



# Laboratorio

---

## *Introducción al diseñador de Xamarin.Android*

Versión: 1.0.0  
Junio de 2017



[Miguel Muñoz Serafín](#)

@msmdotnet





## **CONTENIDO**

### **INTRODUCCIÓN**

#### **EJERCICIO 1: EXPLORANDO EL DISEÑADOR DE XAMARIN.ANDROID**

Tarea 1. Crear un proyecto Android.

Tarea 2. Explorar los componentes del diseñador.

#### **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

---

Xamarin.Android soporta un estilo declarativo de diseño de interfaz de usuario basado en archivos XML, así como un estilo programático donde podemos crear la interfaz de usuario mediante código. Cuando se utiliza el enfoque declarativo, los archivos XML pueden ser modificados manualmente o modificados visualmente mediante el *Diseñador de Xamarin.Android*. El uso de un diseñador permite una retroalimentación inmediata durante la creación de la interfaz de usuario, acelera el desarrollo y hace que el proceso de creación de la interfaz de usuario sea menos laborioso.

En este laboratorio exploraremos las características del *Diseñador de Xamarin.Android*. Describiremos los conceptos básicos del diseñador, mostrando cómo utilizarlo para agregar Widgets visualmente y cómo editar las propiedades de los Widgets.

## Objetivos

Al finalizar este laboratorio, los participantes serán capaces de:

- Describir las diversas partes que componen el Diseñador.
- Crear un diseño de interfaz de usuario con el diseñador de Xamarin.Android.
- Agregar y editar propiedades de Widgets.

## 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: Explorando el Diseñador de Xamarin.Android

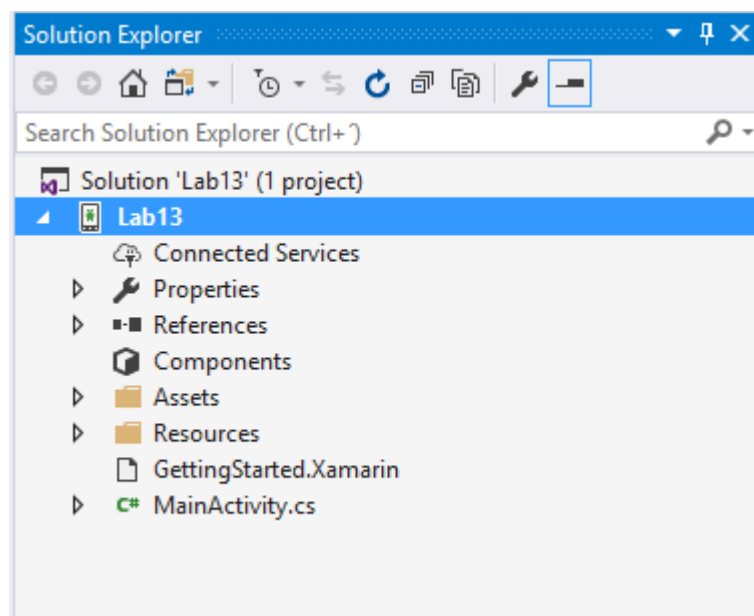
En este ejercicio exploraremos las características del *Diseñador de Android*, la superficie de diseño y el uso de la ventana *Properties* para editar las propiedades de los Widgets.

## Tarea 1. Crear un proyecto Android.

En esta tarea crearás un proyecto Android que te permitirá explorar las características del diseñador de Xamarin.Android.

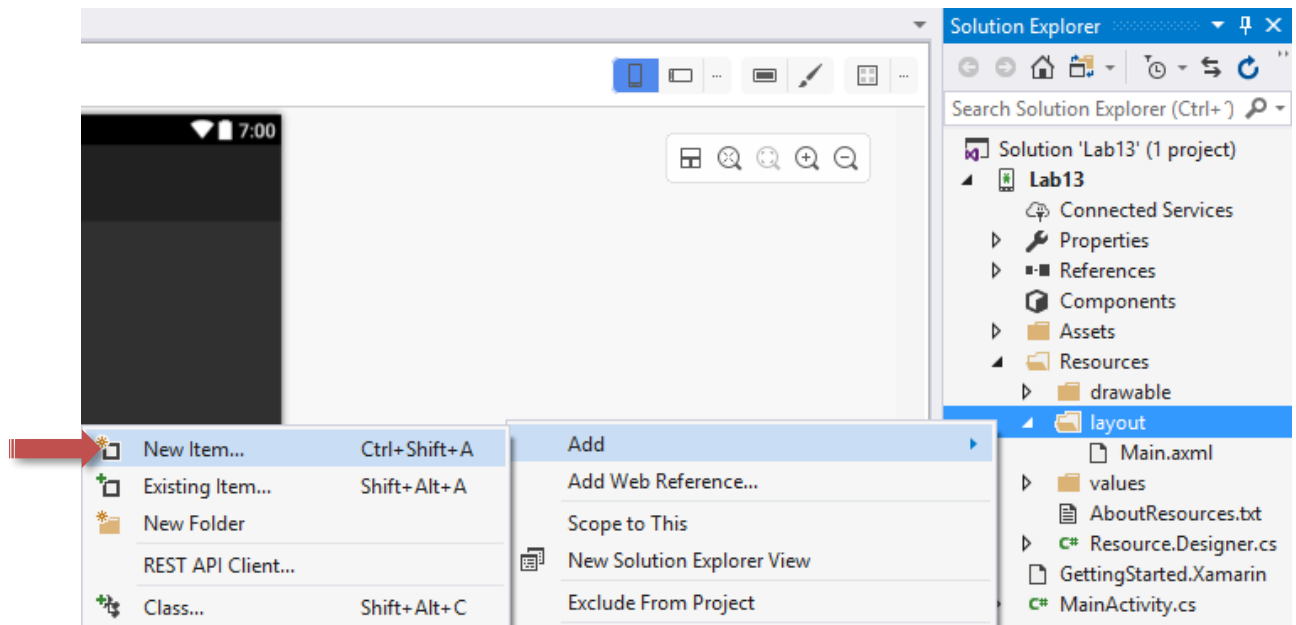
1. Abre Visual Studio bajo el contexto del Administrador.
2. Utiliza la plantilla **Blank App (Android)** para crear una solución con un proyecto *Xamarin.Android* llamado **Lab13**.

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

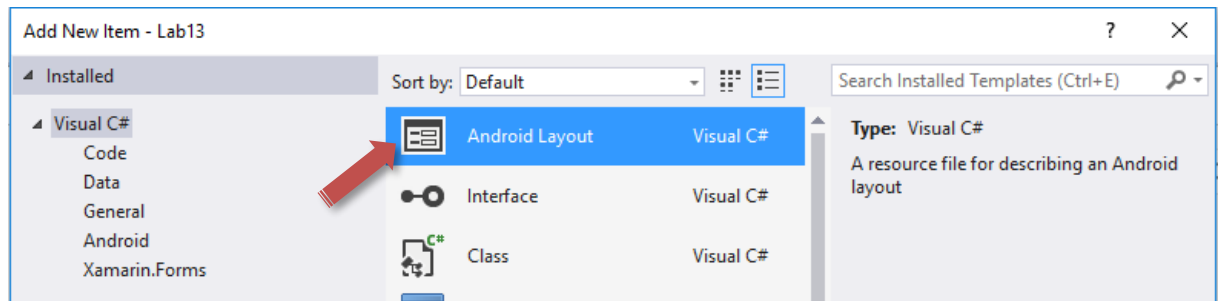


El Diseñador de Xamarin.Android se inicia automáticamente cuando se agrega un nuevo elemento Layout.

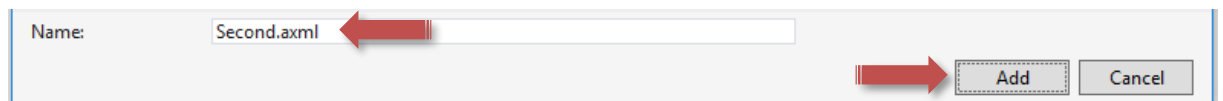
3. Para agregar un nuevo elemento *Layout* al proyecto, haz clic en la opción **Add > New Item** del menú contextual de la carpeta **Resources\layout**.



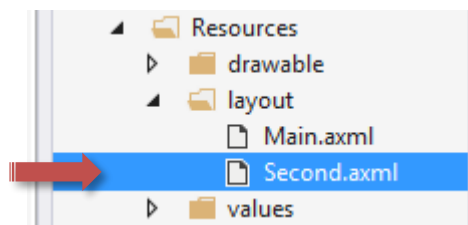
4. En la ventana **Add New Item**, selecciona la plantilla **Android Layout**.

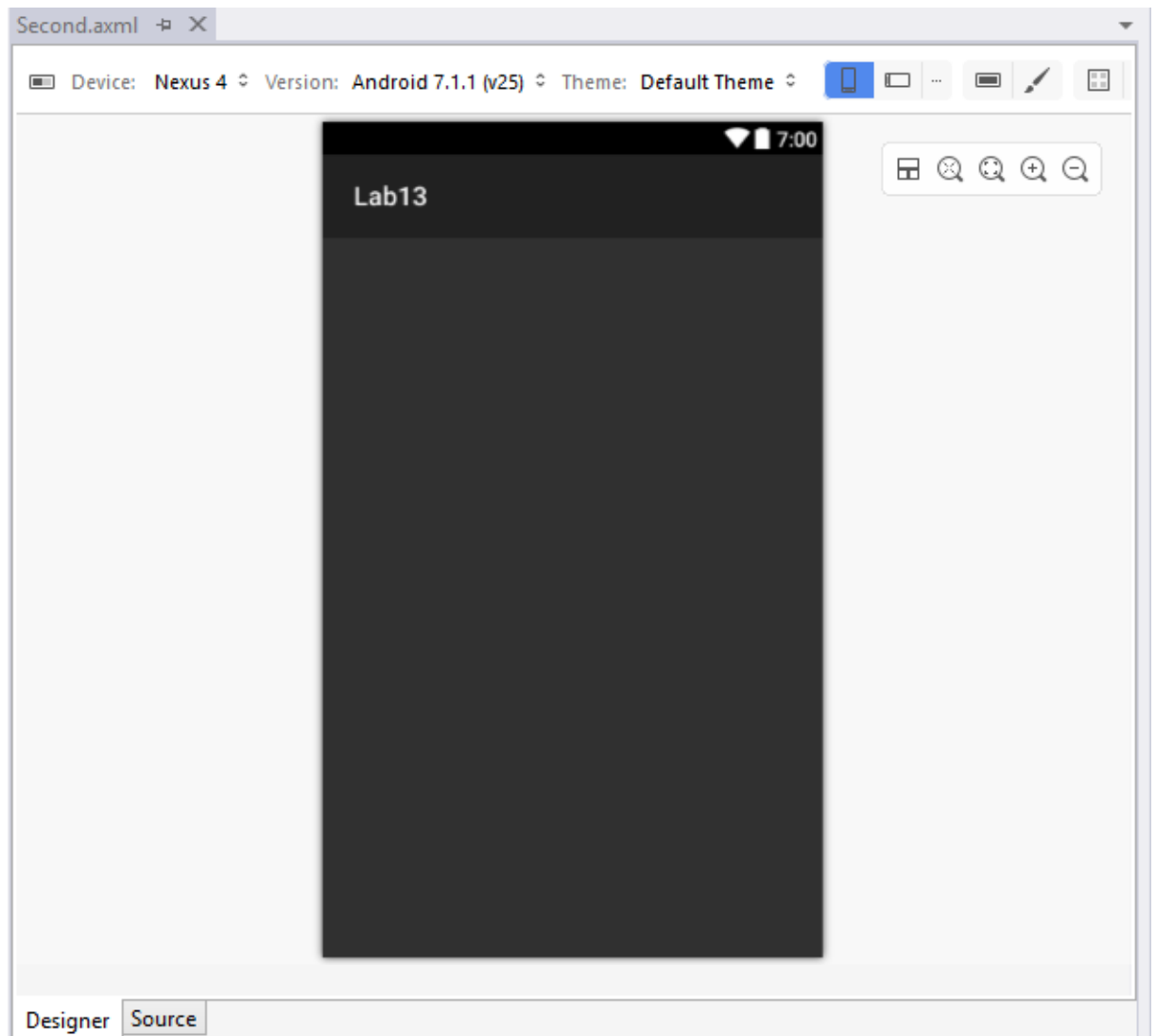


5. Asigna el nombre *Second.xml* al layout y haz clic en *Add* para agregar el nuevo layout.



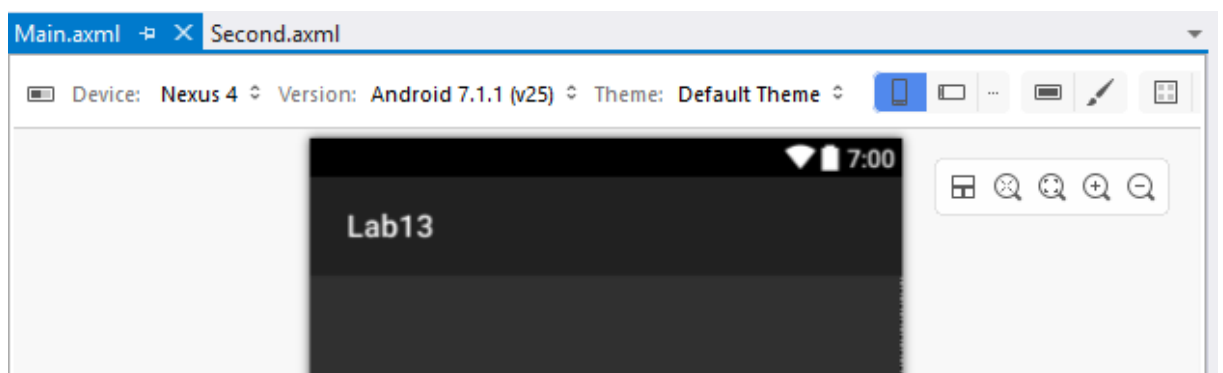
Esto creará un nuevo archivo .xml y será cargado en la superficie de diseño.





El diseñador también puede ser iniciado haciendo doble clic en algún archivo .xml existente.

6. Haz doble clic sobre el archivo *Resources\layout\Main.xml*. El archivo *Main.xml* será cargado en la superficie de diseño.





## Tarea 2. Explorar los componentes del diseñador.

El Diseñador está compuesto por varias secciones que dan soporte a muchas de sus funcionalidades.

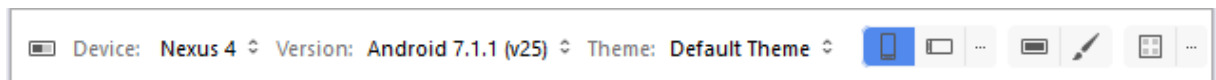
Cuando editamos un *diseño (layout)* en el Diseñador, podemos utilizar los siguientes elementos para crear y dar forma al diseño:

- Barra de herramientas (Toolbar).
- Caja de herramientas (ToolBox).
- Superficie de Diseño.
- Ventana de propiedades.
- Esquema del documento (Document Outline).

En esta tarea explorarás los distintos elementos que componen el Diseñador de Xamarin.Android.

### Barra de herramientas (Toolbar).



1. Localiza la barra de herramientas del diseñador.



La barra de herramientas muestra una lista de selectores de configuración y menús de herramientas.

Elemento	Descripción
	<b>Selector de diseño alternativo.</b> Permite seleccionar entre diferentes versiones de <i>Layout</i> .
Device: Nexus 4 ↕	<b>Selector de dispositivo.</b> Define un conjunto de calificadores asociados con un dispositivo particular tal como el tamaño de la pantalla, la resolución y la disponibilidad del teclado.
Version: Android 7.1.1 (v25) ↕	<b>Selector de versión de Android.</b> Establece la versión de Android para la que se está diseñando el <i>layout</i> . El Diseñador mostrará el <i>layout</i> de acuerdo con la versión de Android seleccionada.
Theme: Default Theme ↕	<b>Selector de tema.</b> Selecciona el tema de la interfaz de usuario para el <i>layout</i> .
	<b>Selector de configuración.</b> Selecciona la configuración del dispositivo tal como horizontal o vertical.
	<b>Opciones de Calificador de Recursos.</b> Abre un cuadro de diálogo que presenta menús desplegables para seleccionar Opciones de idioma, Modo de interfaz de usuario, Modo nocturno y opciones de redondez de pantalla.
	<b>Configuraciones del Action Bar.</b> Configura los ajustes de la barra de acciones para el <i>layout</i> .

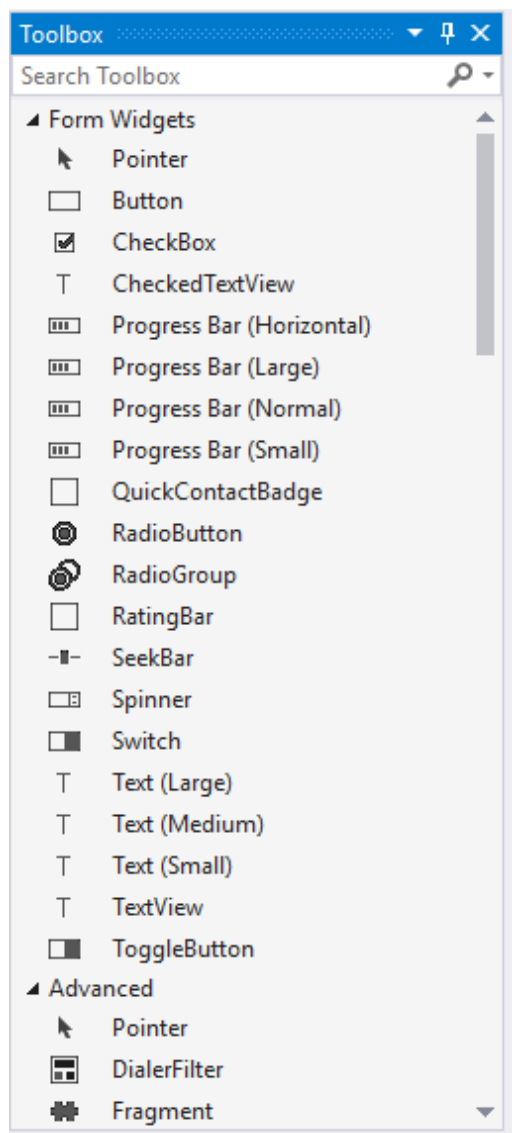


	<b>Editor de temas.</b> Abre el Editor de temas que permite personalizar elementos del tema seleccionado.
	<b>Material Design Grid.</b> Activa o desactiva la cuadrícula <i>Material Design</i> . El menú desplegable a la derecha abre un cuadro de diálogo que permite personalizar la cuadrícula.

En laboratorios posteriores exploraremos con más detalle cada uno de estos elementos.

## Caja de herramientas (ToolBox).

2. Selecciona la opción **View > Toolbox** de la barra de menús de Visual Studio para mostrar la caja de herramientas.



La Caja de Herramientas (Toolbox) proporciona una lista de Widgets y *Layouts* que podemos arrastrar y soltar sobre la superficie de diseño.





## Superficie de Diseño.

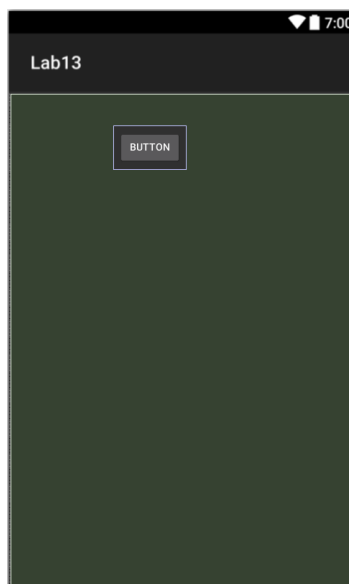
3. Examina los elementos que componen la superficie de diseño.

El Diseñador permite arrastrar y soltar Widgets de la Caja de Herramientas hacia la superficie de diseño.



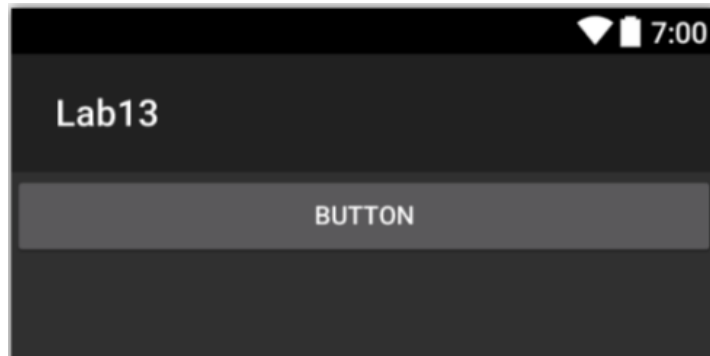
Cuando interactuamos con Widgets en el Diseñador, ya sea añadiendo nuevos Widgets o reposicionando los existentes, se muestran líneas verticales y horizontales para marcar los puntos de inserción disponibles.

4. Arrastra un Widget **Button** de la Caja de Herramientas hacia la superficie de diseño. Antes de soltar el botón podrás ver una imagen similar a la siguiente indicándote el área donde puedes insertar el Widget.



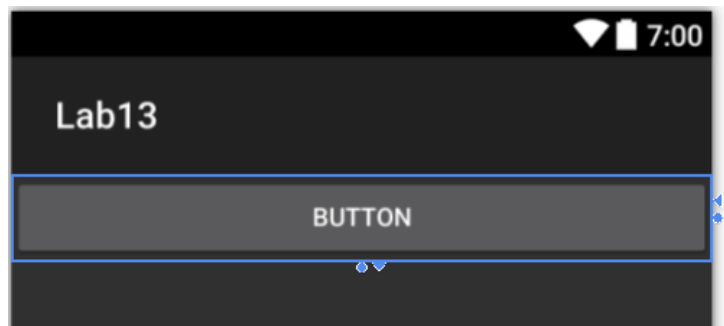


La siguiente imagen muestra la superficie de diseño después de soltar el Widget.

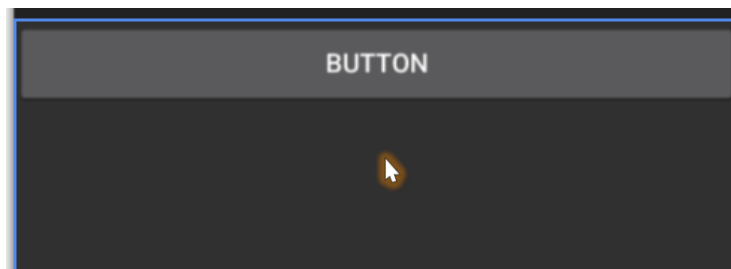


Adicionalmente, para insertar Widgets también podemos copiar uno existente utilizando las opciones *Copiar* y *Pegar*.

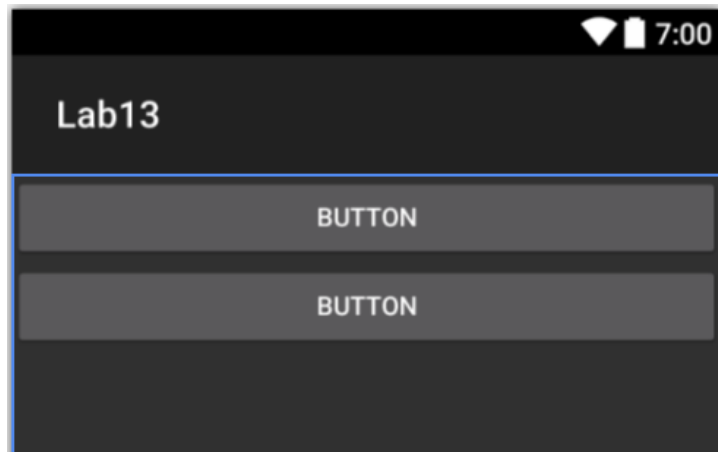
5. Haz clic en el *Button* que acabas de agregar. Nota que se muestra un recuadro sobre el elemento seleccionado.



6. Presiona las teclas *Ctrl-C* para copiar el Widget *Button*.
7. Haz clic en la superficie de diseño para quitar la selección del *Button*. El recuadro desaparece.

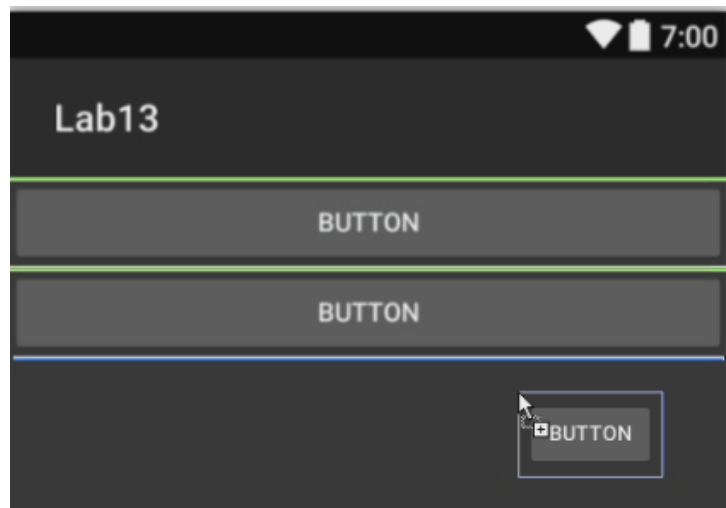


8. Presiona las teclas *Ctrl-V* para pegar la copia del *Button*. Ahora tendrás dos Widgets *Button*.

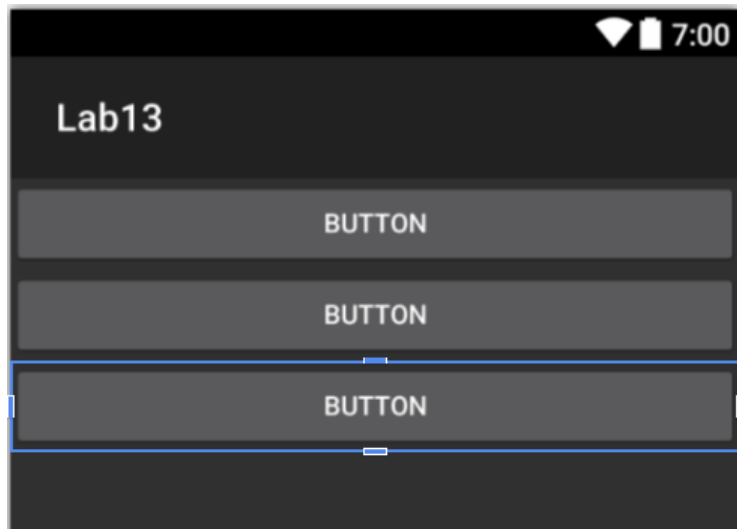


Para insertar un Widget también podemos arrastrar y soltar un Widget existente mientras presionamos la tecla *Ctrl*.

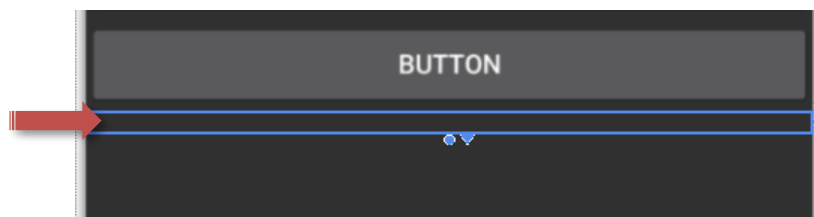
9. Presiona la tecla *Ctrl* y mantenla presionada mientras arrastras uno de los Widgets *Button* hacia otra área de la superficie de diseño. Podrás ver una imagen similar a la siguiente mientras arrastras el Widget.



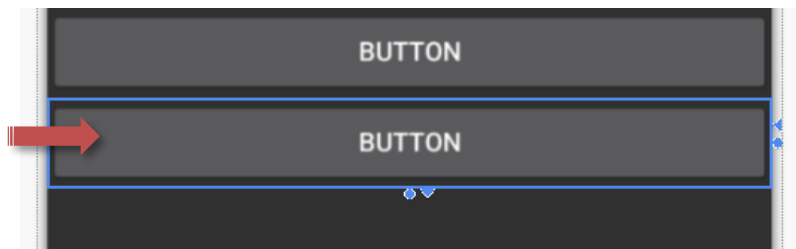
10. Suelta el Widget *Button* para insertarlo en la superficie de diseño.
11. Suelta la tecla *Ctrl*. Podrás ver el nuevo *Button* insertado.



12. Agrega un Widget *LinearLayout (Vertical)* debajo del 3er *Button*.

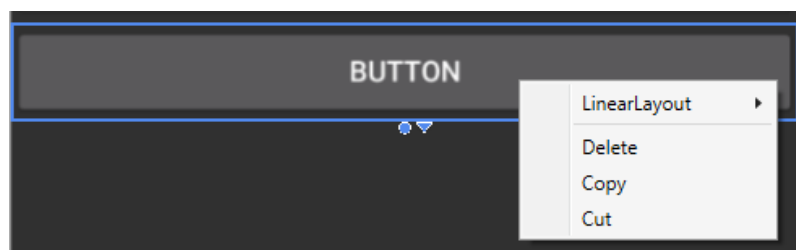


13. Agrega un nuevo Widget *Button* dentro del contenedor *LinearLayout* que acabas de agregar.



¿Puedes seleccionar el contenedor *LinearLayout (Vertical)* del nuevo *Button* agregado?  
Podría ser una tarea difícil.

14. Haz clic con el botón secundario del mouse sobre el primer Widget *Button* para mostrar su menú contextual.

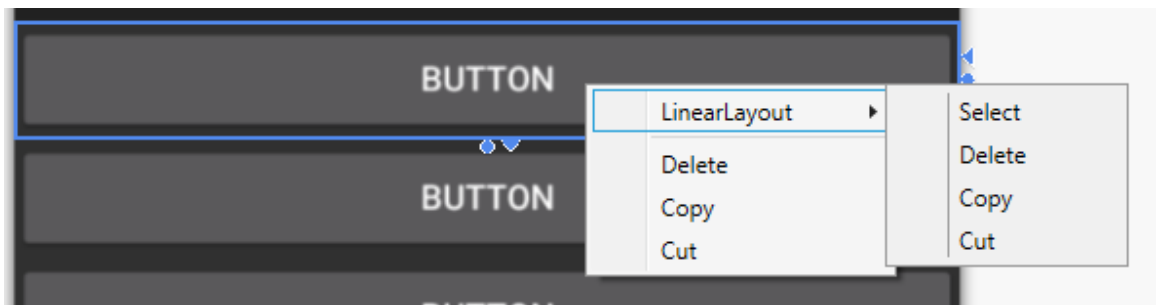




El menú contextual de un Widget muestra los comandos disponibles para el Widget seleccionado y su contenedor, facilitando la realización de operaciones en contenedores (que no siempre son fáciles de seleccionar sobre la superficie de diseño).

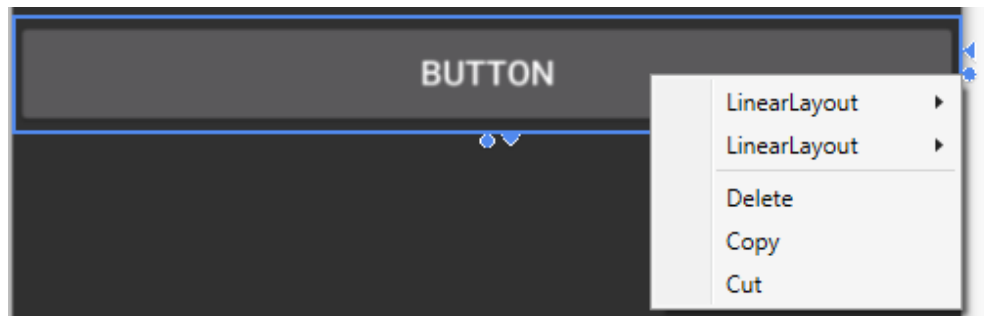
En este ejemplo, el menú contextual proporciona varias opciones:

- **LinearLayout.** Abre un submenú para editar el *LinearLayout* padre del Widget.



- **Delete, Copy y Cut.** Las operaciones que se pueden aplicar al Widget.

15. Abre el menú contextual del Widget que se encuentra dentro del *LinearLayout* que agregaste.




Puedes notar que el menú contextual te permite seleccionar su contenedor padre y el contenedor padre de su contenedor. Haz clic en cada uno para ver el resultado.

16. Localiza los **Controles de Zoom** sobre la superficie de diseño.

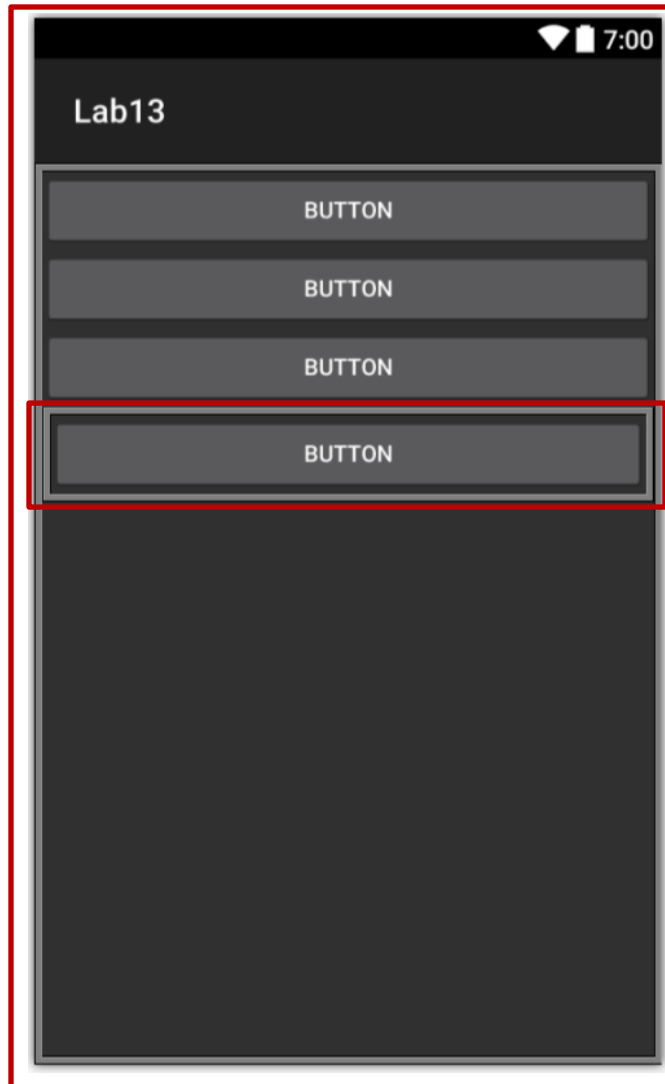



La superficie de diseño soporta Zoom a través de varios controles. Estos controles facilitan ver ciertas áreas de la interfaz de usuario en el Diseñador.



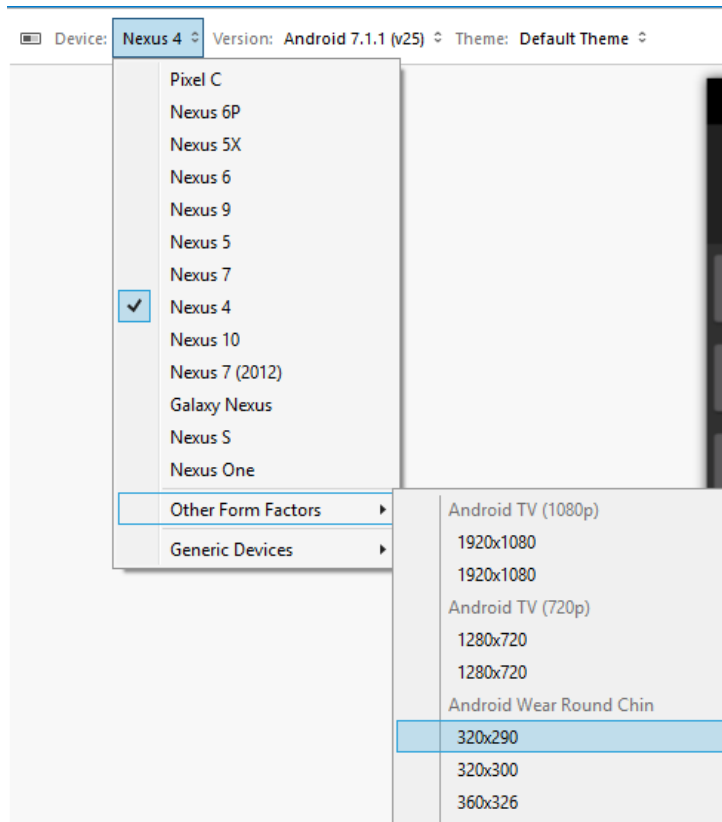
17. Haz clic sobre el control  (*Highlight Containers*).




Este control resalta los contenedores en la superficie de diseño para que sean más fáciles de localizar al momento de acercar y alejar. En este caso puedes ver fácilmente enmarcados los dos contenedores existentes en la interfaz de usuario.



18. Haz clic sobre el control  (*Normal Size*). Este control muestra el diseño para que pueda apreciarse la forma en que el diseño se verá en la resolución del dispositivo seleccionado.

19. Selecciona un dispositivo distinto y observa el diseño que es generado.



20. Haz clic sobre el control  (*Fit to Window*). Este control establece el nivel de Zoom para que todo el diseño sea visible en la superficie de diseño.
21. Haz clic sobre el control  (*Zoom In*). Este control realiza un acercamiento gradual con cada clic ampliando el diseño.
22. Haz clic sobre el control  (*Zoom Out*). Este control realiza un alejamiento gradual con cada clic haciendo que el diseño aparezca más pequeño.
23. Selecciona nuevamente el dispositivo Android inicial del diseño.

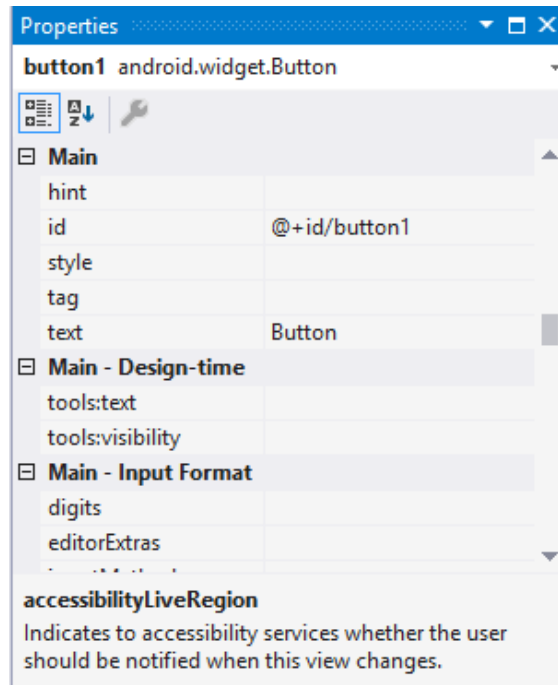
## Ventana de propiedades.

24. Haz clic en el primer Widget *Button* agregado para seleccionarlo.
25. Presiona la tecla **F4** para mostrar la ventana de propiedades.

El Diseñador soporta la edición de propiedades del Widget a través de la ventana **Properties**. Las propiedades que aparecen en la ventana *Properties* cambian dependiendo del Widget



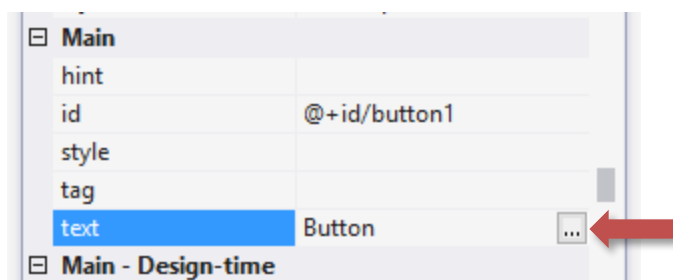
seleccionado en la superficie de diseño. La siguiente imagen muestra las propiedades del Widget *Button* seleccionado.



Las propiedades de la mayoría de los Widgets se muestran en blanco en la ventana de propiedades porque sus valores heredarán del tema Android seleccionado. La ventana de propiedades sólo muestra los valores que se establecen explícitamente para el Widget seleccionado. No mostrará valores heredados del tema.

Algunas propiedades pueden referenciar recursos que son definidos en archivos distintos al archivo .axml del diseño. Los casos más comunes de este tipo son los recursos *string* y *drawable*. Sin embargo, las referencias también se pueden utilizar para otros recursos, como valores *Bool* y *Dimension*. Cuando una propiedad soporta referencia de recurso, se muestra un botón de exploración (puntos suspensivos ...) junto a la caja de texto de la propiedad.

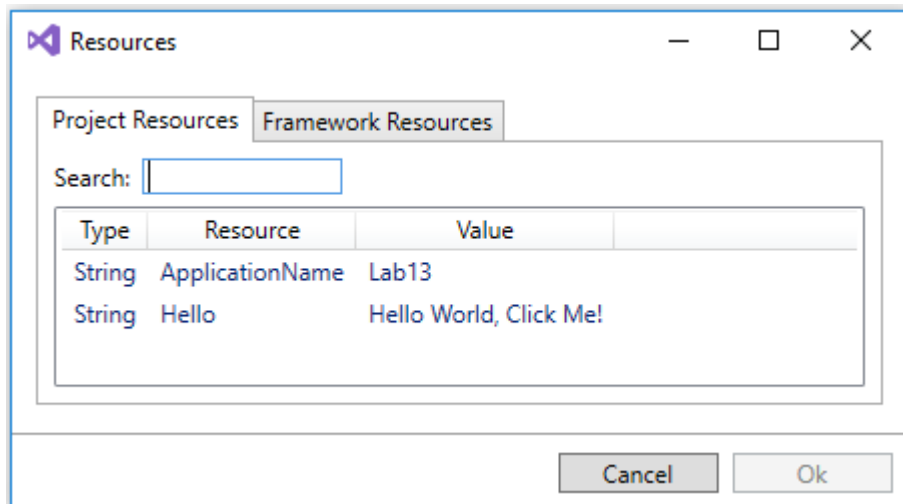
26. Localiza la propiedad **text** del *Button* seleccionado y haz clic en el botón de exploración.





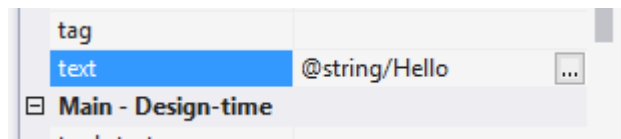


Puedes notar que, al hacer clic en el botón de exploración, se abre el selector de recursos mostrando los recursos disponibles para esa propiedad.



27. Selecciona el recurso *Hello* y haz clic en **Ok** para asignar el valor del recurso *Hello* a la propiedad **text** del *Button*.

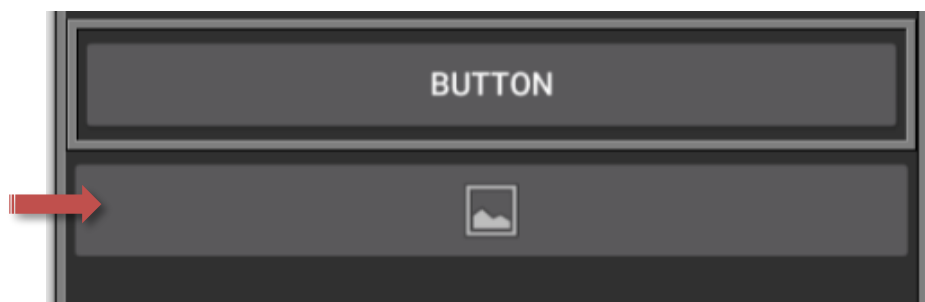
La ventana de propiedades mostrará el valor seleccionado de la propiedad **text**.



El diseñador también mostrará el valor del recurso *Hello* sobre la superficie del botón.



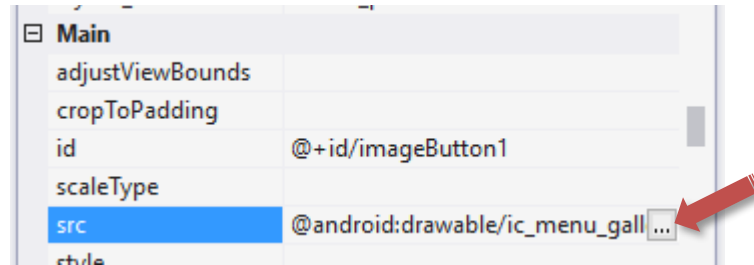
28. Agrega un Widget *ImageButton* a la superficie de diseño.



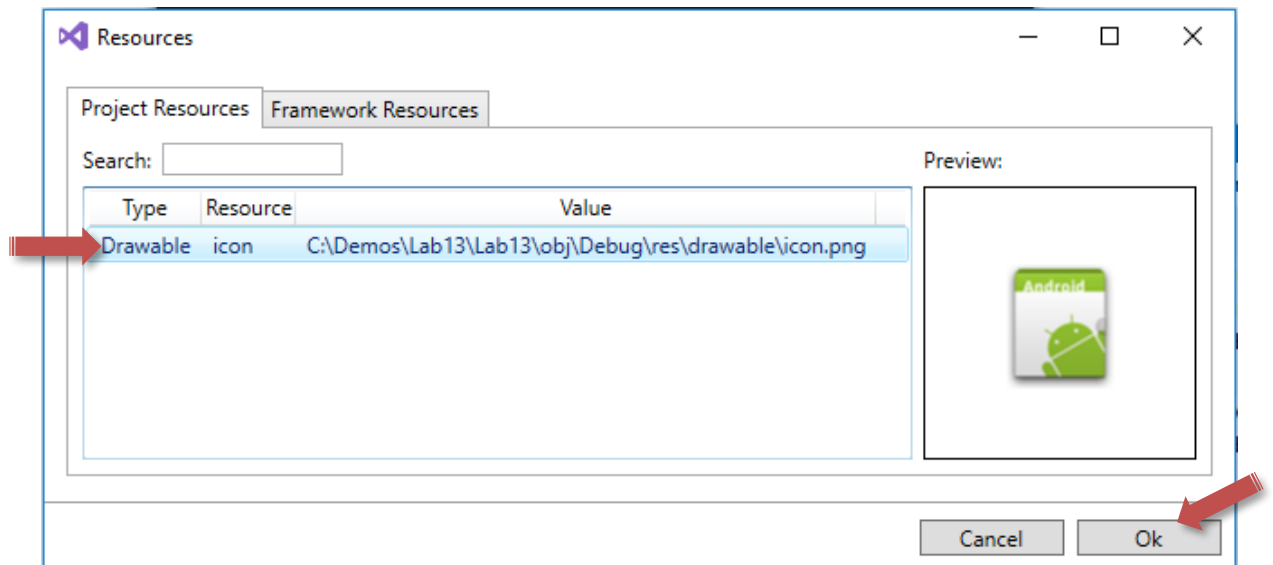


29. Localiza la propiedad **src** del *ImageButton* agregado.

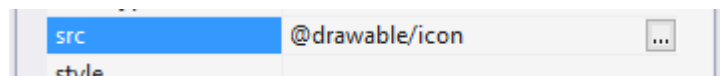
30. Haz clic en el botón de exploración para seleccionar el valor del recurso para la propiedad.



31. Selecciona el recurso *icon* y haz clic en **Ok** para agregar el valor a la propiedad.



32. La ventana de propiedades mostrará el recurso seleccionado.



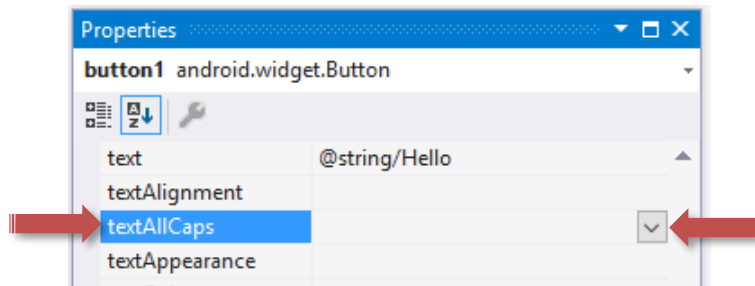
33. El diseñador mostrará el valor del recurso dentro del *ImageButton*.



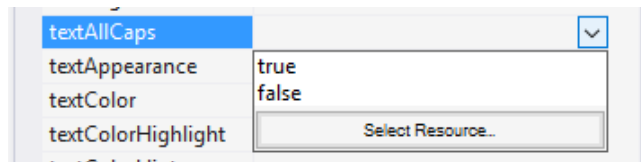
34. Localiza la propiedad **textAllCaps** del primer Widget *Button*.



35. Abre el menú desplegable de la propiedad.



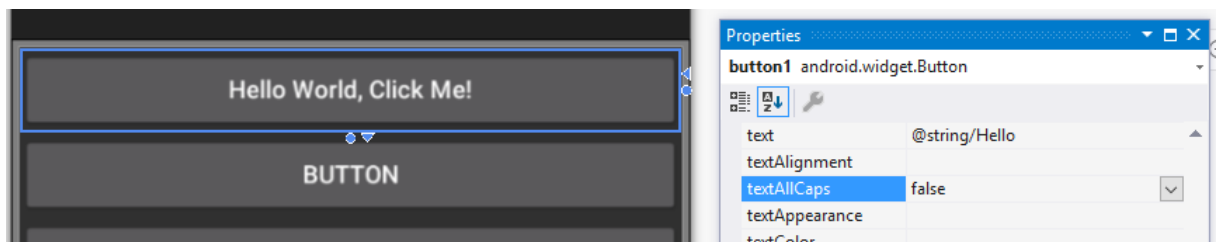
Las propiedades booleanas se seleccionan normalmente de un menú desplegable en la ventana de propiedades.



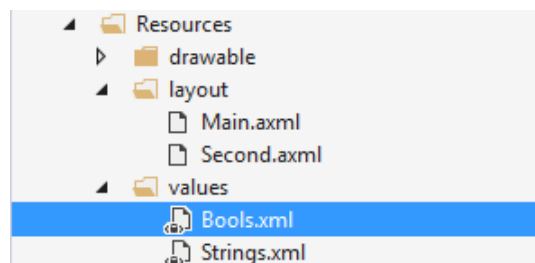
Podemos seleccionar un valor verdadero (*true*) o falso (*false*) o incluso podemos seleccionar una referencia a un recurso haciendo clic en **Select Resource...** También podemos introducir directamente el valor *true* o *false*.

36. Establece el valor *false* a la propiedad **textAllCaps**.

37. La ventana de propiedades mostrará el nuevo valor establecido y reflejará el cambio en el *Button*.



38. Utiliza la plantilla **XML File** para agregar un nuevo archivo XML al directorio *Resources\values* llamado *Bools.xml*.

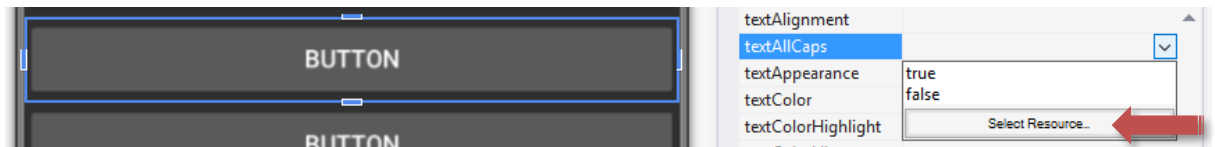




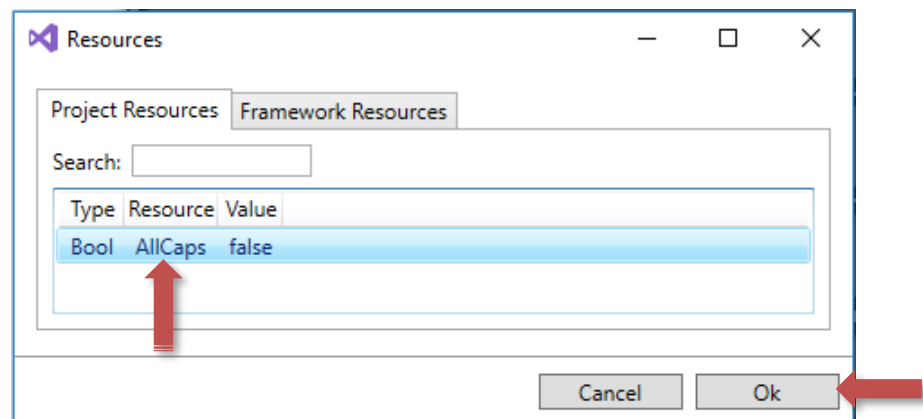
39. Agrega el siguiente código al archivo *Bools.xml* para definir un recurso de tipo *Bool* llamado *AllCaps* con valor *false*.

```
<?xml version="1.0" encoding="utf-8" ?>  
  
<resources>  
  <bool name="AllCaps">false</bool>  
</resources>
```

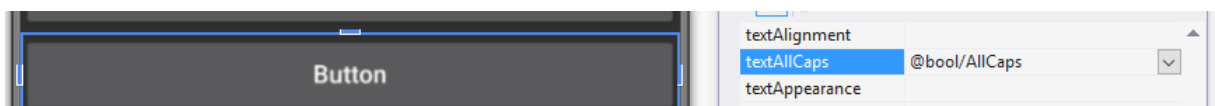
40. Guarda y cierra el archivo.
41. Regresa al diseño *Main.xml* y selecciona el segundo *Button*.
42. Localiza la propiedad ***textAllCaps*** y abre el menú desplegable de la propiedad.
43. Selecciona la opción ***Select Resource...***



44. Selecciona el recurso booleano *AllCaps* que definiste previamente y haz clic en ***Ok*** para asignar el valor del recurso a la propiedad.



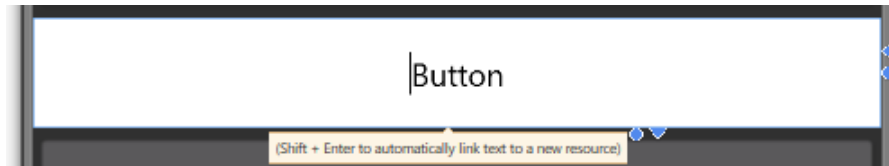
45. La ventana de propiedades mostrará el nuevo valor establecido y reflejará el cambio en el *Button*



El diseñador de Android soporta la edición directa de ciertas propiedades sobre la superficie de diseño por lo que no es necesario buscarlas en la lista de propiedades. Las propiedades que se pueden editar directamente incluyen Texto, Margen y Tamaño.



46. Haz doble clic sobre el segundo *Button*.



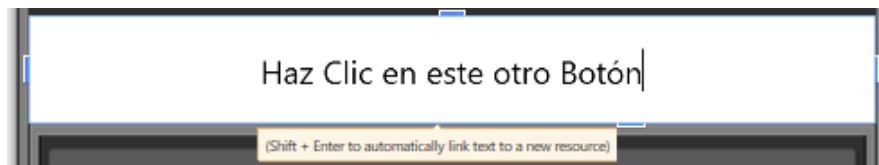
Las propiedades de texto de algunos Widgets como *Button* y *TextView*, se pueden editar directamente en la superficie de diseño. Al hacer doble clic en un Widget lo pondrá en modo de edición como se muestra en la imagen anterior. En modo de edición podemos introducir un nuevo valor de texto o bien, introducir un nuevo recurso *String*.

47. Escribe el texto **Haz Clic en este botón** y presiona la tecla *Enter* para finalizar la edición. La ventana de propiedades mostrará el nuevo valor asignado.



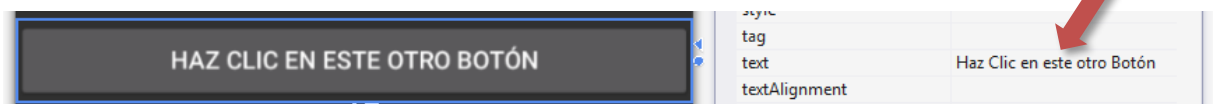
48. Haz doble clic en el tercer botón para entrar en modo de edición.

49. Escribe el texto **Haz Clic en este otro Botón**.



Cuando ingresamos una nueva cadena de texto podemos presionar las teclas *Shift + Enter* para vincular automáticamente el texto ingresado a un nuevo recurso.

50. Presiona las teclas *Shift + Enter* para finalizar la edición. La ventana de propiedades mostrará el nuevo valor asignado.

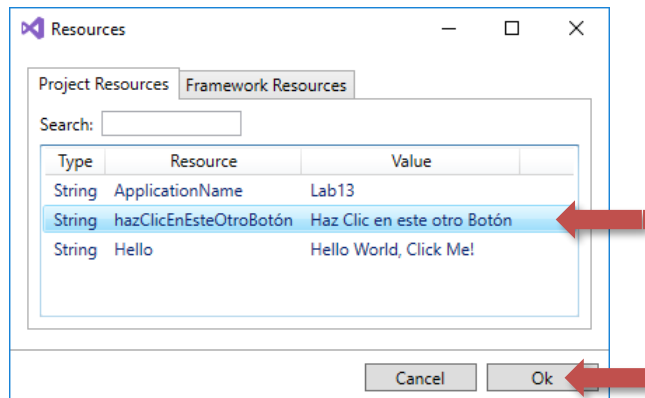


51. Localiza la propiedad **text** del cuarto *Button* y presiona el botón de exploración de recursos.





52. Selecciona el recurso que ligaste anteriormente y haz clic en **Ok** para aceptar el recurso.



La ventana de propiedades mostrará el nuevo valor asignado.



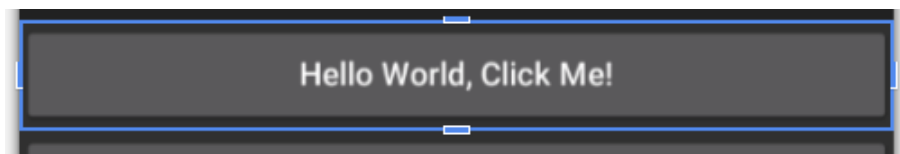
53. Guarda los cambios.

54. Abre el archivo *Resources\values\Strings.xml*. Puedes ver el nuevo recurso agregado.

```
<resources>
  <string name="Hello">Hello World, Click Me!</string>
  <string name="ApplicationName">Lab13</string>
  <string name="hazClicEnEsteOtroBotón">Haz Clic en este otro Botón</string>
</resources>
```

55. Cierra el archivo y regresa al diseñador de Android.

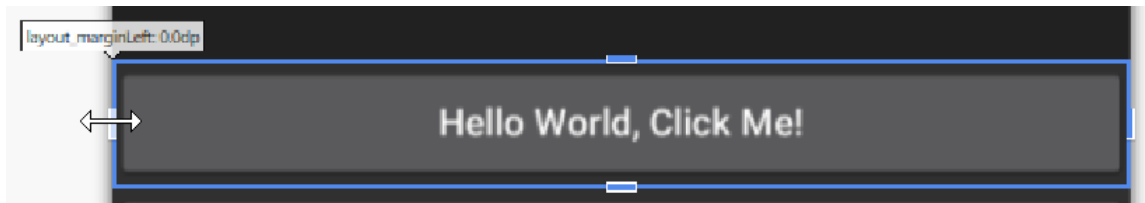
56. Selecciona el primer Widget *Button*.



Cuando seleccionamos un Widget, el Diseñador muestra los controles que nos permiten cambiar el tamaño o el margen del Widget de forma interactiva. Al hacer clic en el Widget mientras esta seleccionado, alterna su estado entre el modo de edición de márgenes y el modo de edición de tamaño.

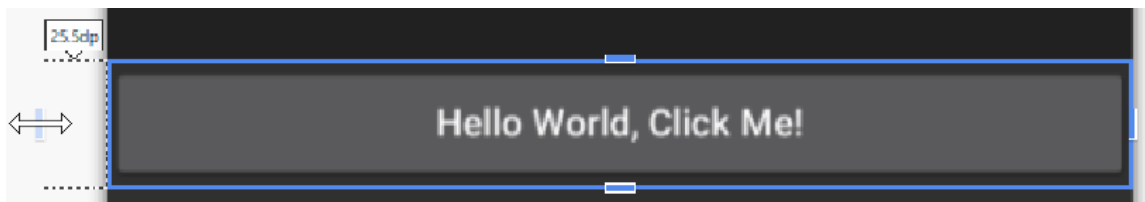
Al hacer clic en un Widget por primera vez, se muestran los indicadores de margen.

57. Mueve el mouse hacia el indicador de margen izquierdo.

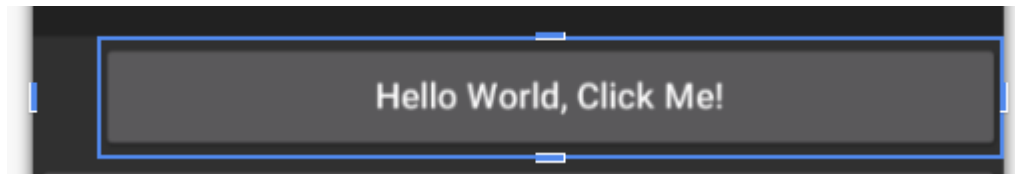


Puedes notar que el cursor cambia de forma y que el Diseñador muestra la propiedad que será modificada, en este caso la propiedad que será modificada es `layout_marginLeft`.

58. Arrastra el marcador hacia la izquierda y suéltalo cuando el valor de la propiedad `layout_marginLeft` este en 25dp aproximadamente.



Al soltar el marcador, podrás ver que se ha establecido un margen izquierdo para el Widget.

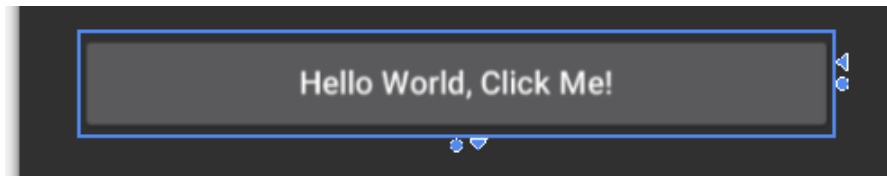


59. Repite el proceso anterior para agregar al primer *Button* un margen superior, derecho e inferior de aproximadamente 25dp.

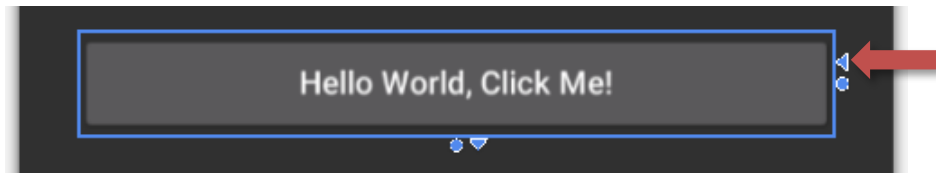


Puedes notar que cuando se ha establecido un margen, el diseñador muestra con líneas punteadas el espacio que ocupa el margen.

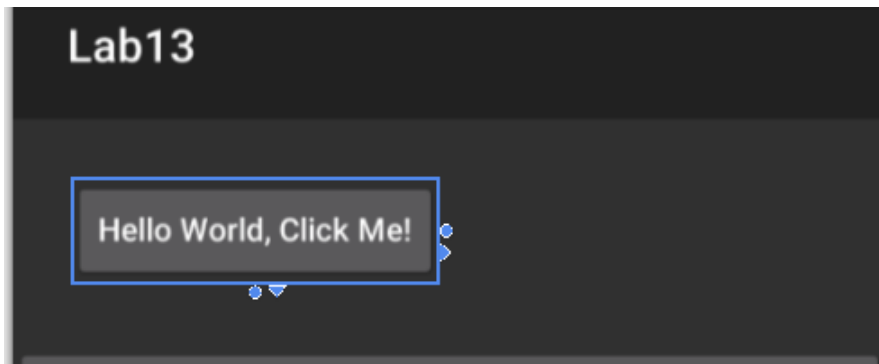
60. Estando en el modo de edición de márgenes, haz clic sobre el *Button*. Podrás notar que el *Button* cambia al modo de edición de tamaño.



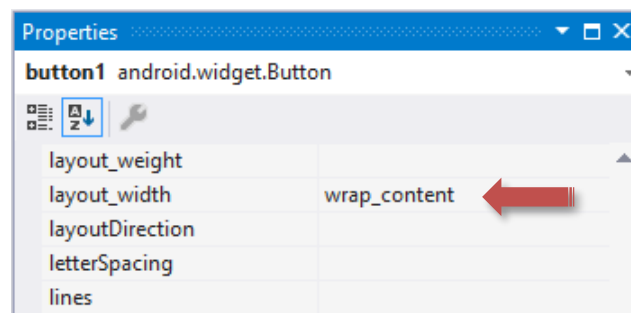
61. Haz clic en el marcador **Wrap Content**.



Puedes notar que el tamaño del Widget es reducido al tamaño mínimo necesario para mostrar el contenido.



También puedes observar que el valor de su propiedad **layout\_width** se ha establecido con el valor **wrap\_content**.



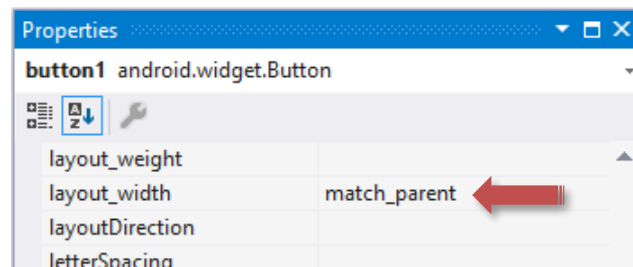
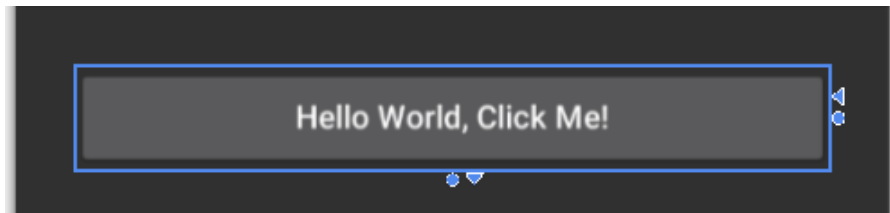
Nota que cuando el valor del tamaño es establecido en **wrap\_content**, el diseñador muestra un marcador triangular que apunta en la dirección opuesta.



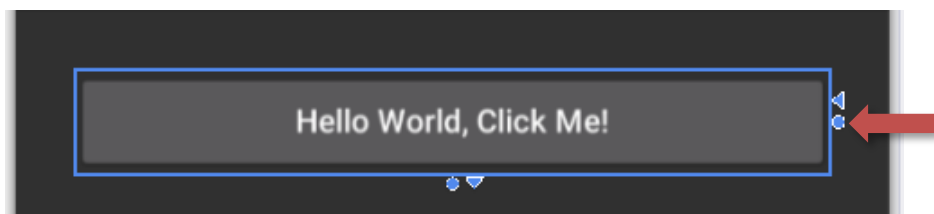




62. Haz clic en el marcador triangular que apunta en la dirección opuesta. Puedes notar que el tamaño cambia a *match\_parent* ajustándose a las dimensiones del contenedor.

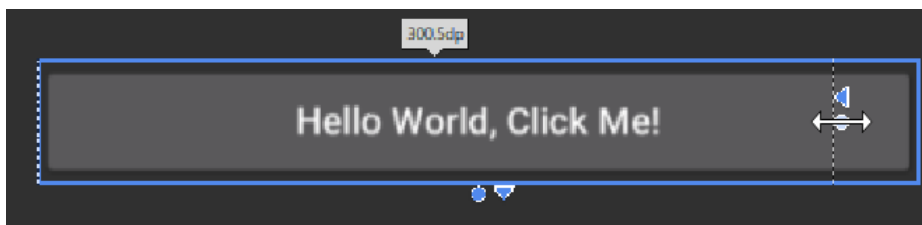


63. Coloca el cursor del mouse sobre el marcador circular de la derecha.



Este marcador nos permite cambiar el tamaño del Widget a un valor *dp* arbitrario.

64. Arrastra el marcador circular de tamaño para asignarle un valor aproximado de 300 dp.



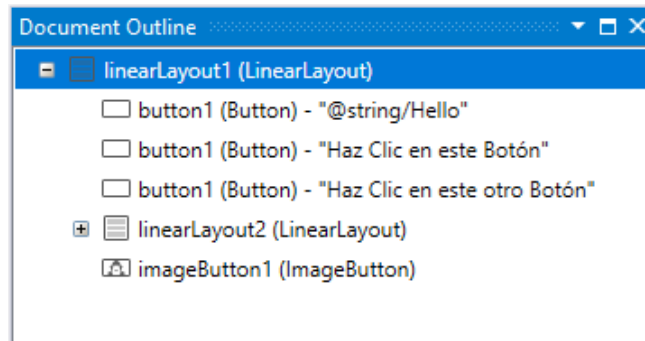
Al soltar el mouse podrás ver su nuevo tamaño.





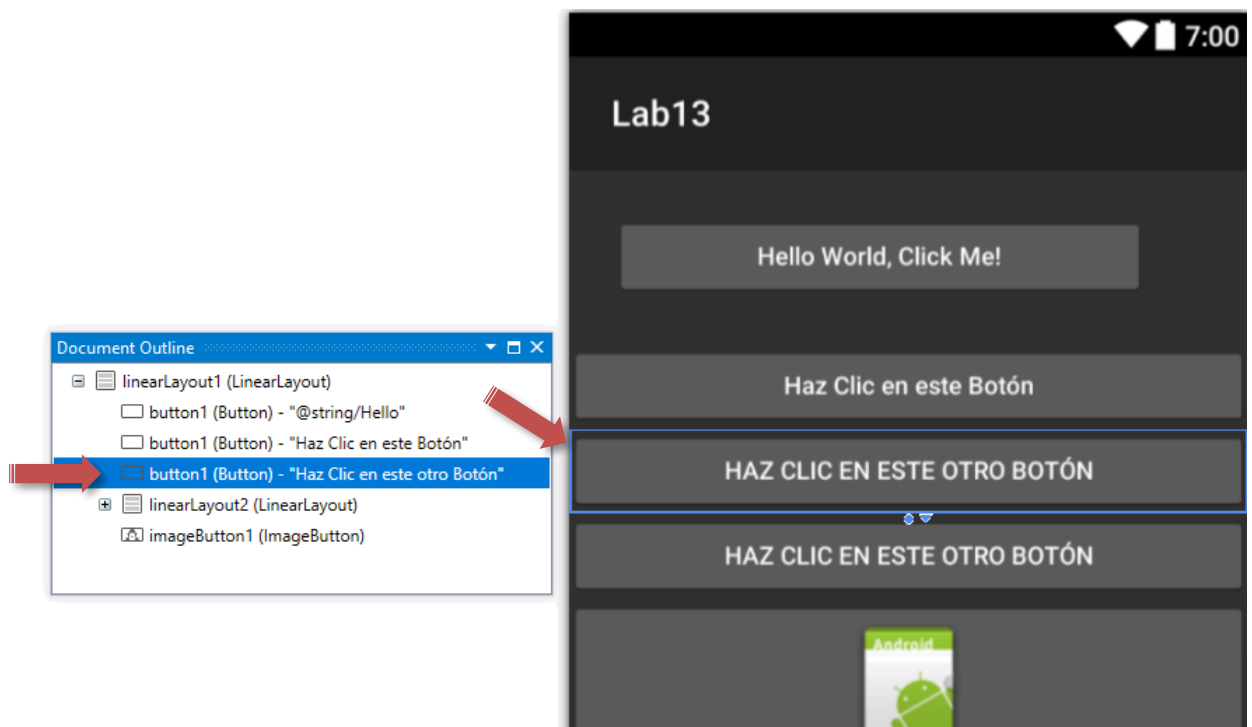
## Esquema del documento (Document Outline).

65. Selecciona la opción **View > Other Windows > Document Outline** de la barra de menús de Visual Studio para mostrar la ventana **Document Outline**.



Como puedes notar, la ventana *Document Outline* muestra la jerarquía de Widgets del layout.

66. Haz clic en el 3er *Button*. Puedes notar que el contorno del Widget seleccionado también es resaltado en la superficie de diseño.

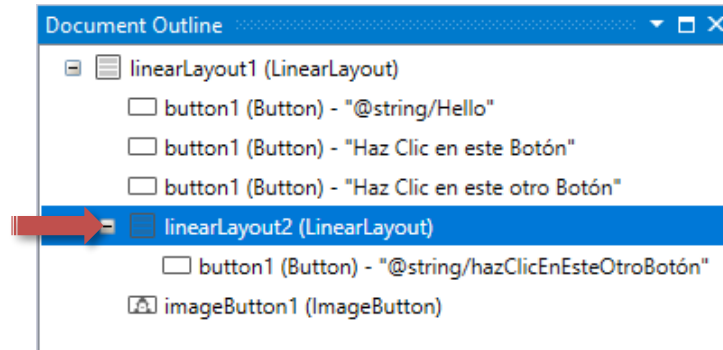


El Widget seleccionado en la ventana *Document Outline* permanece sincronizado con su contraparte en la superficie de diseño. Esto es útil para seleccionar Widgets que no siempre son fáciles de seleccionar en la superficie de diseño.

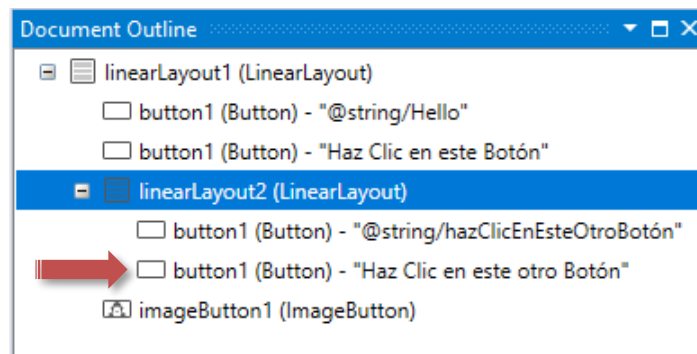


*Document Outline* soporta las características de *Copiar y Pegar*. También soporta la característica de *Arrastrar y Soltar*.

67. Expande el contenedor *LinearLayout* que contiene un único *Button*.



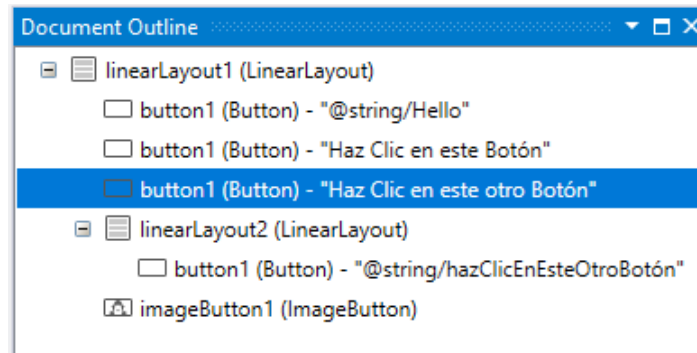
68. En la ventana *Document Outline*, arrastra el tercer *Button* y suéltalo debajo del botón actual dentro del Widget *LinearLayout2*.



Puedes notar que esta acción te permitió mover un Widget a otra posición.

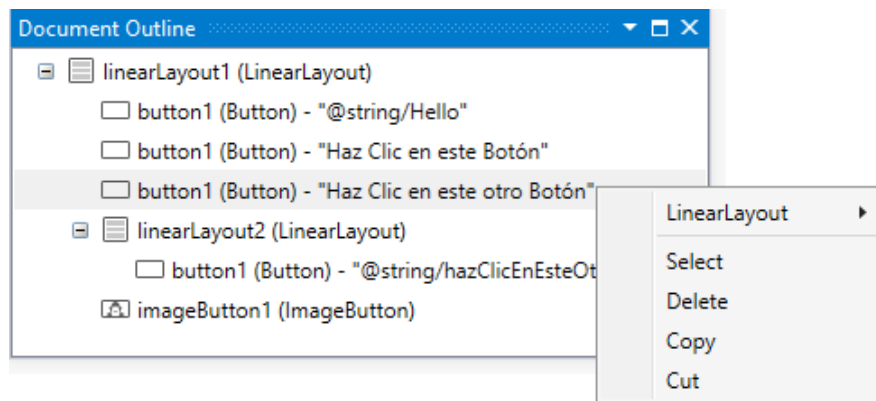
La característica de *Arrastrar y Soltar* es soportada desde la ventana *Document Outline* hacia la superficie de diseño, así como desde la superficie de diseño hacia la ventana *Document Outline*.

69. Desde la superficie de diseño, arrastra el *Button* que moviste y suéltalo en su posición original dentro de la venta *Document Outline*.



Puedes notar que el *Button* puede colocarse fácilmente mediante las acciones de *Arrastrar* y *Soltar* dentro de la ventana *Document Outline* o utilizando la ventana *Document Outline* y la superficie de diseño simultáneamente.

70. En la ventana *Document Outline*, haz clic con el botón secundario del mouse sobre el tercer *Button*.



Puedes notar que el menú contextual también se encuentra disponible dentro de la ventana *Document Outline*, el mismo menú contextual que aparece al hacer clic con el botón secundario del mouse en ese mismo *Widget* sobre la superficie de diseño.

71. Guarda los cambios.
72. Modifica el archivo *MainActivity* para que al ejecutarse la aplicación muestre el *Layout Main*.
73. La pantalla que debe mostrarse será similar a la siguiente.





## 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 **SALLab13.dll** que implementa la capa de acceso a servicio. El archivo **SALLab13.dll** se encuentra disponible junto con este documento.

1. En el proyecto Xamarin.Android, agrega una referencia del ensamblado **SALLab13.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**.

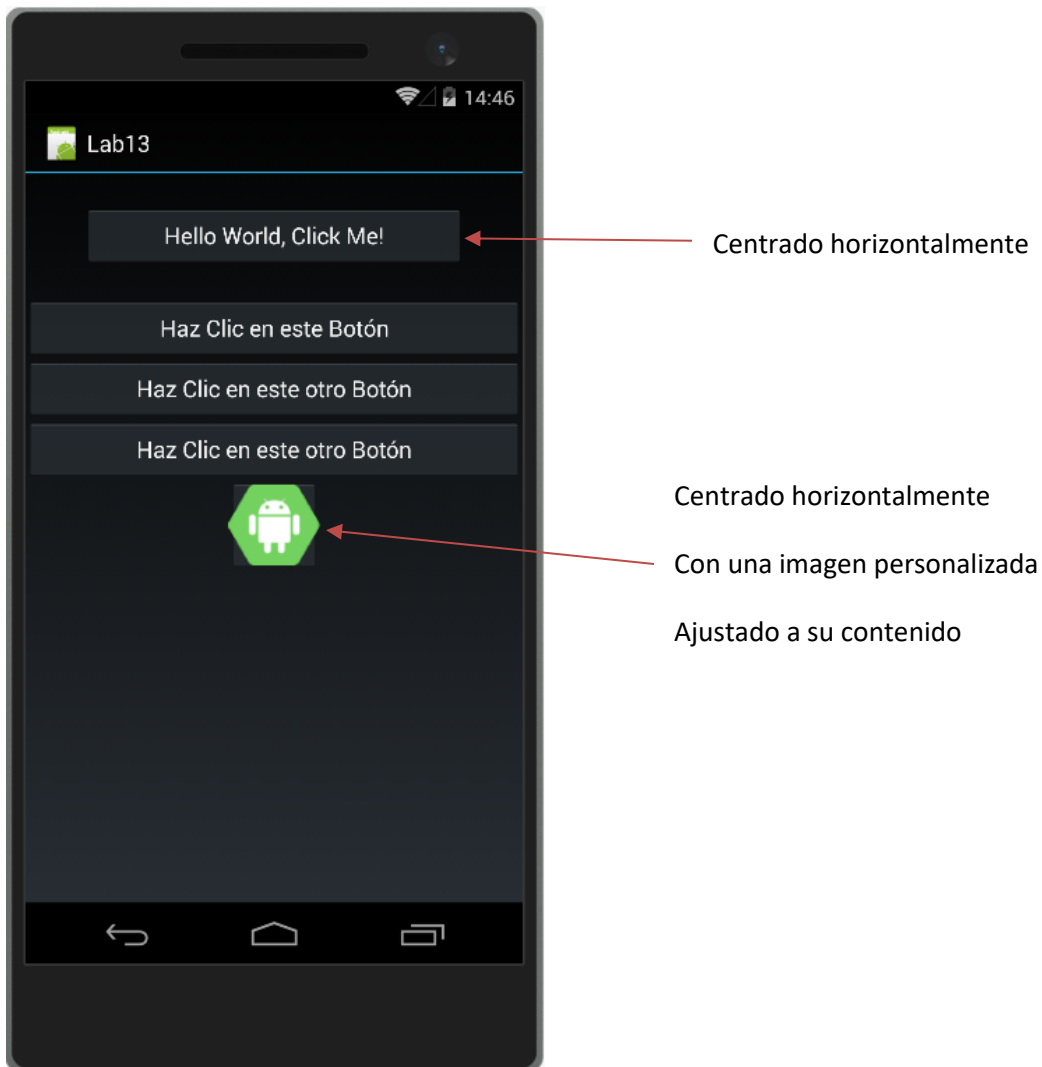
### Tarea 2. Modificar la interfaz de usuario.

Realiza los cambios necesarios para:

- a) Centrar horizontalmente en la pantalla el primer Widget *Button*.
- b) Asignar una imagen personalizada al Widget *ImageButton*.
- c) Ajustar el *ImageButton* al tamaño de su contenido.
- d) Centrar horizontalmente en la pantalla el Widget *ImageButton*.

**TIP:** Para centrar horizontalmente un Widget puedes establecer su propiedad *layout\_gravity* con el valor *center\_horizontal*.

Al ejecutar la aplicación se deberá mostrar una pantalla similar a la siguiente.



### 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 *ImageButton* se ejecute el siguiente código que realizará la autenticación y mostrará el resultado en un cuadro de Diálogo.



```
var Client = new SALLab13.ServiceClient();
string Email = ""; /* Aquí pon tu correo */
string Password = ""; /* Aquí pon tu contraseña */
var Result = await Client.ValidateAsync(this, Email, Password);

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();
```

Cuando tu actividad se haya validado exitosamente puedes ver el estatus en el siguiente enlace:  
<https://ticapacitacion.com/evidencias/xamarin30>.

La evidencia donde se evalúa tu actividad es “Ejecución del laboratorio 13”.

**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 exploramos las características del *Diseñador de Xamarin.Android*. Describimos los conceptos básicos del diseñador y mostramos la forma de utilizarlo para agregar Widgets de forma visual y la forma de modificar las propiedades de los Widgets.

¿Qué te pareció este laboratorio?

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