



---

# Programando con B4X

## Tema 13 – XUI Views

Version 1.0, abril 2021

Autor original: [Prokopis Leon](#)

Traducido al español por [José Miguel López](#)



Anywhere Software

## Tema 13 – XUI Views

🕒 3h

Lo que los estudiantes aprenderán

- Qué es una biblioteca
- Biblioteca XUI
- Dialogs – Diálogos
- Plantillas

Los lenguajes de programación poseen bibliotecas (es frecuente que se traduzca incorrectamente al español como “librerías”). En general, una biblioteca incluye bloques de código, estructuras, clases y métodos que pueden usar los programadores para facilitar el proceso de desarrollo de programas.

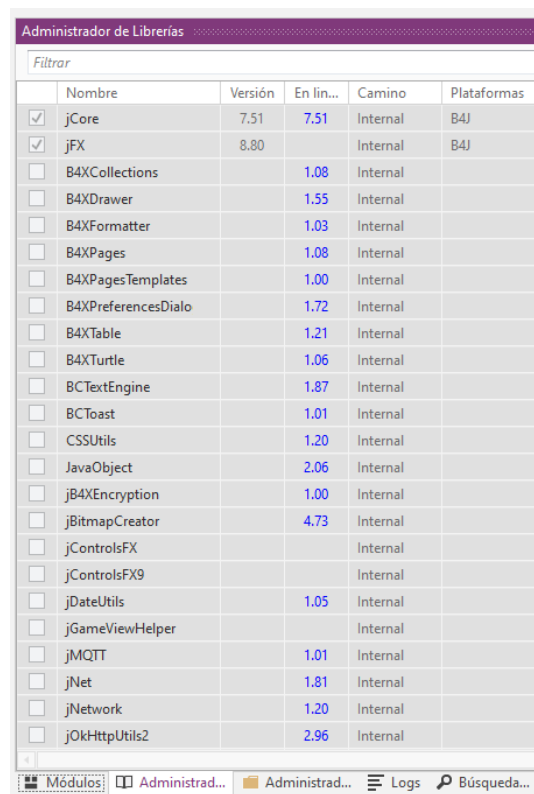
### Bibliotecas en in B4X

Las bibliotecas se suelen dividir en internas (se instalan junto con el propio lenguaje) y externas (creadas por el propio programador u obtenidas de otras fuentes como GitHub). Para usar una biblioteca, únicamente hay que elegirla de la lista de Bibliotecas de B4X.

La lista de bibliotecas ofrece información útil como:

- La versión actual de la biblioteca que está instalada,
- La última versión publicada para poder actualizar,
- Si es una biblioteca interna o externa,
- En qué plataforma funciona.

Para usar una biblioteca debes saber qué hace y cuáles son sus métodos y datos. Cada biblioteca hay información importante sobre su uso en la web del lenguaje en <https://www.b4x.com/android/documentation.html>.



Administrador de Librerías					
Filtrar					
	Nombre	Versión	En lin...	Camino	Plataformas
<input checked="" type="checkbox"/>	jCore	7.51	7.51	Internal	B4J
<input checked="" type="checkbox"/>	jFX	8.80		Internal	B4J
<input type="checkbox"/>	B4XCollections		1.08	Internal	
<input type="checkbox"/>	B4XDrawer		1.55	Internal	
<input type="checkbox"/>	B4XFormatter		1.03	Internal	
<input type="checkbox"/>	B4XPages		1.08	Internal	
<input type="checkbox"/>	B4XPagesTemplates		1.00	Internal	
<input type="checkbox"/>	B4XPreferencesDialog		1.72	Internal	
<input type="checkbox"/>	B4XTable		1.21	Internal	
<input type="checkbox"/>	B4XTurtle		1.06	Internal	
<input type="checkbox"/>	BCTextEngine		1.87	Internal	
<input type="checkbox"/>	BCToast		1.01	Internal	
<input type="checkbox"/>	CSSUtils		1.20	Internal	
<input type="checkbox"/>	JavaObject		2.06	Internal	
<input type="checkbox"/>	JB4XEncryption		1.00	Internal	
<input type="checkbox"/>	jBitmapCreator		4.73	Internal	
<input type="checkbox"/>	jControlsFX			Internal	
<input type="checkbox"/>	jControlsFX9			Internal	
<input type="checkbox"/>	jDateUtils		1.05	Internal	
<input type="checkbox"/>	jGameViewHelper			Internal	
<input type="checkbox"/>	jMQTT		1.01	Internal	
<input type="checkbox"/>	jNet		1.81	Internal	
<input type="checkbox"/>	jNetwork		1.20	Internal	
<input type="checkbox"/>	JOkHttpUtils2		2.96	Internal	

Imagen 1. Administrador de Bibliotecas

Las bibliotecas externas deben copiarse en una carpeta específica de tu ordenador. Desde el menú "Herramientas", "Configurar Rutas", "Librerías Adicionales" puedes elegir la carpeta donde se guardarán.



#### Consejo para el profesor

Más abajo hay algunas "views" importantes. Hay algunas que no están, como el ComboBox, aunque las veremos con más detalle en próximos temas.

## La Biblioteca "XUI Views"

El objetivo de la biblioteca XUI Views es permitir crear aplicaciones para B4J, B4A and B4i de forma similar.

Esta biblioteca permite crear formularios y vistas (views), a saber:

- **Views** (objetos)
  - **B4XComboBox** - ComboBox / Spinner / ActionSheet multi-plataforma.
  - **ScrollingLabel** - Una etiqueta que permite desplazar el texto cuando es más ancho que la etiqueta.
  - **AnotherProgressBar** - Barra de progreso Vertical u horizontal animada.
  - **B4XLoadingIndicator** - 6 distintos indicadores de carga animados.
  - **RoundSlider** - Un deslizador redondeado.
  - **SwiftButton** - Botón 3D
  - **AnimatedCounter**
  - **B4XFloatTextField** - Un TextField / EditText con un consejo flotante.
  - **B4XSwitch** - Un bonito interruptor.
  - **B4XPlusMinus** - Permite al usuario elegir un número o un ítem de una lista previamente fijada.
  - **B4XBreadCrumb** - Control de navegación.
  - **B4XSeekBar** - Barra de búsqueda / deslizador horizontal o vertical.
  - **MadeWithLove** - Muestra tu amor por B4X :)
  - **B4XImageView** - ImageView con útiles modos de redimensionado.
  - **XUIViewsUtils** - Módulo con código estático de métodos útiles.
- **Dialogs** (Cuadros de diálogo)
  - Una clase que ofrece los servicios necesarios para mostrar un diálogo. Hay 3 métodos para mostrar diálogos: Mostrar - Mostrar un diálogo simple con texto, ShowCustom - Permite que se le envíe un diseño personalizado para mostrarlo en el diálogo, ShowTemplate - Muestra un diálogo basado en una clase plantilla. Puedes ver el código fuente para la estructura de la plantilla. Es bastante sencillo.



- **Templates (Plantillas)**

- B4XDateTemplate – Basado en AnotherDatePicker.
- B4XColorTemplate – Selector de colores.
- B4XLongTextTemplate – Texto desplazable.
- B4XListTemplate – Una lista de ítems. El usuario puede elegir uno de los ítems.
- B4XSignatureTemplate – Captura la firma del usuario y añade una marca temporal a la imagen de mapa de bits.
- B4XInputTemplate – Plantilla para entradas de texto y numéricas.
- B4XSearchTemplate – Una lista con un campo de búsqueda.
- B4XTimedTemplate – Una plantilla que agrupa otras plantillas y crea un diálogo que se cierra automáticamente tras un tiempo prefijado con una barra de progreso bonita y animada.

## Uso de XUI Views

Para usar las XUI Views primero hay que comprobar su nombre en la pestaña del Administrador de Librerías. Después vamos al diseñador para crear los objetos que quieras.



### Ejemplo

El programa "Ejemplo" muestra el uso de:

**ScrollingLabel**

**B4XFloatTextField**

**RoundSlider**

**AnotherProgressBar**

**B4XSwitch**

**B4XImageView**

## Etiqueta desplazable

Una etiqueta desplazable es una etiqueta donde el texto se puede desplazar para que se pueda mostrar completamente.

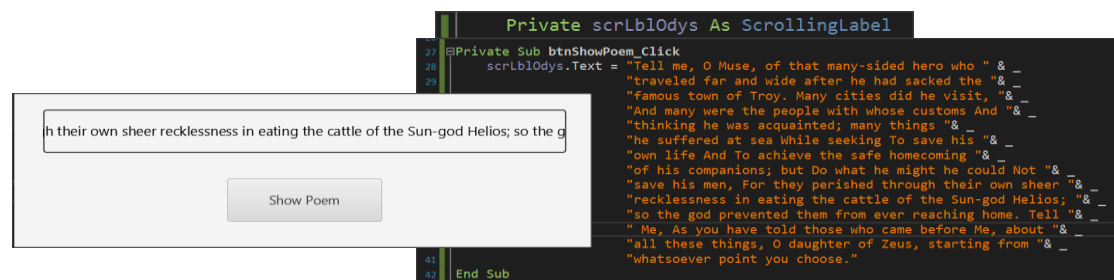


Imagen 2 Etiqueta desplazable

Se usa normalmente para textos muy grandes o para mostrar un mensaje móvil en la pantalla.



## B4XFloatTextField

B4XFloatTextField crea un cuadro de texto en pantalla para introducir datos. Se usa de formar parecida a un cuadro de texto simple, pero muestra una

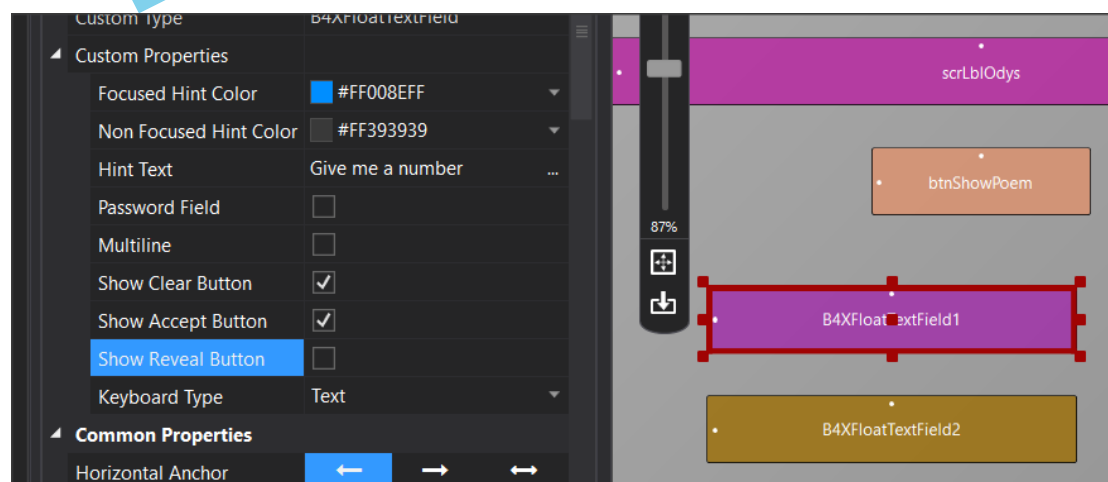


Imagen 3. B4XFloatTextField

etiqueta que ayuda al usuario a identificar el cuadro de texto sin tener que introducir una etiqueta adicional antes del mismo (Hint Text).

Además, permite mostrar los controles necesarios para borrar y activar el cuadro de texto (Show Clear Button, Show Accept button). Finalmente, con la propiedad "Keyboard Type" permite que se introduzca texto, números enteros o decimales.

## RoundSlider

RoundSlider es un marcador que se puede mover con el ratón alrededor de un círculo. Para cada punto en el círculo muestra un valor dentro del rango de valores especificado en las propiedades Minimum – Maximum en el Diseñador.

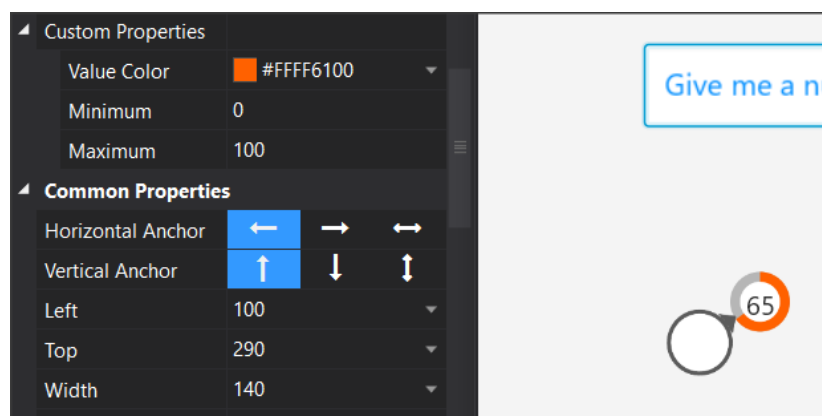


Imagen 4 RoundSlider

Cada vez que el valor del RoundSlider se cambia, se dispara el evento "\_ValueChanged" y genera un valor que puede usar el programador.

## AnotherProgressBar

AnotherProgressBar muestra una barra con valores entre 0 y 100 y se usa para representar porcentaje u otras proporciones. Su uso es sencillo; tras declarar la variable que representa al objeto, se usa la propiedad ".value" para fijar un valor.

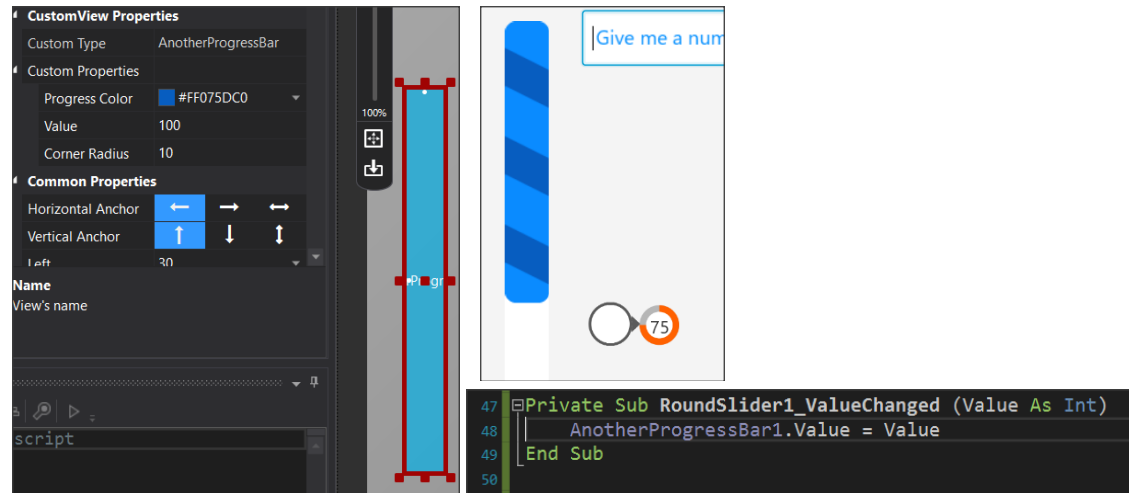


Imagen 5. AnotherProgressBar

En el ejemplo de la imagen, progressBar varía en función del valor recibido de RoundSlider.

## B4XSwitch

B4XSwitch crea un interruptor en la pantalla.

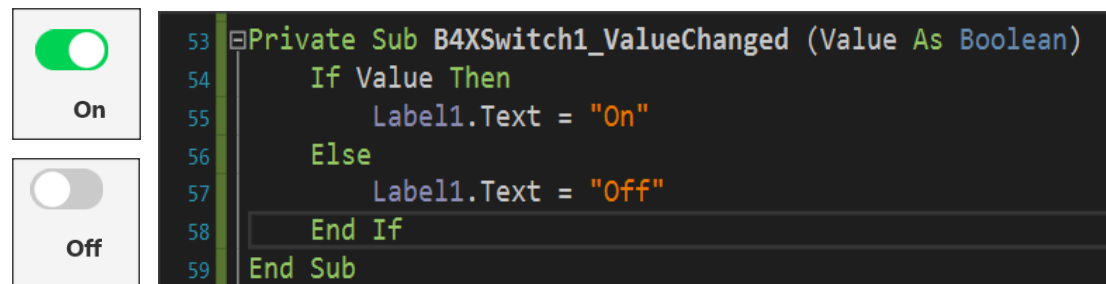


Imagen 6. B4XSwitch

En el ejemplo de la imagen, cuando el valor devuelto es True se muestra el texto "On" en la etiqueta "Label1"; en caso contrario muestra "Off".

## B4XImageView

B4XImageView muestra una imagen. Permite elegir la imagen que se mostrará mediante código o mediante las opciones del objeto en el Diseñador.

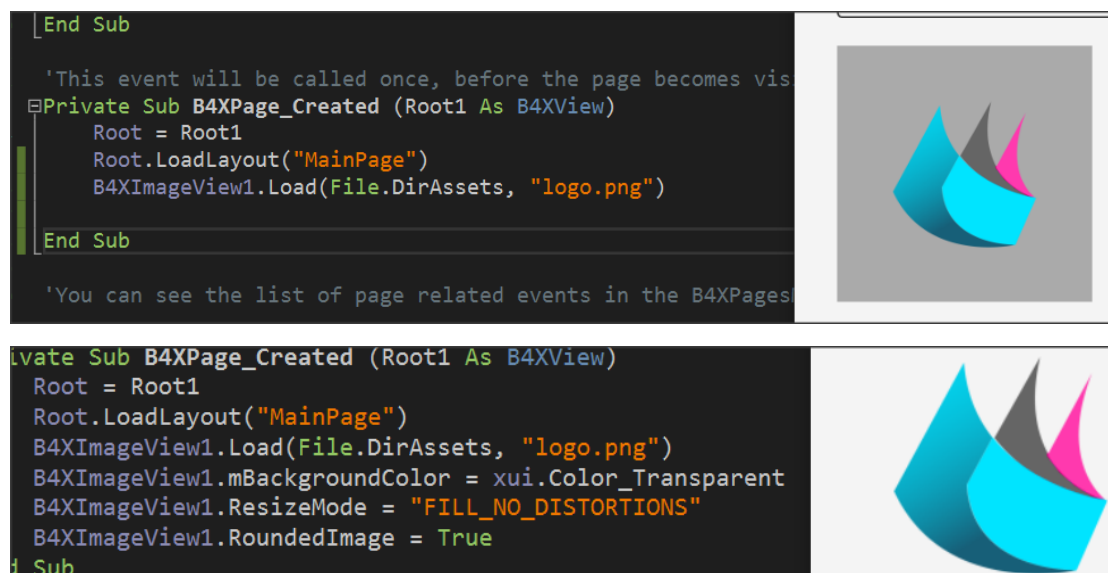
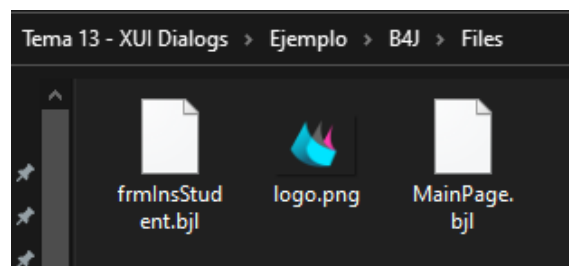


Imagen 7. B4XImageView

En el método "Load", además de la imagen, hay que incluir como primer argumento la carpeta donde está almacenada la imagen. En este caso, la carpeta "Files" (Files.DirAsset) es la que se usa por defecto (ver imagen):



Esta carpeta no es accesible para guardar nuevos ficheros cuando se crea la aplicación. En temas posteriores veremos con más detalle estas carpetas.

Imagen 8. Carpeta "Files" (DirAsset)

Para mostrar una imagen almacenada en la carpeta "Files" debe declararse también en la pestaña "Administrador de Archivos".

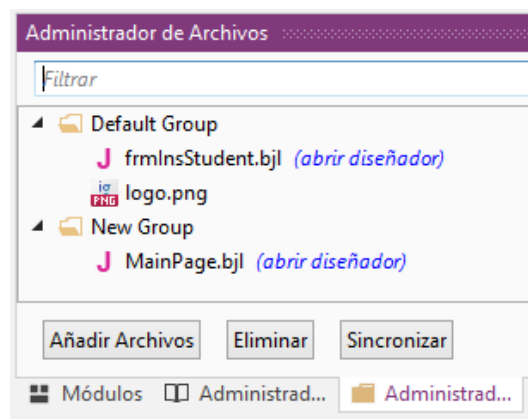


Imagen 9 Files Manager

## B4XDialogs

Los cuadros de diálogo son pantallas que se usan para informar al usuario de un evento o para obtener datos de los usuarios sin necesidad de crear otra B4XPage.

### MsgBoxAsync

Ya se ha usado anteriormente en otros temas. Es la función **xui.MsgBoxAsync** que muestra un mensaje en pantalla.

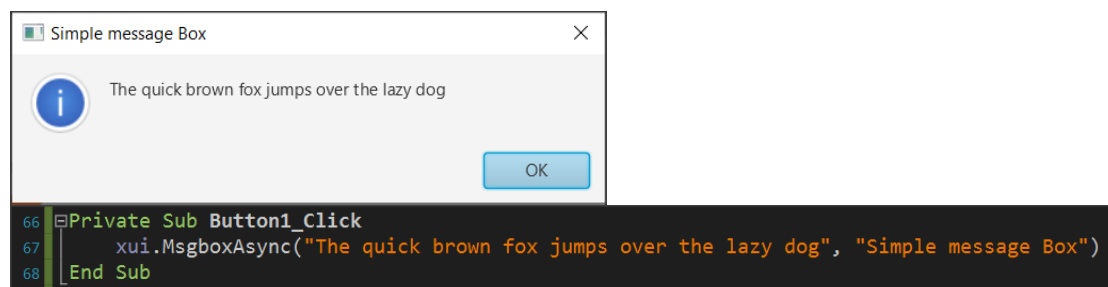


Imagen 10. MsgBoxAsync

Como se ve en la imagen anterior, también se muestra un botón OK para cerrar el mensaje y volver al programa.

### MsgBox2Async

El uso de MsgBox2Async es más complejo y permite a los programadores mostrar mensajes y elegir entre 3 diferentes funciones pulsando el botón adecuado.

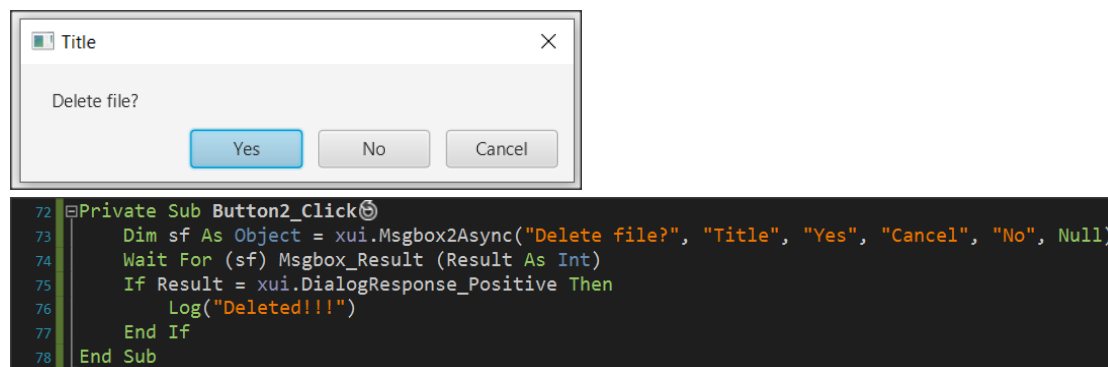


Imagen 11, MsgBox2Async

MsgBox2Async devuelve un objeto ("sf" en el ejemplo) que espera a un evento de respuesta por parte del usuario.

- **DialogResponse\_Positive** (si se ha pulsado "Yes" en el ejemplo)
- **DialogResponse\_Negative** (si se ha pulsado "No")
- **DialogResponse\_Cancel** (si se ha pulsado "Cancel")

Los valores anteriores se controlan mediante una sentencia "if" y así se decide qué acción realizar.



Las palabras que aparecen en los botones ("Yes", "No" y "Cancel") se pueden cambiar, pero las respuestas siempre van a seguir el orden de "Positive", "Cancel" y "Negative". Eso es importante tenerlo en cuenta.

Puedes ignorar un botón si pones una cadena vacía "", con lo que el botón no aparecerá en cuadro de diálogo.

El último parámetro permite mostrar un icono a la izquierda del mensaje; con Null no se muestra nada.

## Wait For

**Wait For** detiene la ejecución de una acción hasta que un evento es disparado.

```
Wait For (<remitente>) <event signature>
```

Donde **remitente** es el objeto sobre el que se va a esperar y

**Event signature** es el evento que se debe generar. En el caso de MsgBox2 Async estos eventos pueden ser DialogResult\_Positive, DialogResult\_Cancel o DialogResult\_Negative.

El tipo de evento se comprueba con una sentencia "if".

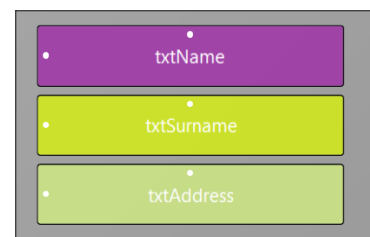
## CustomDialog

Crear un diálogo personalizado necesita varios pasos que comienzan en el Diseñador.



Imagen 12. Pasos para crear un diálogo personalizado

Siguiendo con el primer ejemplo, creamos un nuevo formulario en el Diseñador y lo guardamos con el nombre **frmInsStudent**. También creamos las variables para los TextFields desde "Generar Miembros"



```
87 Private Sub Button3_Click
88     dialog.Initialize (Root)
89     dialog.Title = "My Custom Dialog" 1
90
91     Dim p As B4XView = xui.CreatePanel("")
92     p.SetLayoutAnimated(0, 0, 0, 350dip, 250dip) 2
93
94     p.LoadLayout("frmInsStudent")
95     dialog.PutAtTop = True 'put the dialog at the top of the screen 3
96
97     Wait For (dialog.ShowCustom(p, "OK", "", "CANCEL")) Complete (Result As Int) 4
98     If Result = xui.DialogResult_Positive Then
99         Log(txtName.text & " " & txtSurname.text & " " & txtAddress.text)
100     End If
101 End Sub
```

Imagen 13. Diálogo personalizado

1. Dentro de **GlobalSub** creamos un objeto llamado "dialog" del tipo B4XDialog. Después usamos la función "dialog.Initialize(Root)" para inicializarlo. Con "dialog.Title" establecemos el título del cuadro de diálogo que vamos a crear.
2. Creamos un objeto "p" de tipo "B4XView" donde mostraremos los ítems del formulario que hemos diseñado. Fijamos su tamaño en píxeles.
3. Cargamos el formulario con "p.LoadLayout" donde "p" es el Paneque creamos antes y le decimos que aparezca encima de todas las demás ventanas ("p.PutAtTop=True").
4. Esperamos a que se pulse un botón y comprobamos el resultado.

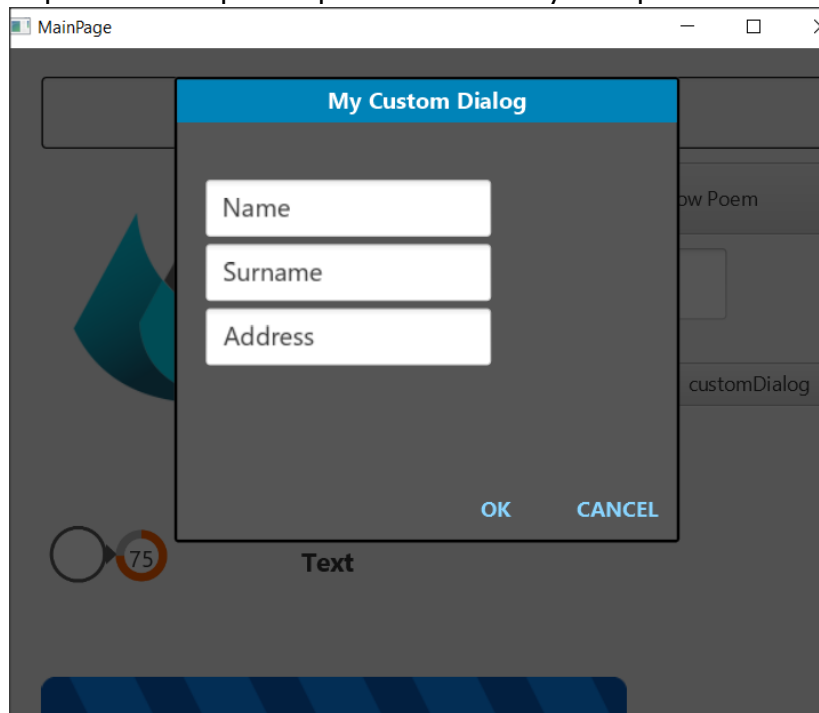


Imagen 14. El cuadro de diálogo

## Templates - Plantillas

B4X posee plantillas prediseñadas para crear cuadros de diálogo entre las que destacamos:

- B4XDateTemplate
- B4XColorTemplate
- B4XLongTextTemplate



### Consejo para el profesor

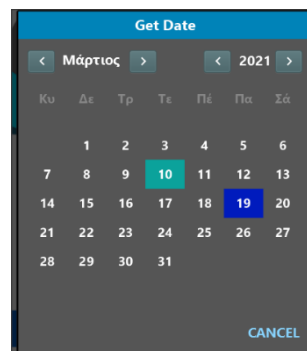
Más funciones se usarán en los siguientes temas a medida que se expliquen conceptos como los de lista, diccionarios, etc.



## B4XDateTemplate

Crea un cuadro de diálogo para elegir la fecha. Este cuadro sólo necesita un botón cancelar porque cuando se elige una fecha se vuelve automáticamente el valor introducido al programa. Cuando el formulario se abre ya tiene prefijada la fecha actual en un color distinto,

B4XDateTemplate se basa en una plantilla de diálogo que se debe declarar e inicializar como con los diálogos personalizados vistos antes.



```
108 Private Sub btnDate_Click
109     dialog.Title = "Get Date" 1
110     DateTemplate.Initialize
111     DateTemplate.MinYear = 2010 2
112     DateTemplate.MaxYear = 2030
113     'only CANCEL needed
114     Wait For (dialog.ShowTemplate(DateTemplate, "", "", "CANCEL")) Complete (Result As Int)
115     If Result = xui.DialogResponse_Positive Then
116         btnDate.Text = DateTime.Date(DateTemplate.Date) 3
117     End If
118 End Sub
```

Imagen 15. Crear un cuadro de diálogo para fecha

1. Inicializar y fijar el título del diálogo. En el ejemplo la inicialización se hace en la rutina Class\_Globals.
2. Indicamos el límite de años.
3. Esperar a que se elija la fecha. Cuando se haga, mostramos la fecha en "btnDate" o bien lo podemos usar como queramos.



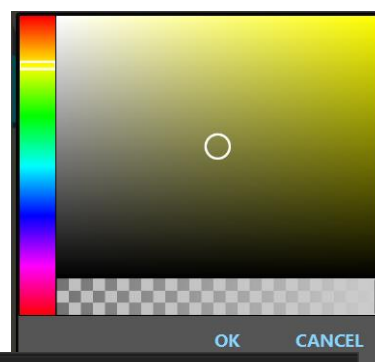
### Recuerda

La fecha se devuelve como "tics" que representan los milisegundos pasados desde el 1/1/1970. Los milisegundos se convierten a la fecha con la función "DateTime.Date".

## B4XColorTemplate

B4XColorTemplate es similar al de la fecha, pero para elegir un color. De nuevo el código espera con el comando Wait For a que se elija el color y cuando se pulsa "OK" se devuelve el color.

En el ejemplo se usa el color elegido para cambiar el fondo de la pantalla principal con el comando "Root.Color".



```
123 Private Sub btnColor_Click
124     ColorTemplate.Initialize
125     Wait For (dialog.ShowTemplate(ColorTemplate, "OK", "", "CANCEL")) Complete (Result As Int)
126     If Result = xui.DialogResponse_Positive Then
127         Root.Color = ColorTemplate.SelectedColor
128     End If
129 End Sub
```

Imagen 16. Plantilla para crear color

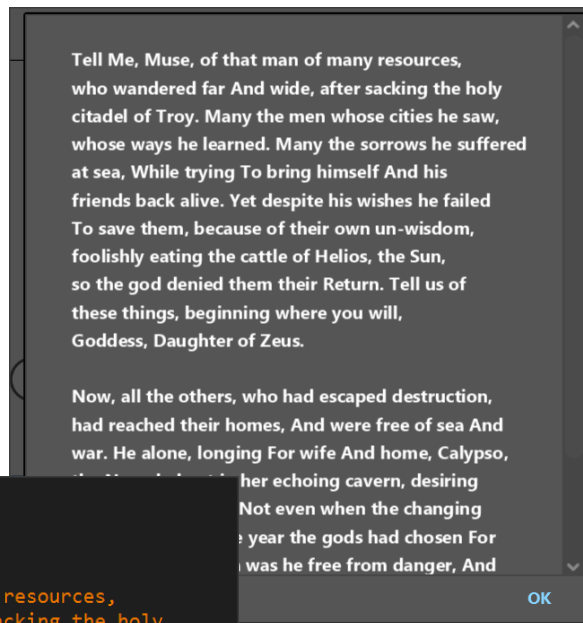
## B4XLongTextTemplate

B4XLongTextTemplate crea una ventana en la pantalla para mostrar un texto largo. Añade además un barra de navegación para desplazarse por el texto.

Como en anteriores plantillas, se crea una variable del tipo B4XLongTextTemplate y después se inicializa fijando el tamaño de la ventana que vas a crear con el comando "Resize".

Finalmente, en la propiedad "Text" ponemos el texto que

```
134 Private Sub btnReadPoem_Click
135     LongTextTemplate.Initialize
136     LongTextTemplate.Resize(500, 500)
137     LongTextTemplate.Text = "$"
138     Tell Me, Muse, of that man of many resources,
139     who wandered far And wide, after sacking the holy
140     citadel of Troy. Many the men whose cities he saw,
141     whose ways he learned. Many the sorrows he suffered
142     at sea, While trying To bring himself And his
143     friends back alive. Yet despite his wishes he failed
144     To save them, because of their own un-wisdom,
145     foolishly eating the cattle of Helios, the Sun,
146     so the god denied them their Return. Tell us of
147     these things, beginning where you will,
148     Goddess, Daughter of Zeus.
149     "$
150     dialog.ShowTemplate(LongTextTemplate, "OK", "", "")
151 End Sub
```



queremos mostrar e invocamos al método "ShowTemplate".

Imagen 17. LongTextTemplate

## Ejercicios

1. Crea un programa que pida dos fechas entre 2000 y 2021 y muestre cuántos días, meses y años de diferencia hay entre ellas.

*Pista:*

```
DateUtils.PeriodBetween(fecha1, fecha2).Days  
DateUtils.PeriodBetween(fecha1, fecha2).Months  
DateUtils.PeriodBetween(fecha1, fecha2).Years
```

2. Crea un programa que lea la siguiente información de un cliente:
  - a. Nombre
  - b. Apellidos
  - c. Teléfono
  - d. Saldo del teléfono (en €)

Después crea un botón en el formulario con el texto "pago" y que al pulsarlo muestre el mensaje "¿Desea pagar?". Si se responde "Sí", pone a 0 el contenido de la variable que guarda el saldo del teléfono.

3. Crea un programa que muestre un resumen de 5 libros en una ventana. Los libros se elegirán de un conjunto de 5 botones que tengan una imagen de ellos encima. Los resúmenes y las imágenes de los 5 libros se pueden encontrar en los materiales de apoyo a este ejercicio.

4. Crea un formulario que contenga estos elementos:
  - a. Nombre
  - b. Apellidos
  - c. Teléfono

También incluirá un interruptor (switch) que, cuando el formulario esté abierto, tenga un color de fondo gris claro ("light\_gray") con los campos de texto en color blanco. Cuando se cierre, el fondo será gris oscuro (dark\_grey) con los campos de texto en gris claro. Añade un etiqueta que diga "día" o "noche" con letras negras o blancas, respectivamente.

*Pista: Usa **JFX library** para establecer los colores de los floatTextFields*

```
Private fx As JFX ` dentro de Class_Globals  
lblDiaNoche.TextColor = fx.Colors.Black ` donde quieras cambiar el color
```

5. Un depósito de agua mide 5 metros de altura y tiene una base de 3x3 metros. Escribe un programa que:
  - A. Lea continuamente el nivel del agua en metros y muestre una barra que represente la altura de llenado. El campo de texto usado sólo permitirá introducir valores decimales, pero no texto.
  - B. Si la altura del agua está por encima de 5 metros mostrará el mensaje "¡Peligro de desbordamiento!".
  - C. Si la altura es menor de 0'5 metros mostrará el mensaje "Peligro: no hay suficiente agua en el depósito".
  - D. Cada vez que se cambie la altura, se mostrará la cantidad total de agua expresada en metros cúbicos.