

A collection of approximately 15 squares in light blue, medium blue, and grey, scattered across the top half of the slide.

MDPA

Máster en Diseño y Programación de Apps

Entornos de desarrollo

Parte 3: Xamarin

Presentación asignatura/sesión

Entorno de desarrollo Xamarin



■ Descripción y temario

- En esta primera parte de Xamarin, veremos el entorno usado para trabajar, Visual Studio y Xamarin Studio, los requisitos de hardware y software para cada tipo de proyecto, los diferentes tipos de proyecto que podemos crear y una breve introducción a como funciona cada uno con herramientas como MVVMCross.

■ Profesor

- Oriol Noya - *Freelance Frontend Web/Mobile Developer & Creative Director*

■ Evaluación

- Se realizará un ejercicio práctico al final de las sesiones donde se pondrán en práctica todos los conceptos aprendidos.

■ Entrega

- **07/01/2018:** Práctica final de las sesiones de Xamarin

Objetivos

- Esta primera asignatura del curso busca conseguir los conocimientos básicos de los lenguajes y herramientas a usar durante el resto del curso.
- Se desarrollará una pequeña práctica, para poner en funcionamiento los conceptos aprendidos hasta el momento.

Metodología

Durante las clases se expondrá información y se realizarán explicaciones teóricas acompañadas de ejemplos prácticos en vivo que los alumnos podrán seguir y desarrollar por sus propios medios.

Se intentará que dentro de lo posible todas las clases sean lo más prácticas posible.

Sistema de evaluación

Esta primera asignatura del curso se evaluará mediante la realización de una práctica en la que los alumnos deberán usar los conceptos aprendidos:

La práctica consistirá en una aplicación Xamarin Android, con la estructura básica de MVVMCross implementada que muestre una pantalla con un texto de bienvenida

La práctica deberá ser entregada el 7 de Enero como muy tarde.

Introducción a Xamarin 1 (de 2)

Teoría (1a mitad de la clase):

- Qué es Xamarin?
- Xamarin.Classic
- Xamarin.Forms
- Ecosistema Xamarin

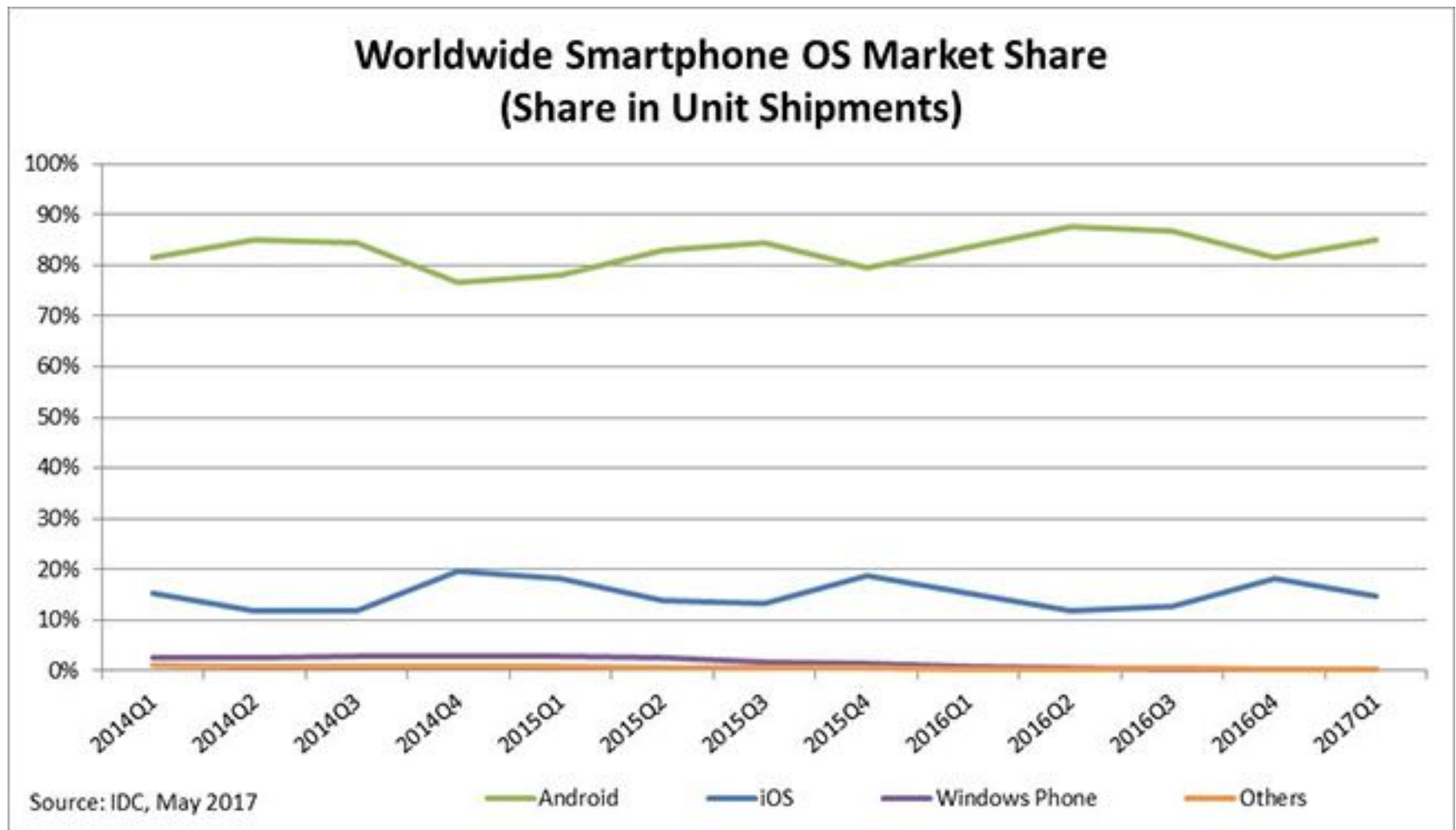
Práctica (2a mitad de la clase):

- Instalación + 1r proyecto con Xamarin.Classic

El problema actual de los desarrolladores



El problema actual de los desarrolladores



Y 3 maneras de enfocar la solución:

1. Enfoque Nativo



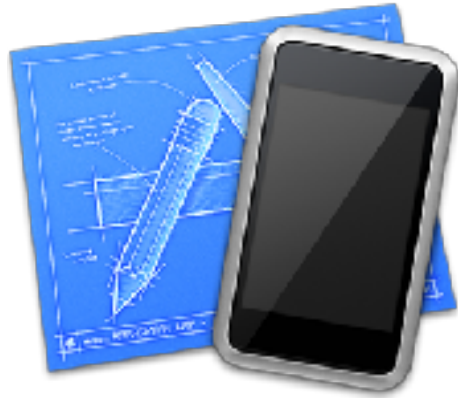
2. Enfoque Híbrido



3. Enfoque Nativo Multiplataforma



1. Enfoque nativo



iOS

Objective-C
Swift
Xcode



Android

Java
Kotlin
Android Studio

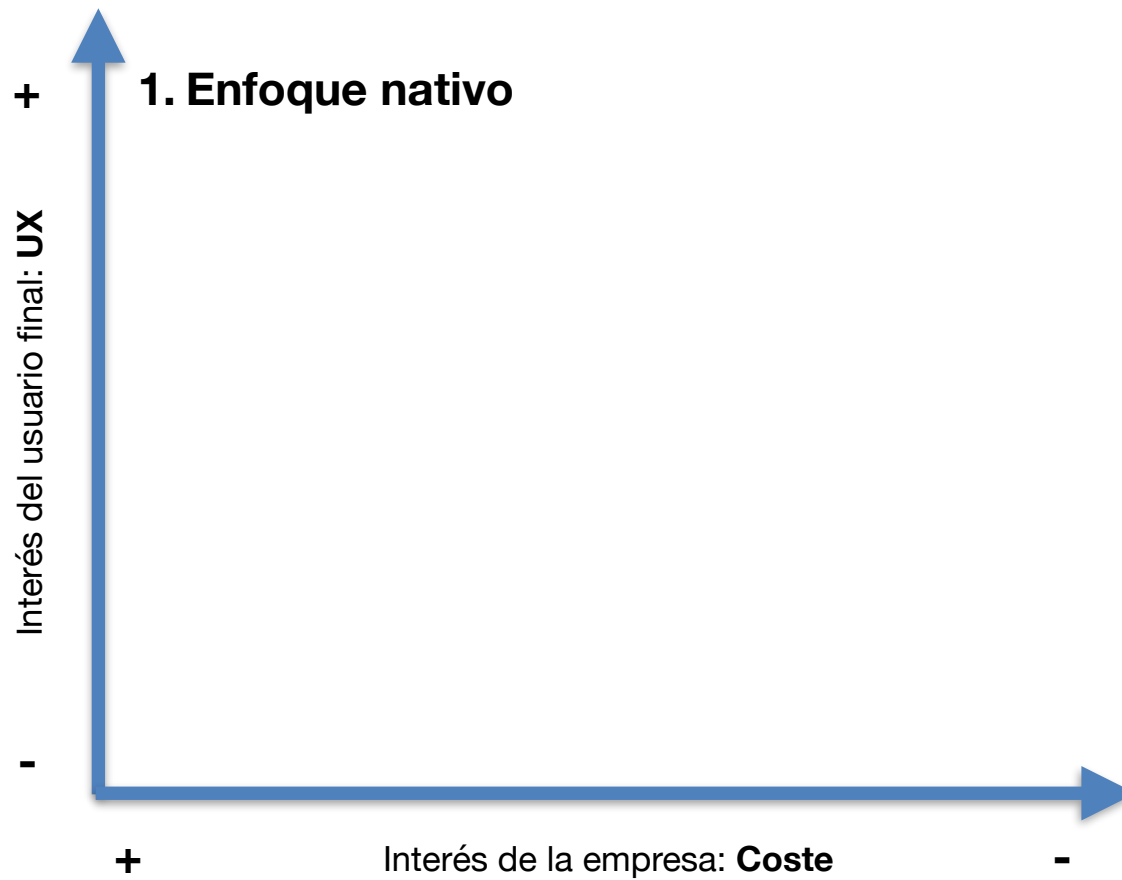


Windows

C#
C++
Visual Studio

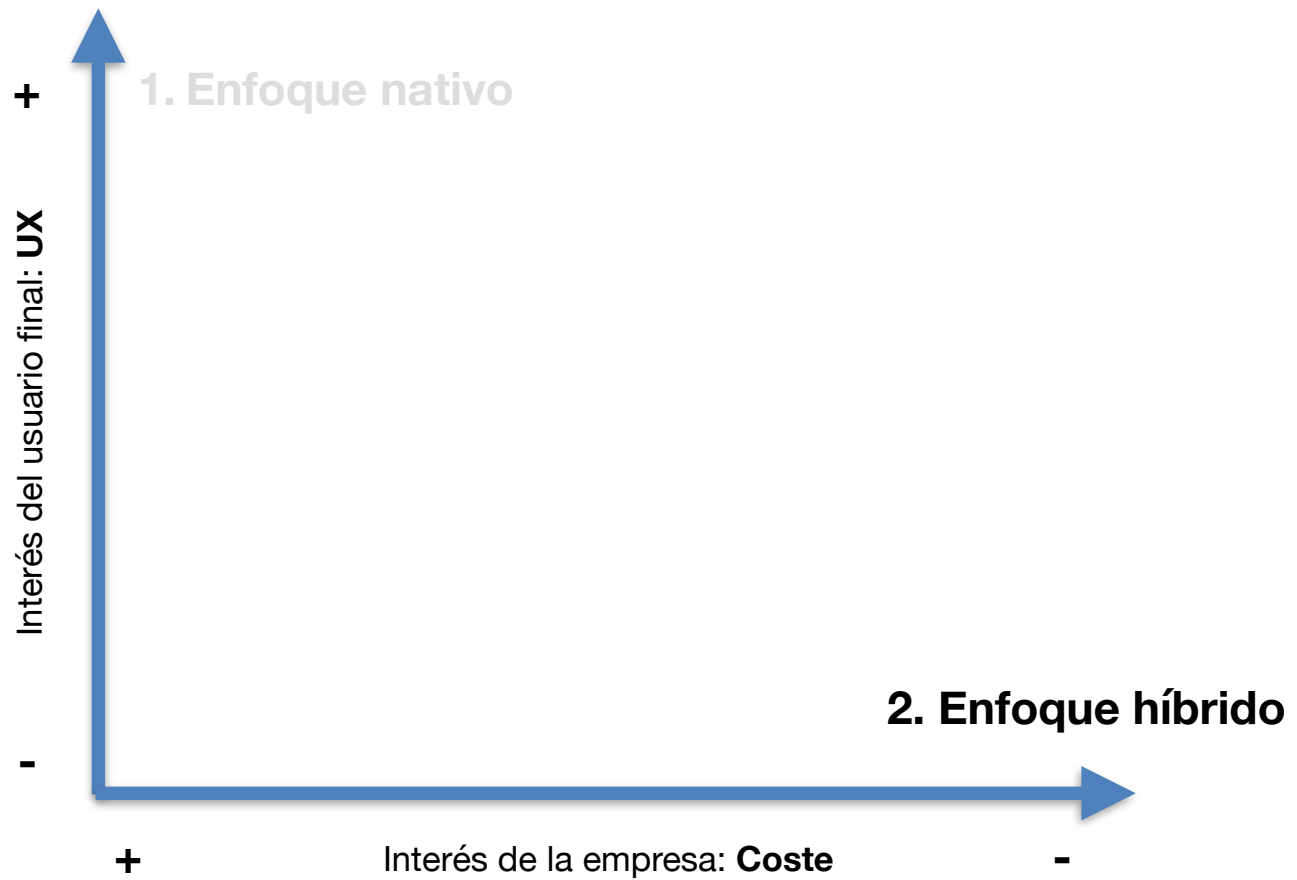
1. Enfoque nativo

- ✓ Mejor rendimiento → **Mejor UX**
- ✓ Elementos UI nativos → más intuitiva → **Mejor UX**
- ✓ Acceso fácil y total al HW del teléfono (GPS, Calendario, etc)
- ✓ Acceso a APIs nativas
- ✗ Dificultad para la reutilización de código → Trabajo por duplicado → **Mayor coste**
- ✗ Mantenimiento de 2 o más equipos de desarrolladores → **Mayor coste**
- ✗ Pérdida de control sobre la alineación de las diferentes versiones
- ✗ Know-how diseminado
- ✗ Dificultad para ser ágiles al cambio → Trabajo por duplicado → Mayor tiempo → **Mayor coste**



2. Enfoque híbrido

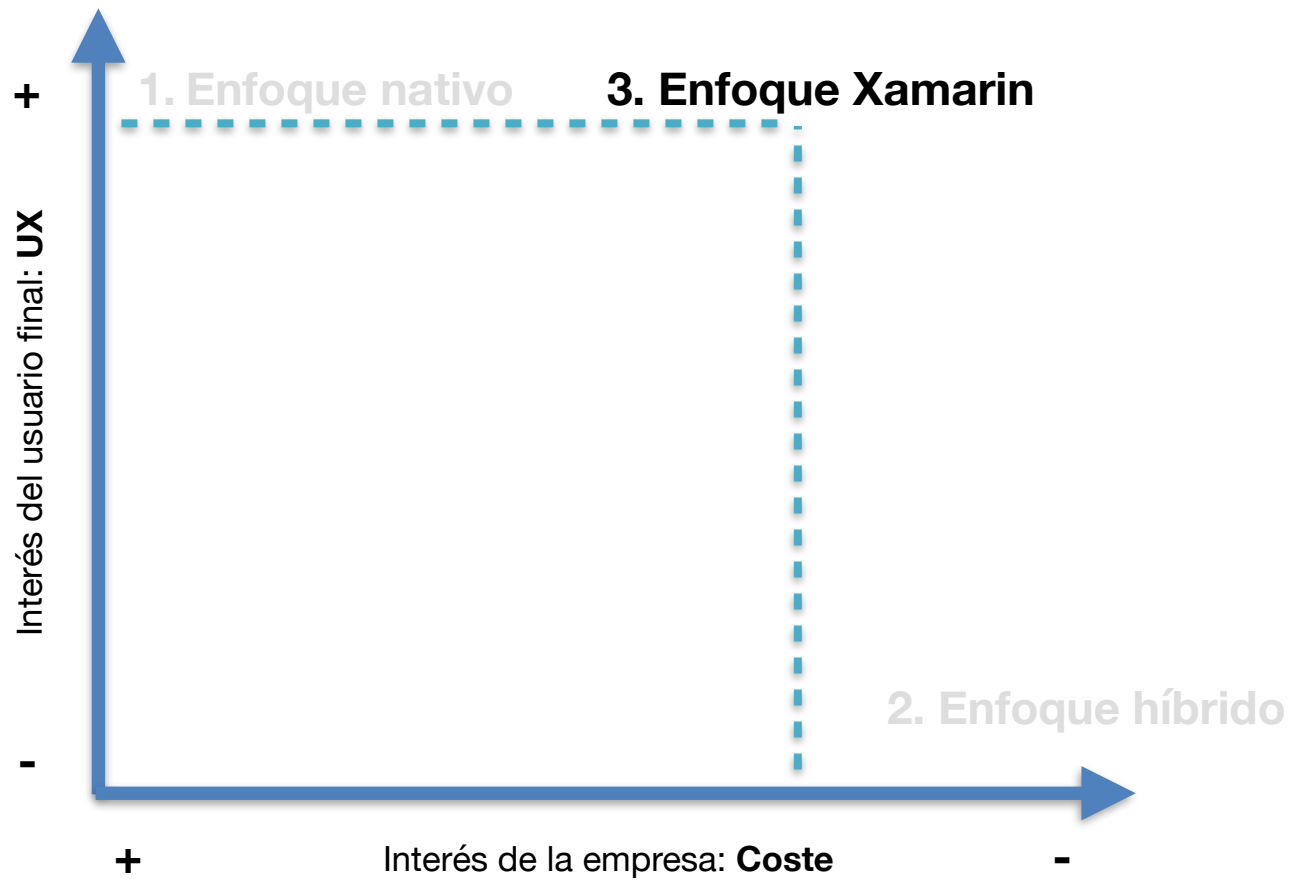
- ✓ 1 sólo codebase → **Menor coste**
- ✓ Mayor agilidad a los cambios → **Menor coste**
- ✓ Lanzamientos más rápidos (sobretudo al inicio) → **Menor coste**
- ✗ Rendimiento no óptimo → **Peor UX**
- ✗ UI no nativa → **Peor UX**
- ✗ Alta dificultad para acercarse a “lo nativo” → **Mayor coste**



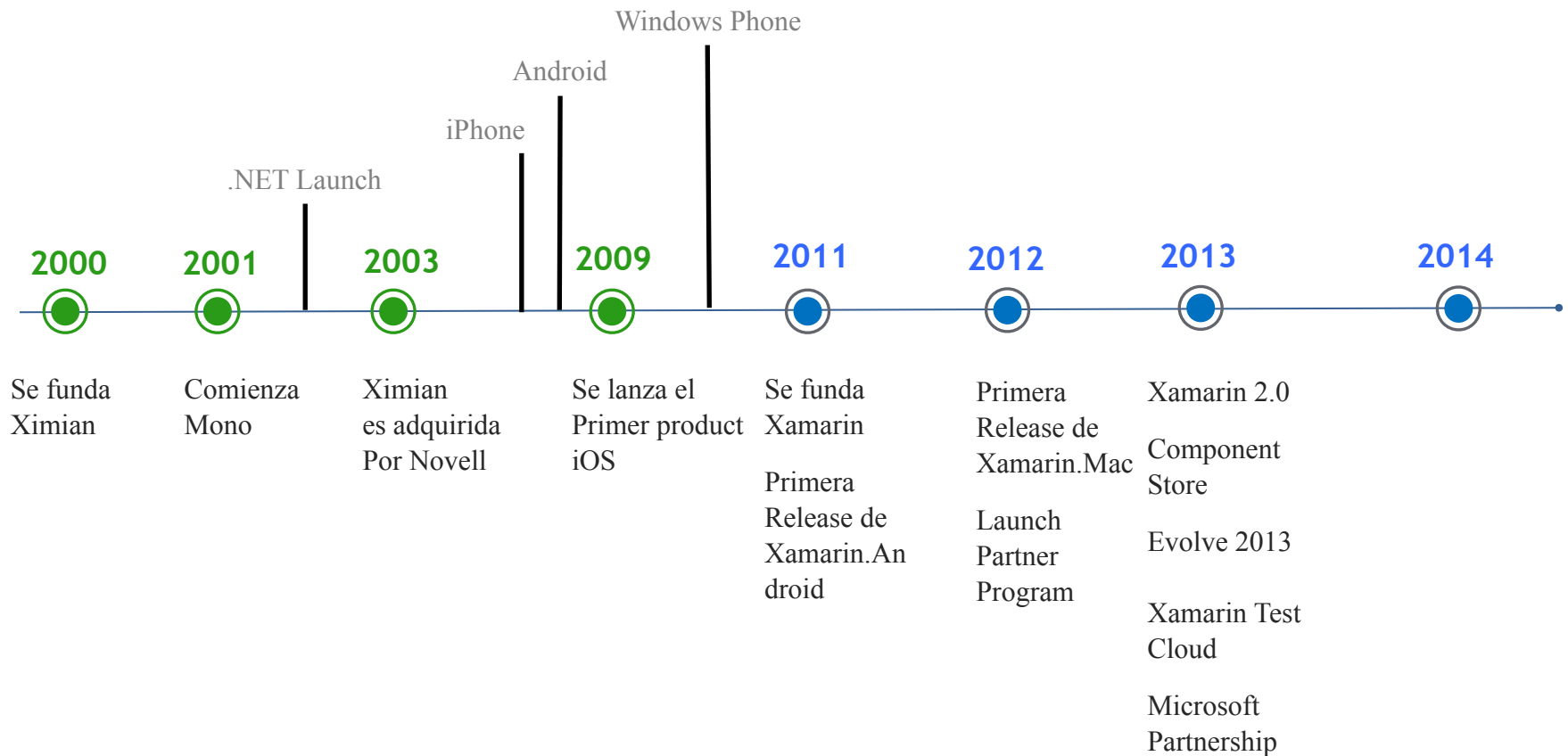
3. Enfoque Xamarin

Lo mejor de los 2 enfoques:

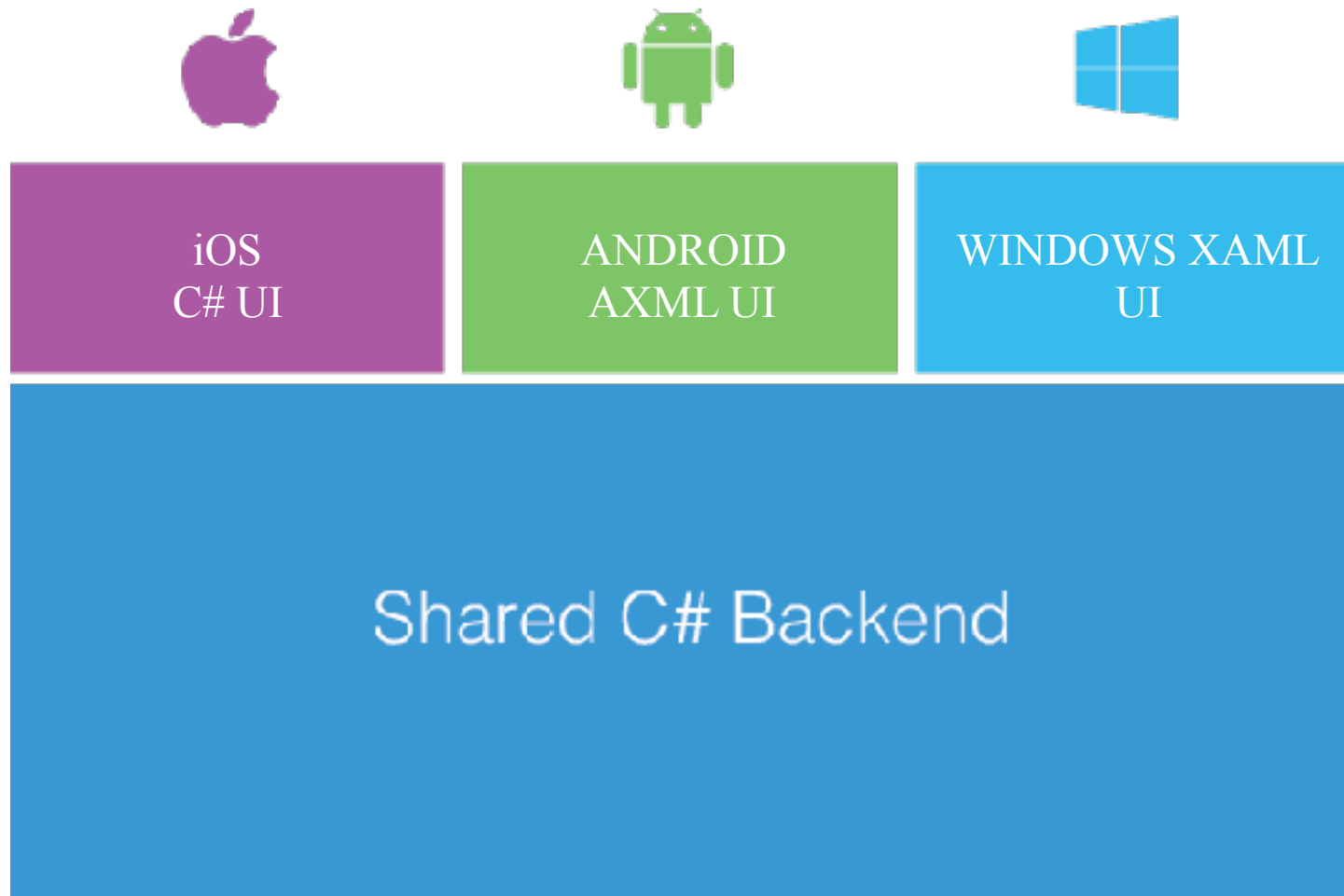
Compartir código, pero sin comprometer la UX



Un poco de historia...



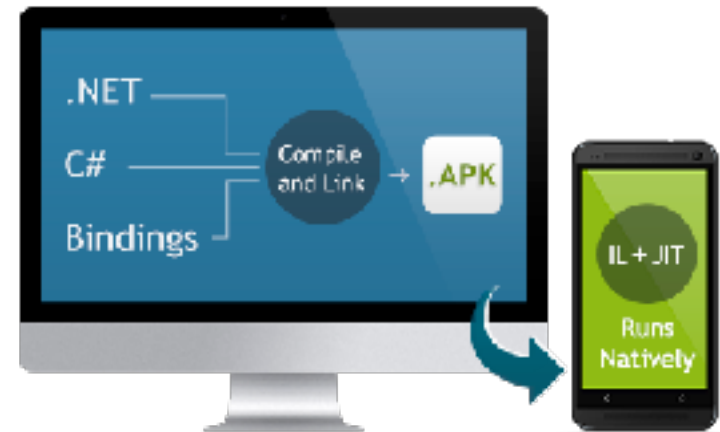
Xamarin.Classic



¿Dónde ocurre la magia?

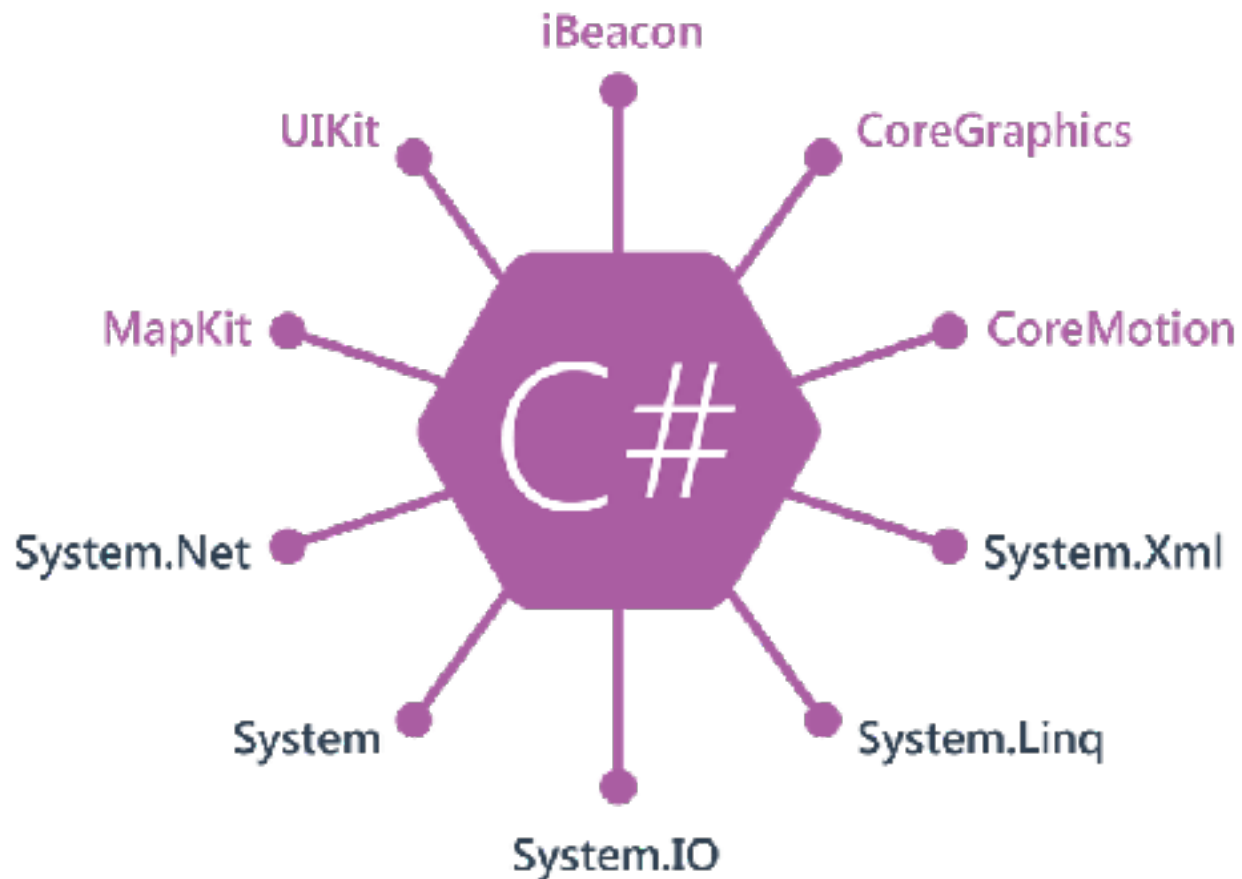


Xamarin.iOS realiza una compilación Ahead Of Time (AOT) para crear un binario ARM para la App Store.

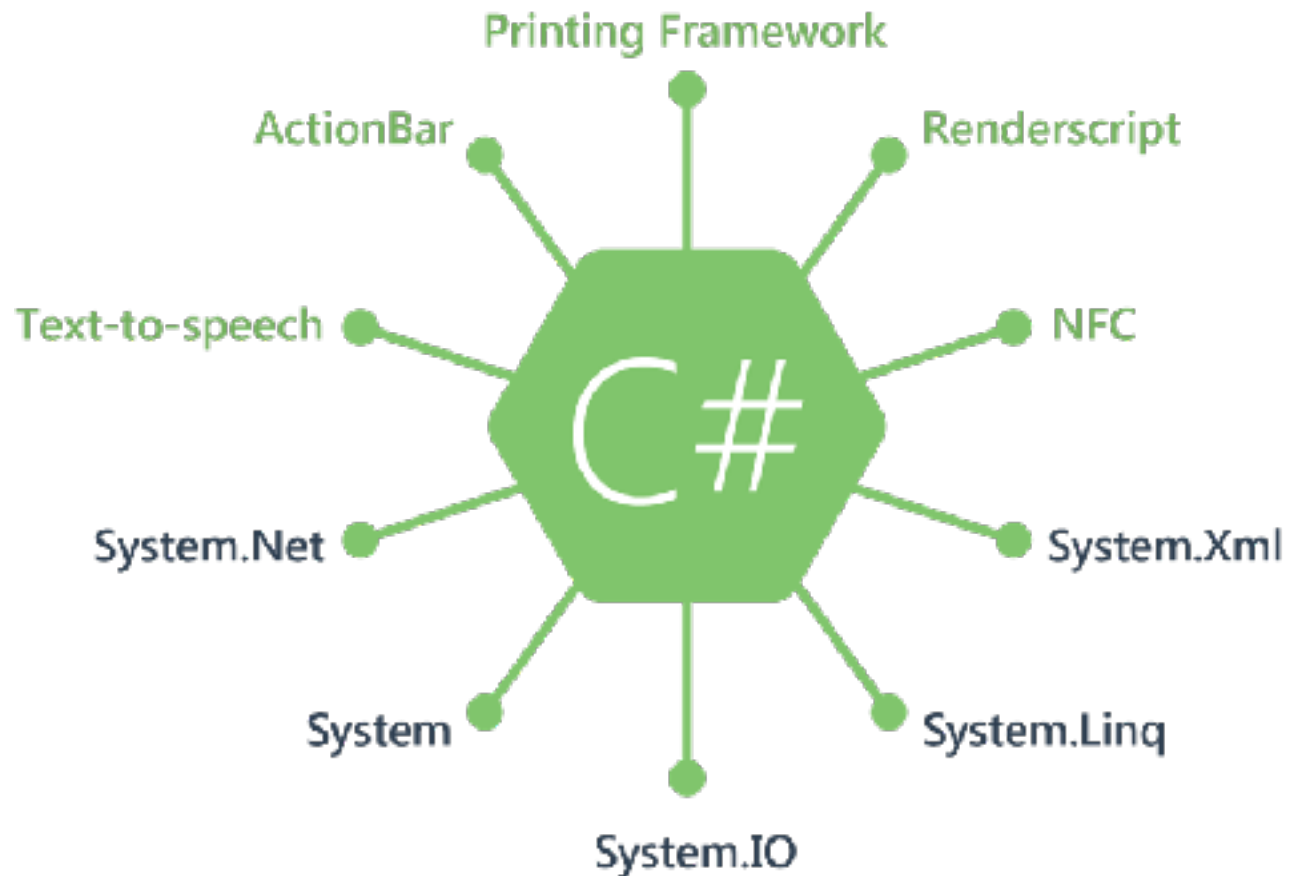


Xamarin.Android usa la compilación Just In Time (JIT).

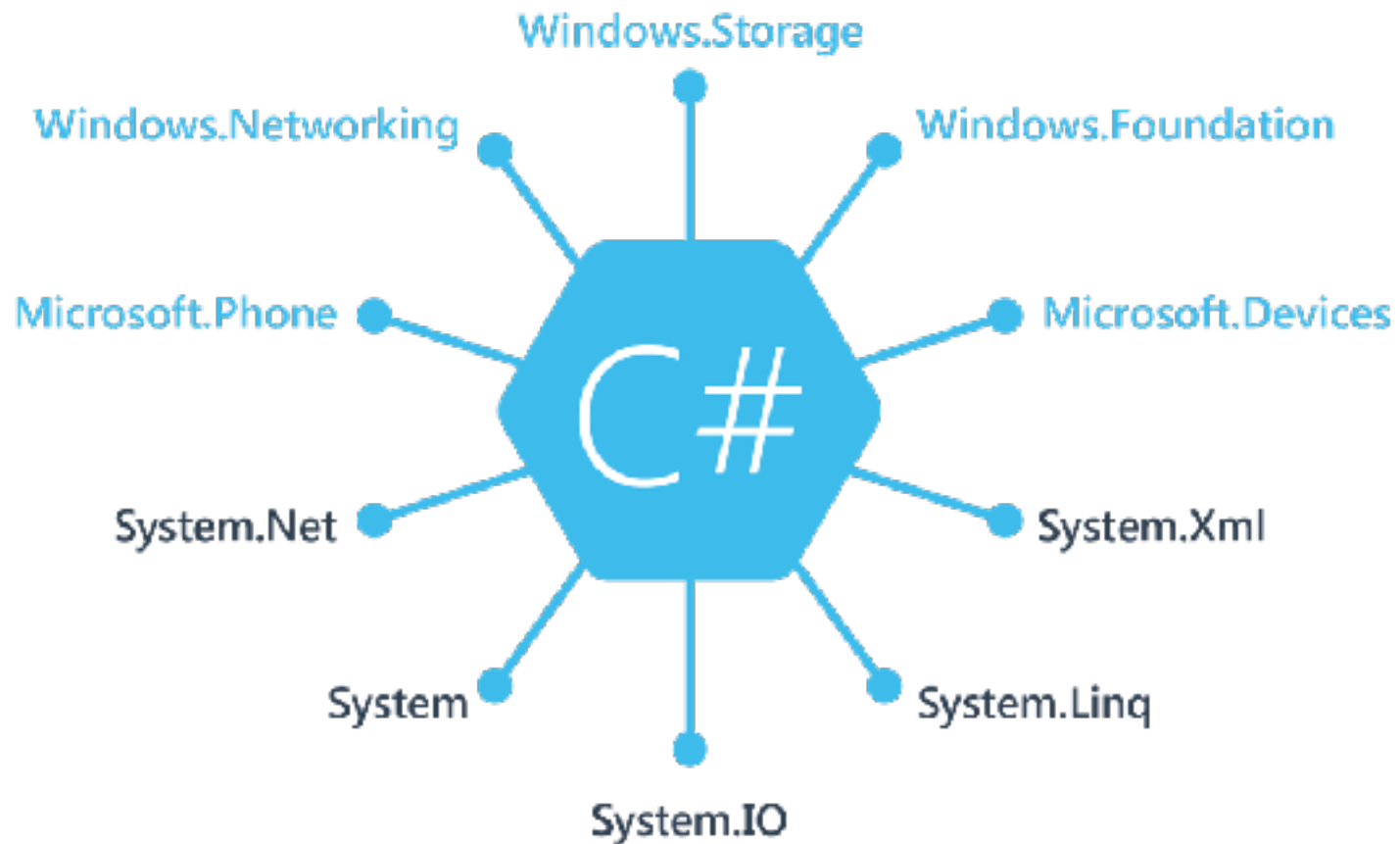
APIs iOS | 100% de Convergencia



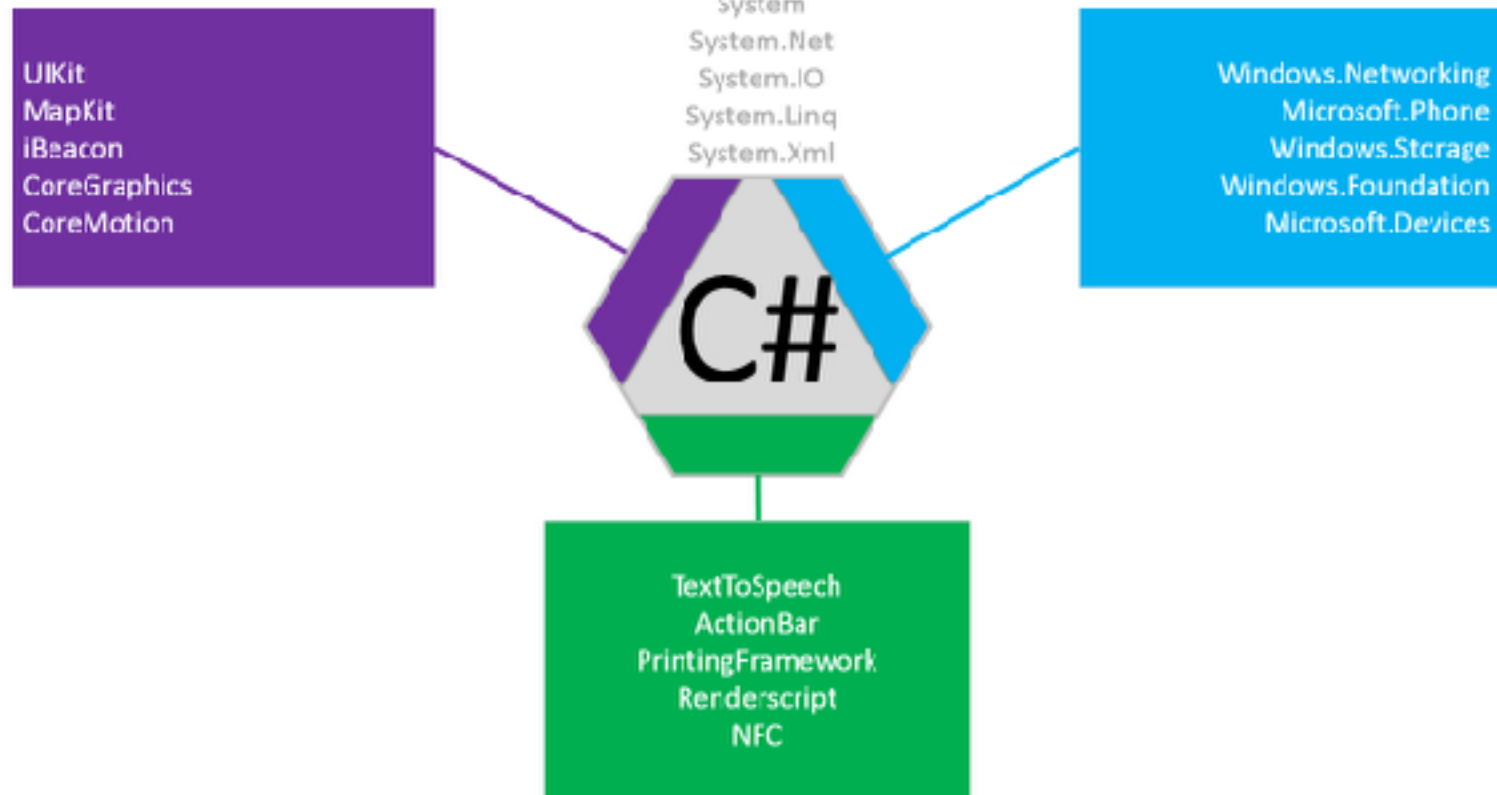
APIs Android | 100% de Convergencia



APIs Windows



CORE .NET API



Es decir, cualquier cosa que puedas hacer
en Objective-C, Swift o Java
se puede hacer en C# con Xamarin usando
Visual Studio

Desarrollo
nativo

iOS	Android	Win Phone
Xcode	Android Studio	Visual Studio
ObjC/Swift	Java	C#
Storyboard	AXML	XAML
MVC	MVC	MVVM

Desarrollo
Xamarin

iOS	Android	Win Phone
Visual Studio	Visual Studio	Visual Studio
C#	C#	C#
Storyboard	AXML	XAML
MVVM	MVVM	MVVM

Xamarin.Forms



Shared UI Code C#/XAML

Shared C# Backend

Xamarin.Forms

- Permite crear facilmente y con rapidez interfaces de usuario nativas compartidas
- Los elementos de Xamarin.Forms son mapeados a elementos nativos y behaviors propios de cada plataforma
- Podemos mezclar Xamarin.Forms con APIs nativas

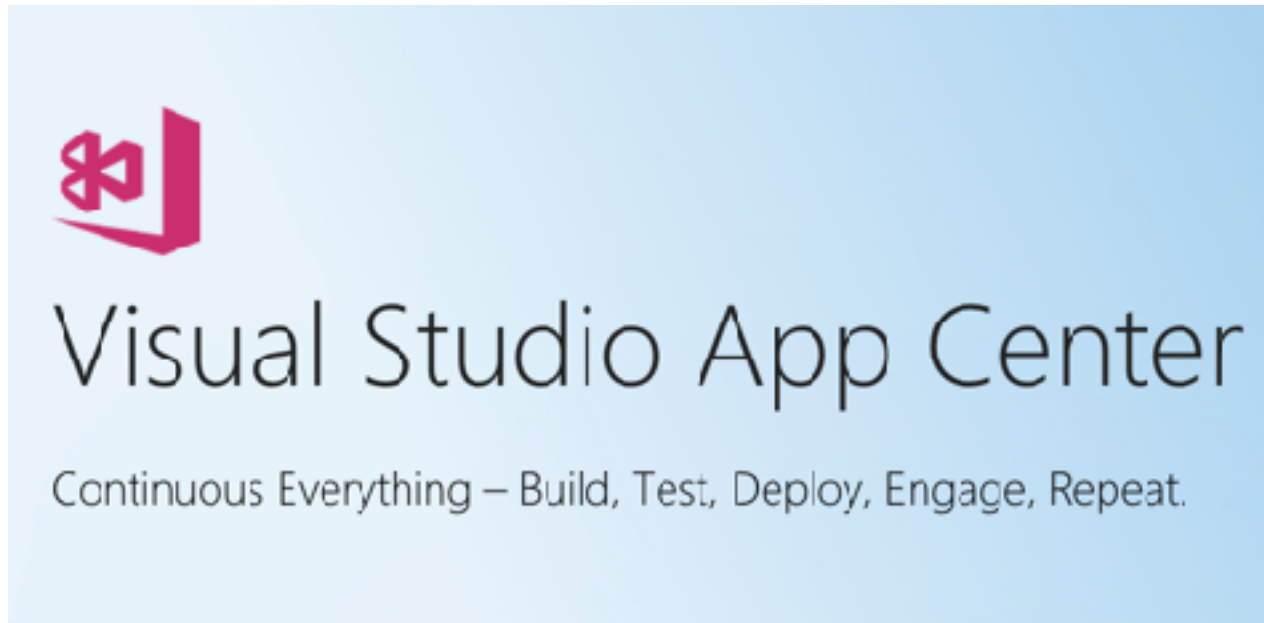
Desarrollo
Xamarin.Classic

iOS	Android	Win Phone
Visual Studio	Visual Studio	Visual Studio
C#	C#	C#
Storyboard	AXML	XAML
MVVM	MVVM	MVVM

Desarrollo
Xamarin.Forms

iOS	Android	Win Phone
Visual Studio	Visual Studio	Visual Studio
C#	C#	C#
XAML	XAML	XAML
MVVM	MVVM	MVVM

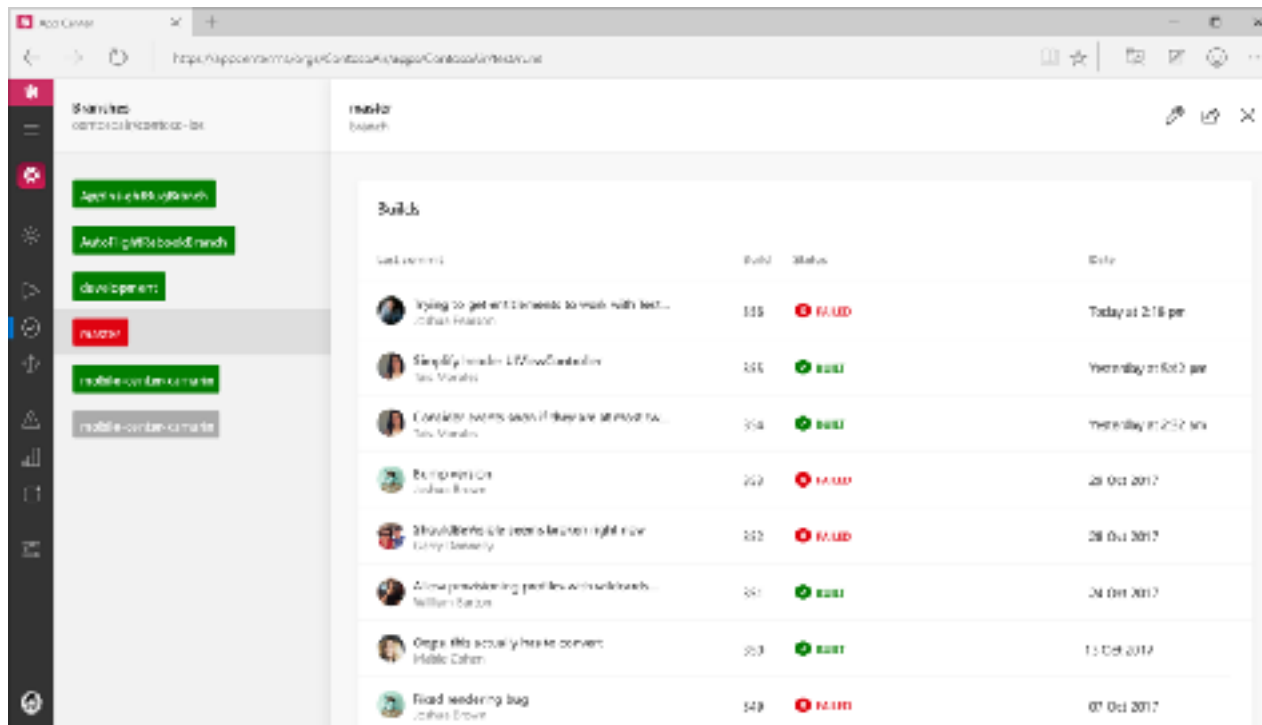
Visual Studio App Center



- Conjunto de herramientas para automatizar el ciclo de desarrollo de aplicaciones móviles
- También funciona para apps desarrolladas en nativo y en React Native

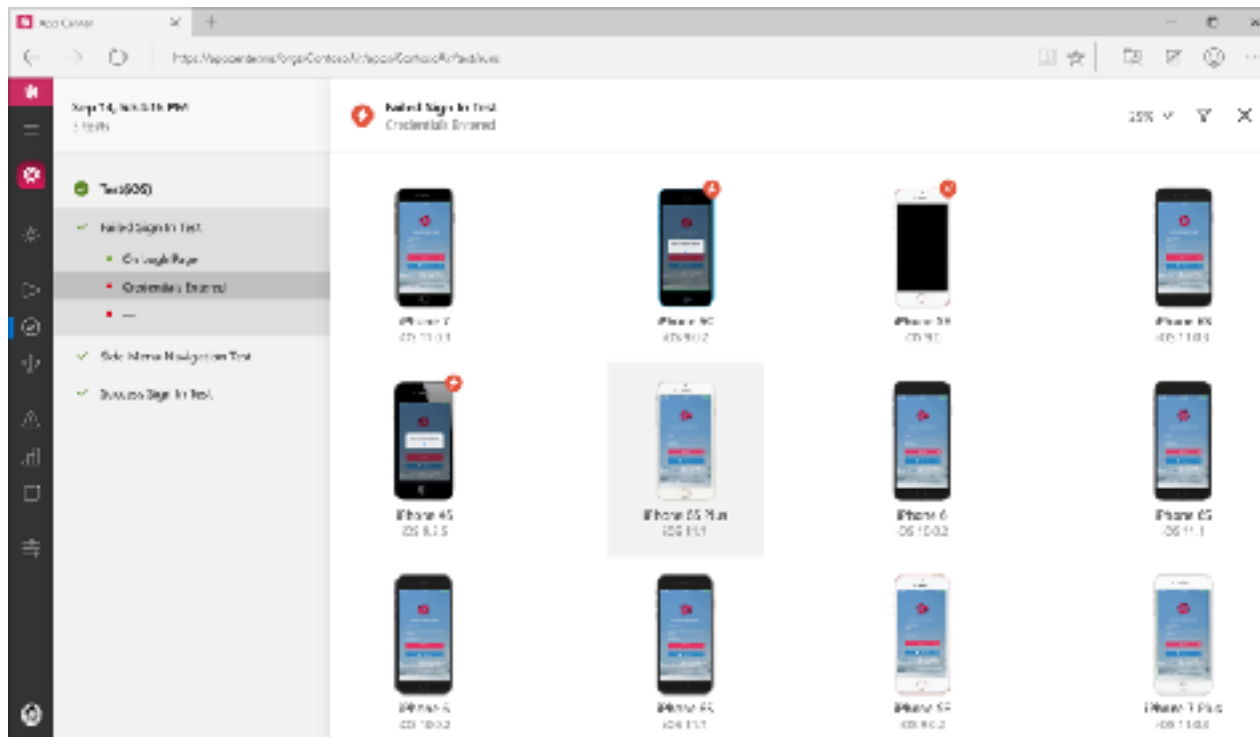
<https://www.visualstudio.com/app-center/>

Visual Studio App Center - Integración continua (CI)



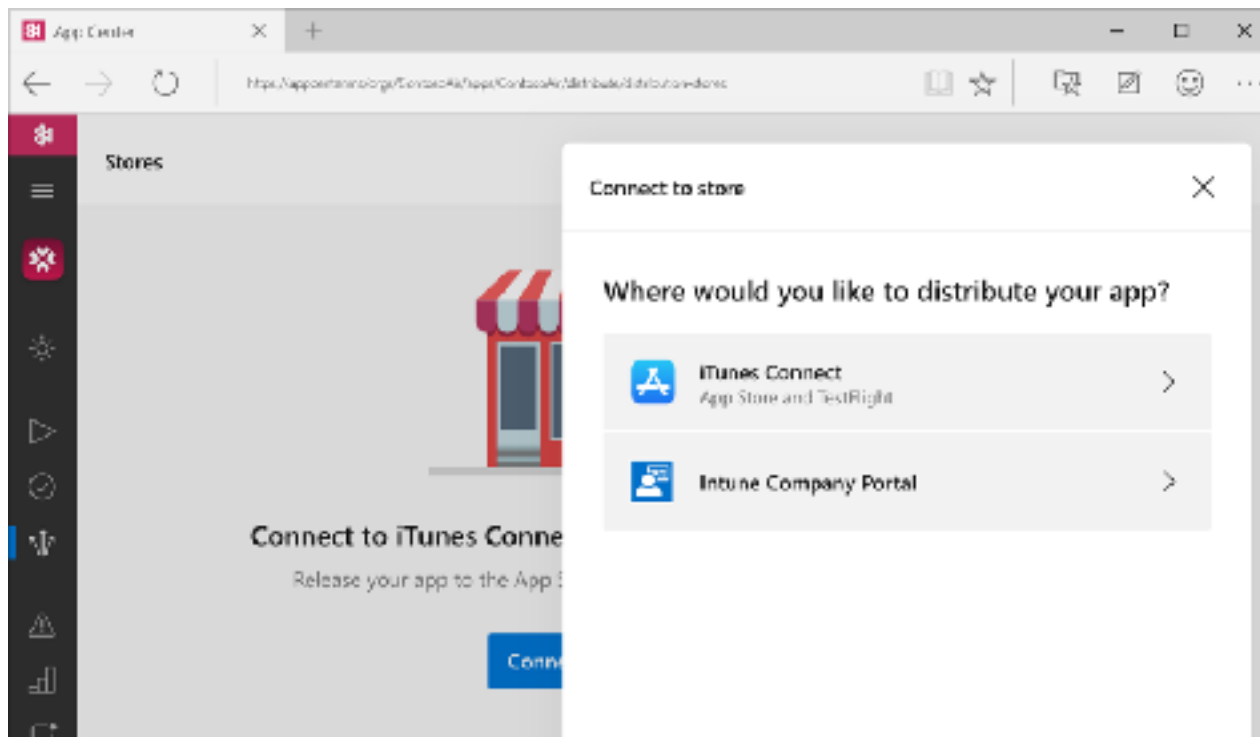
- Conectando tu repositorio el servicio genera automáticamente una build de la aplicación cada vez que alguien hace un pull, permitiendo rápidamente la detección de bugs.

Visual Studio App Center - UI Testing



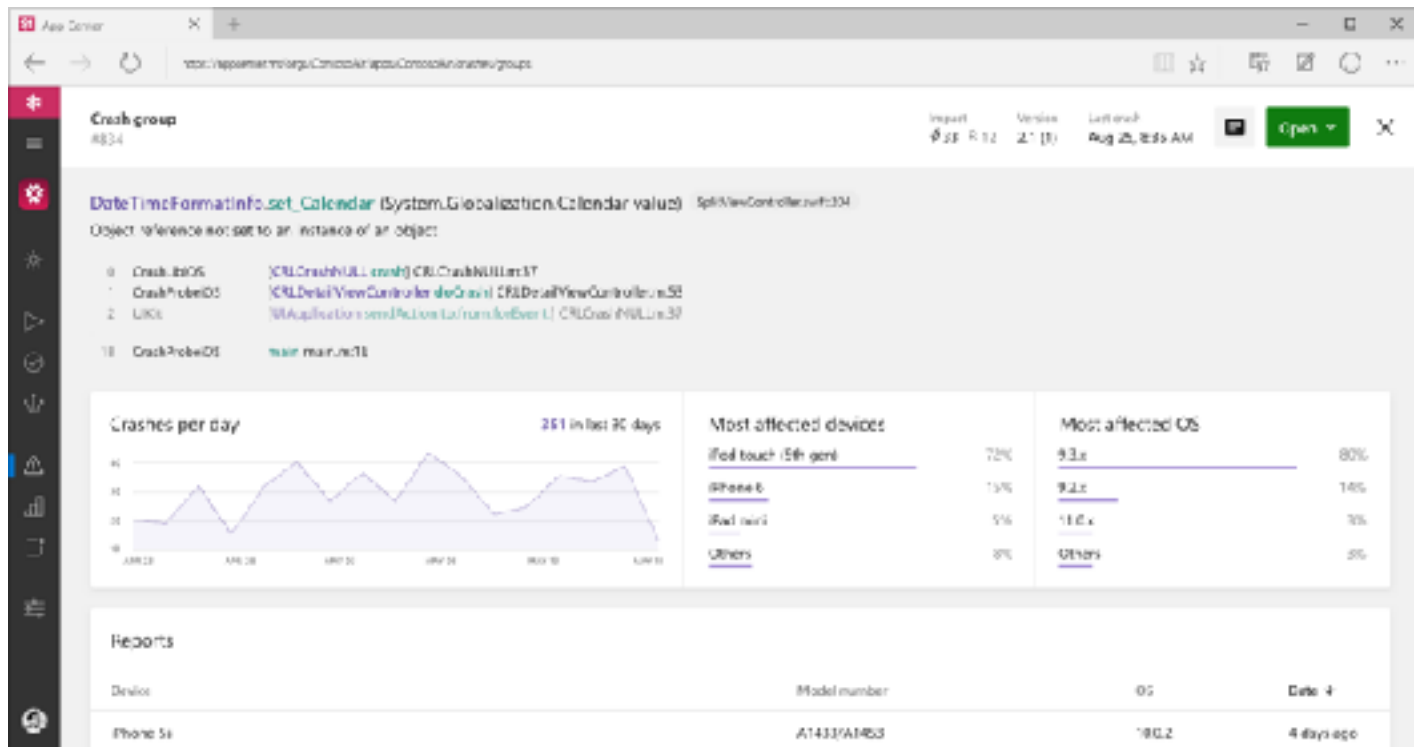
- Permite testear tu UI en dispositivos reales, automatizando las interacciones que quieres probar.
- Te da información tanto sobre bugs visuales como de rendimiento.

Visual Studio App Center - Despliegue continuo (CD)



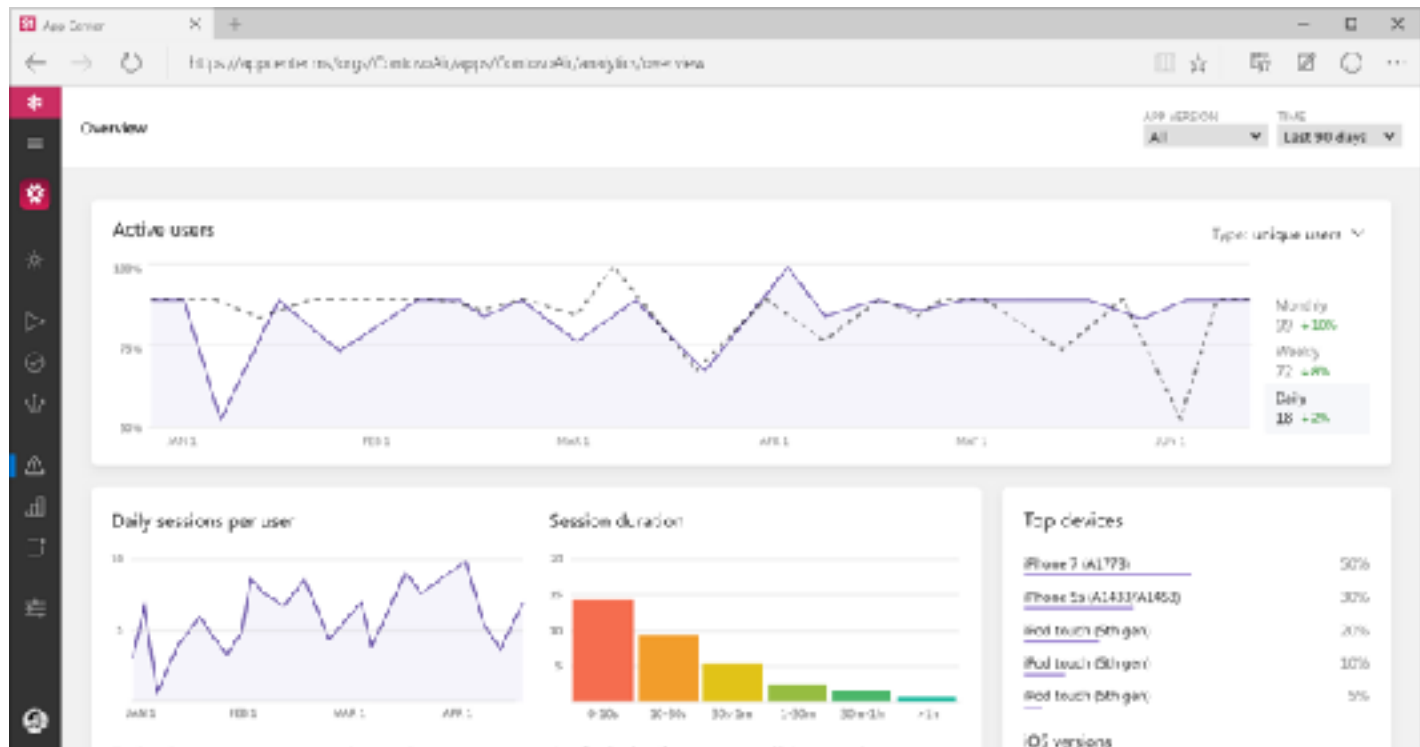
- Permite distribuir automáticamente tu aplicación a usuarios beta durante el desarrollo, así como gestionar la publicación de las releases finales a las respectivas Stores.

Visual Studio App Center - Crash reporting



- Permite monitorizar el comportamiento de tu aplicación una vez instalada en dispositivos reales y por tanto facilita la detección y corrección de bugs.

Visual Studio App Center - Analytics



- Permite monitorizar el comportamiento de tus usuarios a nivel de patrones de uso.

Xamarin University

- Es la mejor manera (exceptuando el master que estáis cursando :) de seguir evolucionando como desarrollador Xamarin.
- Tiene más de 80 clases repartidas en 10 itinerarios.
- Hay clases que se pueden hacer individualmente, pero hay otras que son en vivo con un profesor.
- Hay la opción de certificarse como Xamarin Developer siguiendo un determinado itinerario de clases obligatorias y superando un examen.
- Cuesta unos 80€ al mes pero tenéis 30 días gratuitos.

<https://www.xamarin.com/university>

Recursos Xamarin

- En el Xamarin Developer Center tenéis a vuestra disposición una infinidad de recursos para que vuestra entrada al desarrollo Xamarin sea más fácil:
 - Guides: <https://developer.xamarin.com/guides/>
 - Recipes: <https://developer.xamarin.com/recipes/>
 - Samples: <https://developer.xamarin.com/samples-all/>
 - APIs: <https://developer.xamarin.com/api/>
 - Workbooks: <https://developer.xamarin.com/workbooks/>
 - Videos: <https://developer.xamarin.com/videos/>
 - Foros: <https://forums.xamarin.com/>
- El blog es muy activo y sirve para estar al día de las últimas novedades: <https://blog.xamarin.com/>

Recursos Xamarin

- Otros links interesantes:

- Blog de James Montemagno: <https://montemagno.com/>
- Blog de Javier Suarez: <https://javiersuarezruiz.wordpress.com/>

Descanso