Universal Windows Platform Y Xamarin

Cubriendo todas las plataformas del mercado

Bravent
IT consulting company

Xamarin

Why Xamarin?

¿Por qué elegir Xamarin? Principales razones

- Desarrollo usando un mismo lenguaje (C# o F#) para las tres plataformas que dominan el mercado: iOS, Android y Windows Phone.
- Ahorro de tiempo y dinero. Las empresas pueden reducir tiempos de desarrollo debido a la gran facilidad que ofrece Xamarin para compartir código entre plataformas.
- Soporte constante de nuevas APIS. Xamarin ofrece un soporte bastante rápido con respecto a nuevas APIS que puedan publicar Apple o Google. Google Glass, Apple Watch, etc están soportadas por Xamarin.
- Resultados nativos. Xamarin compila a una aplicación nativa para todas las plataformas. Esto da a lugar a una mejor experiencia de usuario que otras opciones multiplataforma que apuestan por apps hibridas.

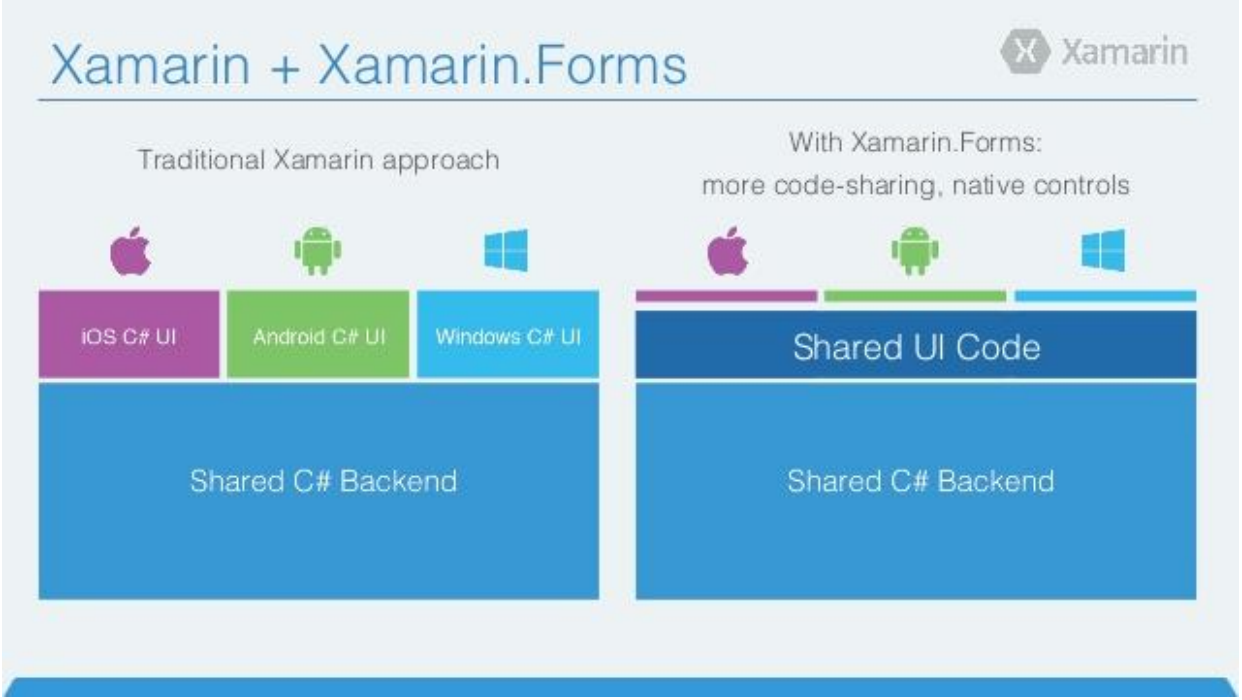
- El número de bugs de tu código puede reducirse considerablemente ya que gran cantidad del mismo se comparte.
- Con Xamarin.Forms el desarrollo puede ser incluso más rápido ya que se puede compartir hasta la interfaz de usuario.

Why Xamarin?

Comparativa entre diferentes herramientas multiplataforma del mercado

	Xamarin	Native	React Native	Native Script (Telerik)	PhoneGap	Titanium
Native apps / UI / performance	Yes	Yes	Yes	Yes	No	No
Languages	C# and F#	Swift, Objective-C, Java	JavaScript	JavaScript	JavaScript	JavaScript
Cross-platform Code Sharing	Yes	No	Yes	Yes	Yes	Yes
Active Developer Community	Yes	Yes	No	No	Yes	No
Commercially Licensed	Yes	No	No	Yes	No	Yes
Results on Stackoverflow	27,145	78,450	5,811	411	45,650	13,641

Arquitecturas Xamarin: ¿Que opciones tenemos?

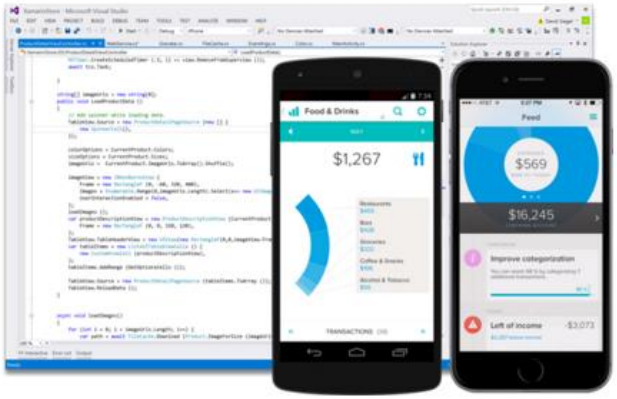


- Xamarin.Forms es una buena opción cuando la interfaz de usuario no es demasiado compleja
- Xamarin accede a las APIS nativas y nos permite cualquier tipo de personalización que puede ofrecer una app implementada en nativo

Xamarin 4. What's new

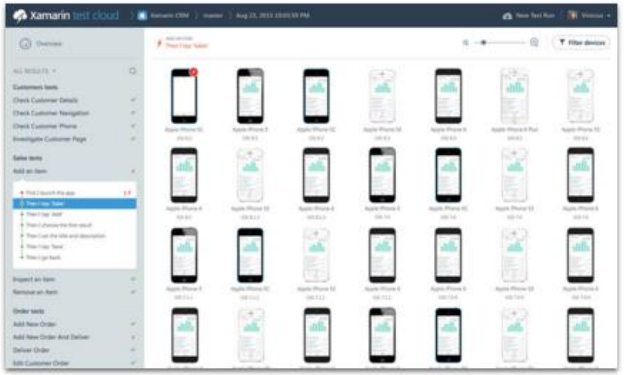
Xamarin 4

Xamarin 4 es un paquete completo de soluciones para el desarrollo móvil



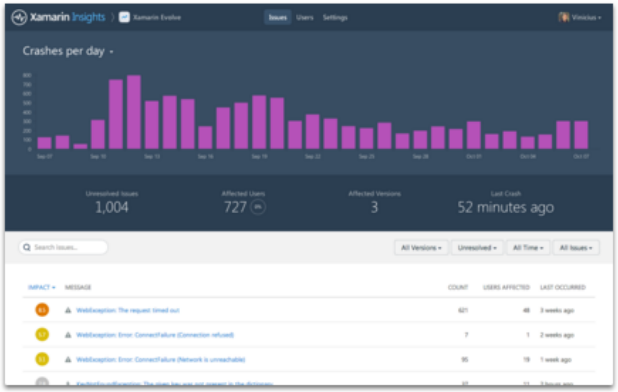
Xamarin Platform

Build apps in C# for iOS, Android and Windows Phone.



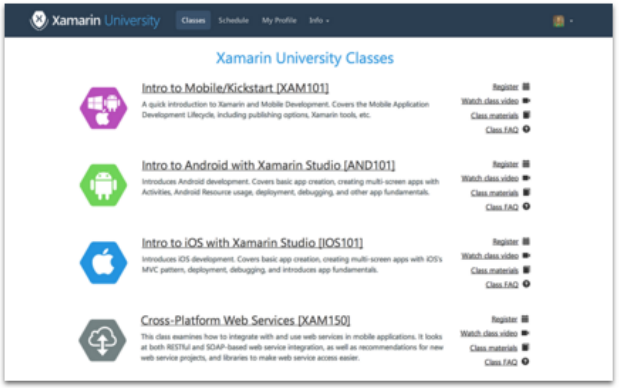
Xamarin Test Cloud

Automatically test your app on hundreds of mobile devices.



Xamarin Insights

Real-time monitoring to improve your apps.



Xamarin University

Unlimited, live mobile app development training from mobile experts.

Xamarin 4
What's new

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 Insights ya no es Beta

- Solución multiplataforma para monitorizar nuestras apps

Xamarin Test Recorder y Xamarin.UITest

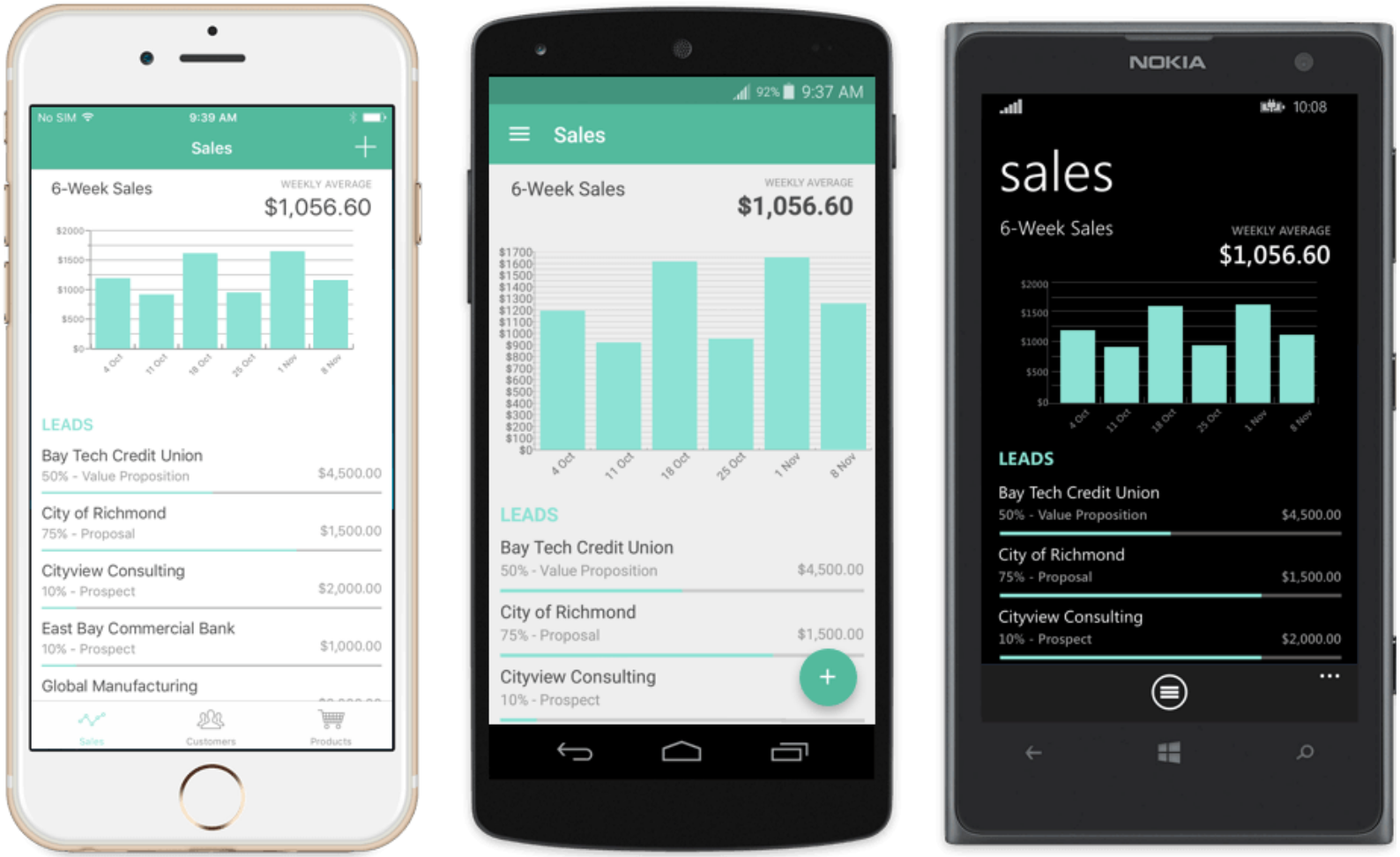
- Herramienta para grabar nuestras pruebas funcionales.
- Integrada con Xamarin Test Cloud

Xamarin Inspector Preview

- Similar a las herramientas de desarrollo de los navegadores web.
- Inspector en combinación con Continius (nuget package open source) podemos modificar diseño y código en tiempo real sin tener que volver a compilar.

Bravent
IT consulting company

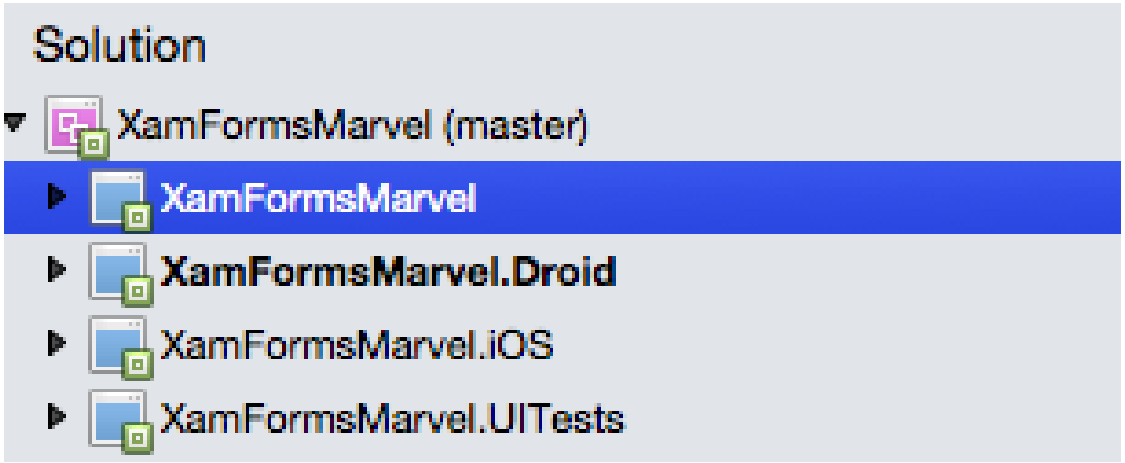
Xamarin Forms 2.0



Xamarin.Forms

Estructura de un proyecto de Xamarin.Forms

- Un proyecto Core, pero que no solo incluye la lógica de negocio sino también las Vistas de las app.
- Un proyecto para cada plataforma.
- Un proyecto de UITest.



Bravent
IT consulting company

Xamarin.Forms

¿Que 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
    - CustomRenderes

Bravent
IT consulting company

Xamarin.Forms

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);
    }
}
```

Xamarin.Forms

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
    }
}
```

Xamarin.Forms

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
    {

    }
}
```

Xamarin.Forms

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();
            }

        }
```

Bravent
IT consulting company

Xamarin.Forms

Gorilla Player:

http://gorillaplayer.com/

- Se trata de una herramienta recién salida del horno que nos ayudará a maquetar XAML en Xamarin.Forms
- Podremos tener varios simuladores abiertos con el preview del XAML que estemos editando en Xamarin Studio y en Visual Studio
- Si queréis saber como manejarlo, en la web de Bravent hemos publicado un post recientemente.

Bravent
IT consulting company