

# Laboratorio

---

*¡Hello Xamarin!*

Versión: 1.0.0  
Octubre de 2016



[Miguel Muñoz Serafín](#)  
@msmdotnet



---

**CONTENIDO****INTRODUCCIÓN****EJERCICIO 1: INSTALANDO LA PLATAFORMA XAMARIN**

Tarea 1. Instalar Xamarin en la Mac.

Tarea 2. Instalar Xamarin en Windows.

Tarea 3. Habilitar el inicio de sesión remoto en la Mac.

Tarea 4. Conectar a la Mac

**EJERCICIO 2: CREANDO UNA APLICACIÓN iOS**

Tarea 1. Crear la aplicación.

**EJERCICIO 3: CREANDO UNA APLICACIÓN ANDROID**

Tarea 1. Crear la aplicación.

**EJERCICIO 4: CREANDO UNA APLICACIÓN WINDOWS**

Tarea 1. Crear la aplicación Windows 8.1/Windows Phone 8.1.

Tarea 2. Crear la aplicación de la Plataforma Universal de Windows.

**RESUMEN**

# Introducción

---

La instalación y configuración del hardware y software dependerá de las plataformas móviles sobre las cuales estemos pensando desarrollar. Además, dependerá de la plataforma de cómputo sobre la cual nos sintamos más cómodos, una Mac o una PC.

Los requerimientos de Xamarin.Forms no son diferentes de los requerimientos para utilizar Xamarin.iOS o Xamarin.Android o para programar para las plataformas Windows.

Si vamos a desarrollar para iPhone, necesitaremos una Mac. Apple requiere que una Mac sea utilizada para compilar aplicaciones iPhone y otras aplicaciones iOS. Necesitamos instalar Xcode en esa máquina y por supuesto, la plataforma Xamarin que incluye las bibliotecas necesarias y Xamarin Studio. Después de esto, podemos utilizar Xamarin Studio y Xamarin.Forms en la Mac para desarrollar aplicaciones iOS.

Una vez que ya tengamos una Mac con Xcode y la plataforma Xamarin instalada, podemos también instalar la plataforma Xamarin sobre una PC y desarrollar para el iPhone utilizando Visual Studio. La PC y la Mac deben estar conectadas a través de la Red. Visual Studio se comunica con la Mac a través de una interfaz SSH (Secure Shell) y utiliza la Mac para compilar la aplicación y ejecutar el programa sobre un dispositivo físico o simulador.

También es posible realizar programación Android en Xamarin Studio sobre la Mac o con Visual Studio sobre la PC.

Si deseamos desarrollar para las plataformas Windows, necesitaremos Visual Studio 2015. Podemos desarrollar para todas las plataformas en un único IDE ejecutando Visual Studio 2015 sobre una PC conectada a la Mac a través de la red. La otra opción es ejecutar Visual Studio en una máquina virtual sobre la Mac.

Es posible probar nuestras aplicaciones en dispositivos reales conectados a la computadora de desarrollo a través de un cable USB, o podemos probar nuestros programas con emuladores.

Existen ventajas y desventajas en cada enfoque. Un dispositivo real es esencial para probar una interacción táctil compleja o para probar los tiempos de arranque o respuesta de la aplicación. Sin embargo, los emuladores nos permiten ver como una aplicación se adapta a una variedad de tamaños y factores de forma.

Los emuladores de iPhone y iPad se ejecutan sobre la Mac. Sin embargo, debido a que las Mac de escritorio no tienen monitores táctiles, necesitamos utilizar un mouse o un TrackPad para simular los gestos táctiles. Los gestos táctiles, del TrackPad de la Mac no se trasladan hacia el emulador. También podemos conectar un iPhone real a la Mac, pero necesitamos provisionarlo como un dispositivo de desarrollo.

Históricamente, los emuladores de Android proporcionados por Google han tendido a ser lentos, aunque a menudo son extremadamente versátiles en la emulación de una amplia gama de dispositivos Android actuales. Afortunadamente, Visual Studio ahora tiene su propio emulador de Android que trabaja mucho mejor. También es muy fácil conectar un dispositivo Android real a la PC o a la Mac para realizar pruebas. Todo lo que necesitamos hacer es habilitar la depuración USB en el dispositivo.

Los emuladores de Windows Phone pueden manejar diferentes resoluciones de pantalla y también tienden a ejecutarse bastante bien, aunque consumen demasiada memoria. Si ejecutamos el emulador de Windows Phone sobre un monitor táctil, podemos utilizar los gestos táctiles sobre la pantalla del emulador. Conectar un Windows Phone real a la PC es bastante fácil, pero requiere que habilitemos el teléfono para desarrollo desde la sección de **Configuración**. Si queremos desbloquear más de un teléfono, necesitaremos una cuenta de desarrollador.

En este laboratorio, instalarás el software necesario para poder empezar a desarrollar aplicaciones multiplataforma utilizando Xamarin.

Después de haber realizado la instalación, tendrás la oportunidad de probar tu ambiente de desarrollo creando una aplicación para cada plataforma: iOS, Android y Windows.

## Objetivos

Al finalizar este laboratorio, los participantes serán capaces de:

- Configurar un ambiente de desarrollo para crear aplicaciones iOS, Android y Windows utilizando la plataforma Xamarin y Visual Studio.
- Crear una aplicación iOS simple desde Visual Studio.
- Crear una aplicación Android simple desde Visual Studio.
- Crear una aplicación Windows simple desde Visual Studio.

## Requisitos

Para la realización de este laboratorio es necesario contar con lo siguiente:

- Un equipo de desarrollo con sistema operativo Windows 10 y Visual Studio 2015 Community, Professional o Enterprise.
- Un equipo Mac con OS X Yosemite (10.10) o posteriores accesible por la PC a través de la red con la última versión de Xcode. Xcode puede descargarse desde la tienda de aplicaciones de Mac o desde el [Centro de desarrollo de iOS](#).

Tiempo estimado para completar este laboratorio: **120 minutos**.

# Ejercicio 1: Instalando la plataforma Xamarin

Antes de escribir aplicaciones para Xamarin.Forms, necesitamos instalar la plataforma Xamarin en la Mac, PC o ambas.

La siguiente tabla muestra el tipo de aplicaciones que podemos desarrollar dependiendo del sistema operativo e IDE que utilicemos.

Sistema Operativo	Mac con Mac OS X Yosemite (10.10) y posteriores.	PC con Windows	
Ambiente de desarrollo	Xamarin Studio	Visual Studio	Xamarin Studio
Xamarin.iOS	Si	Si (con conexión a una Mac para compilación y depuración)	No
Xamarin.Android	Si	Si	Si
Xamarin.Forms	iOS y Android únicamente	Android, Windows, Windows Phone (iOS con la computadora Mac)	Android únicamente

De la tabla anterior podemos concluir que, si deseamos desarrollar aplicaciones para las plataformas iOS, Android y Windows, lo recomendable es configurar un equipo de desarrollo PC con Windows y Visual Studio. Para compilar y depurar aplicaciones iOS, necesitaremos una Mac accesible por la PC a través de la Red.

Los productos Xamarin dependen de los SDKs de las plataformas Apple y Google para crear aplicaciones iOS y Android respectivamente.

En este ejercicio instalarás la plataforma Xamarin en la Mac e instalarás la plataforma Xamarin sobre Visual Studio 2015 en la PC con Windows 10.

## Tarea 1. Instalar Xamarin en la Mac.

La manera más simple de instalar Xamarin en la Mac es a través del instalador universal de Xamarin. El instalador de Xamarin automáticamente detecta, descarga e instala cualquier componente Xamarin requerido para completar la instalación.

Realiza los siguientes pasos para instalar la plataforma Xamarin sobre la Mac.

1. Desde el navegador Web de tu equipo Mac, accede al siguiente enlace:

<http://xamarin.com/download>.

Te será mostrada una pantalla similar a la siguiente.



Download Xamarin Studio Community.

Nice! You are about to download Xamarin Studio Community so you can write your apps entirely in C# and share the same code on iOS, Android, Windows, Mac and more.

Full name

Email

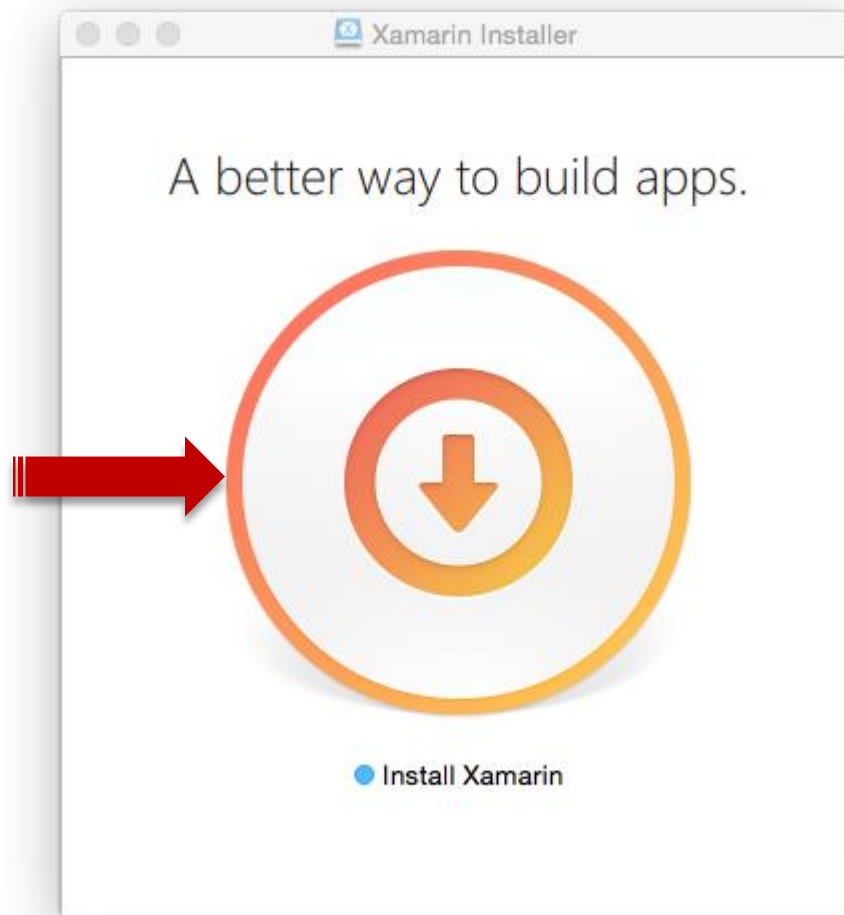
Company

☐ I agree to the [Terms of Use](#) and [Privacy Statement](#)

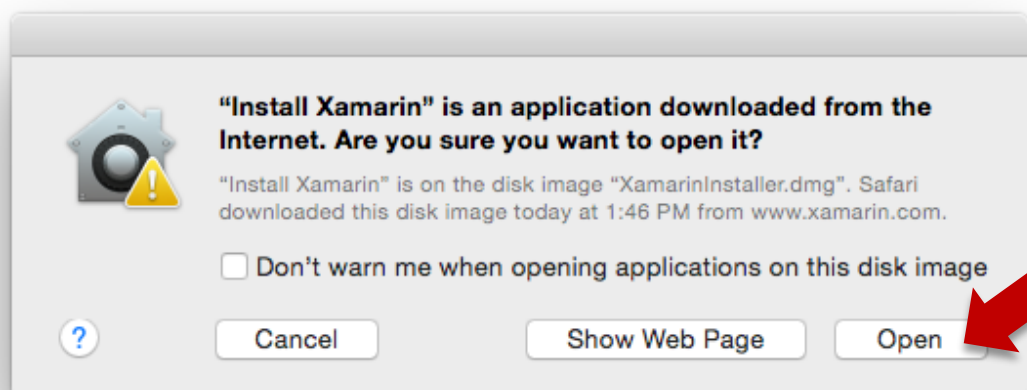
**Download Xamarin Studio Community**

[or Visual Studio Community with Xamarin for Windows](#)

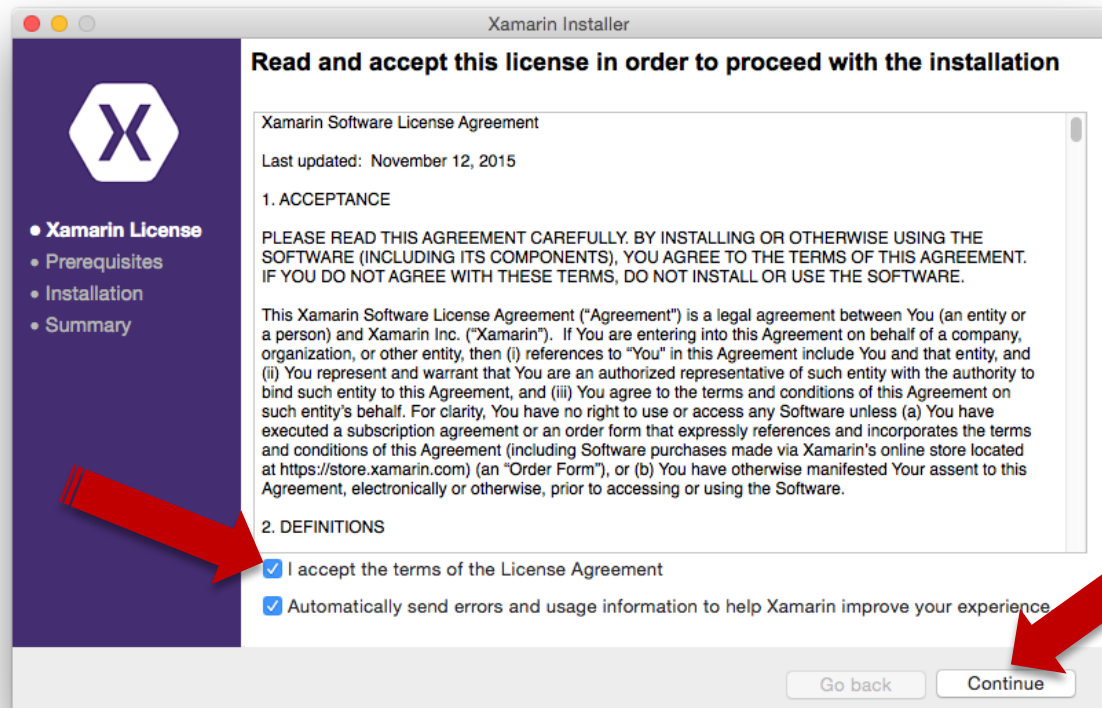
2. Llena el formulario con los datos solicitados y haz clic en el botón “**Download Xamarin Studio Community**” para descargar el instalador.
3. Una vez que el instalador haya sido descargado, ejecútalo dando doble clic sobre el logo del instalador.



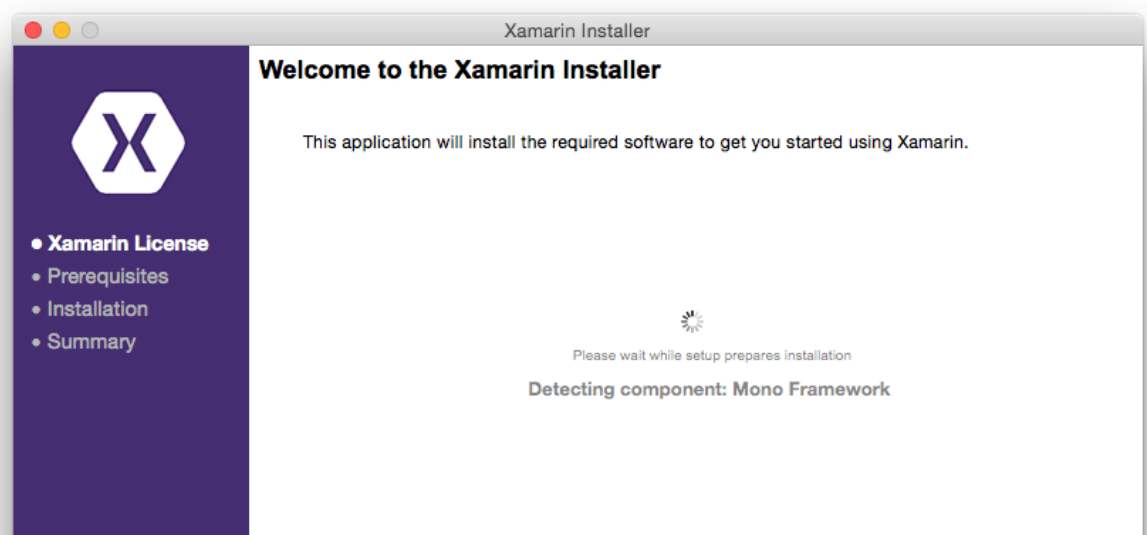
4. En el cuadro de diálogo de seguridad haz clic en el botón **Open** para permitir la ejecución de la aplicación.



5. El primer paso de la instalación requiere que revises y aceptes la licencia para poder continuar. haz clic en la casilla de verificación y presiona **Continue**.



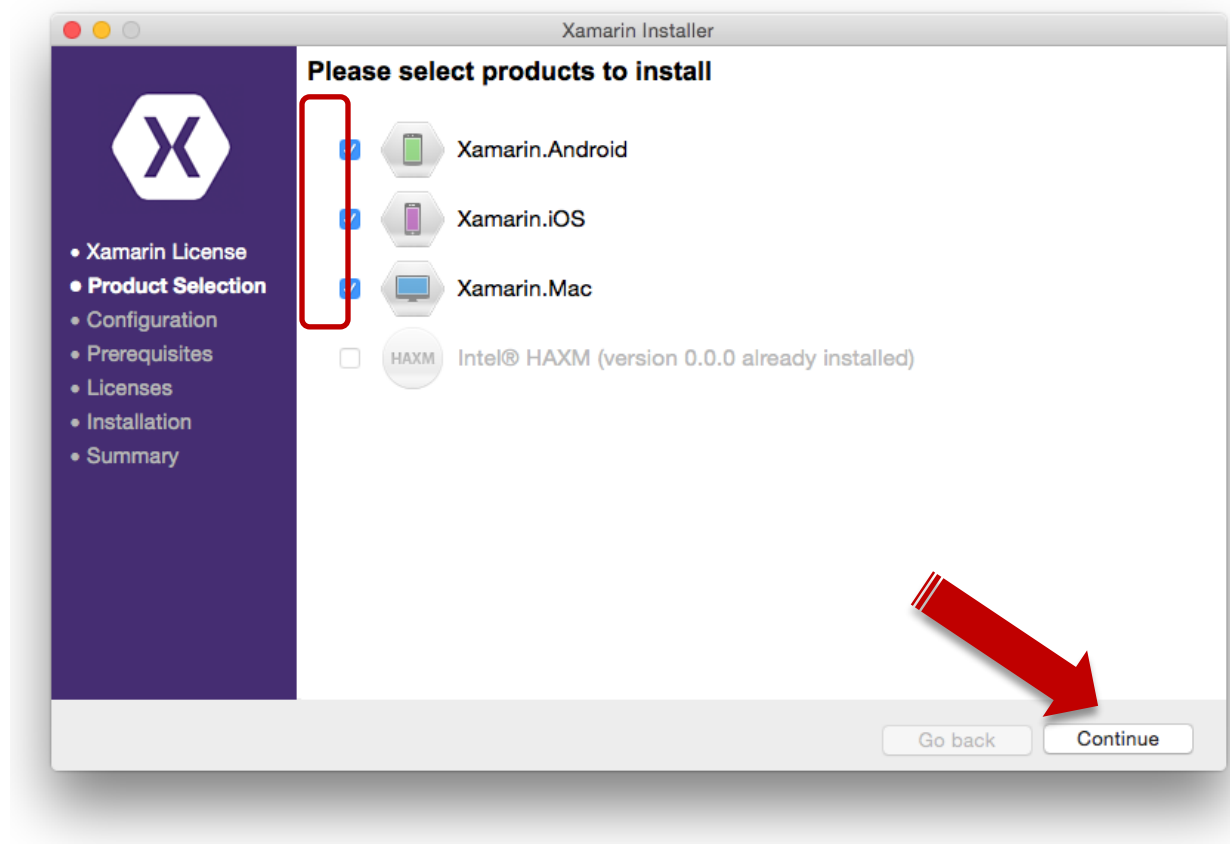
A continuación, el instalador inspeccionará el sistema para determinar los componentes requeridos que no están instalados y que necesitan ser descargados e instalados.





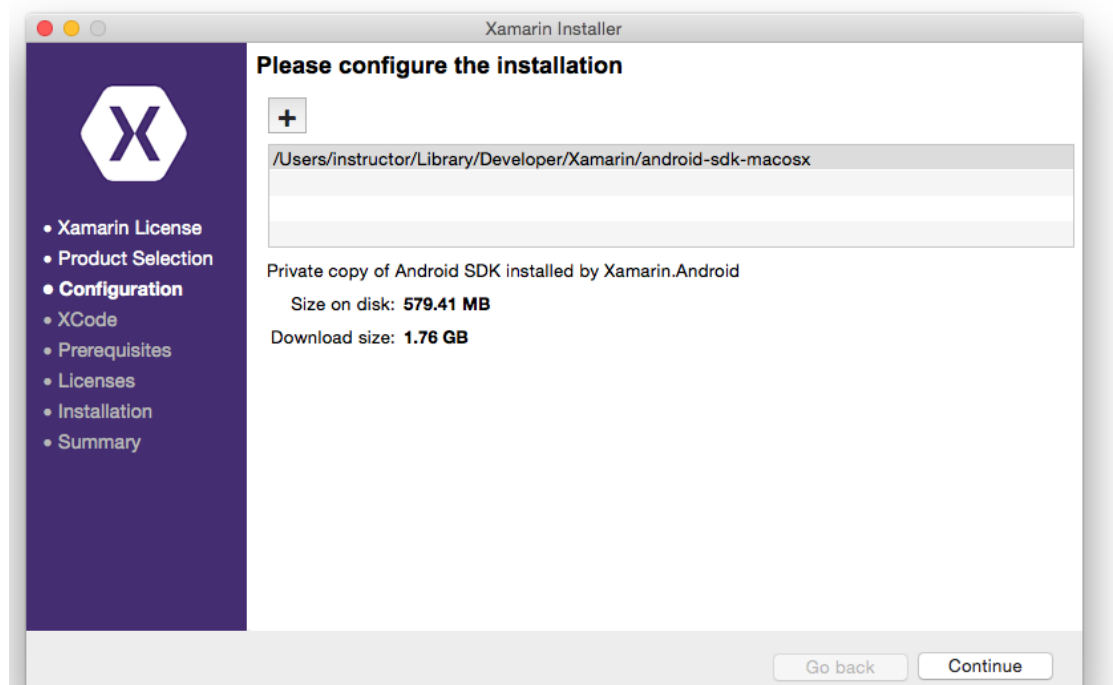
6. En la siguiente pantalla aparecerá la lista de los productos a descargar. Los productos que selecciones en esta pantalla dependerán de las plataformas para las cuales deseas desarrollar aplicaciones. Si únicamente deseas utilizar la Mac para compilar y depurar aplicaciones iOS desde Visual Studio, puedes seleccionar únicamente Xamarin.iOS.

Para este laboratorio, selecciona todos los productos disponibles y da clic en **Continue** para iniciar la descarga e instalación.

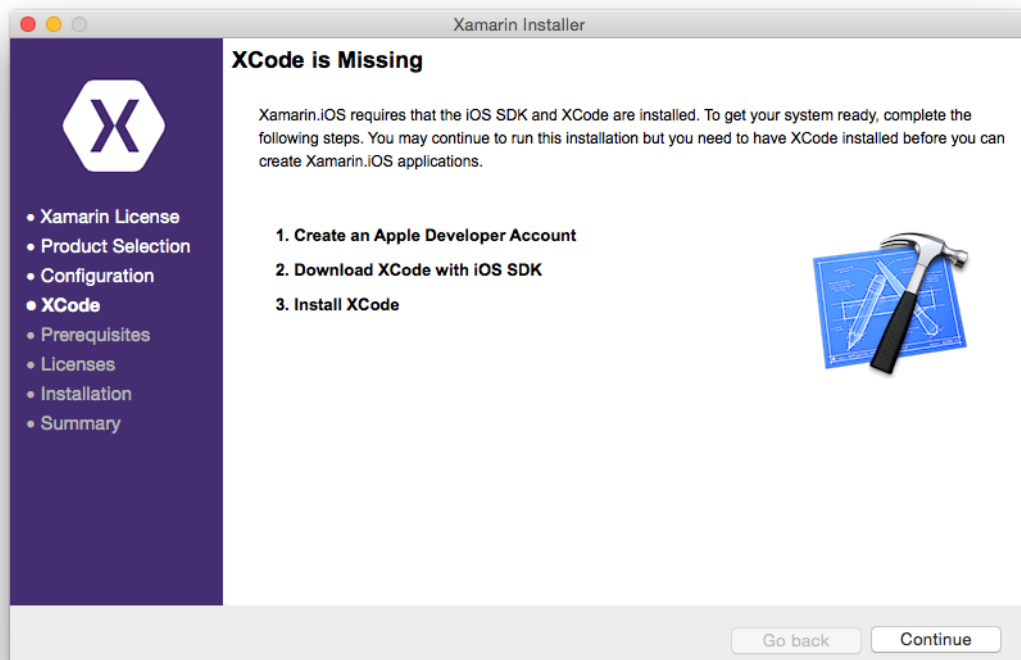


7. Después de identificar los componentes faltantes, el instalador de Xamarin descargará y ejecutará los instaladores de las dependencias de las plataformas seleccionadas.

Al instalar el SDK de Android te será mostrada una pantalla donde podrás seleccionar la ubicación de instalación. Haz clic en **Continue** para aceptar la ubicación sugerida.

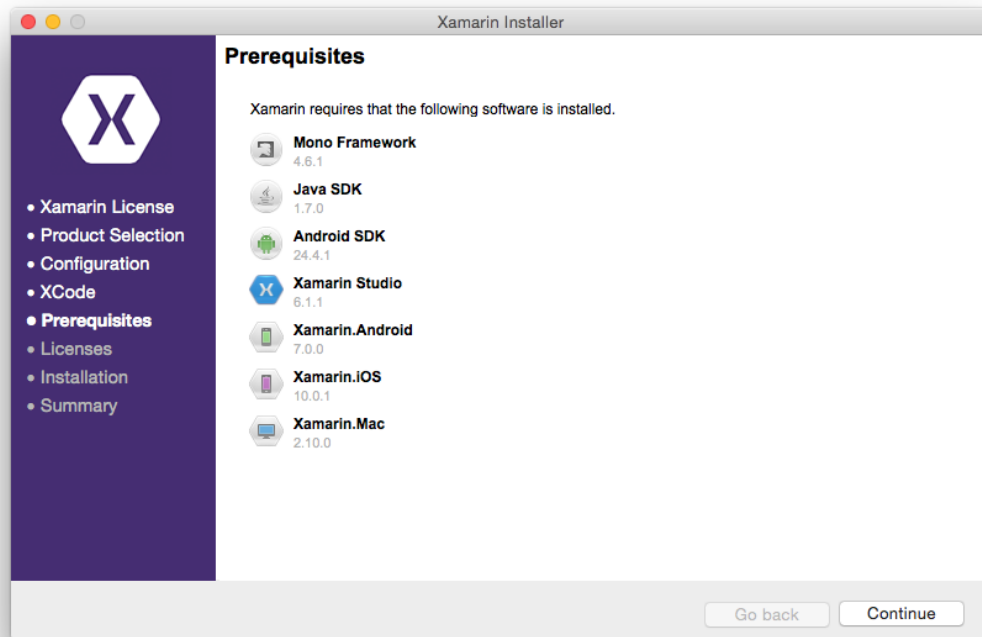


8. En caso de que no hayas instalado Xcode, te será mostrada una pantalla indicándote los pasos que debes seguir para instalar Xcode. Xamarin.iOS requiere que el SDK de iOS y Xcode estén instalados.

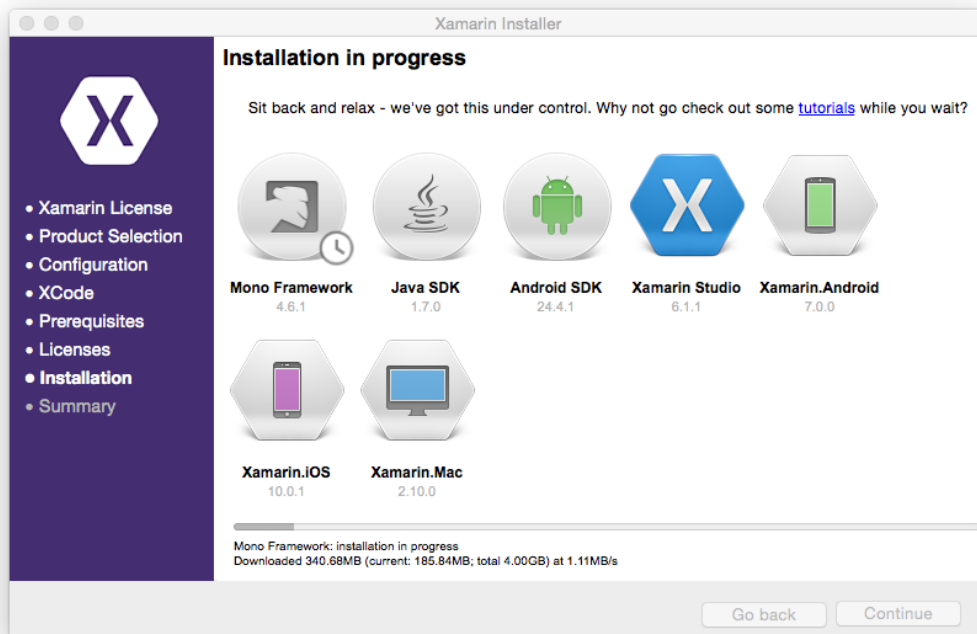


Haz clic en **Continue** para continuar con la instalación.

9. El instalador te mostrará los elementos que serán descargados e instalados. Haz clic en **Continue** para iniciar el proceso.

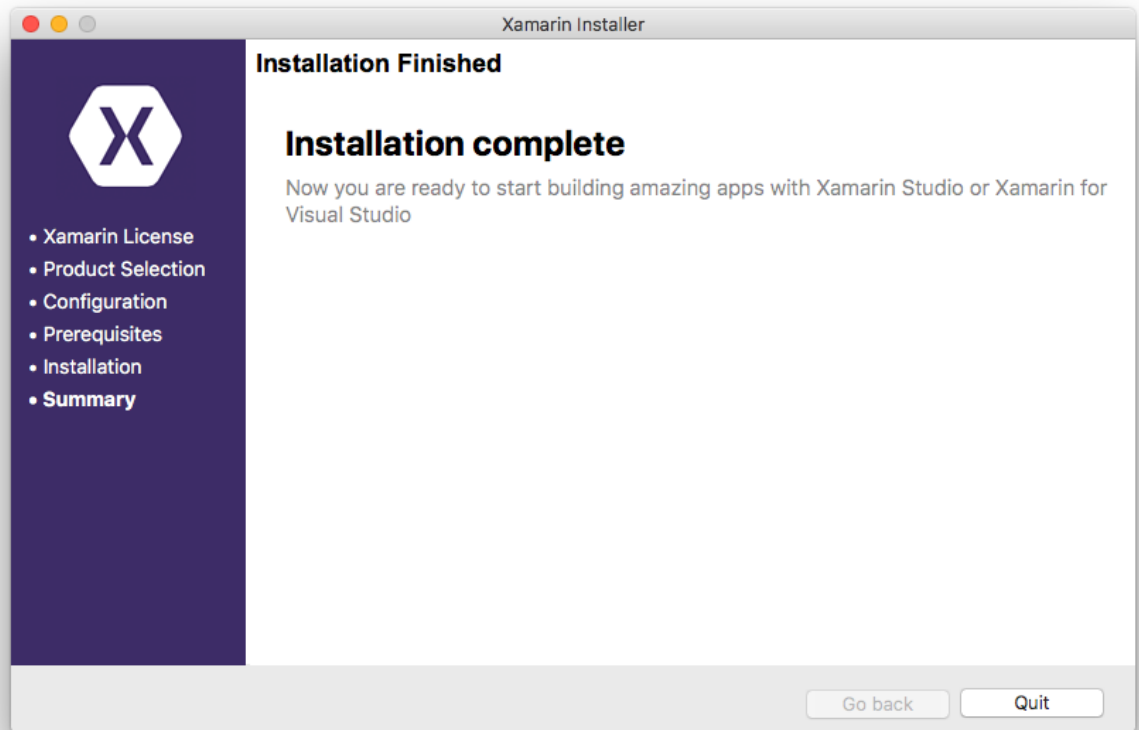


El instalador de Xamarin iniciará el proceso de descarga e instalación de los elementos seleccionados.



Es probable que, durante el proceso de instalación, te sea solicitado aceptar el acuerdo de licencias y elevar los permisos necesarios para componentes individuales que se requieren para completar la instalación.

10. Al finalizar la instalación, te será mostrada la pantalla siguiente.



11. Da clic en **Quit** para cerrar la ventana del instalador de Xamarin.

## Tarea 2. Instalar Xamarin en Windows.

Durante su instalación, Visual Studio 2015 nos permite instalar las herramientas Xamarin para Visual Studio.

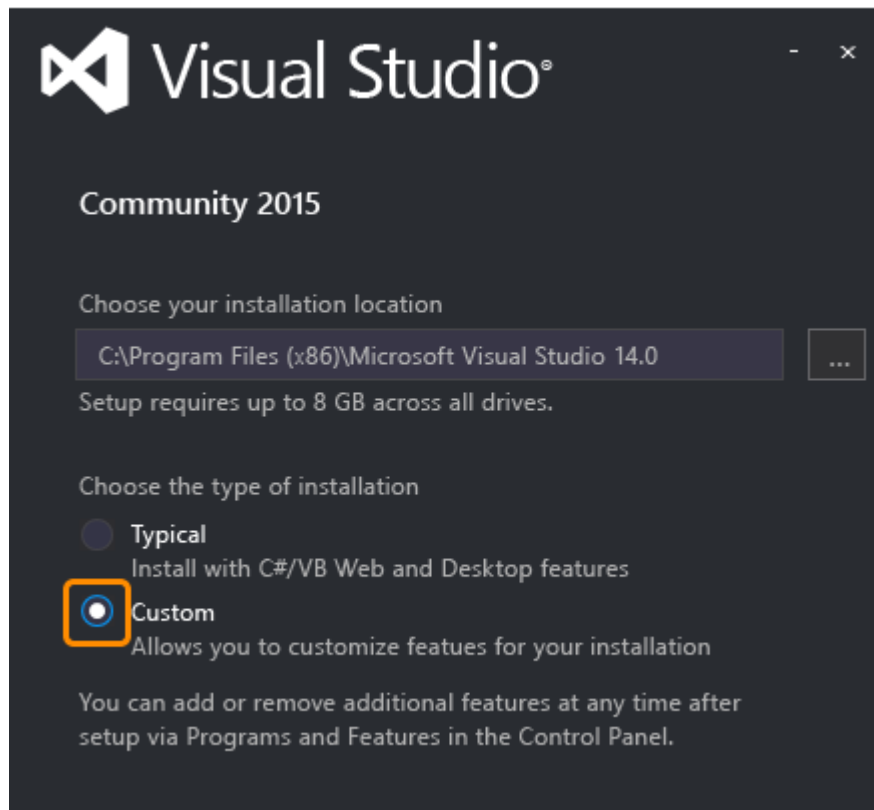
Realiza los siguientes pasos para instalar y configurar Xamarin en Visual Studio.

1. Si aún no tienes instalado Visual Studio, accede al siguiente enlace para descargar y ejecutar el instalador para cualquier edición de Visual Studio (Community, Professional o Enterprise).

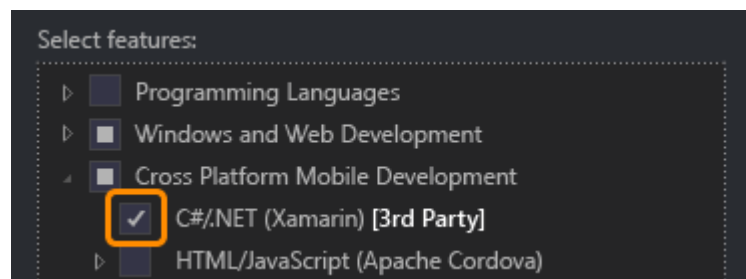
<https://www.visualstudio.com/es/downloads/>

Si ya tienes instalado Visual Studio, abre **Control Panel > Programs and Features**, selecciona el elemento **Visual Studio 2015** y da clic en la opción **Change**. Cuando el instalador se abra, da clic en **Modify** y continua con el paso 3 de esta tarea.

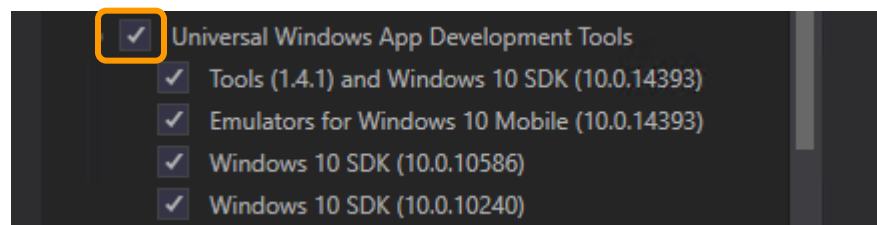
2. En el instalador, selecciona la opción **Custom**.



3. Selecciona las siguientes opciones.
- a. **Cross-Platform Mobile Development > C#/.NET (Xamarin)**. Esta opción seleccionará automáticamente varias herramientas Android en la sección **Common Tools and Software Development Kits**.

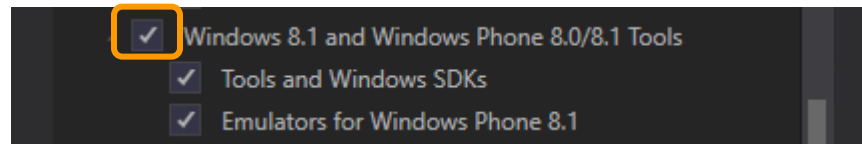


- b. **Windows and Web Development > Universal Windows App Development Tools**.



Al seleccionar esta opción podrás desarrollar aplicaciones para la Plataforma Universal de Windows.

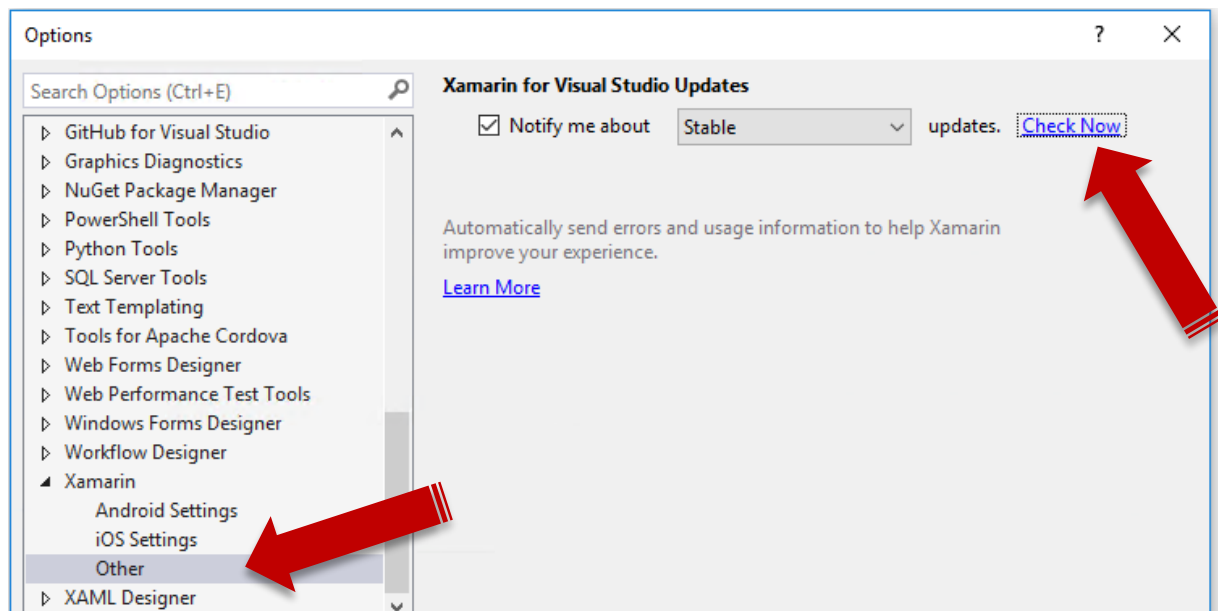
c. **Windows 8.1 and Windows Phone 8.0/8.1 Tools.**



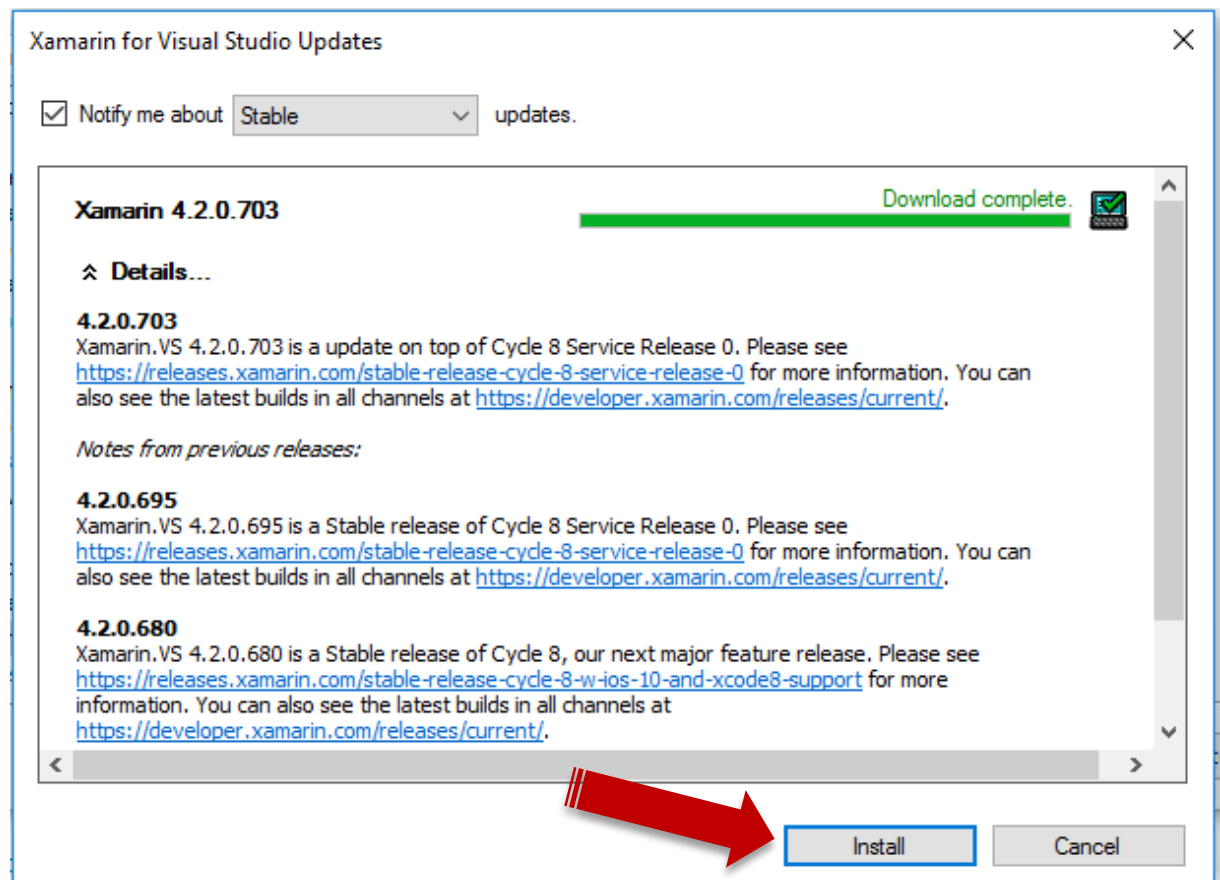
Al seleccionar esta opción podrás desarrollar aplicaciones Windows 8.1 y Windows Phone 8.1.

4. Haz clic en el botón correspondiente para iniciar el proceso de instalación.
5. Una vez que la instalación haya concluido, lanza Visual Studio e inicia sesión si te es requerido.
6. A partir de marzo 31 de 2016, Xamarin es incluido con todas las ediciones de Visual Studio de forma gratuita y no requiere de una licencia adicional. Las instalaciones existentes de Visual Studio que fueron configuradas con licencias anteriores de Xamarin, deben ser actualizadas a la versión 4.0.3.214 o posteriores.

Accede a la opción **Tools > Options > Xamarin > Other** y da clic en **Check Now** para descargar alguna actualización existente de **Xamarin**.



7. En caso de que existan actualizaciones, haz clic en **Install** y sigue las indicaciones para realizar la actualización.



### Tarea 3. Habilitar el inicio de sesión remoto en la Mac.

Xamarin.iOS para Visual Studio nos permite crear, compilar y depurar aplicaciones iOS sobre una computadora Windows utilizando el IDE Visual Studio. Sin embargo, esto no lo puede hacer únicamente el IDE por sí mismo, las aplicaciones iOS no pueden ser creadas sin el compilador de Apple y no pueden ser desplegadas sin los certificados y herramientas para firma de código de Apple. Esto significa que la instalación de Xamarin.iOS para Visual Studio requiere una conexión hacia una computadora Mac OS X en la red que realice esas tareas por nosotros.

Visual Studio se conecta a la Mac sobre SSH lo cual proporciona los siguientes beneficios:

- Visual Studio puede lanzar y controlar directamente al agente de compilación.
- El nuevo administrador de conexión de Visual Studio descubrirá, autenticará y recordará la Mac.
- Debido a que toda la comunicación es realizada de forma segura mediante SSH, únicamente una sola conexión al puerto 22 es requerida.

- Visual Studio es notificado de los cambios tan pronto ocurren. Por ejemplo, cuando un dispositivo iOS es conectado, la barra de herramientas se actualizará inmediatamente para indicar la presencia del dispositivo.
- Múltiples instancias de Visual Studio pueden conectarse de forma simultánea.
- La conexión no interrumpirá el desarrollo. Solo preguntará por la conexión a la Mac cuando estemos realizando una operación en la que sea requerida la Mac, por ejemplo, para depuración o para utilizar el diseñador de iOS.

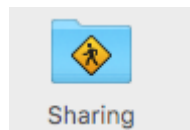
Para configurar un host Mac, debemos habilitar la comunicación entre la extensión Xamarin para Visual Studio y la Mac. Para hacer esto, necesitamos habilitar la opción **Remote Login** en la Mac.

Realiza los siguientes pasos para habilitar la opción **Remote Login** en la Mac.

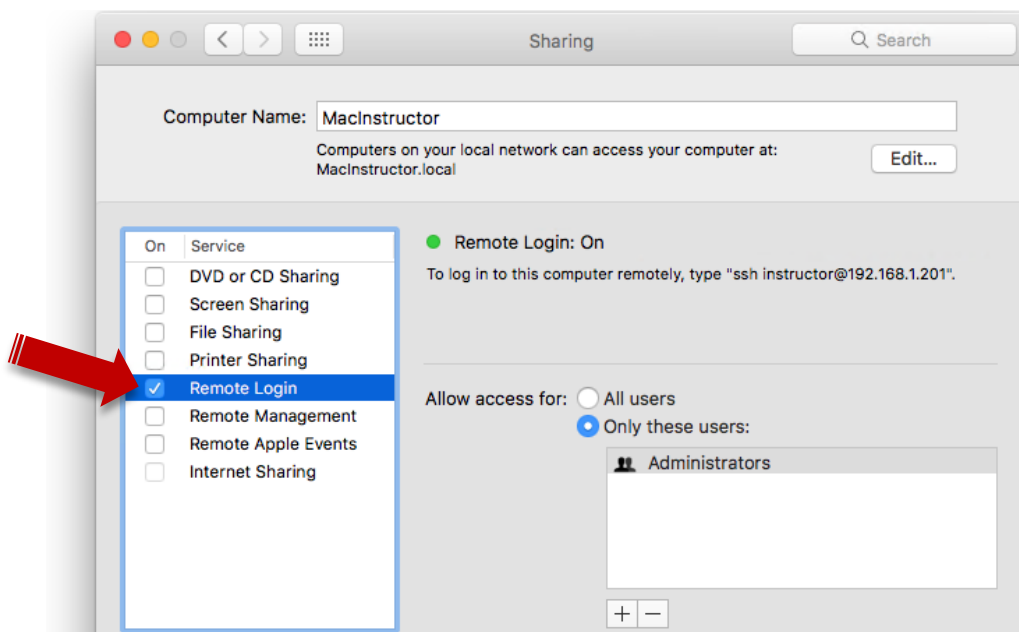
1. Abre la opción **System Preferences** en la Mac.



2. En la pantalla **System Preferences**, abre la opción **Sharing**.



3. En la lista de servicios de la pantalla **Sharing** selecciona la opción **Remote Login**.





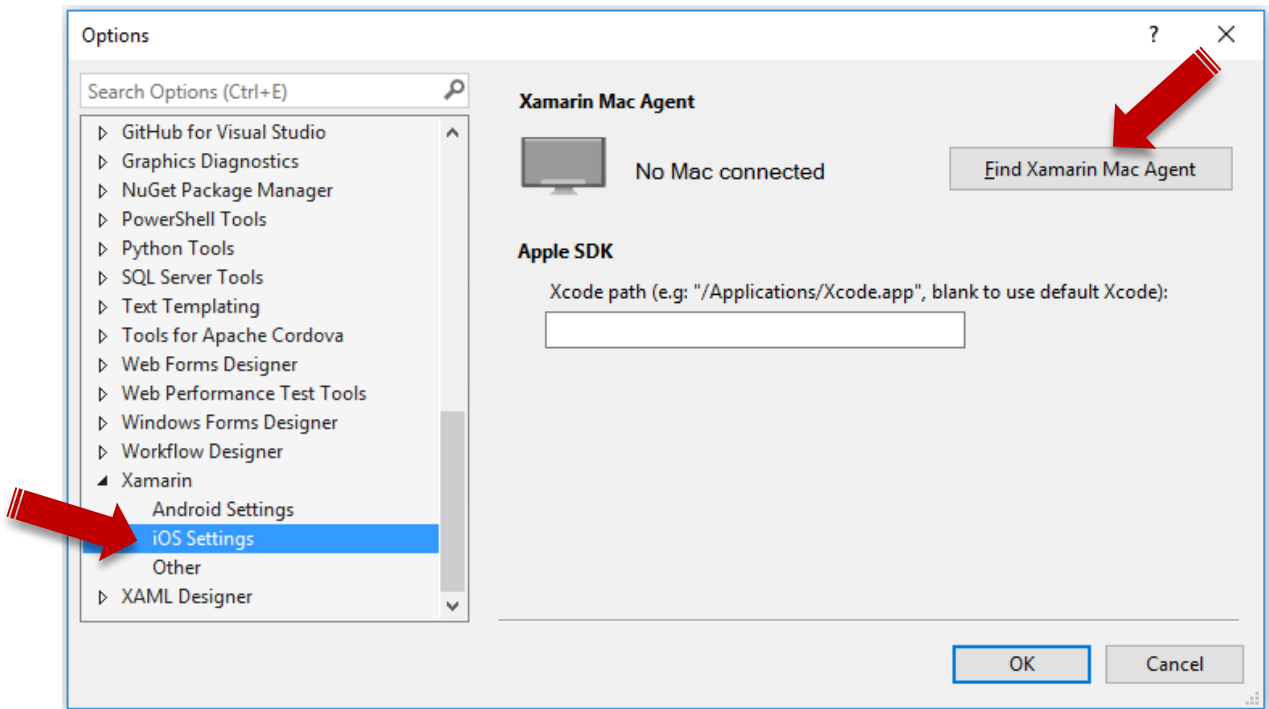
4. Asegúrate de la opción **Remote Login** esté permitida para todos los usuarios o que tu usuario o grupo Mac esté incluido en la lista de usuarios permitidos en la lista de la derecha.

Si tienes el firewall de OS X configurado para bloquear aplicaciones firmadas de manera predeterminada, deberás permitir al proceso **mono-sgen** recibir conexiones entrantes. Un dialogo de alerta aparecerá preguntándote cuando este sea el caso.

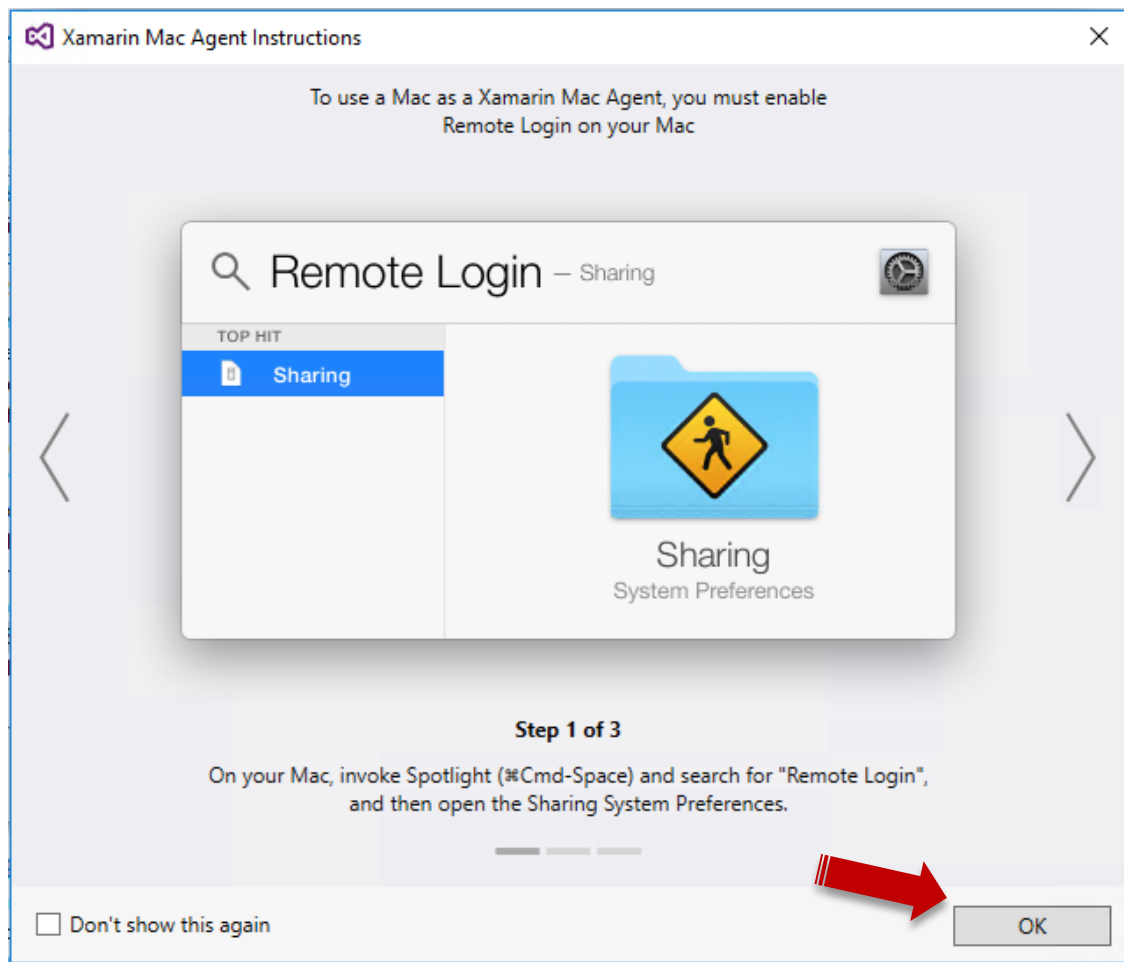
#### Tarea 4. Conectar a la Mac.

Realiza los siguientes pasos para conectar el IDE Visual Studio a la computadora Mac.

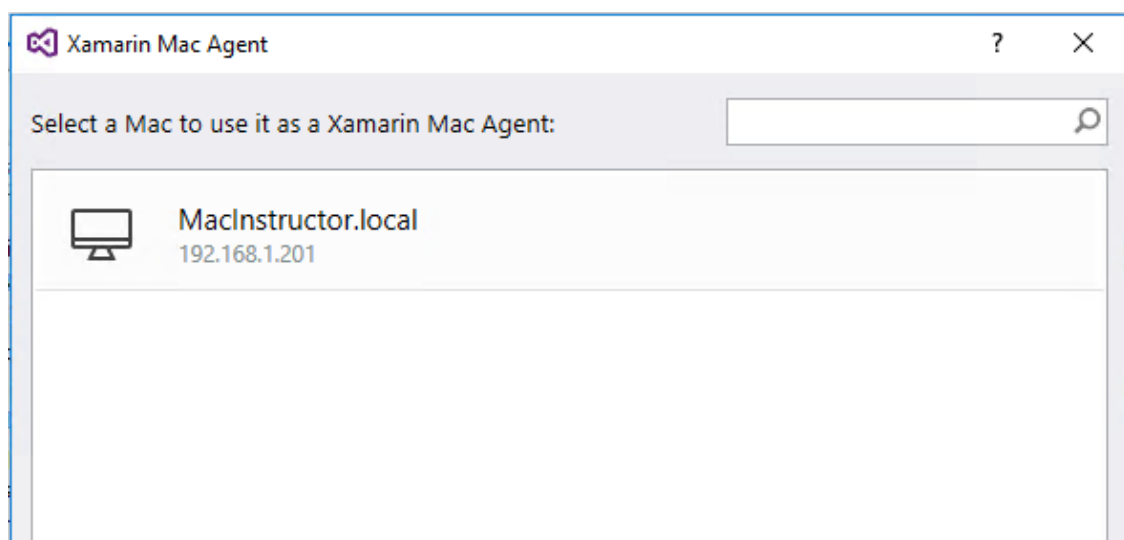
1. Desde Visual Studio accede a **Tools > Options**.
2. En la ventana **Options** selecciona **Xamarin > iOS Settings** y haz clic en el botón **Find Xamarin Mac Agent** para buscar la computadora Mac.



3. En caso de que te sea mostrada la ventana **Xamarin Mac Agent Instructions**, haz clic en el botón **OK** para cerrarla. Esta ventana te proporciona información para habilitar el inicio de sesión remota en la Mac. Este paso lo realizaste en la tarea anterior.



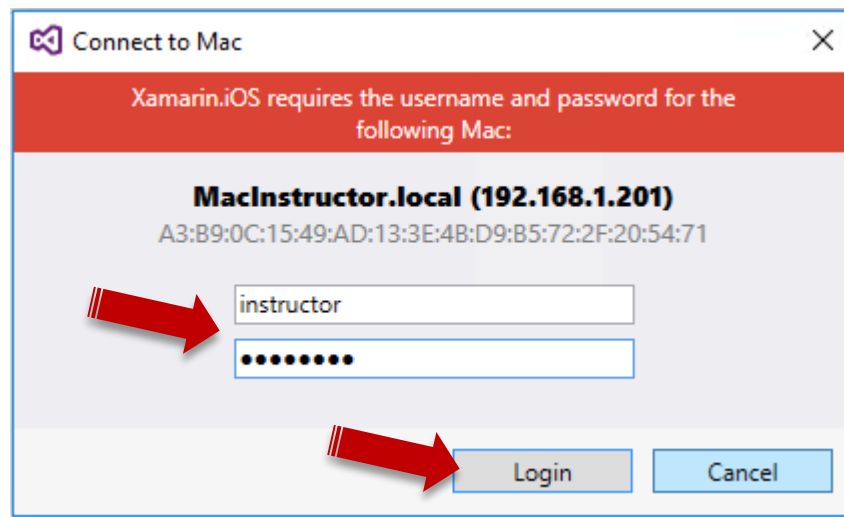
4. La ventana **Xamarin Mac Agent** te mostrará las computadoras Mac encontradas en la red.



5. Haz doble clic sobre la Mac para conectarte a ella.

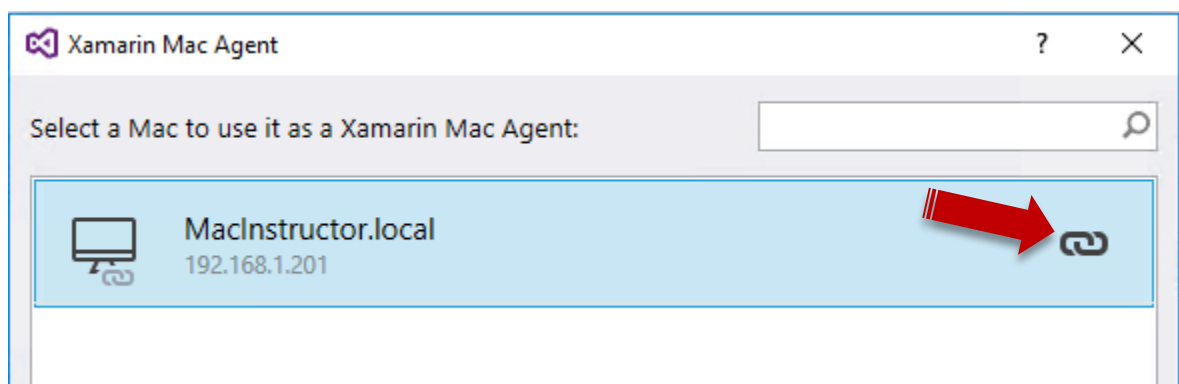
6. La primera vez que te conectes a una Mac, te serán solicitadas tus credenciales. Las credenciales que proporcionas deben ser de un usuario configurado en la Mac con derecho de inicio de sesión remota.

Proporciona tus credenciales en la ventana de dialogo **Connect to Mac** y haz clic en el botón **Log In** para iniciar sesión.

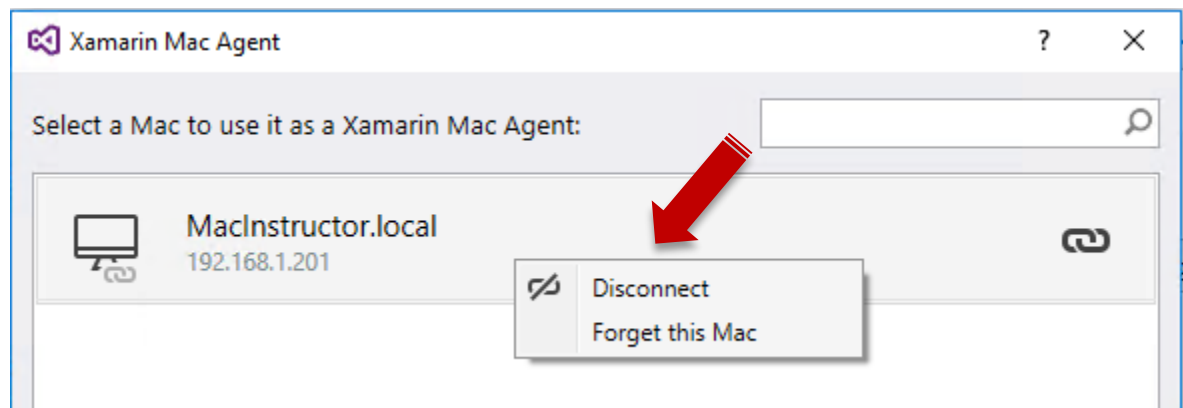


El agente utilizará esas credenciales para crear una nueva conexión SSH hacia la Mac. Si la conexión tiene éxito, una llave SSH será creada y será registrada en el archivo **authorized\_keys** en la Mac. En conexiones subsecuentes, el agente utilizará el nombre de usuario y el archivo con la llave para conectarse al host de compilación al que nos hayamos conectado más recientemente.

Cuando la conexión haya sido exitosa, la Mac será mostrada con un icono de conexión a la derecha.



Cada Mac en la lista tiene un menú contextual que nos permite conectarnos, desconectarnos u olvidar los datos de conexión.



7. Haz clic en el botón **Close** para cerrar la ventana **Xamarin Mac Agent**.
8. Haz clic en **OK** para cerrar la ventana **Options**.

Es probable que ya tengas ansias de crear tu primera aplicación Xamarin.Forms, pero antes de que lo hagas, deberás intentar crear proyectos Xamarin normales para iPhone, Android, Windows, Windows Phone y Windows Mobile.

Si estás experimentando problemas utilizando Xamarin.iOS, Xamarin.Android o Windows, ese no es problema de Xamarin.Forms. Primero necesitarás resolver esos problemas antes de utilizar Xamarin.Forms.

## Ejercicio 2: Creando una aplicación iOS

Si tienes interés en utilizar Xamarin.Forms para desarrollar aplicaciones iPhone, es recomendable que primero te familiarices con los documentos introductorios desde el sitio de Xamarin.

[https://developer.xamarin.com/guides/ios/getting\\_started/](https://developer.xamarin.com/guides/ios/getting_started/)

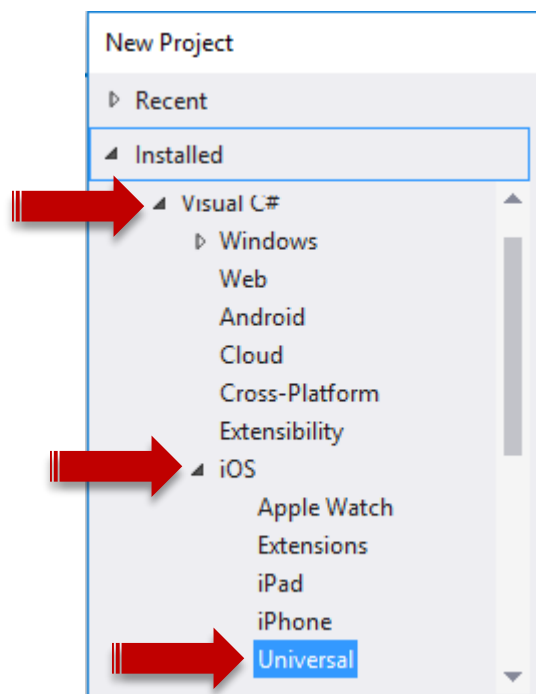
Esto te proporcionará una guía en el uso de la biblioteca Xamarin.iOS para desarrollar una aplicación iPhone en C#.

En este ejercicio crearás una aplicación iPhone sencilla que será desplegada en un simulador de iPhone.

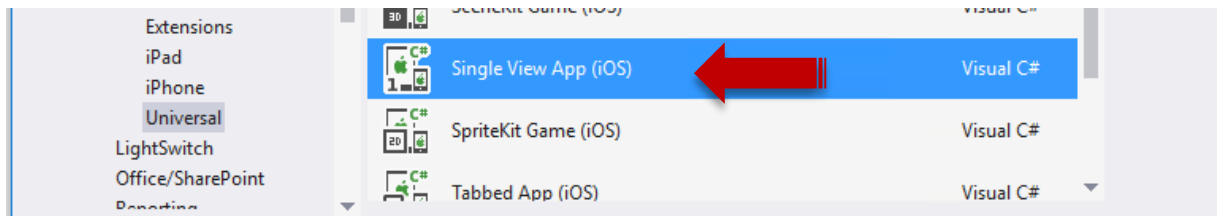
### Tarea 1. Crear la aplicación.

Realiza los siguientes pasos para crear una aplicación iOS desde Visual Studio.

1. Selecciona **File > New > Project** desde Visual Studio.
2. En el panel izquierdo de la ventana **New Project** selecciona **Visual C# > iOS > Universal** para indicar que deseas crear una aplicación para iPhone y iPad.



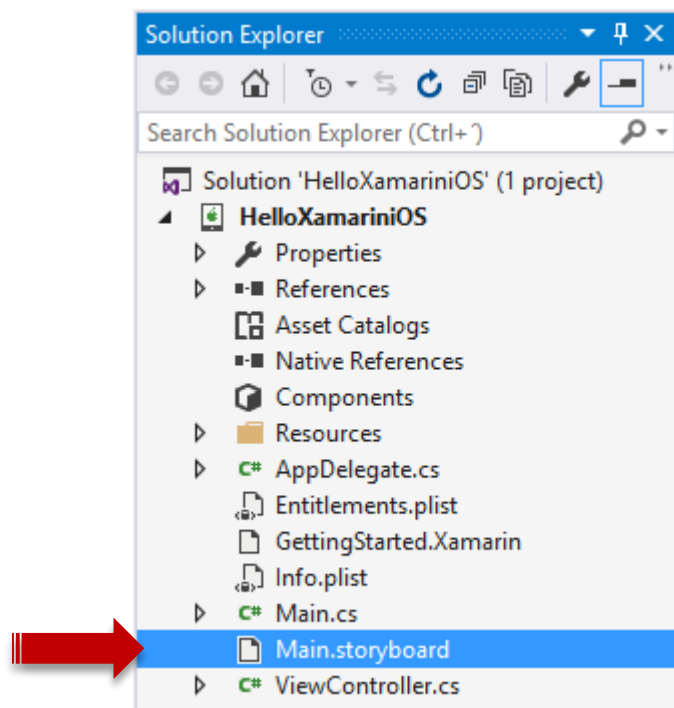
3. En el panel derecho, selecciona la plantilla **Single View App (iOS)**.



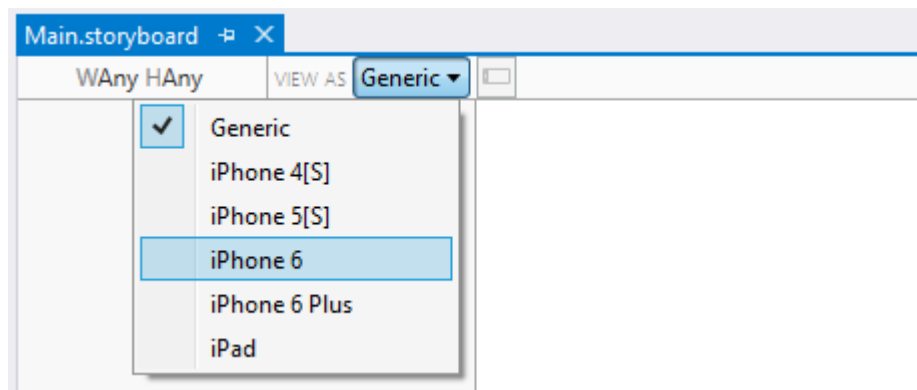
4. Proporciona el nombre, ubicación y haz clic en **OK** para crear el proyecto.



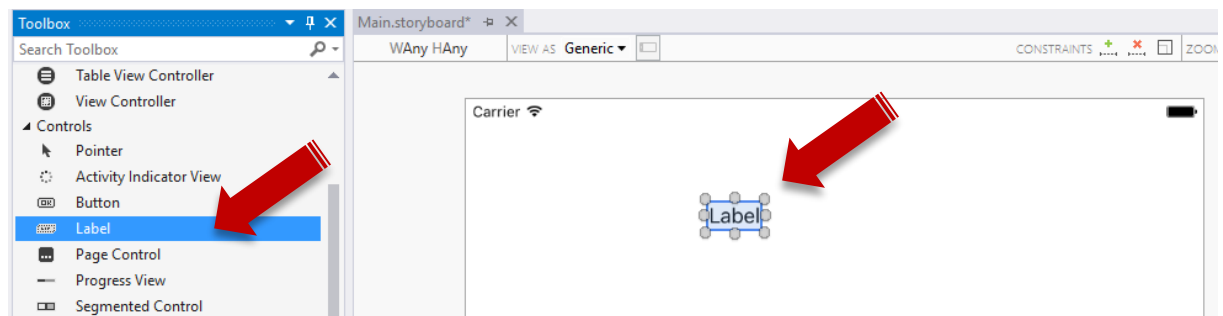
5. Haz doble clic sobre el archivo **Main.storyboard** desde la ventana **Solution Explorer**.



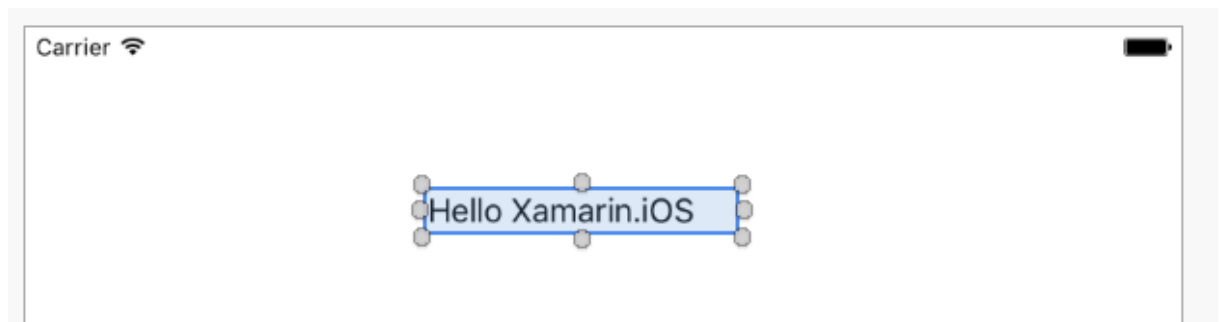
6. Selecciona la opción **iPhone 6** desde **View As** en la barra de herramientas del diseñador.



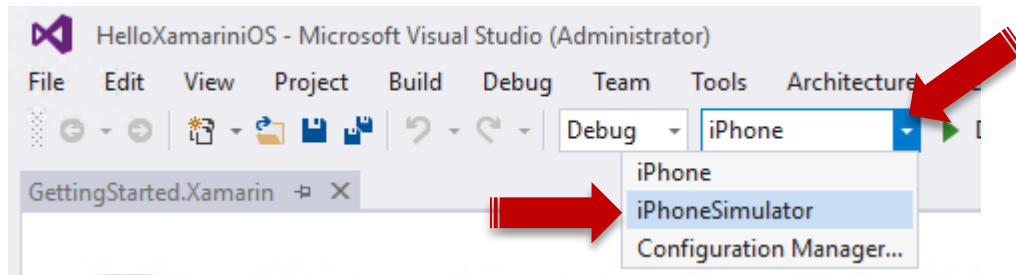
7. Selecciona la opción **View > Toolbox** para mostrar la caja de herramientas.
8. Arrastra el control **Label** hacia el área de diseño.



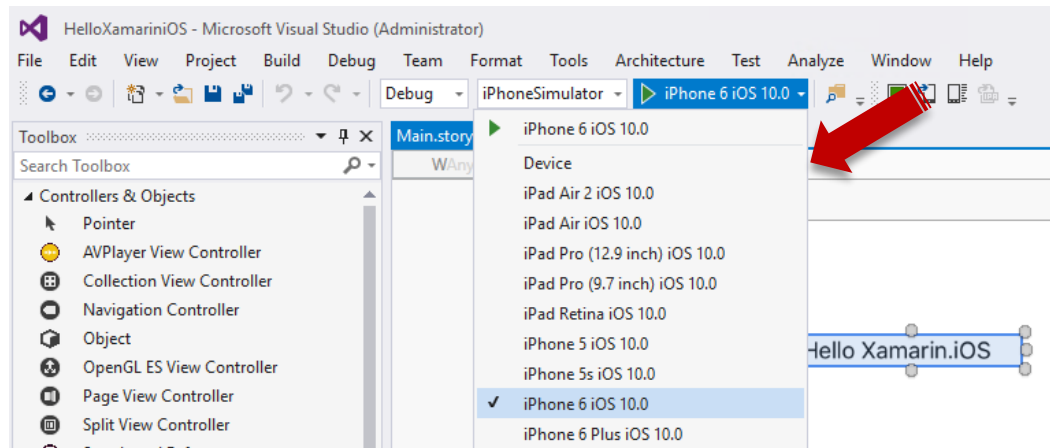
9. Utiliza la ventana de propiedades para cambiar la propiedad **Text** del control **Label** por el texto **"Hello Xamarin iOS"**. También puedes editar directamente el texto dentro del control **Label** dando doble clic sobre el control.



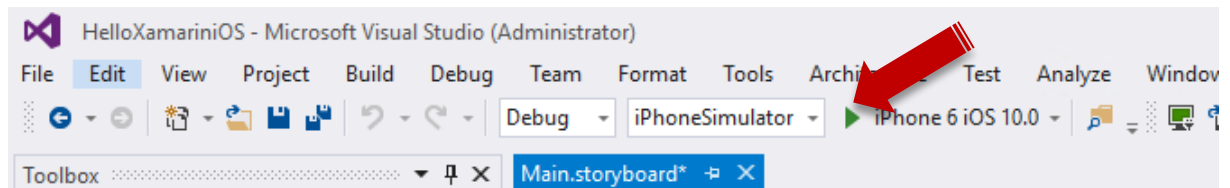
10. Selecciona la opción **iPhone Simulator** para indicar que deseas desplegar la aplicación iOS en el simulador de iPhone.



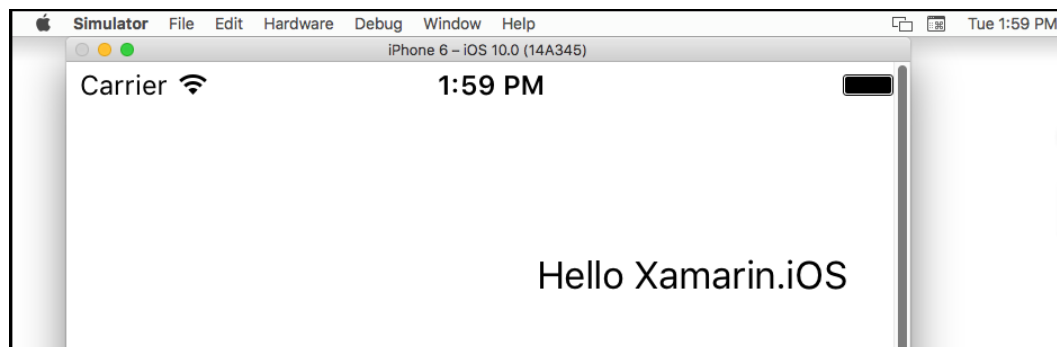
11. En la lista de simuladores selecciona alguno donde deseas desplegar la aplicación.



12. Haz clic en **Start** para desplegar la aplicación en el simulador seleccionado.



Si todo funciona correctamente, la Mac mostrará la aplicación en el simulador como en la siguiente imagen.



13. Regresa a Visual Studio y detén la ejecución.



# Ejercicio 3: Creando una aplicación Android

Si tienes interés en utilizar Xamarin.Forms para desarrollar aplicaciones para dispositivos Android, es recomendable que primero te familiarices con los documentos introductorios desde el sitio de Xamarin.

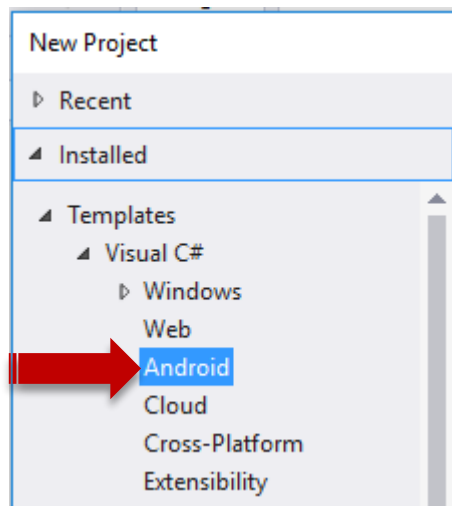
[https://developer.xamarin.com/guides/android/getting\\_started/](https://developer.xamarin.com/guides/android/getting_started/)

En este ejercicio crearás una aplicación Android sencilla que será desplegada en un simulador de Android.

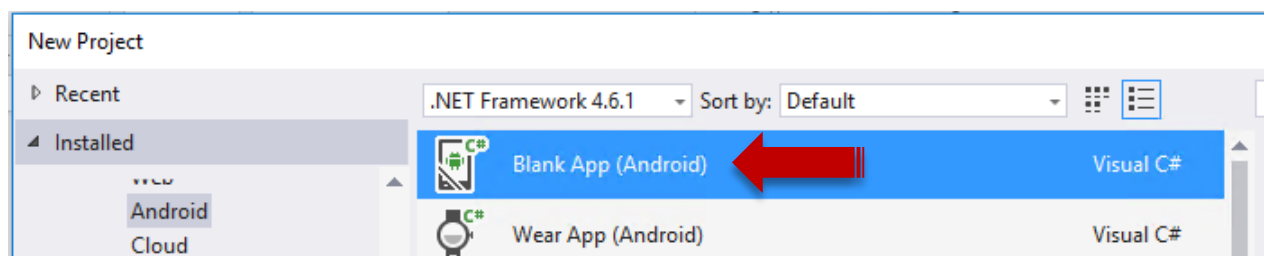
## Tarea 1. Crear la aplicación.

Realiza los siguientes pasos para crear una aplicación Android desde Visual Studio.

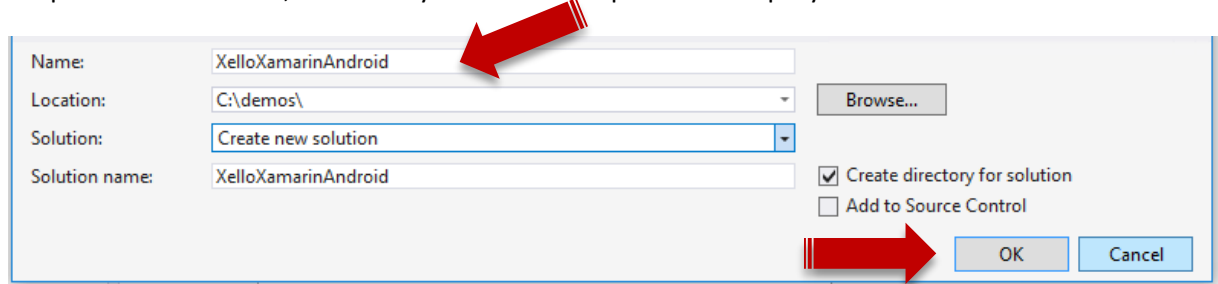
1. Selecciona **File > New > Project** desde Visual Studio.
2. En la ventana **New Project**, selecciona **Visual C# > Android**.



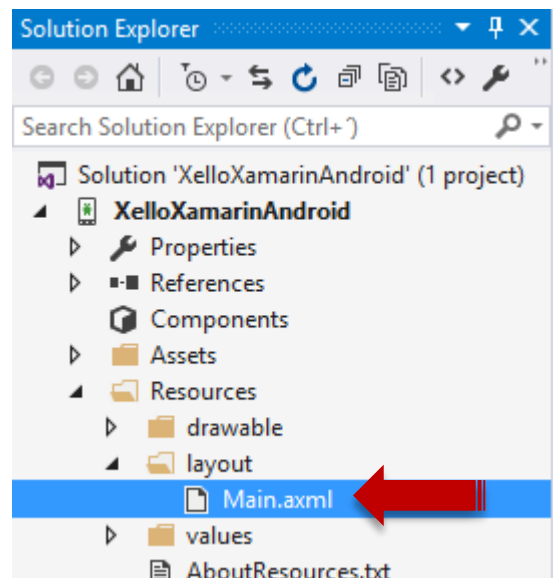
3. En el panel derecho, selecciona la plantilla **Blank App (Android)**.



4. Proporciona el nombre, ubicación y haz clic en **OK** para crear el proyecto.



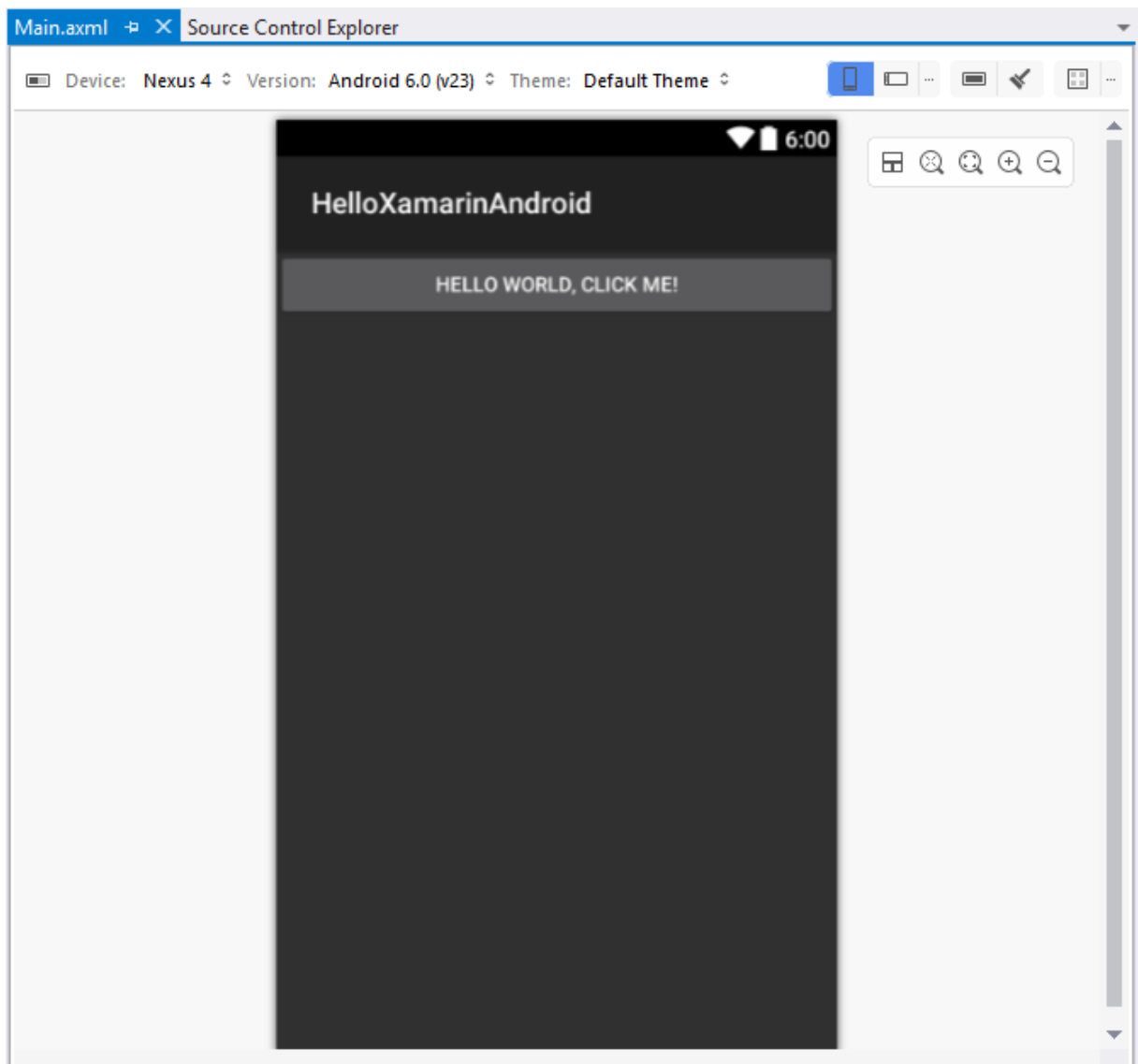
5. Desde la ventana **Solution Explorer**, haz doble clic sobre el archivo **Resources > layout > Main.xml** para abrirlo en el diseñador de Android.



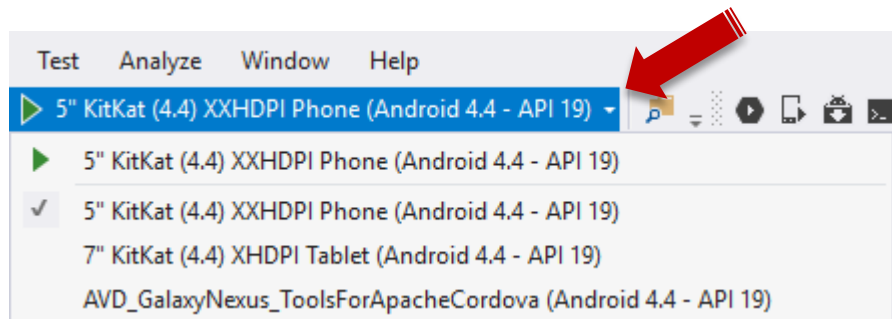
En caso de que te sea mostrada una pantalla similar a la siguiente, deberás actualizar el SDK de Android que tengas instalado a través del enlace **Open Android SDK**.



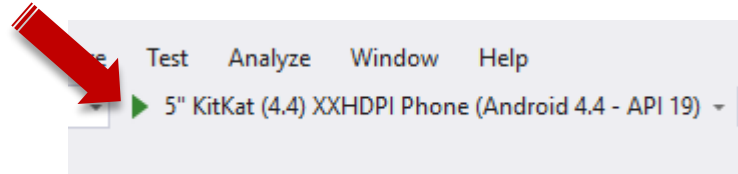
Cuando el SDK de Android esté actualizado, podrás ver la pantalla del diseñador de Android similar a la siguiente.



6. En la lista de simuladores selecciona alguno donde deseas desplegar la aplicación.



7. Haz clic en **Start** para desplegar la aplicación en el simulador seleccionado.



Si todo funciona correctamente, la aplicación será mostrada en el simulador de Android que elegiste.



8. Haz clic en el botón **Hello World, Click Me!**  
Se mostrará una pantalla similar a la siguiente.



9. Regresa a Visual Studio y detén la ejecución.

# Ejercicio 4: Creando una aplicación Windows

Si tienes interés en utilizar Xamarin.Forms para desarrollar aplicaciones para dispositivos Windows, Windows Phone o Windows 10 Mobile, es recomendable que primero te familiarices con el uso de Visual Studio para desarrollar aplicaciones Windows desde el sitio de Microsoft.

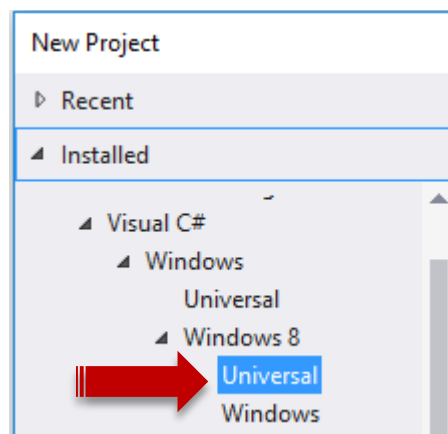
<http://dev.windows.com/>

En este ejercicio crearás una aplicación Windows 8.1, una aplicación Windows Phone 8.1 y una aplicación Windows 10 Mobile sencilla que será desplegada en un simulador.

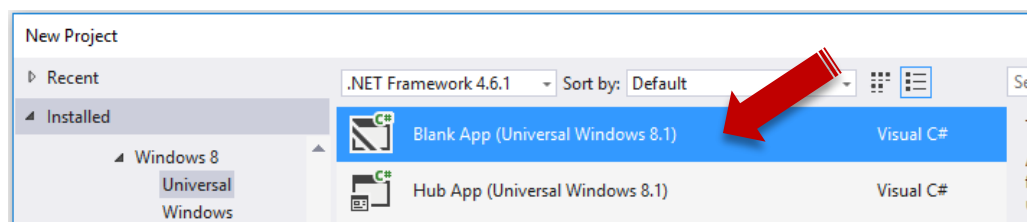
## Tarea 1. Crear la aplicación Windows 8.1/Windows Phone 8.1.

Realiza los siguientes pasos para crear una aplicación para las plataformas Windows 8.1 y Windows Phone 8.1.

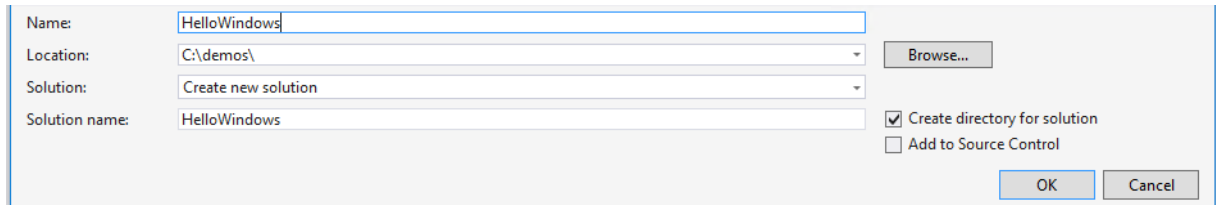
1. Selecciona **File > New > Project** desde Visual Studio.
2. En la ventana **New Project**, selecciona **Visual C# > Windows > Windows 8 > Universal**.



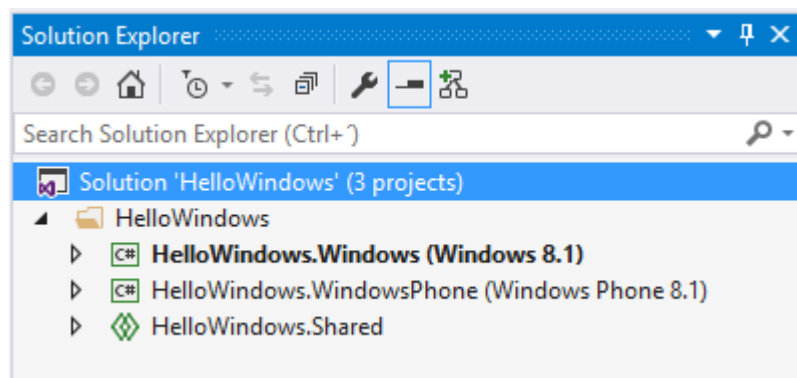
3. En el panel derecho, selecciona la plantilla **Blank App (Universal Windows 8.1)**.



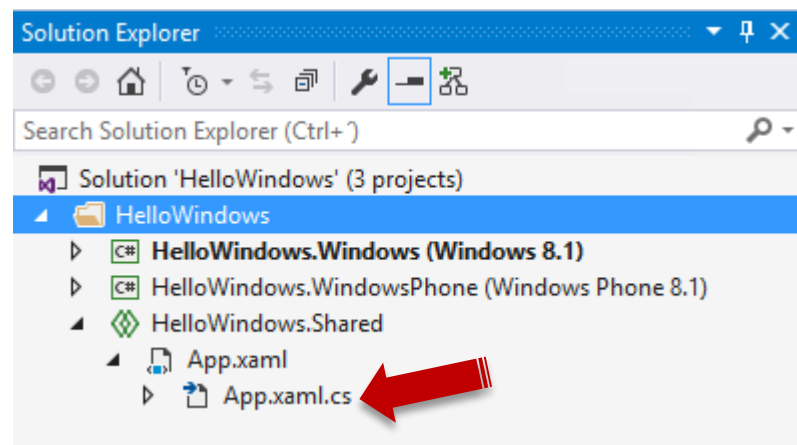
4. Proporciona el nombre, ubicación y haz clic en **OK** para crear el proyecto.



Esta plantilla creará dos proyectos, un proyecto para una aplicación Windows 8.1 y otro proyecto para una aplicación Windows Phone 8.1. La plantilla además agrega un tercer proyecto con código compartido.



10. Desde la ventana **Solution Explorer**, haz doble clic sobre el archivo **App.xaml.cs** del proyecto compartido para abrirlo en la ventana de código.

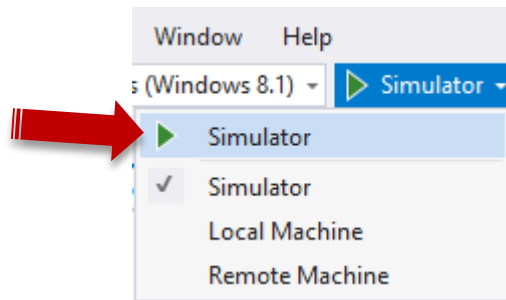


11. Agrega el siguiente código dentro del método **OnLaunched**, justo después de la última línea **Window.Current.Activate();**

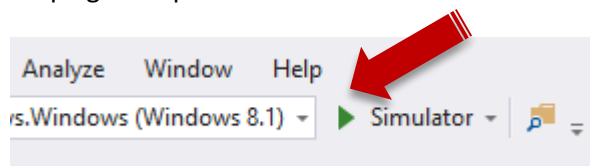
```
Window.Current.Activate();
```

```
var msg = new Windows.UI.Popups.MessageDialog("Hello Windows!");  
msg.ShowAsync();
```

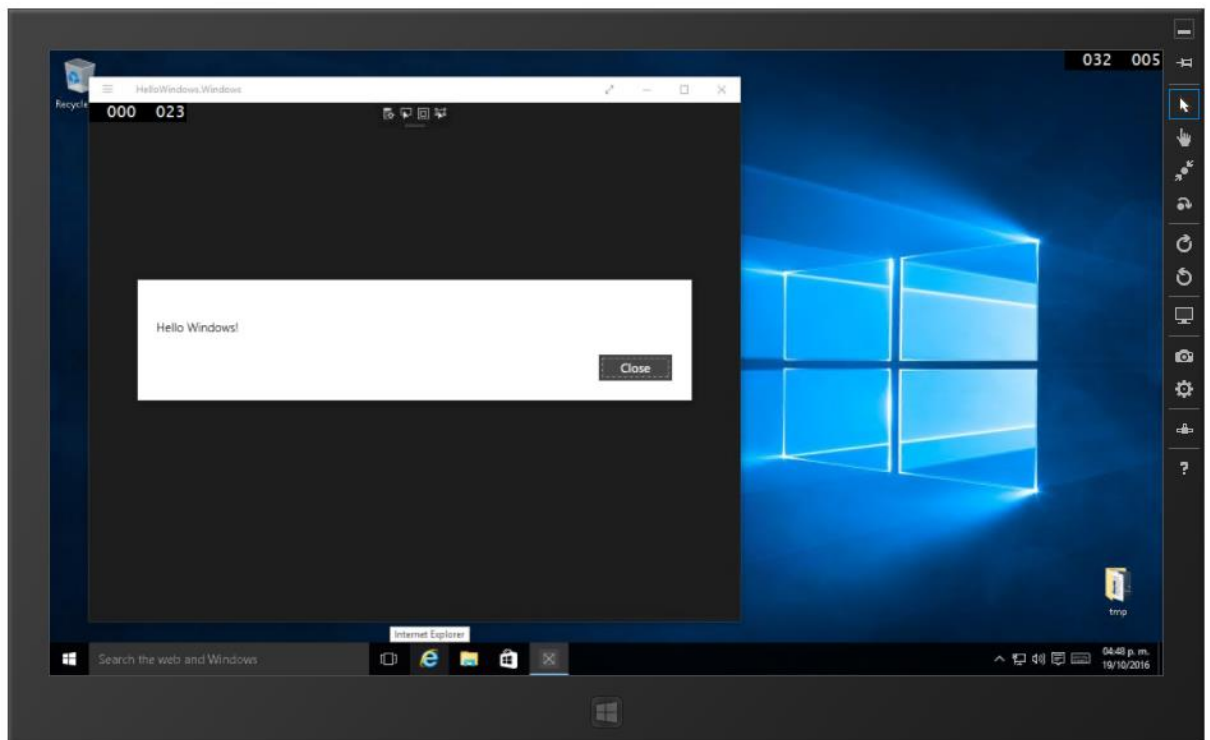
12. Selecciona **Simulator** para desplegar la aplicación Windows en el simulador.



13. Haz clic en **Start** para desplegar la aplicación en el simulador seleccionado.



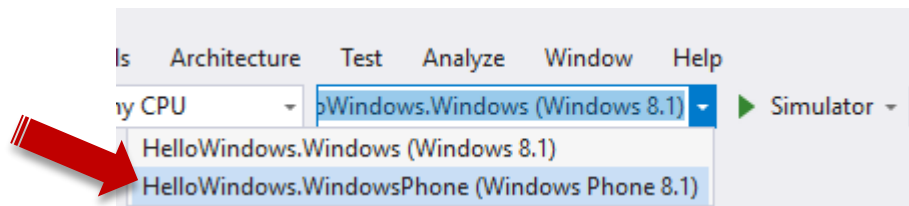
Si todo funciona correctamente, la aplicación será mostrada en el simulador que elegiste. La aplicación mostrará el mensaje “Hello Windows!”.



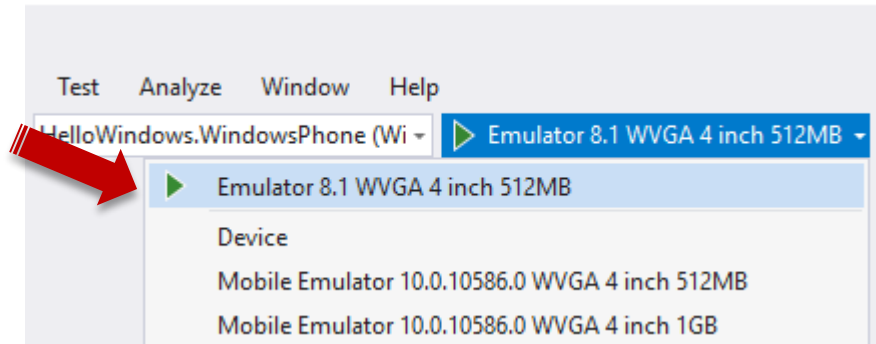
14. Regresa a Visual Studio y detén la ejecución.

15. Selecciona el proyecto Windows Phone como proyecto de inicio desde la barra de herramientas.





16. Selecciona un emulador de la lista de emuladores.



17. Haz clic en **Start** para desplegar la aplicación en el emulador seleccionado.



Si todo funciona correctamente, la aplicación será mostrada en el emulador que elegiste. La aplicación mostrará el mensaje “Hello Windows!”.

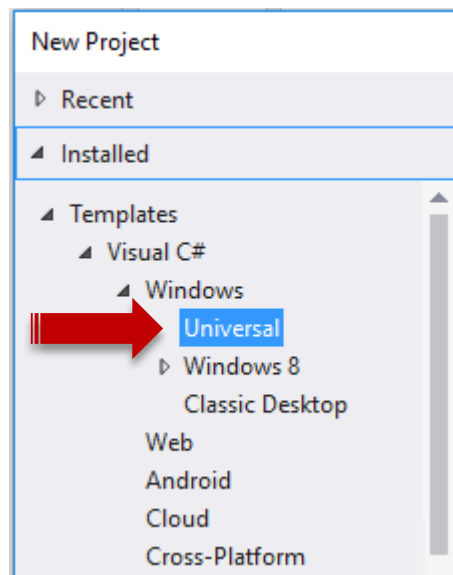


18. Regresa a Visual Studio y detén la ejecución.

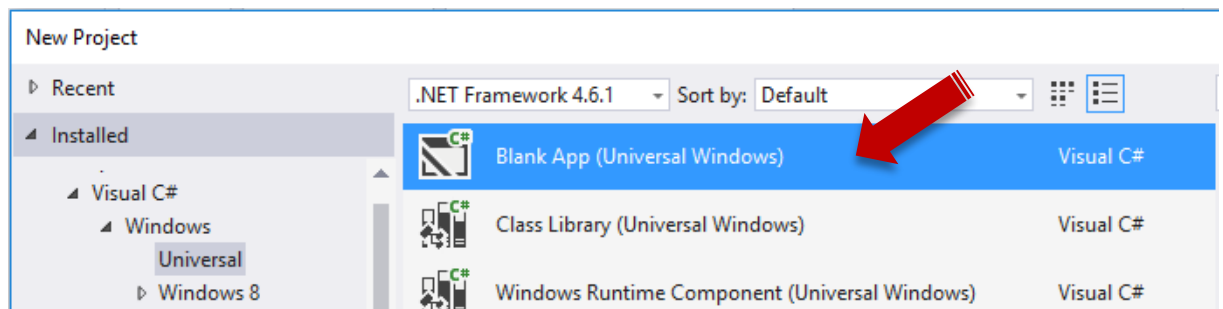
## Tarea 2. Crear la aplicación de la Plataforma Universal de Windows.

Realiza los siguientes pasos para crear una aplicación Windows 10 Mobile.

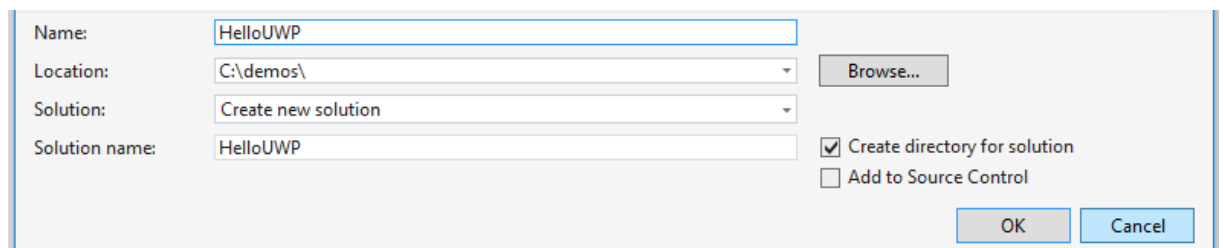
1. Selecciona **File > New > Project** desde Visual Studio.
2. En la ventana **New Project**, selecciona **Visual C# > Windows > Universal**



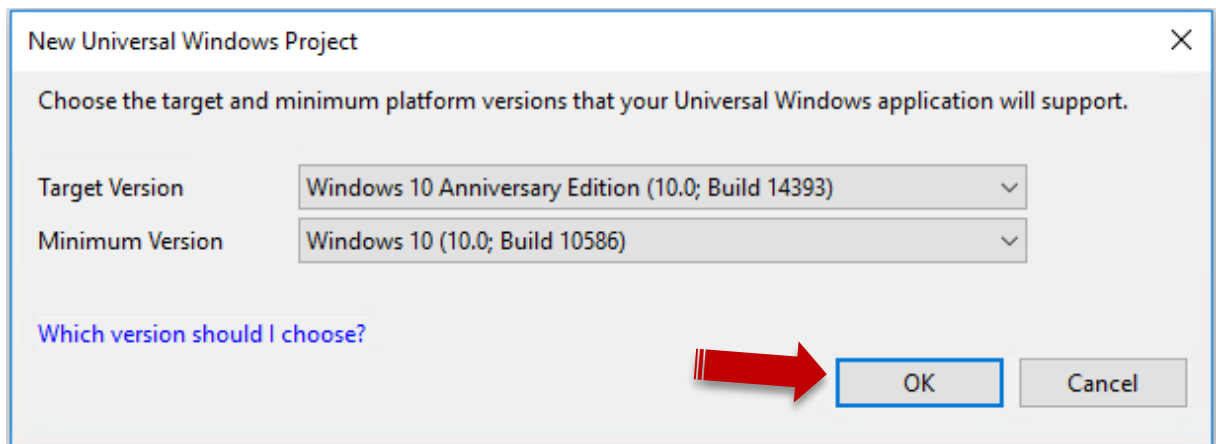
3. En el panel derecho, selecciona la plantilla **Blank App (Universal Windows)**.



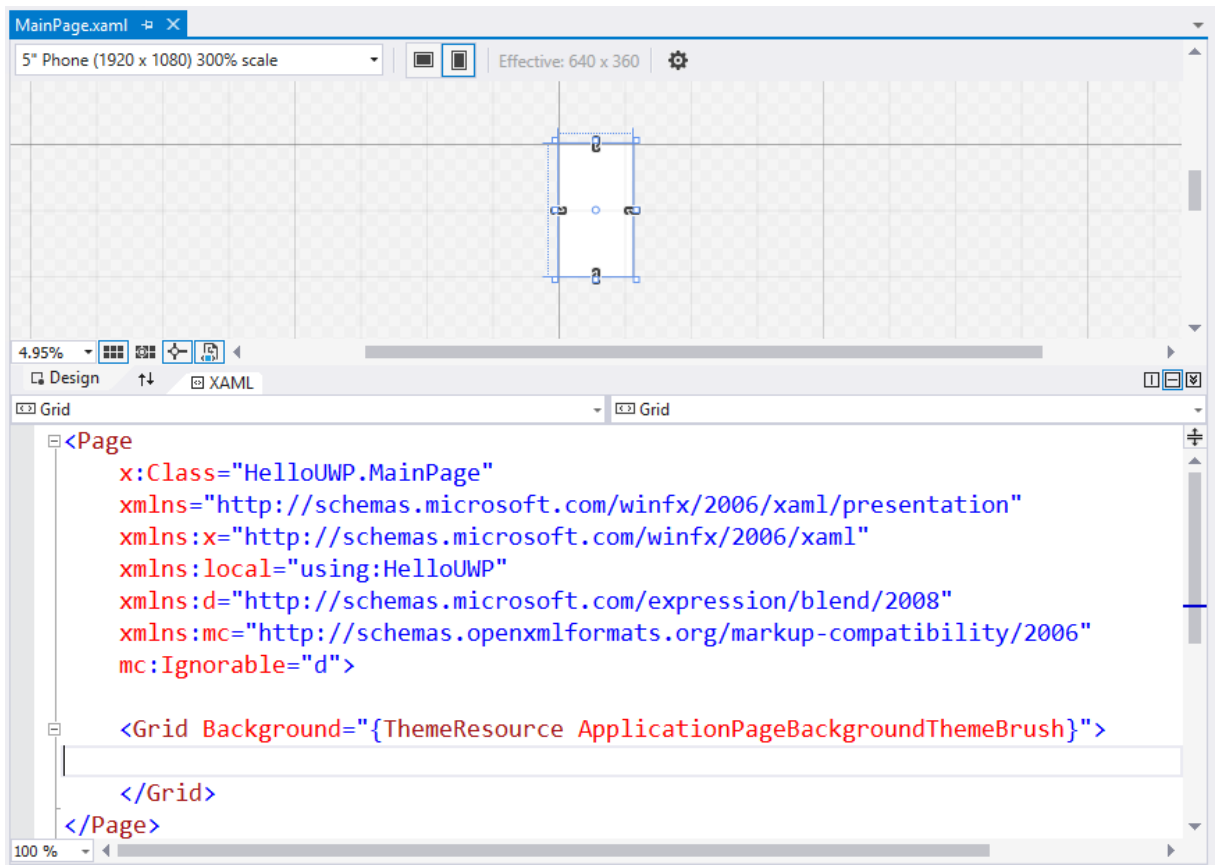
4. Proporciona el nombre, ubicación y haz clic en **OK** para crear el proyecto.



5. En el cuadro de dialogo **New Universal Windows Project** haz clic en **OK** para seleccionar las versiones de plataforma sugeridas.



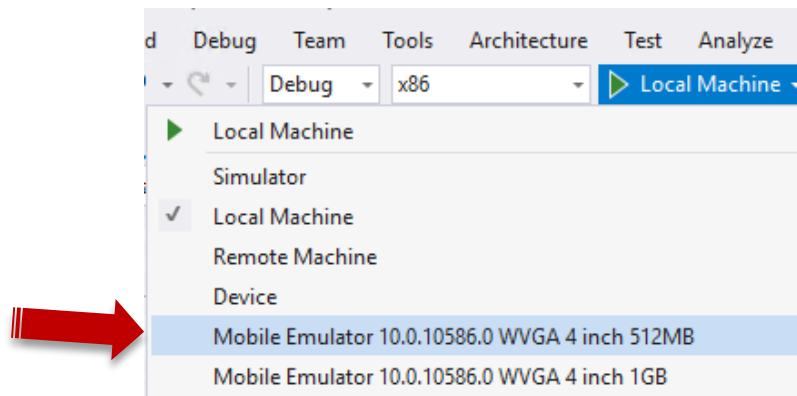
6. Desde la ventana **Solution Explorer**, haz doble clic sobre el archivo **MainPage.xaml** para abrirlo en el diseñador. Se mostrará una pantalla similar a la siguiente.



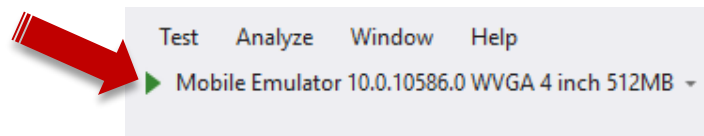
7. Agrega el siguiente código dentro del cuerpo del elemento **Grid**.

```
<TextBlock Text="Hello UWP!" HorizontalAlignment="Center"/>
```

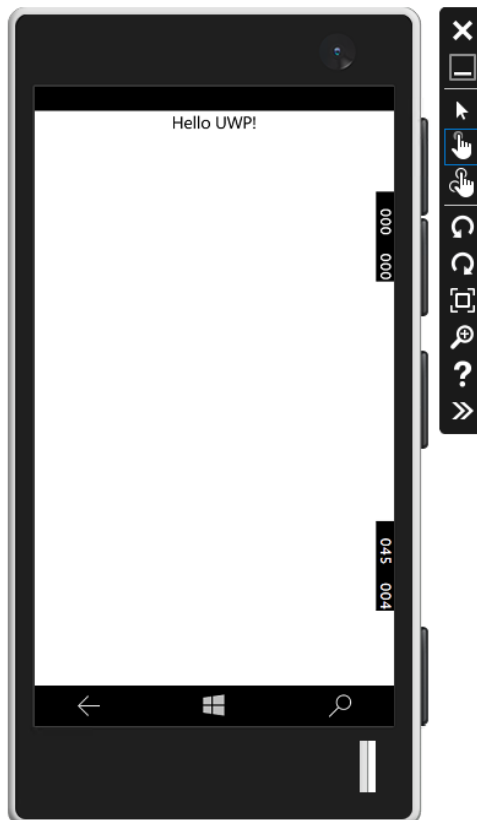
8. Selecciona un emulador Mobile de la lista de emuladores.



9. Haz clic en **Start** para desplegar la aplicación en el emulador seleccionado.



10. Si todo funciona correctamente, la aplicación será mostrada en el simulador que elegiste. La aplicación mostrará el mensaje "Hello UWP!".



11. Regresa a Visual Studio y detén la ejecución.

# Resumen

---

En este laboratorio preparaste tu entorno de desarrollo Xamarin para crear aplicaciones móviles multiplataforma iOS, Android y Windows desde el entorno de desarrollo Visual Studio 2015.

Configuraste también tu entorno de desarrollo Mac para permitir la compilación y depuración de aplicaciones iOS desde Visual Studio.

Si has podido construir aplicaciones Xamarin.iOS, Xamarin.Android y Windows, entonces tienes todo listo para crear tu primera aplicación Xamarin.Forms.

Es tiempo de decir “Hello, Xamarin.Forms” a una nueva era en el desarrollo de aplicaciones móviles multiplataforma.

Cuando hayas finalizado este laboratorio publica el siguiente mensaje en Twitter y Facebook:

***¡He finalizado el #Lab01 del #XamarinDiplomado y tengo listo mi entorno de desarrollo!***