

# XAMARIN

## Qué es

---

Es una plataforma de desarrollo gratuita incluida en visual studio que nos permite construir aplicaciones usando las APIs de los sistemas operativos más importantes estamos hablando de **iOS, Android y Windows Mobile**. Las apps son codificadas utilizando C# y .NET

## ¿Por qué elegir Xamarin en lugar de las herramientas de desarrollo de cada plataforma?

---

Brevemente te cuento que Xamarin es un entorno de desarrollo Lanzado por primera vez en 2011 para crear aplicaciones nativas en cada plataforma móvil. Estamos hablando de **iOS, Android y Windows Mobile**. Microsoft incentiva a desarrollar en Xamarin ya que una de sus novedades es que puedes crear apps para cada plataforma, utilizando lenguaje C# y .net y nada más.

Para crear aplicaciones iOS se necesita escribir código en Objective-C y para crear Apps Android necesitamos conocer Java; Xamarin viene a unificar estas diferencias con su IDE Xamarin Studio, ya que solamente necesitamos dominar C# para crear aplicaciones iOS, Android y Windows Phone.

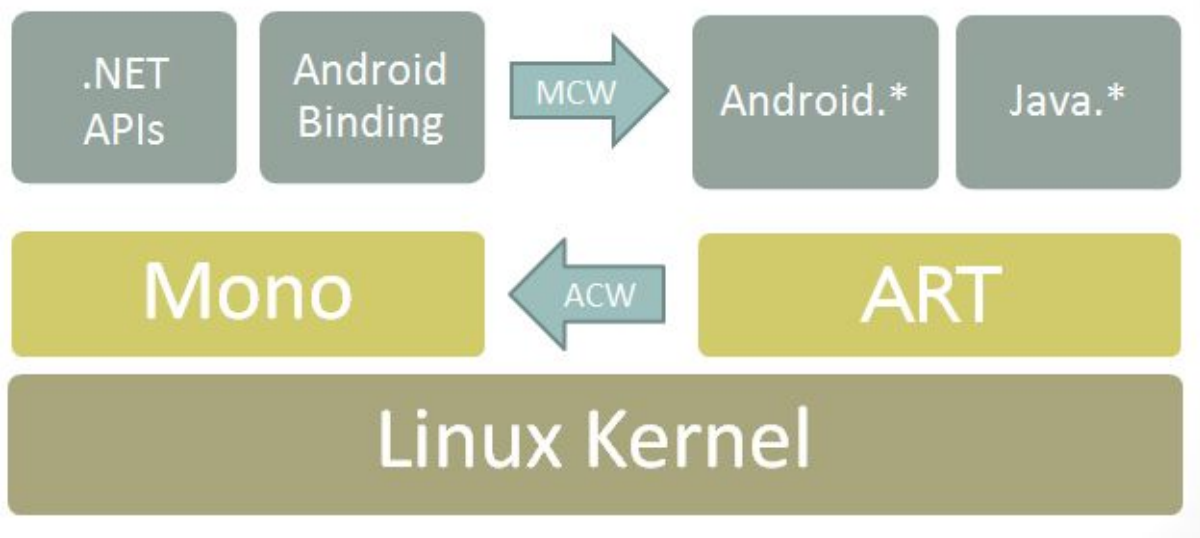
En resumidas cuentas este interesante IDE toma el código C# y lo compila como código nativo para cada plataforma

## Arquitectura de Xamarin

Las aplicaciones de Xamarin. Android se ejecutan en el entorno de ejecución de mono. Este entorno de ejecución se ejecuta en paralelo con la máquina virtual en tiempo de ejecución (ART) de Android. Ambos entornos en tiempo de ejecución se ejecutan sobre el kernel de Linux y exponen varias API al código de usuario que permite a los desarrolladores tener acceso al sistema subyacente. El entorno de ejecución mono se escribe en el lenguaje C.

En Android, la mayoría de los recursos del sistema como audio, gráficos, OpenGL y telefonía no están disponibles directamente en las aplicaciones nativas, solo se exponen a través de las API de Java en tiempo de ejecución de Android que residen en uno de los espacios de nombres Java. \* o Android. \* espacios de nombres.

Imagen aproximada de la arquitectura de Xamarin:



¿Cuáles son sus principales características?

- Las interfaces de usuario que se desarrollan con Xamarin son nativas.

- El código se comparte entre todas las plataformas.
- Las interfaces se pueden desarrollar mediante XAML.
- Open Source y en constante innovación.
- Está basado en Mono.
- **Lenguaje C#.** Xamarin utiliza un único lenguaje, C#, para crear aplicaciones de todas las plataformas móviles. Xamarin está compilado de forma nativa, lo que lo convierte en una opción para crear aplicaciones de alto rendimiento con aspecto nativo.
- **Basado en .NET framework.** Como C# es uno de los lenguajes del framework.NET, puede ser usado con un gran número de características útiles de .NET como Lambdas, LINQ, y programación Asynchronous (Asynk).

Para los desarrolladores que no tienen experiencia en el desarrollo móvil, pero dominan estas tecnologías .NET; tienen la posibilidad de adaptarse fácilmente al entorno de desarrollo de aplicaciones móviles con Xamarin.

## VENTAJAS

### Múltiples plataformas

Con Xamarin podemos crear aplicaciones con una interfaz de usuario totalmente nativa. No solo cumpliendo los estándares, sino también el comportamiento que se espera de cada plataforma. Por otro lado también tenemos acceso a toda la API nativa de cada plataforma; con lo cual podemos acceder a todas las funcionalidades que expone el dispositivo. Todo esto aprovechando la aceleración de hardware de la plataforma específica para compilar las aplicaciones y tener un rendimiento nativo.

Si eres desarrollador y no conoces XAMARIN; te sorprenderá saber que con Xamarin también podemos desarrollar aplicaciones de escritorio para macOS y Windows. Esto supone una gran ventaja por encima de la competencia. Más para las empresas y desarrolladores que les gusta

trabajar en diferentes plataformas. Y como plus final Xamarin también soporta Relojes Inteligentes.

## **Tecnología .NET**

Como C# es uno de los lenguajes de .NET framework, se puede usar varias funciones .NET útiles como Lambdas, LINQ, programación asincrónica, etc.

Para los desarrolladores que no tienen experiencia en el desarrollo móvil, pero dominan estas tecnologías .NET; tienen la posibilidad de adaptarse fácilmente al entorno de desarrollo de aplicaciones móviles con Xamarin.

## **Dos Enfoques**

Dentro de Xamarin podemos desarrollar aplicaciones bajo diferentes enfoques en la arquitectura del proyecto. Uno de estos enfoques te permite compartir la interfaz de usuario para cada plataforma. El otro te permite compartir la lógica de tu aplicación entre proyectos de cada plataforma.

Si queremos una aplicación que comparta los mismos controles de la interfaz de usuario entre plataformas entonces podemos utilizar Xamarin.Forms para el desarrollo de nuestra aplicación. Por otro lado si queremos compartir código común (Recursos) entre plataformas pero no compartir la interfaz de usuario entonces podemos elegir un proyecto con Xamarin Tradicional.

## **Xamarin.Forms**

Xamarin.Forms es una característica que permite utilizar los elementos de diseño de interfaces nativos de cada plataforma. Esto hará que tu aplicación se vea muy bien en Android o iOS, ya que se ajustará

visualmente al estilo de Material Design en el caso de Android y Apple Design para iOS.

### **ciclo de vida**

Cubre todo el ciclo de vida de una app: Desde el desarrollo, testeo hasta la puesta en producción, gracias a herramientas como Xamarin Test Cloud .

### **Soporte y mantenimiento de la plataforma estable**

Una fortaleza de Xamarin es que su ciclo de mantenimiento y mejora está asegurado, lo que hace que tu inversión de tiempo y recursos para desarrollar una app con esta plataforma no esté en riesgo

### **Documentación**

- Antes de comenzar a desarrollar utilizando Xamarin, dentro de su sitio se encuentra buena documentación para iniciarse en la creación de tu primera aplicación.
- Independientemente del sistema operativo para el cual deseas iniciarte, esta documentación y tutoriales te serán de mucha ayuda.

<https://docs.microsoft.com/es-es/xamarin/>

## **DESVENTAJAS**

Xamarin es una tecnología y plataforma para desarrollo de aplicaciones móviles fantástica pero al mismo tiempo hay que reconocerle algunos puntos débiles antes de elegirlo como plataforma de desarrollo para aplicaciones móviles.

1. **Soporte en las actualizaciones:** Desafortunadamente, Xamarin nos trae las últimas actualizaciones pero ofrece un soporte lento y a menudo tardío. Normalmente depende del equipo de

desarrolladores. No se pueden ofrecer herramientas de terceros con soporte inmediato para las últimas versiones de Android o iOS. El equipo de Xamarin necesita algún tiempo en implementar todos los nuevos cambios que ofrecen estas nuevas versiones. Por eso siempre las últimas actualizaciones llegan más tarde a Xamarin. No obstante, el equipo en mayor medida cumple que todas las nuevas características estén disponibles el día del lanzamiento oficial.

2. Xamarin tiene bibliotecas de código abierto. Pero a veces, estas tienen **limitado el acceso**.
3. **Una enorme comunidad, pero pequeña a su vez:** No hay que pensar que Xamarin no tiene una gran comunidad, no sería cierto. Pero si la comparamos con las comunidades nativas de Android o iOS es irremediable pensar que es pequeña. Esto a su vez también dificulta poder encontrar desarrolladores con experiencia en el desarrollo de aplicaciones en Xamarin. La plataforma está creciendo y la fuerza que ejerce Microsoft por detrás hace que crezca de forma rápida. Los desarrolladores de Xamarin con altos conocimientos son considerablemente más difícil de conseguir, comparado con la demanda de Android o iOS. Xamarin ofrece una plataforma de formación Xamarin University que aunque un nivel básico puede ser adquirido de forma gratuita, una vez más volvemos a entrar en un gasto necesario para aprender un nivel más profesional.
4. **Conocimiento de un lenguaje nuevo C#:** Desde su principio el concepto de Xamarin fue poder programar en ambas plataformas sin aprender un lenguaje nuevo (JAVA, Swift, Objective C) y poder desarrollar en C#. Pero si no tienes conocimientos de C# entonces esto deja de ser una ventaja y se convierte un inconveniente. Si por el contrario no tenemos conocimiento de ningún otro lenguaje entonces volvería a ser ventajoso. Por lo que, según la situación de cada uno este punto puede tener una diferente interpretación.
5. **No se puede compartir código de interfaz de usuario:** El beneficio de Xamarin es que podemos compartir código entre las diferentes plataformas. Pero solo se puede compartir la lógica del proyecto. Todo desarrollo orientado a interfaces de usuario tiene que

ser realizado específicamente para cada plataforma. Por consecuencia, al desarrollar juegos, enriquecidas interfaces de usuario, etc son más complicadas porque Xamarin no permite compartir la misma cantidad de código en estos casos.

## **6. Mayor tamaño de nuestra app**

Las aplicaciones compiladas en Xamarin tienen mayor tamaño que una aplicación nativa. Es una realidad que algunas aplicaciones nativas llegan a tener la mitad de tamaño que la misma aplicación en Xamarin.

## **7. Problemas de compatibilidad con aplicaciones de terceros**

Xamarin tiene problemas de compatibilidad con las bibliotecas de terceros. Te puedes encontrar algunos problemas mientras estamos integrando recursos de terceros en nuestra aplicación de Xamarin. La mayoría de las herramientas y bibliotecas que te ofrece con soporte completo para tecnologías nativas, pero al mismo tiempo el soporte de Xamarin no está siempre disponible. Xamarin tiene su propia tienda de componentes pero es posible que necesites añadir algunas funcionalidades a través de alguna otra aplicación.

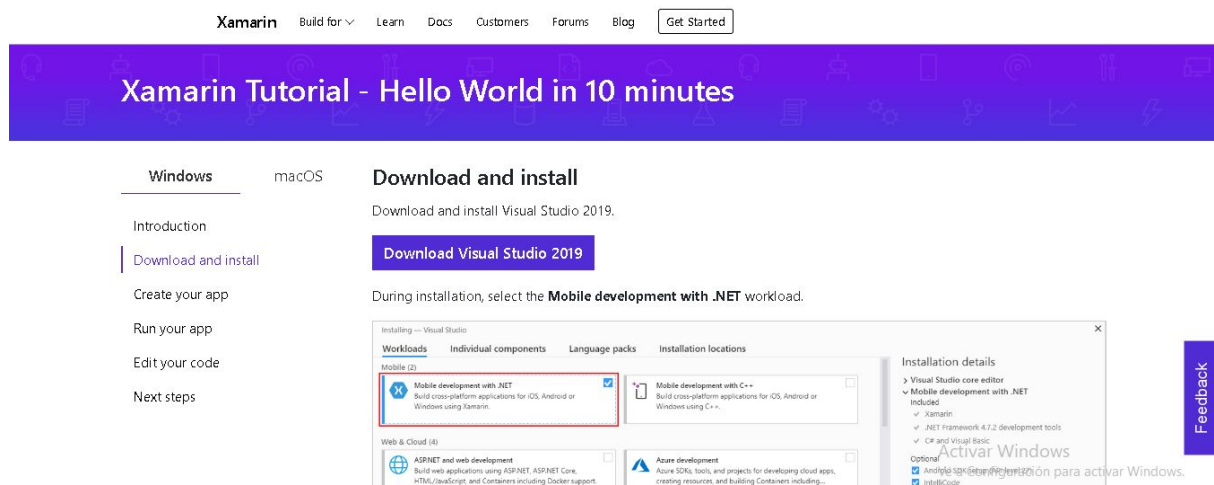
## **Herramientas que debes usar para crear una aplicación con Xamarin**

---

Antes de comenzar tu camino en el desarrollo de aplicaciones con C#, .NET y Xamarin es indispensable contar con las herramientas adecuadas para programar.

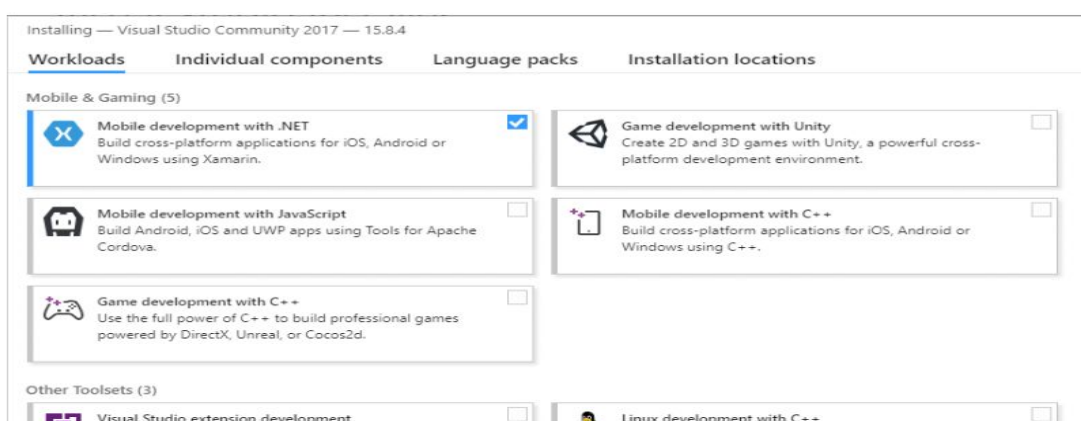
## Entorno de Desarrollo (IDE) Visual Studio:

- Actualmente si eres usuario del **sistema operativo Windows** debes instalar el entorno de desarrollo Visual Studio para programar una app. Puedes ocupar la versión Community o Professional.
- Si eres usuario de **Mac** tendrás que instalar Visual Studio for Mac y también Xcode.



- También podemos utilizar Xamarin Studio Es necesario descargar el IDE desde su página oficial, debiendo saber que su licencia no es gratis, pero podemos tener la opción de instalar la herramienta durante 29 días. Este IDE se puede instalar en tu PC o Mac.

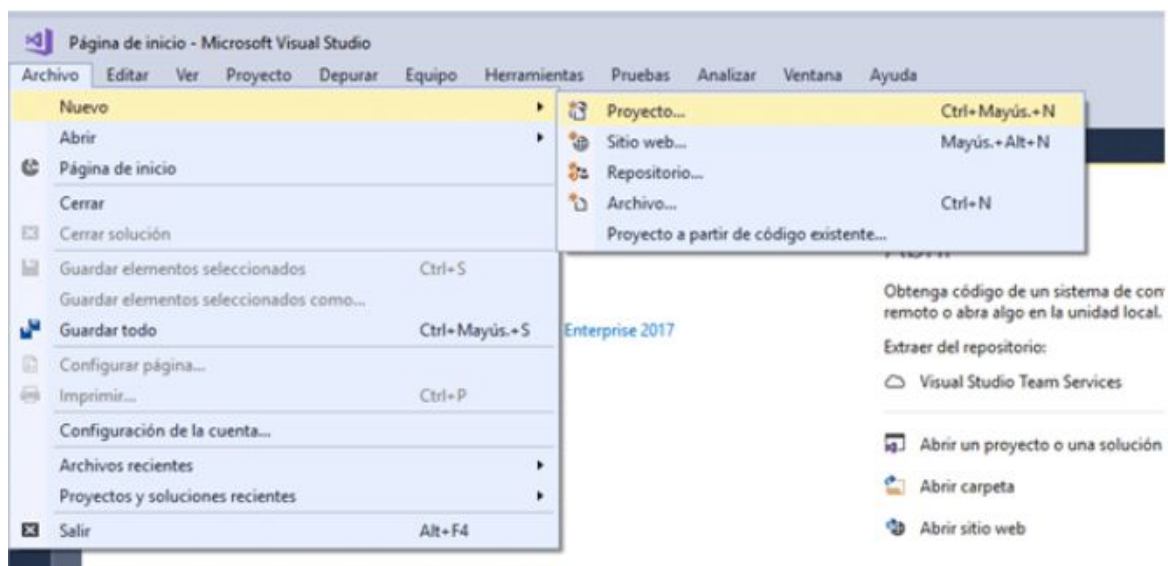
## Bibliotecas





- Además de instalar el IDE también tendrás que instalar algunas bibliotecas. Cuando inicies un proyecto en Visual Studio con Xamarin, el mismo programa te ayudará a instalar y hacer las configuraciones necesarias para comenzar a trabajar.

¿Cómo crear un proyecto con Xamarin.Forms?



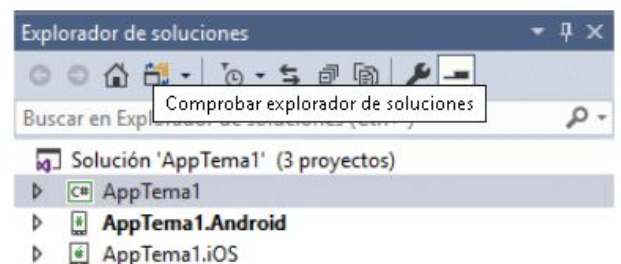
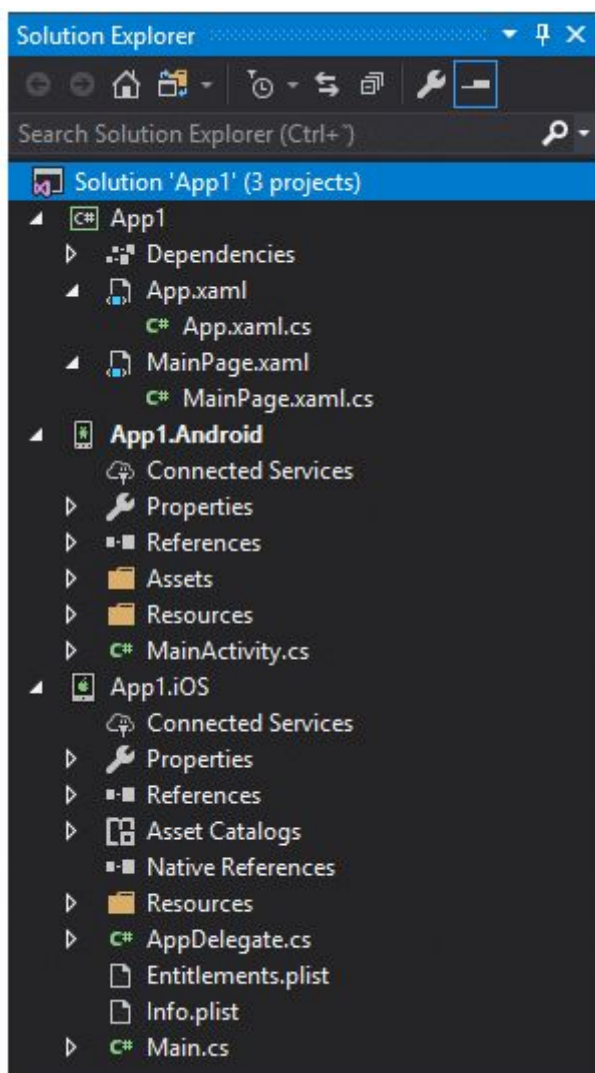
Deberemos elegir entre las diferentes plantillas disponibles a la hora de crear el proyecto. Para ello, seleccionaremos Visual C#, Cross-Platform y dentro Cross Platform App (Xamarin)

Llegados a este punto se abrirá un cuadro de diálogo en el que debemos elegir entre Aplicación en blanco y marcar .NET Standard.

La aplicación en blanco es una plantilla para generar proyectos en Xamarin.Forms, que generará el esqueleto necesario para realizar una aplicación de iOS y Android.

¿Cómo analizar la estructura del proyecto creado?

Si no estamos familiarizados con Visual Studio, lo primero que debemos saber es que se ha generado una Solución, que es el nombre con el que se denominan los proyectos de aplicaciones.



De forma general, la estructura es la siguiente:

- Proyecto de código común (.NET Standard o shared project según configuración)

Contiene las dependencias comunes, el SDK, la raíz de la aplicación App (App.xaml y App.xaml.cs), la página de la vista principal MainPage.xaml y todas aquellas que vayan a formar parte de la interfaz de usuario.

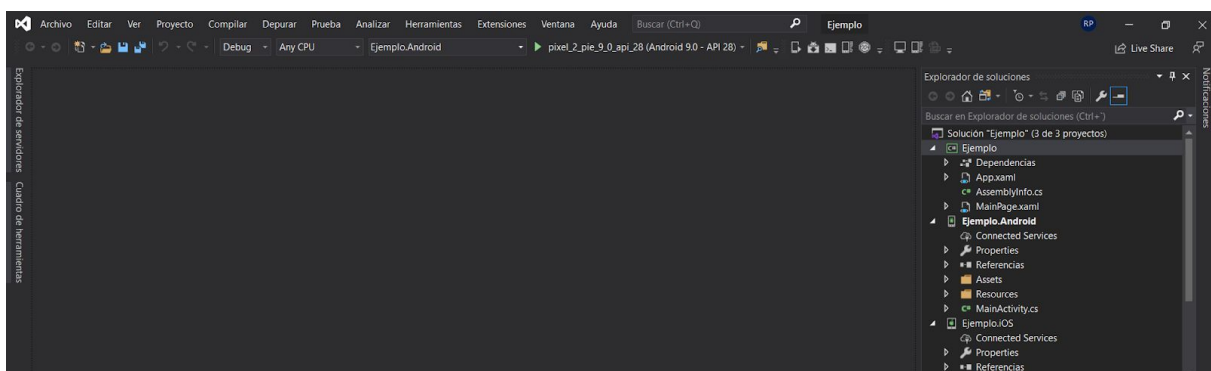
- **App de Android**

Contiene las librerías específicas necesarias para este sistema operativo, los directorios donde almacenar recursos específicos, tipos de letra, música, vídeo, imágenes, etc. y el archivo principal

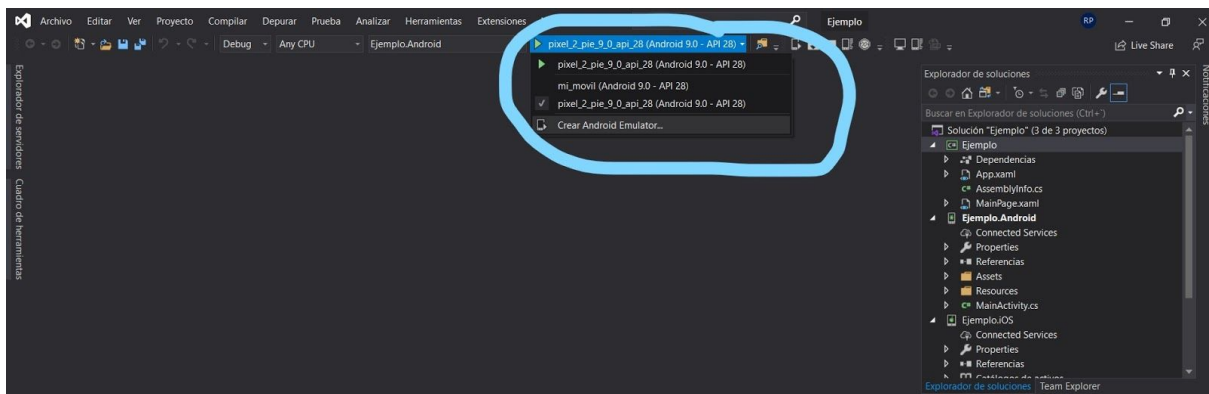
MainActivity.cs que contiene la instrucción Xamarin.Forms.Forms.Init() que inicializa los componentes necesarios para que todo funcione.

## Correcta Configuración del Emulador

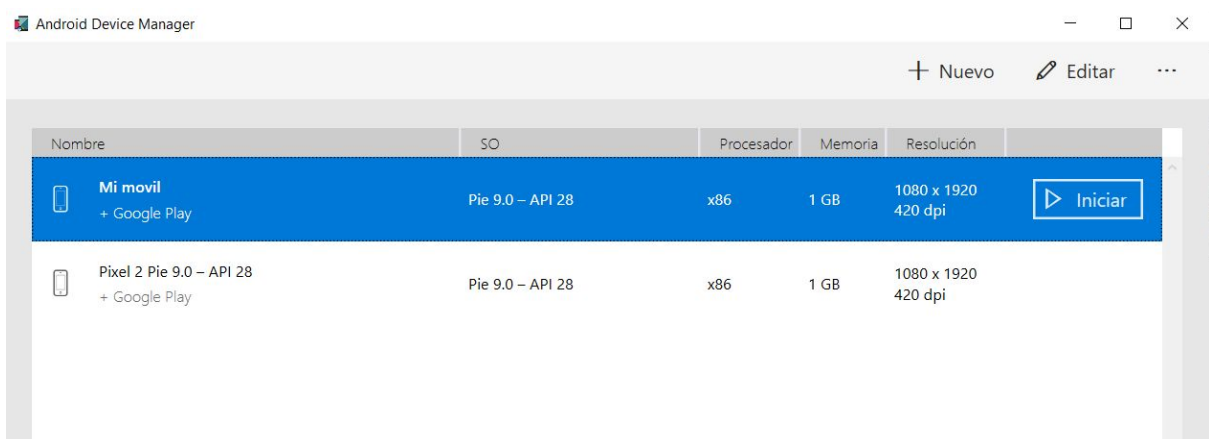
Una vez creamos un nuevo proyecto con xamarin.forms se nos verá el visual studio de la siguiente manera:



Por defecto aparece un emulador creado pero podemos crear uno nosotros con las opciones que queramos, para ello debemos hacer click en la flecha hacia abajo que hay al lado derecho del emulador.



A continuación le daremos a crear android emulador y se abrirá esta interfaz.



Una vez aquí le daremos a nuevo y se pondrá esta interfaz para editar el dispositivo y personalizarlo a nuestro gusto.

Propiedad	Valor
disk.dataPartition.size	2G
fastboot.forceColdBoot	<input type="checkbox"/>
fastboot.forceFastBoot	<input checked="" type="checkbox"/>
hw.accelerometer	<input checked="" type="checkbox"/>
hw.arc	<input type="checkbox"/>
hw.audioInput	<input checked="" type="checkbox"/>
hw.battery	<input checked="" type="checkbox"/>
hw.camera.back	virtualescene
hw.camera.front	emulated
hw.cpu.ncore	4
hw.dPad	<input type="checkbox"/>
hw.gps	<input checked="" type="checkbox"/>
hw.gpu.mode	auto
hw.keyboard	<input checked="" type="checkbox"/>
hw.lcd.density	420
hw.lcd.height	1920
hw.lcd.width	1080
hw.mainKeys	<input type="checkbox"/>
hw.ramSize	1536
hw.sdCard	<input checked="" type="checkbox"/>
hw.sensor.orientation	<input checked="" type="checkbox"/>

**disk.dataPartition.size**  
 Data partition size.  
 Default: 0  
 Specifies the size of the user data partition in bytes. If size is a simple integer, it specifies the size in bytes. You can also specify the size in kilobytes, megabytes, and gigabytes by appending K, M, or G to size. The minimum size is 9M and the maximum size is 1023G.

Para finalizar la creación le daremos a crear y ya estaría nuestro emulador de android.

## Conclusión

En definitiva, si lo que quieres es desarrollar una aplicación móvil que sirva para todas las plataformas y que no implique demasiados costes ni esfuerzos, Xamarin es tu opción

## GLOSARIO

**.NET:** Por .NET se hace referencia al conjunto de tecnologías de Microsoft destinadas al desarrollo de aplicaciones informáticas

Lo verdaderamente interesante es que estas tecnologías se basan en lo que se conoce como un **modelo de desarrollo unificado** podrán ser utilizadas tanto para el desarrollo de aplicaciones que se ejecuten en entorno de escritorio (un ordenador, generalmente con sistema operativo Windows), como para las que se vayan a ejecutar en entorno web, o incluso para aquellas que se vayan a ejecutar en un dispositivo móvil.

**XAML:** Es un lenguaje desarrollado por Microsoft que tiene como objetivo describir gráficamente las interfaces de usuario. Los archivos XAML son producidos por herramientas de diseño visual como Microsoft Blend.

**Aplicación Nativa:** Es el software desarrollado utilizando el lenguaje de programación oficial de un sistema operativo.

**Xamarin.forms:** Es un framework que se agrega en un proyecto de Xamarin para poder crear interfaces de usuario que aprovechen los lenguajes visuales nativos de cada sistema operativo.