



# Laboratorio

---

## *Vistas de Diseño Alternativas*

Versión: 1.0.0  
Junio de 2017



[Miguel Muñoz Serafín](#)  
@msmdotnet





## CONTENIDO

### INTRODUCCIÓN

#### **EJERCICIO 1: EJERCICIO 1: TRABAJANDO CON DISEÑOS ALTERNATIVOS**

Tarea 1. Crear un proyecto Android.

Tarea 2. Crear Diseños Alternativos.

Tarea 3. Editar Diseños Alternativos.

Tarea 4. Realizar cambios locales.

Tarea 5. Manejar conflictos.

#### **EJERCICIO 2: VALIDANDO TU ACTIVIDAD**

Tarea 1. Agregar los componentes de la Capa de acceso a Servicio.

Tarea 2. Modificar la interfaz de usuario.

Tarea 3. Agregar la funcionalidad para validar la actividad.

### RESUMEN



# Introducción

---

Las vistas de diseño (*Layout*) en Xamarin.Android también pueden ser versionadas mediante el uso de calificadores de recursos. Por ejemplo, puede haber una versión de un *layout* que sólo se utilice cuando el dispositivo esté en modo *Landscape* y una versión de *layout* que sólo sea para el modo *Portrait*.

En este laboratorio conocerás la forma de crear vistas de diseño alternativas que podrán ser utilizadas dependiendo de la configuración o estado actual del dispositivo.

## Objetivos

Al finalizar este laboratorio, los participantes serán capaces de:

- Crear diseños alternativos.
- Editar diseños alternativos.
- Manejar conflictos entre versiones de diseños alternativos.

## Requisitos

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

- Un equipo de desarrollo con Visual Studio. Los pasos descritos en este laboratorio fueron realizados con Visual Studio 2017 y Windows 10 Professional, sin embargo, los participantes pueden utilizar la versión de Visual Studio 2015 que ya tengan instalada.
- Xamarin para Visual Studio.

Tiempo estimado para completar este laboratorio: **60 minutos**.



# Ejercicio 1: Trabajando con Diseños Alternativos

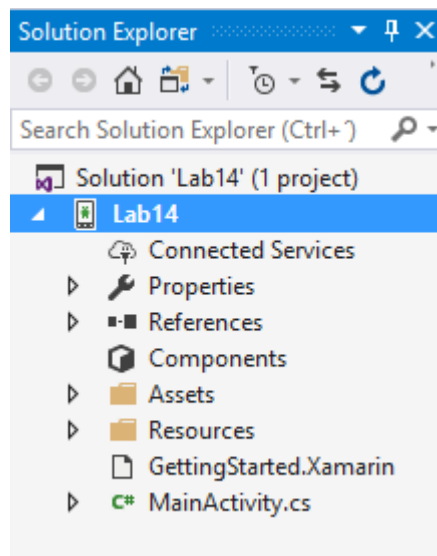
En este ejercicio exploraremos la forma de diseñar vistas alternativas para mostrar la interfaz de usuario en modo horizontal y vertical.

## Tarea 1. Crear un proyecto Android.

En esta tarea crearás un proyecto Android que te permitirá explorar la forma de trabajar con Diseños alternativos.

1. Abre Visual Studio bajo el contexto del Administrador.
2. Utiliza la plantilla **Blank App (Android)** para crear una solución con un proyecto *Xamarin.Android* llamado **Lab14**.

El explorador de soluciones deberá mostrar algo similar a lo siguiente.



3. Agrega al directorio *Resources\drawable*, el archivo *XamarinDiplomado.jpg* que viene adjunto a este documento.
4. Haz doble clic en el archivo *Resources\layout\Main.axml* para abrirlo en el diseñador de Xamarin.Android.
5. Construye la interfaz de usuario agregando los siguientes Widgets y configurando sus propiedades.



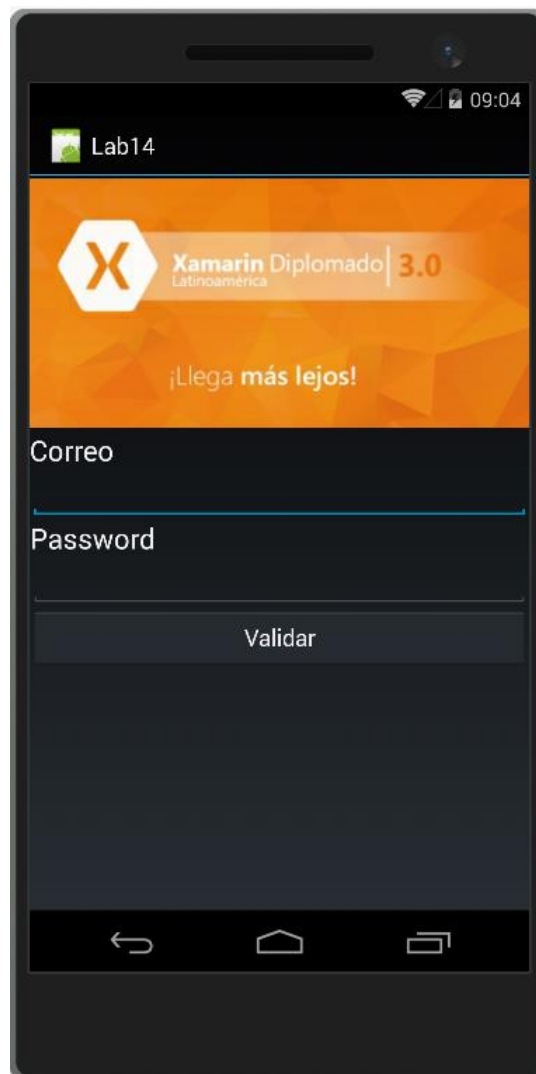
Widget	Propiedad	Valor
<b>ImageView</b>	id	@+id/imageViewLogo
	src	@drawable/xamarindiplomado
<b>Text (Large)</b>	id	@+id/textViewEMail
	text	Correo
<b>E-mail</b>	id	@+id/editTextEmail
<b>Text (Large)</b>	id	@+id/textViewPassword
	text	Password
<b>Password</b>	id	@+id/editTextPassword
<b>Button</b>	id	@+id/buttonValidate
	text	Validar

La interfaz de usuario se verá de la siguiente forma.

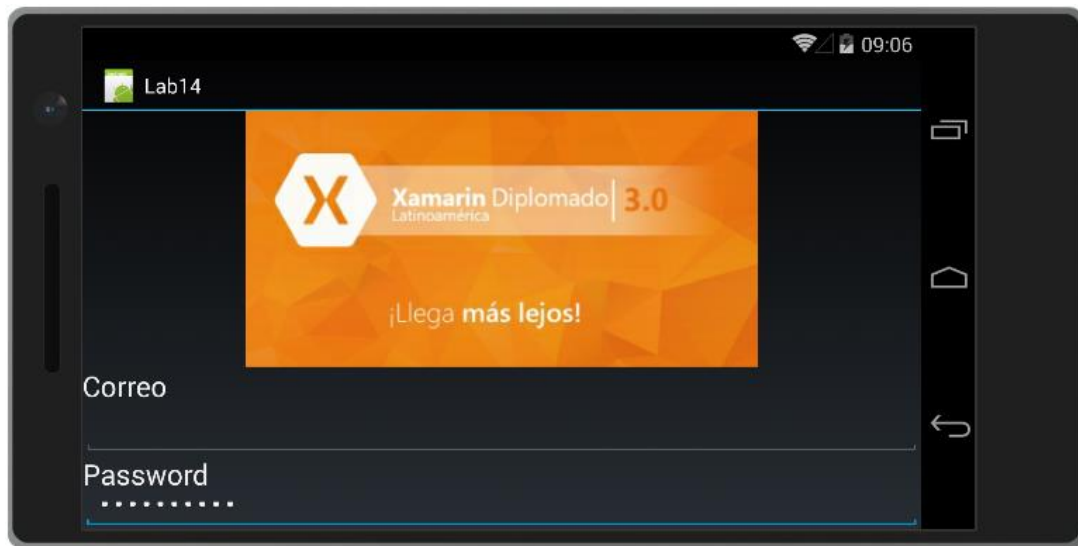




6. Guarda los cambios.
7. Modifica el archivo *MainActivity.cs* para establecer el layout *Main* como contenido de la *Activity*.
8. Ejecuta la aplicación. Se mostrará una pantalla similar a la siguiente.



9. Gira el dispositivo para establecerlo en modo horizontal.  
  
Puedes notar que la interfaz de usuario no se muestra correctamente.

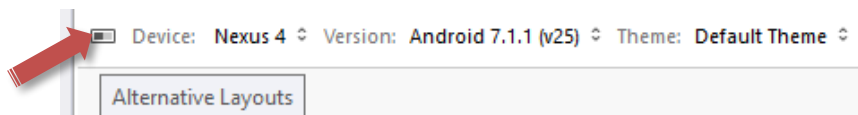


10. Regresa a Visual Studio y detén la ejecución.

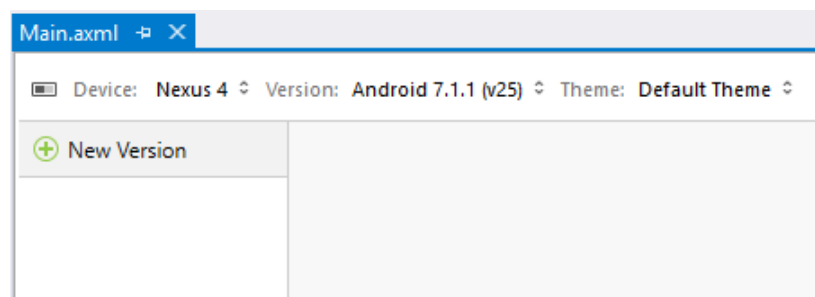
## Tarea 2. Crear Diseños Alternativos.

Los diseños alternativos nos permiten crear interfaces de usuario que se adapten a distintos tamaños de pantalla o a distintas configuraciones de dispositivos. En esta tarea crearás diseños alternativos para que la interfaz de usuario se muestre apropiadamente en modo horizontal, en modo vertical y para dispositivos con una configuración regional específica.

1. En la ventana del diseñador, haz clic en el icono *Alternative Layouts*.



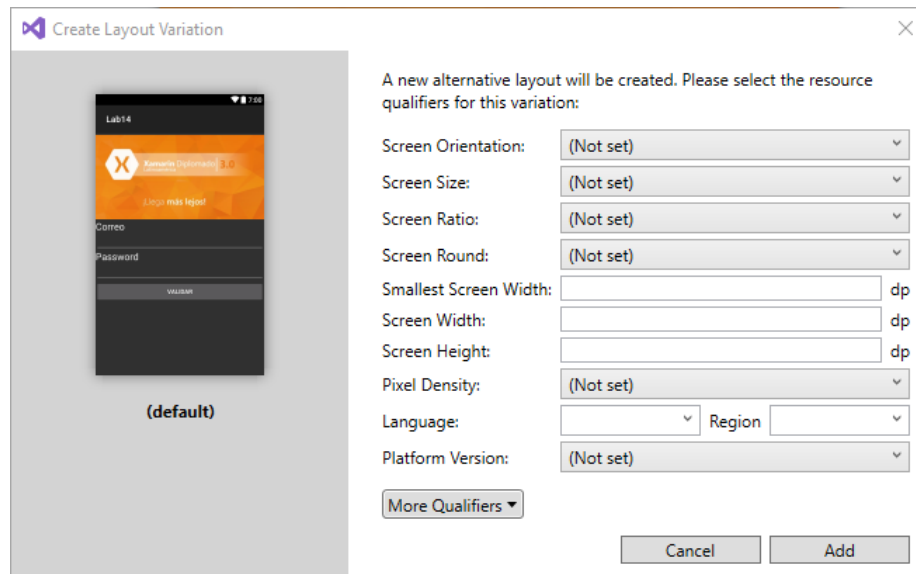
Al hacer clic en el icono *Alternative Layouts*, se abre un panel de vista previa para mostrar los diseños alternativos disponibles en el proyecto. Debido a que en este momento no tienes diseños alternativos, el panel aparece vacío.



La opción **New Version** nos permite crear vistas de diseño alternativas.

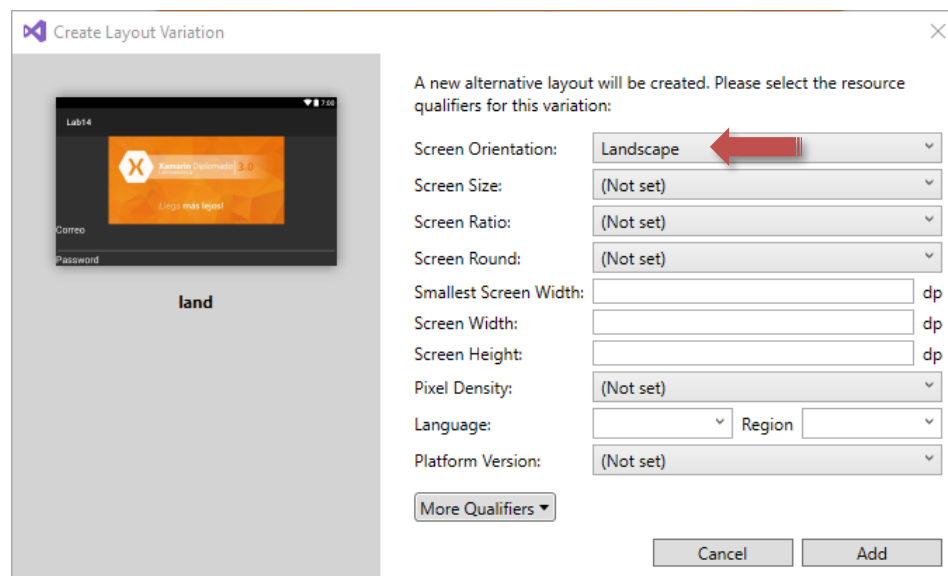


- Haz clic en la opción **New Version**. La ventana de diálogo **Create Layout Variation** se abrirá para que puedas seleccionar los calificadores de recursos para esta alternativa de diseño.



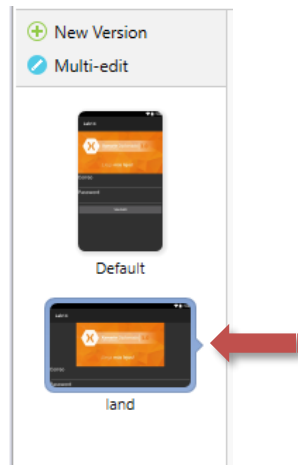
En esta ventana se pueden seleccionar los calificadores de recursos para los cuales será utilizado este nuevo diseño alternativo. Por ejemplo, si queremos diseñar un *layout* que sea utilizado cuando el dispositivo se encuentre en una orientación horizontal podemos seleccionar el valor *Landscape* para el calificador **Screen Orientation**.


- Selecciona el valor *Landscape* para el calificador **Screen Orientation**.

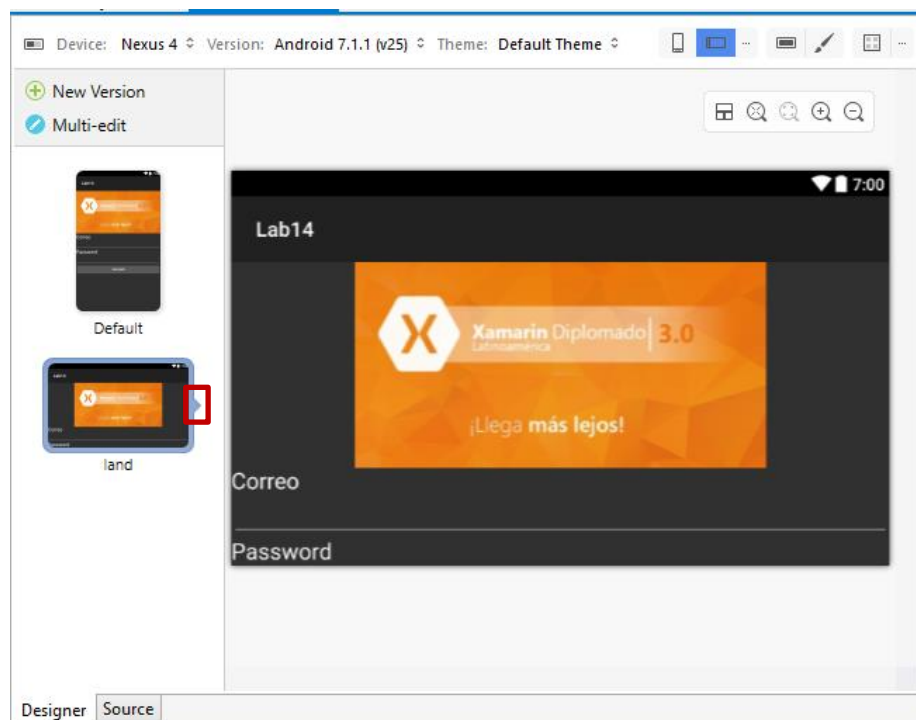


- Haz clic en **Add** para crear la nueva versión del *layout* llamado *land*. En el panel de diseños alternativos puedes ver el nuevo *layout* generado junto con el *layout Default*.





La superficie de diseño mostrará el nuevo layout creado. El panel de diseños alternativos indicará cual *layout* está visible en el diseñador mediante un pequeño puntero  dirigido hacia la derecha.

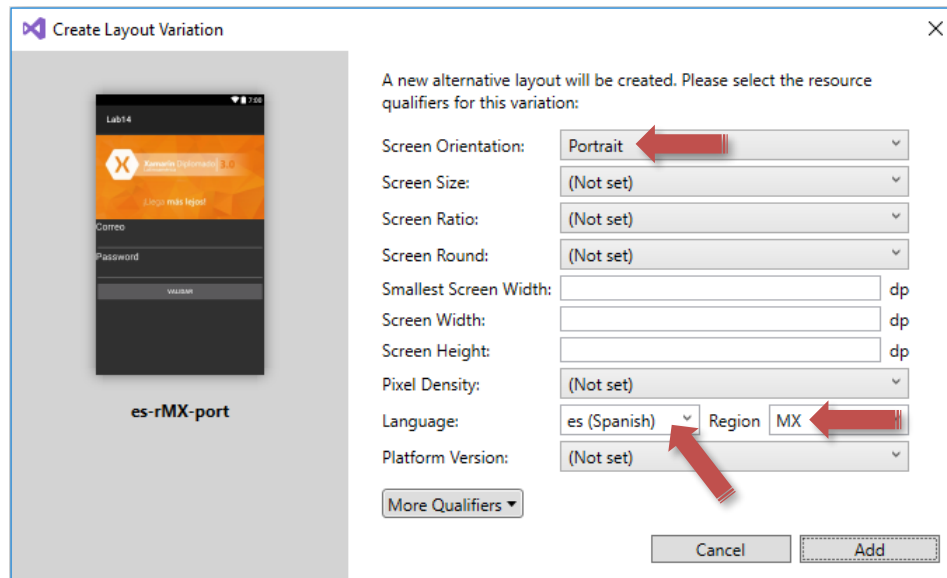


Es posible agregar más diseños alternativos dependiendo de nuestras necesidades. Por ejemplo, podemos crear una vista alternativa que sea utilizada en dispositivos con orientación de pantalla vertical con la configuración regional español de México.

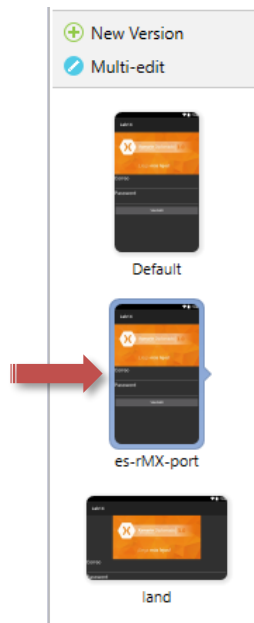
5. Haz clic en la opción **New Version** para abrir la ventana **Create Layout Variation**.



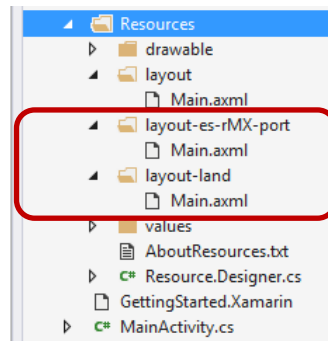
- En la ventana **Create Layout Variation** selecciona la orientación de pantalla **Portrait**, el lenguaje **es (Spanish)** y la región **MX**.



- Haz clic en **Add** para crear el layout **es-rMX-port**. En el panel de diseños alternativos puedes ver el nuevo *layout* generado junto con el *layout* **land** y **Default**



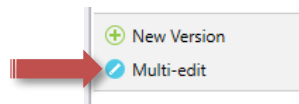
- Guarda los cambios realizados. Puedes notar que al guardar los cambios se generan las carpetas de los nuevos *layouts* alternativos siguiendo la nomenclatura de los calificadores de recursos.



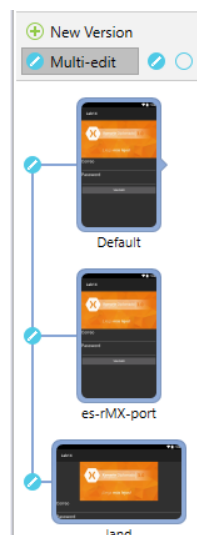
### Tarea 3. Editar Diseños Alternativos.

Cuando creamos *layouts* alternativos, a menudo es deseable realizar un solo cambio que se aplique a todas las versiones ramificadas de un *layout*. Por ejemplo, podríamos querer cambiar el color del texto de la interfaz de usuario a *Naranja* para todos los diseños. Si tenemos un gran número de *layouts* y necesitamos hacer un mismo cambio a todas las versiones, el mantenimiento puede volverse rápidamente engorroso y estar propenso a errores.

Para simplificar el mantenimiento de varias versiones del *layout*, el diseñador ofrece un modo de edición múltiple que propaga los cambios a través de uno o más *layouts*. Cuando hay más de un *layout*, aparece el icono de edición múltiple.



1. Haz clic en la opción *Multi-edit*. Puedes notar que, al hacer clic en el icono de edición múltiple, aparecen líneas que indican que los diseños están enlazados, esto significa que cuando realicemos un cambio en un *layout*, ese cambio se propagará a cualquier diseño enlazado.



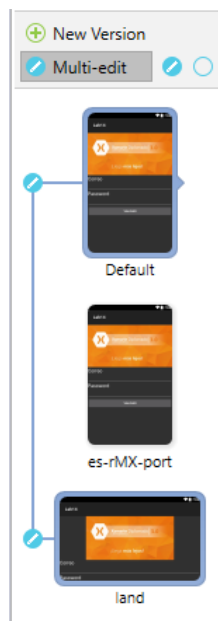


Si tenemos más de dos *layouts*, podemos alternar selectivamente el botón de edición a la izquierda de cada vista previa de *layout* para determinar qué *layouts* están vinculados entre sí.

2. Haz clic en el botón de edición del *layout* **es-rMX-port** para desligarlo de los otros 2.

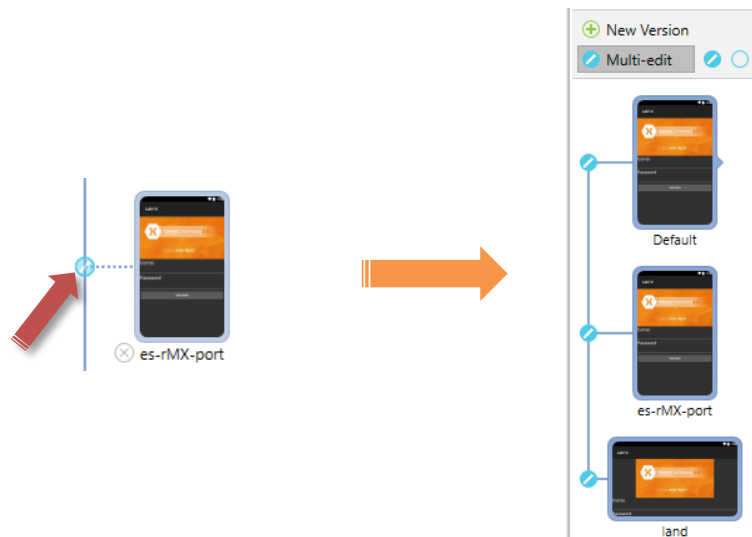


El panel se mostrará de la siguiente manera indicando que solo están enlazados 2 de los *layouts*.



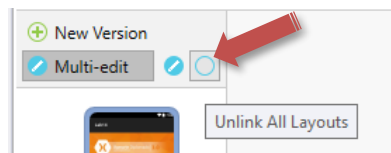
Cualquier cambio que realicemos será propagado únicamente a los *layouts* **Default** y **land**.

3. Haz nuevamente clic en el botón de edición del *layout* **es-rMX-port** para volver a ligarlo a los demás *layouts*.



Ahora nuevamente los cambios que realices serán propagados a los 3 *layouts*.

Es posible desvincular todos los *layouts* haciendo clic en el icono con el círculo vacío (*Unlink All Layouts*).



4. Haz clic en el icono *Unlink All Layouts* y observa que los *layouts* se han desvinculado.

También es posible enlazar todos los *layouts* haciendo clic en el icono con el círculo de edición (*Link All Layouts*).

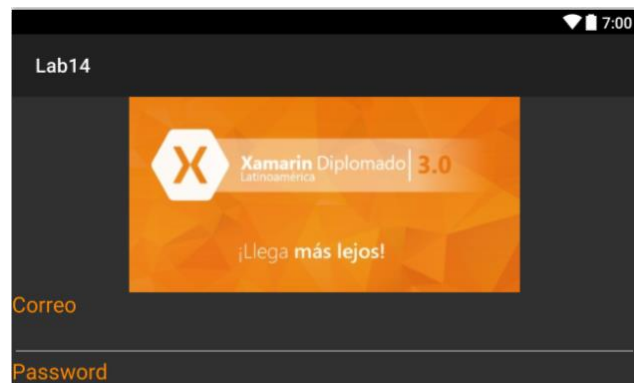


5. Haz clic en el icono *Link All Layouts* y observa que los *layouts* se han vinculado nuevamente.
6. Haz clic sobre el layout **Default** para seleccionarlo.
7. Selecciona el Widget del texto *Correo*.



8. Asigna el valor `@android:color/holo_orange_dark` a la propiedad **textColor** del Widget seleccionado.

9. Selecciona el Widget del texto *Password*.
10. Asigna el valor **@android:color/holo\_orange\_dark** a la propiedad **textColor** del Widget seleccionado.
11. Haz clic en cada *layout* y observa que los cambios se han propagado en los 3 *layouts*.



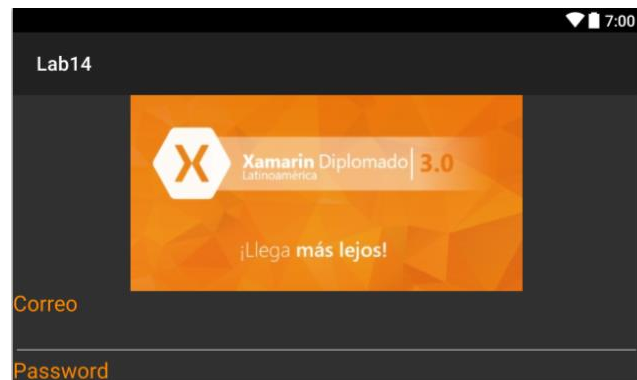
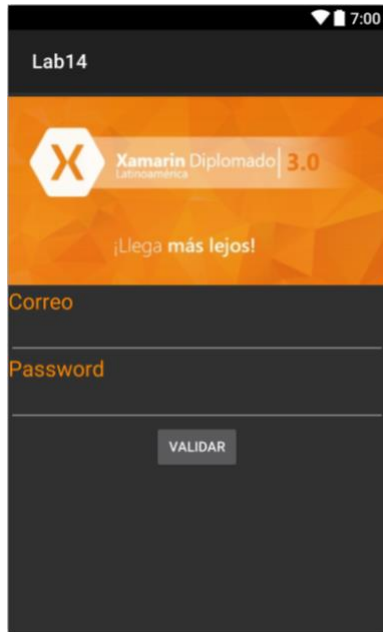
En caso de proporcionar un valor incorrecto en las propiedades, es probable que el diseñador no refleje los cambios esperados e incluso pueda congelarse. Si crees haber proporcionado los valores correctamente y no ves reflejado los cambios, realiza alguna de las siguientes acciones:

- Guarda los cambios, cierra la ventana del diseñador y vuelve a abrir el archivo.
- Si la acción anterior no resuelve el problema, cierra Visual Studio y vuelve a cargar la solución.

12. Asegúrate que los 3 *layouts* sigan vinculados y selecciona el Widget *Button*.
13. Asigna el valor ***wrap\_content*** a la propiedad ***layout\_width*** del *Button*.
14. Asigna el valor ***center\_horizontal*** a la propiedad ***layout\_gravity*** del *Button*.



15. Verifica que el tamaño y posición del *Button* se ha actualizado en los 2 *layouts* verticales y que no se aprecia en el *layout* horizontal.



Pero, ¿Qué ocurre si deseamos realizar un cambio local en un único *layout*, es decir, que el cambio no se propague en los otros *layouts*? Para hacer esto, debemos desvincular el *layout* que deseamos cambiar antes de modificarlo.

#### Tarea 4. Realizar cambios locales.

Podemos notar que la interfaz de usuario para el *layout* en modo horizontal no se muestra correctamente. Podemos necesitar realizar algunos cambios en el *layout* horizontal pero que no sean propagados a los otros *layouts*, por ejemplo, disminuir el tamaño de la imagen y reacomodar los controles de otra forma.

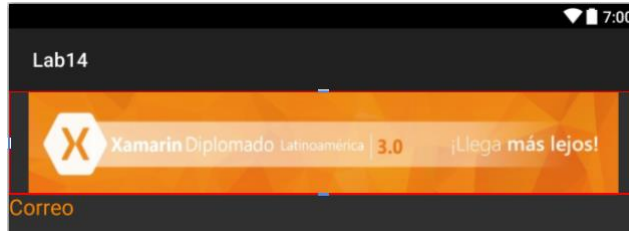
Si realizamos este cambio en el *layout land* mientras los *layouts* están enlazados, el cambio se realizará también en los demás *layouts*. Por lo tanto, primero debemos desvincular los *layouts* antes de realizar el cambio.

1. Agrega al directorio *Resources\drawable*, el archivo *XamarinDiplomadoH.jpg*.
2. En el diseñador de Android, desvincula los *layouts* y selecciona el *layout land*.





3. Establece el valor **@drawable/xamarindiplomadoh** en la propiedad **src** del Widget **ImageView**. El diseñador notifica este cambio con un marco rojo para indicar que el cambio es local en el *layout land* y que no se propaga a los otros *layouts*.



4. Haz clic en el *layout Default*. Puedes notar que la imagen original se mantiene.



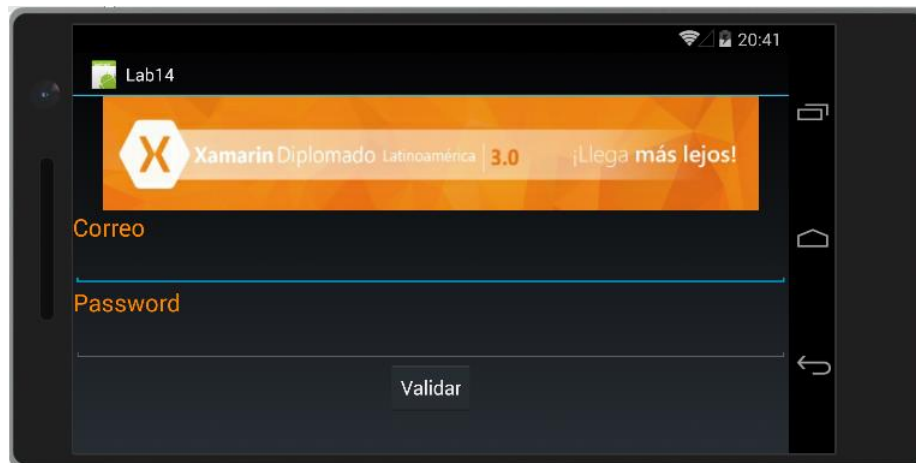
**land**



**Default**

5. Guarda los cambios y ejecuta la aplicación.
6. Puedes notar que ahora el *layout land* se ve sin problemas.





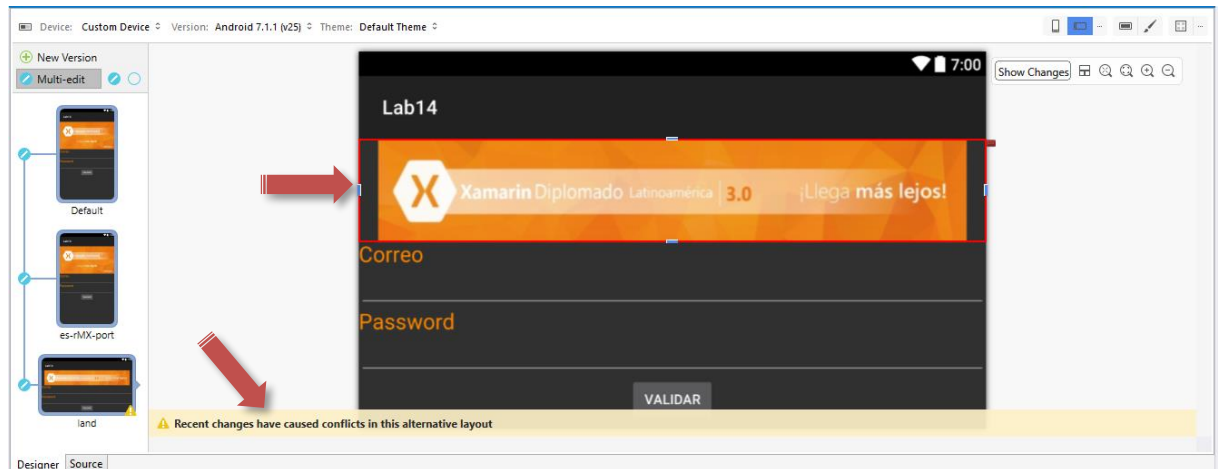
### Tarea 5. Manejar conflictos.

En esta tarea modificarás la propiedad **src** del *ImageView* del *layout es-rMX-port* para que muestre una imagen distinta. Al modificar la propiedad **src** y propagarse los cambios, se generará un conflicto con el *layout land* ya que previamente modificaste también su propiedad **src** y corre el riesgo de sobrescribir ese cambio previo.

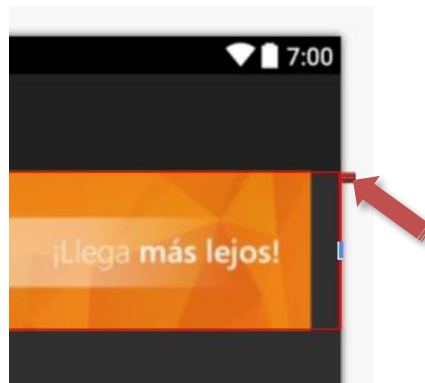
1. Agrega al directorio *Resources\drawable*, el archivo *XamarinDiplomadoM.jpg*
2. En el diseñador de Android, haz clic nuevamente en **Multi-edit** para enlazar los *layouts*.
3. Haz clic en el *layout es-rMX-port* para seleccionarlo.
4. Selecciona el *ImageView*.
5. Establece el valor **@drawable/xamarindiplomadom** en la propiedad **src** del Widget *ImageView*. Puedes notar que aparece un icono de advertencia en el diseño vinculado *land*. Este icono informa de un conflicto ya que la propiedad **src** del *layout land* previamente fue modificado y corre el riesgo de sobrescribirse con la propagación de cambios originada por el *layout es-rMX-port*.



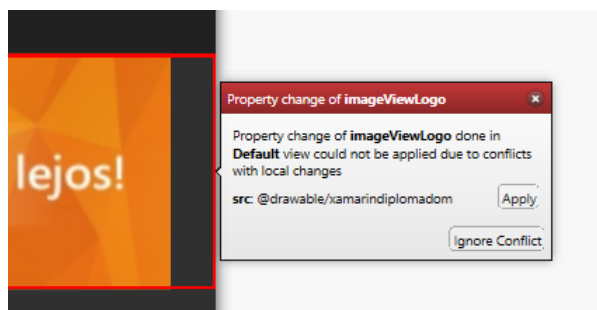
6. Haz clic sobre el *layout land* para seleccionarlo. Al mostrarse el diseño, el conflicto es revelado. El Widget que causó el conflicto se resalta con un marco rojo y se muestra el mensaje: **Recent changes have caused conflicts in this alternative layout.**



Observa que a la derecha del Widget con conflicto se muestra un indicador de conflicto.

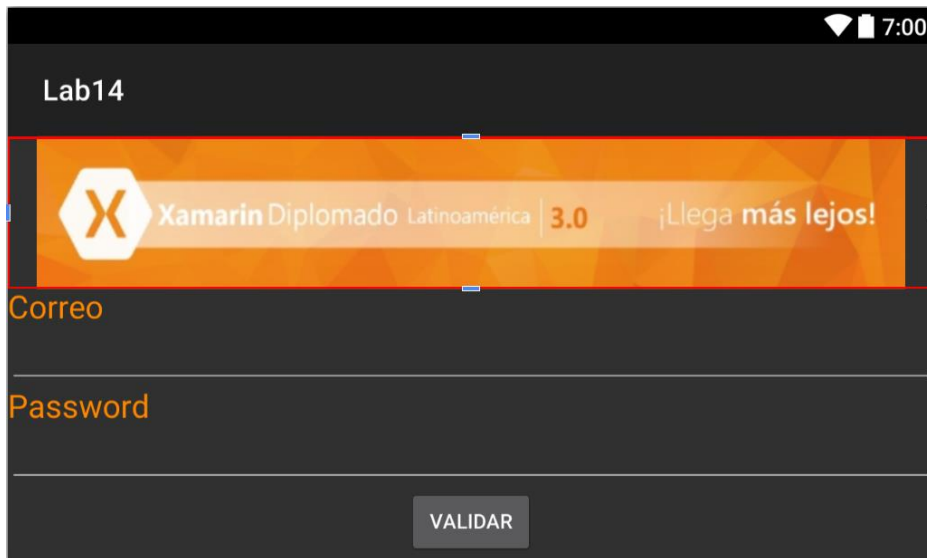


7. Haz clic sobre el indicador de conflicto para abrir el cuadro de dialogo que describe el conflicto.

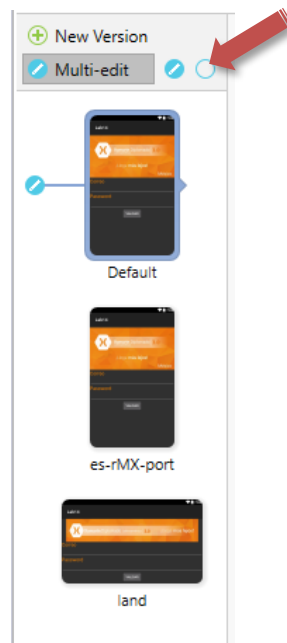


El cuadro de conflicto muestra la lista de propiedades que han cambiado y despliega sus valores. Al hacer clic en **Ignore Conflict** los cambios no son propagados a ese Widget. Al hacer clic en **Apply**, los cambios son propagados a ese Widget (perdiendo su valor actual).

8. Haz clic en *Ignore Conflict* para que el *ImageView* del *layout land* no sea modificado. Puedes notar que el conflicto desaparece.

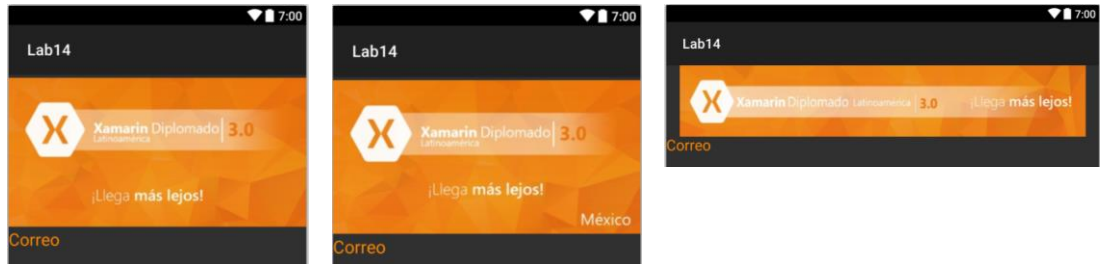


9. Haz clic en el layout *Default*. Puedes notar que la imagen ha cambiado y muestra la palabra *México*. La imagen es la misma que se propago cuando modificaste la imagen del *layout es-rMX-port*.
10. Haz clic en el icono *Unlink All layouts* para desvincular los *layouts*.



11. Establece el valor **@drawable/xamarindiplomado** en la propiedad **src** del Widget *ImageView*.

Puedes notar que ahora cada *layout* tiene una imagen distinta.



12. Ejecuta la aplicación y verifica que se muestren los layouts correspondientes al modo vertical, horizontal y a la configuración regional es-MX.



## Ejercicio 2: Validando tu actividad

---

En este ejercicio agregarás funcionalidad a tu laboratorio con el único propósito de enviar una evidencia de la realización del mismo.

La funcionalidad que agregarás consumirá un ensamblado que representa una Capa de acceso a servicio (SAL) que será consumida por tu aplicación Android.

Es importante que realices cada laboratorio del diplomado ya que esto te dará derecho a obtener el diploma final del mismo.

### Tarea 1. Agregar los componentes de la Capa de acceso a Servicio.

En esta tarea agregarás una referencia al ensamblado **SALLab14.dll** que implementa la capa de acceso a servicio. El archivo **SALLab14.dll** se encuentra disponible junto con este documento.

1. En el proyecto Xamarin.Android, agrega una referencia del ensamblado **SALLab14.dll**.

Este componente realiza una conexión a un servicio de Azure Mobile, por lo tanto, será necesario agregar el paquete NuGet **Microsoft.Azure.Mobile.Client 4.0.0**.

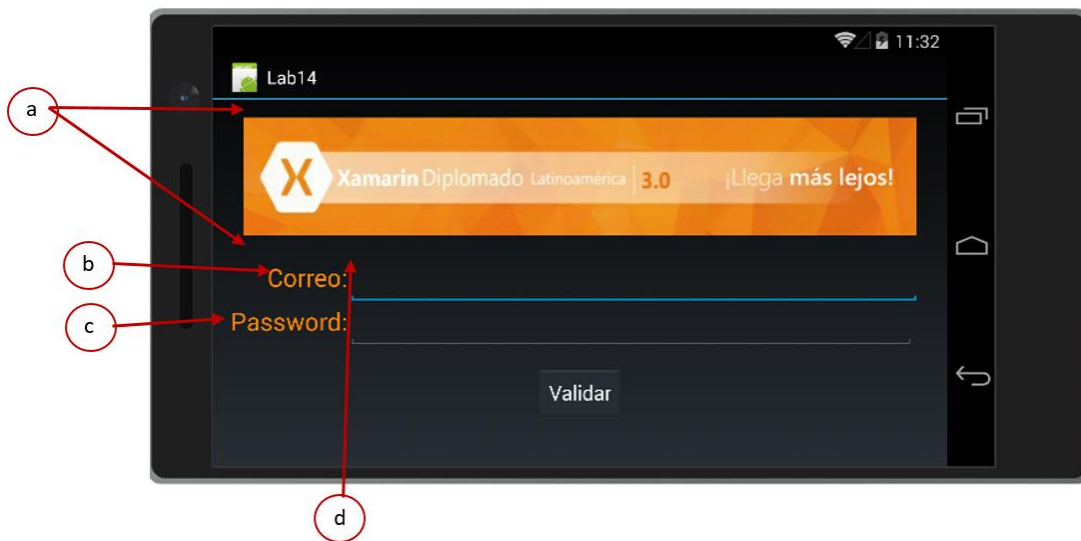
2. En el proyecto Xamarin.Android, instala el paquete NuGet **Microsoft.Azure.Mobile.Client 4.0.0**.

### Tarea 2. Modificar la interfaz de usuario.

Realiza los cambios necesarios para que en la presentación horizontal del dispositivo:

- a) Exista un margen encima y debajo del Widget *imageViewLogo*
- b) El Widget *textViewEMail* se ubique a la izquierda del Widget *editTextEmail*
- c) El Widget *textViewPassword* se ubique a la izquierda del Widget *editTextPassword*
- d) Los Widgets *textViewEMail* y *textViewPassword* deben estar alineados hacia la derecha.

La interfaz de usuario será similar a la siguiente.



**TIPS:**

- El ViewGroup *LinearLayout (Horizontal)* es un contenedor que alinea a sus Views hijos de forma horizontal.
- La propiedad **gravity** de un Widget *TextView* permite alinear su contenido horizontal y verticalmente.

**Tarea 3. Agregar la funcionalidad para validar la actividad.**

El componente DLL que agregaste te permite registrar tu actividad en la plataforma de TI Capacitación y Microsoft. El componente se comunica con la plataforma de TI Capacitación para autenticarte y posteriormente envía un registro a la plataforma Microsoft.

1. Agrega la funcionalidad necesaria para que, al lanzar tu aplicación y hacer clic en el **buttonValidate** se ejecute el siguiente código que realizará la autenticación y mostrará el resultado en un cuadro de Diálogo.

```
var Client = new SALLab14.ServiceClient();
var Result = await Client.ValidateAsync(this);

Android.App.AlertDialog.Builder Builder =
    new AlertDialog.Builder(this);
AlertDialog Alert = Builder.Create();
Alert.SetTitle("Resultado de la verificación");
Alert.SetIcon(Resource.Drawable.Icon);
Alert.SetMessage(
    $"{Result.Status}\n{Result.FullName}\n{Result.Token}");
Alert.SetButton("Ok", (s, ev) => { });
Alert.Show();
```

2. Ejecuta tu aplicación en modo vertical.



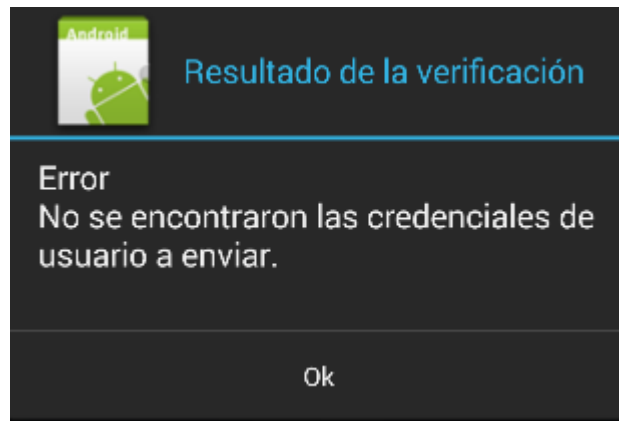
3. Proporciona tus credenciales.
4. Haz clic en el botón para validar. Esto registrará tu actividad en el modo **Portrait** de la aplicación. Puedes ver el resultado de tu actividad en el siguiente enlace:

<https://ticapacitacion.com/evidencias/xamarin30>

Ejecución del laboratorio 14 (Portrait)  
Fecha Límite: 30 de Junio de 2017 0:00 UTC

Aprobado

En caso de que obtengas el siguiente mensaje, verifica que hayas seguido las especificaciones de nombres de los Widgets utilizados en el laboratorio.



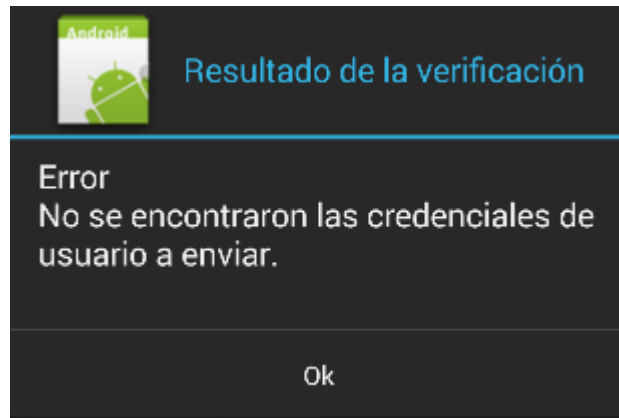
5. Gira el dispositivo a modo horizontal.
6. Proporciona tus credenciales.
7. Haz clic en el botón para validar. Esto registrará tu actividad en el modo **Landscape** de la aplicación. Puedes ver el resultado de tu actividad en el siguiente enlace:

<https://ticapacitacion.com/evidencias/xamarin30>

Ejecución del laboratorio 14 (Landscape)  
Fecha Límite: 30 de Junio de 2017 0:00 UTC

Aprobado

En caso de que obtengas el siguiente mensaje, verifica que hayas seguido las especificaciones de nombres de los Widgets utilizados en el laboratorio.



**Nota:** Es probable que recibas un correo similar al siguiente.

***Tu código de lab no es válido, revisa que estés utilizando un código de reto válido. Si tienes preguntas o dudas por favor contacta a [dxaudmx@microsoft.com](mailto:dxaudmx@microsoft.com)***

Puedes hacer caso omiso al mensaje.

Si encuentras problemas durante la realización de este laboratorio, puedes solicitar apoyo en los grupos de Facebook siguientes:

<https://www.facebook.com/groups/iniciandoconxamarin/>

<https://www.facebook.com/groups/xamarindiplomadoitc/>





# Resumen

---

En este laboratorio conociste la forma de crear vistas de diseño alternativas que pueden ser utilizadas dependiendo de la configuración o estado actual del dispositivo.

¿Qué te pareció este laboratorio?

Comparte tus comentarios en twitter y Facebook utilizando el hashtag **#XamarinDiplomado**.