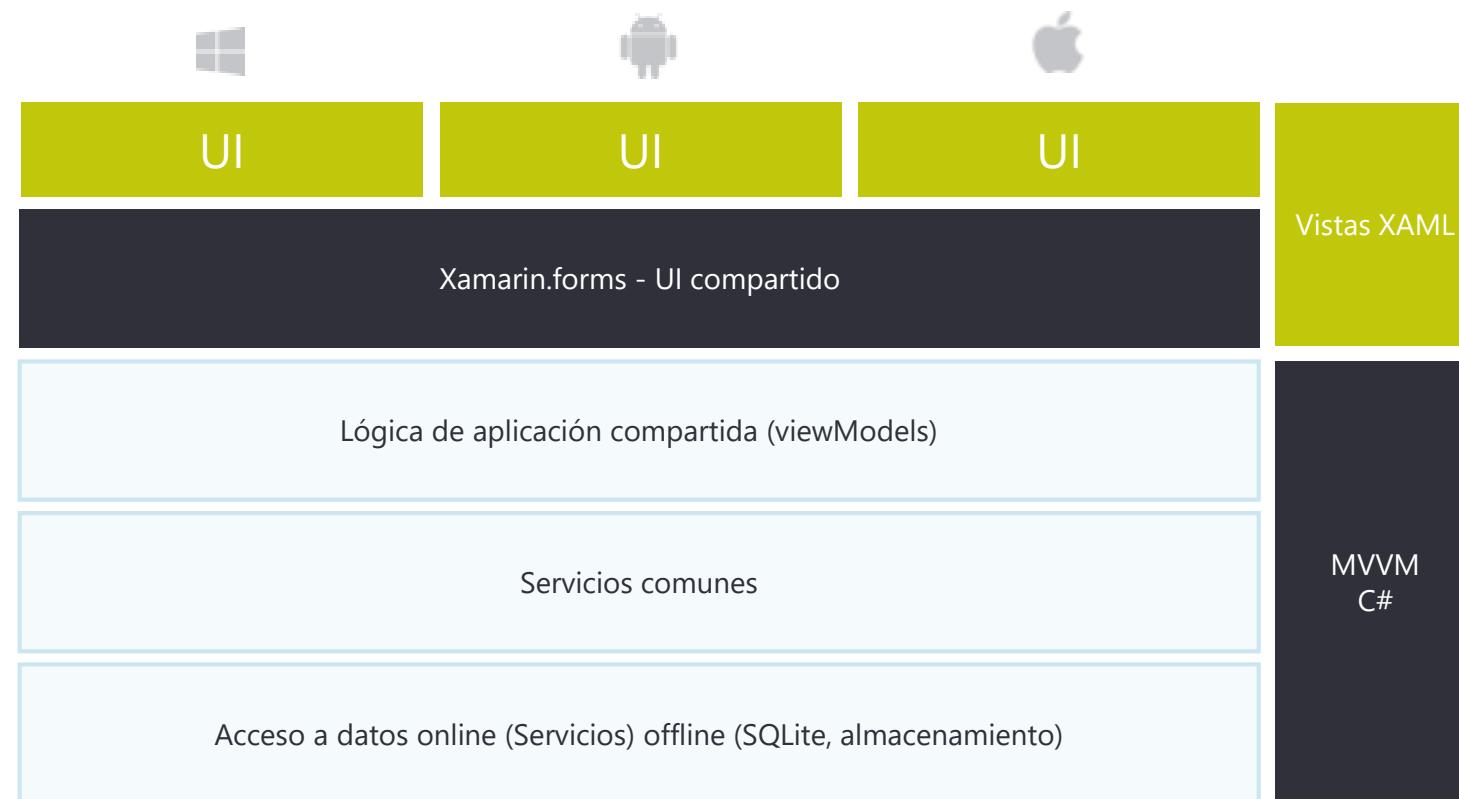


Xamarin.forms

Xamarin Forms es un conjunto de herramientas creadas principalmente con el objetivo de agilizar el desarrollo multiplataforma, centrándose principalmente en maximizar la cantidad de código compartido entre plataformas.

Un paso más en el desarrollo de aplicaciones móviles.

¿Qué es Xamarin.forms?





Estructura de un proyecto Xamarin.Forms

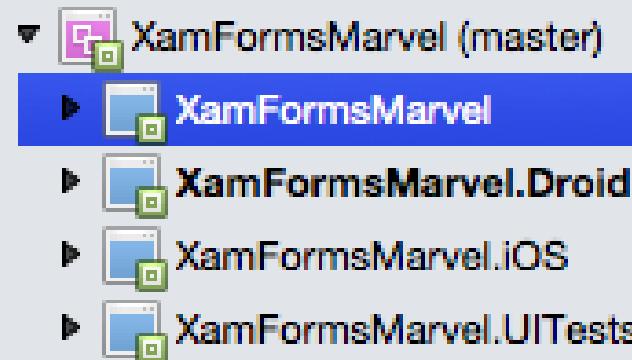
Muy similar a la de MvvmCross.

Un proyecto Core, pero en este caso no solo incluye la lógica de negocio sino también las Vistas de las app.

Un proyecto para cada plataforma.

Un proyecto de UITest.

Solution





¿Qué hacer cuando queremos implementar código específico en cada plataforma?

- ✓ Aunque la mayoría del código es compartido, van a existir situaciones en las que tendremos que implementar cierta funcionalidad.
- ✓ Tenemos dos herramientas para realizar esto:
 - DependencyService
 - CustomRenderers



DependencyService



DependencyService

- Service Locator proporcionado por Xamarin Forms llamado DependencyService.
- La idea es definir una interfaz en Core
- Y la implementación de la misma en cada plataforma.

```
namespace XamFormsMarvel.Services
{
    public interface IOpenWebService
    {
        void OpenUrl(string url);
    }
}
```



DependencyService



Implementación interfaz en Android

```
[assembly: Xamarin.Forms.Dependency (typeof (OpenWebService))]
namespace XamFormsMarvel.Droid.Services
{
    public class OpenWebService : IOpenWebService
    {
        #region IOpenWebService implementation

        public void OpenUrl (string url)
        {
            var uri = Android.Net.Uri.Parse (url);
            var intent = new Intent (Intent.ActionView, uri);
            intent.SetFlags (ActivityFlags.NewTask);
            Application.Context.StartActivity (intent);
        }

        #endregion
    }
}
```

Implementación interfaz en iOS

```
[assembly: Xamarin.Forms.Dependency (typeof (OpenWebService))]
namespace XamFormsMarvel.iOS.Services
{
    public class OpenWebService : IOpenWebService
    {
        public OpenWebService ()
        {
        }

        #region IOpenWebService implementation

        public void OpenUrl (string url)
        {
            UIApplication.SharedApplication.OpenUrl(new NSUrl(url));
        }

        #endregion
    }
}
```





CustomRenderers

CustomRenderers

- Los utilizaremos para aplicar un diseño que no es posible en Forms.
- La idea es crear un Control en el Core.
- E implementar como se va a renderizar en cada plataforma.

```
namespace XamFormsMarvel.Controls
{
    public class ImageCircle : Image
    {
    }
}
```



CustomRenderers



Implementación interfaz en Android

```
[assembly: ExportRenderer(typeof(ImageCircle), typeof(ImageCircleRenderer))]
namespace XamFormsMarvel.Droid.Renderers
{
    public class ImageCircleRenderer : ImageRenderer
    {
        protected override void OnElementPropertyChanged (object sender,
            System.ComponentModel.PropertyChangedEventArgs e)
        {

            base.OnElementPropertyChanged(sender, e);
            if (e.PropertyName == Image.IsLoadingProperty.PropertyName && !this.Element.IsLoading
                && this.Control.Drawable != null)
            {

                using (var sourceBitmap = Bitmap.CreateBitmap(this.Control.Drawable.IntrinsicWidth,
                    this.Control.Drawable.IntrinsicHeight, Bitmap.Config.Argb8888))
                {
                    var canvas = new Canvas(sourceBitmap);
                    this.Control.Drawable.SetBounds(0, 0, canvas.Width, canvas.Height);
                    this.Control.Drawable.Draw(canvas);
                    this.Control.SetImageBitmap (GetRoundedShape (sourceBitmap));
                }

            }
        }
    }
}
```

Implementación interfaz en iOS

```
[assembly: ExportRenderer(typeof(ImageCircle), typeof(ImageCircleRenderer))]
namespace XamFormsMarvel.iOS.Renderers
{
    public class ImageCircleRenderer : ImageRenderer
    {
        protected override void OnElementPropertyChanged(object sender, PropertyChangedEventArgs e)
        {
            base.OnElementPropertyChanged(sender, e);

            if (Control == null) return;

            if (e.PropertyName == Image.IsLoadingProperty.PropertyName
                && !this.Element.IsLoading && this.Control.Image != null)
            {
                DrawOther();
            }
        }
    }
}
```

Links de ayuda



Xamarin Forms

[Documentación Xamarin.Forms](#)

[Xamarin Plugins](#)

[What's Forms 2.0new Xamarin](#)

Material Design

<http://www.materialpalette.com/>

[Tutorial](#)



Demo Xamarin Forms

Ejemplo demo Xamarin forms 2.0



DEMO 3

Xamarin Forms 2.0





Preguntas

Tiempo para dudas,
preguntas y respuestas





Contacto

Estamos a tu disposición en
Madrid y Málaga



Ven a conocernos

Madrid Avenida de Manoteras 38, Edificio B, Oficina 008
28050 Madrid.

Málaga Calle Pirandello 16, Oficina 1-7
29010 Málaga.



Llámanos

+34 91 240 47



Visita nuestra web

www.bravent.net



Síguenos

