# 

**Programando con B4X**

Tema 13 – XUI Dialogs

Version 1.0, abril 2021

Autor original: [Prokopis Leon](https://github.com/pliroforikos)

Traducido al español por [José Miguel López](https://github.com/Lamashino)

[](https://creativecommons.org/licenses/by/4.0/)

# Tema 13 – XUI Views

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

Lo que los estudiantes aprenderán

**3h**

Los lenguajes de programación poseen bibliotecas (es frecuente que se traduzca mal 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 programados u obtenidas de otras fuentes como GutHub). Para usar una biblioteca, únicamente hay que elegirla de la lista de Bibliotecas de B4X.

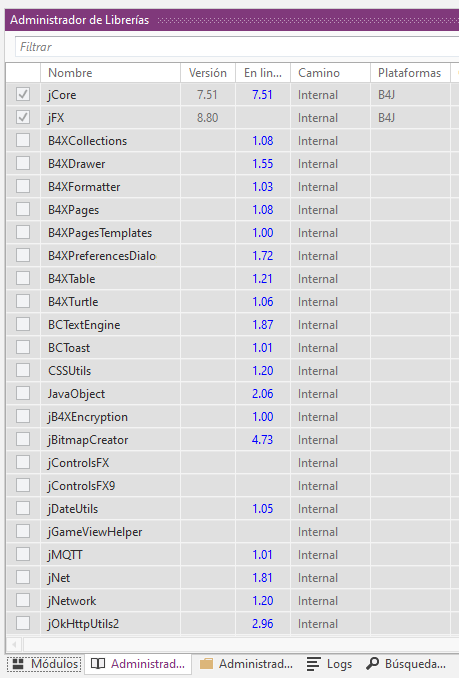


Imagen 1. Administrador de Bibliotecas

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>.

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 vremos 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 multiplataforma.
  + **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.

## Usando 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.

|  |  |
| --- | --- |
| Teacher with solid fill | **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.

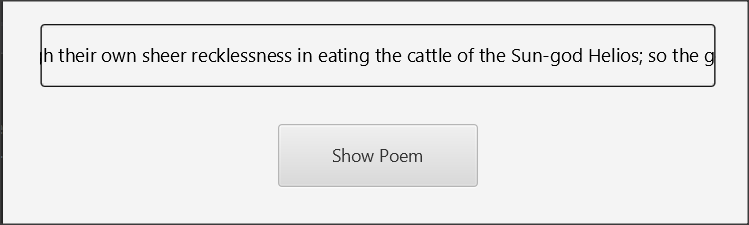
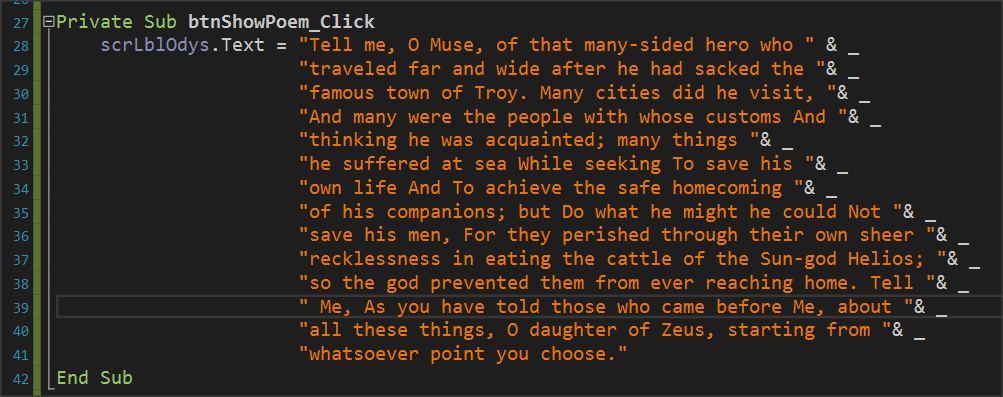


Imagen 2 Etiqueta desplazable

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

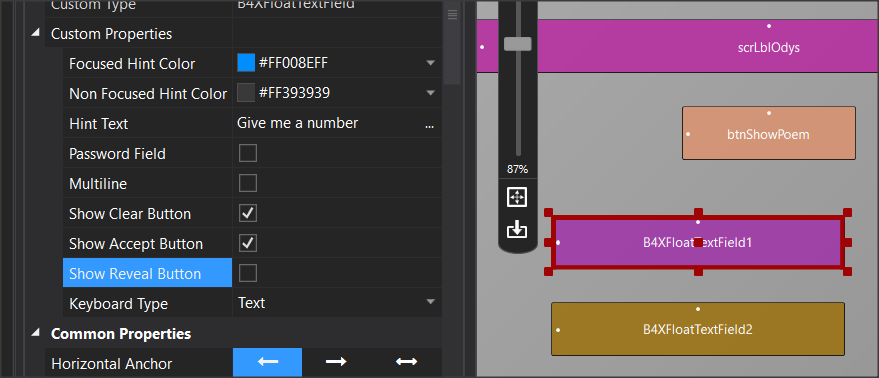


Imagen 3. B4XFloatTextField

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 controlador 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.

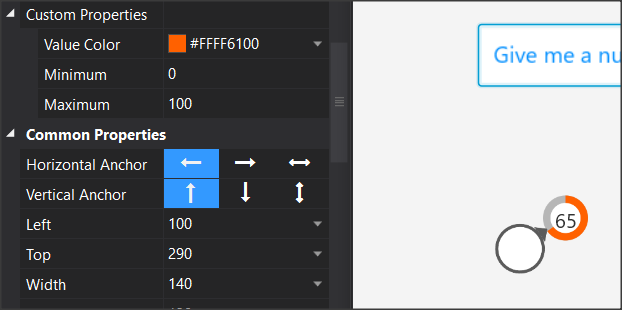


Imagen RoundSlider

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.

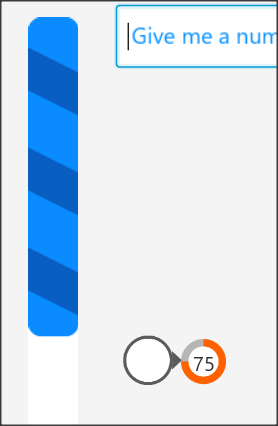
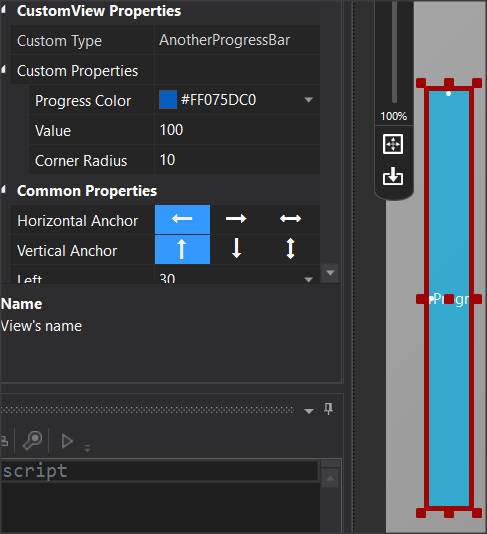
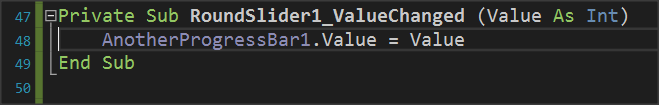


Imagen . AnotherProgressBar

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

B4XSwitch

B4XSwitch crea un botón deslizable en la pantalla.

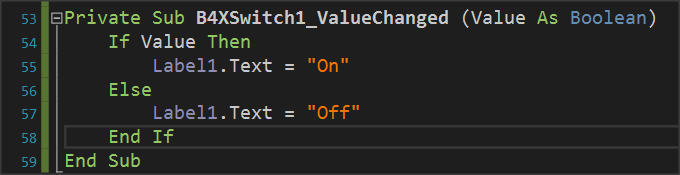
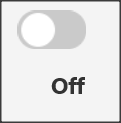


Imagen . B4XSwitch

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.

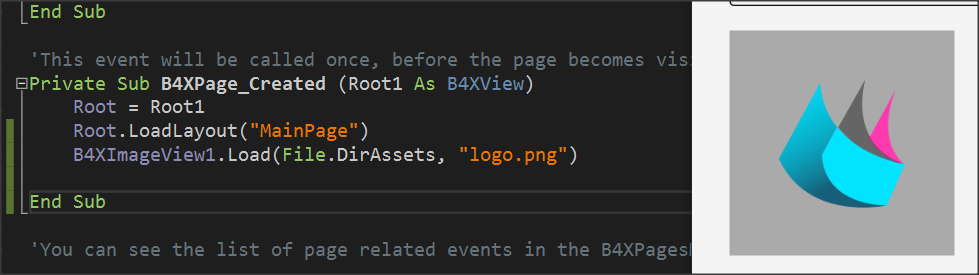
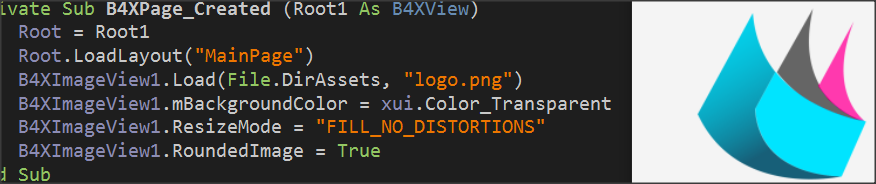


Imagen . B4XImageView

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):

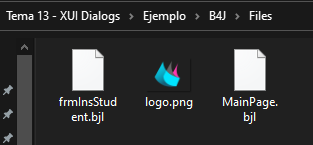


Imagen 8. Carpeta “Files” (DirAsset)

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



Imagen Files Manager

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

## 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 meustra un mensaje en pantalla.

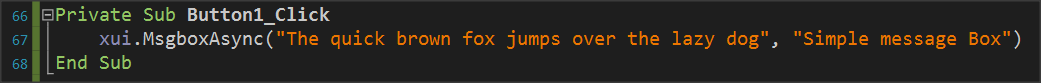
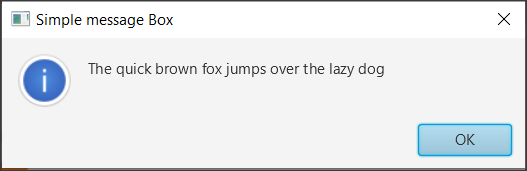


Imagen 10. MsgBoxAsync

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.

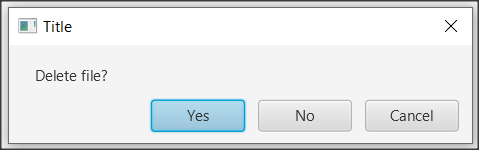
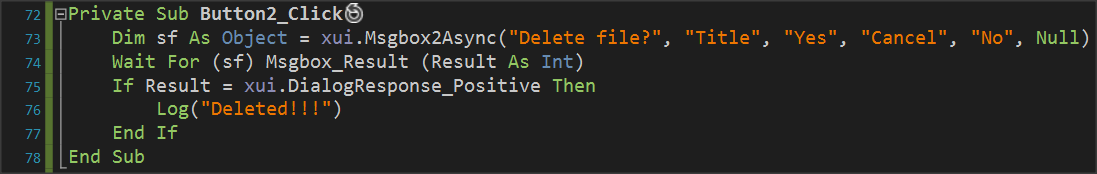


Imagen 11, MsgBox2Async

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 DialogResponse\_Positive, DialogResponse\_Cancel o DialogResponse\_Negative.

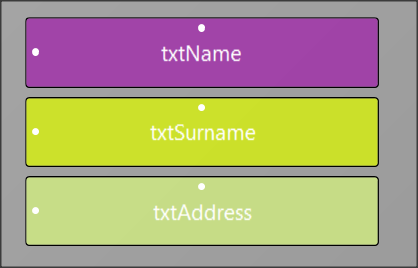
El tipo de evento se comprueba con una sentencia “if”.

## 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”

Picture containing text, black, screenshot, silver

Automatically created description

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.

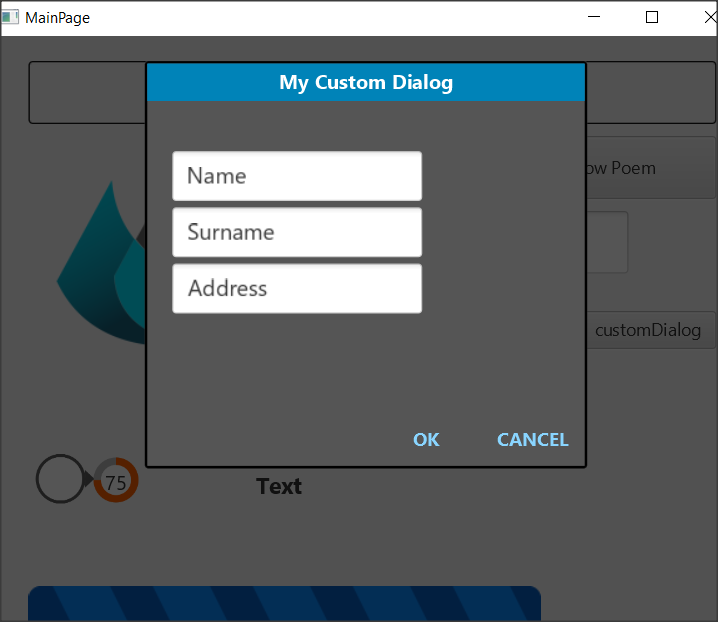


Imagen 14. El cuadro de diálogo

## 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.

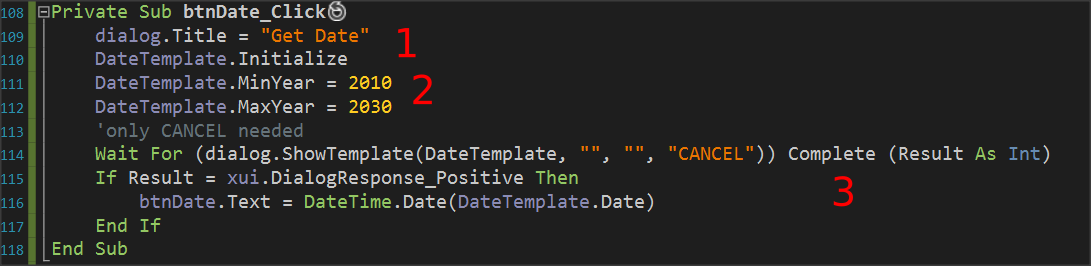


Imagen 15. Crear un cuadro de diálogo para fecha

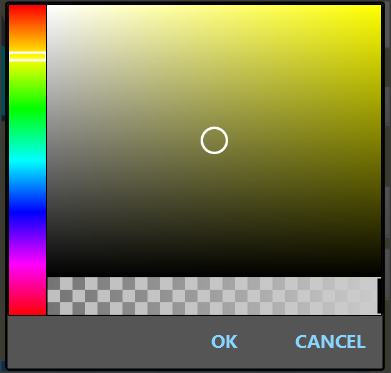
1. Inicializar y fijar el título del diálogo. En el ejemplo la inicialización se hace en la rutina Class\_Globals.
2. Indicamos el límite de años.
3. Esperar a que se elija la fecha. Cuando se haga, mostramos la fecha en “btnDate” o bien lo podemos usar como queramos.

**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

B4XColorTamplate 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.

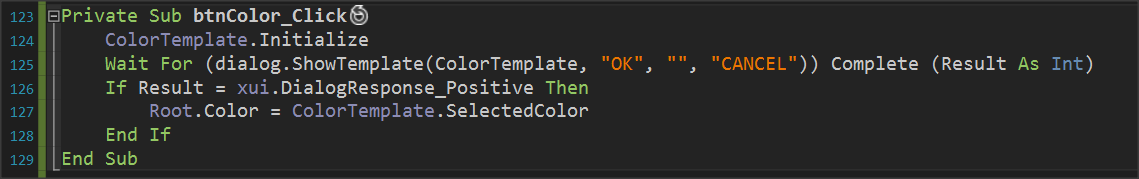
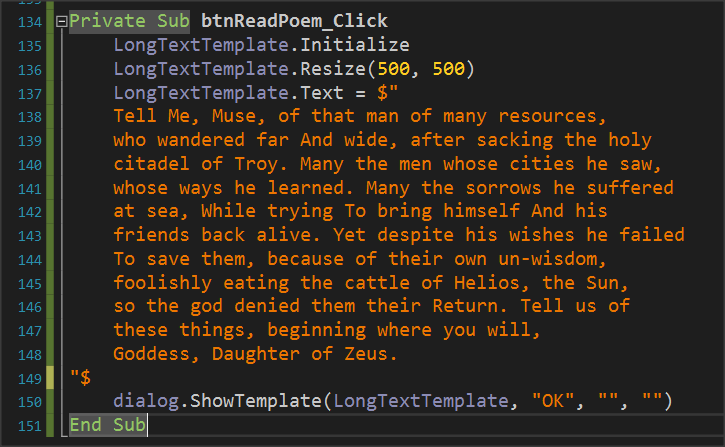
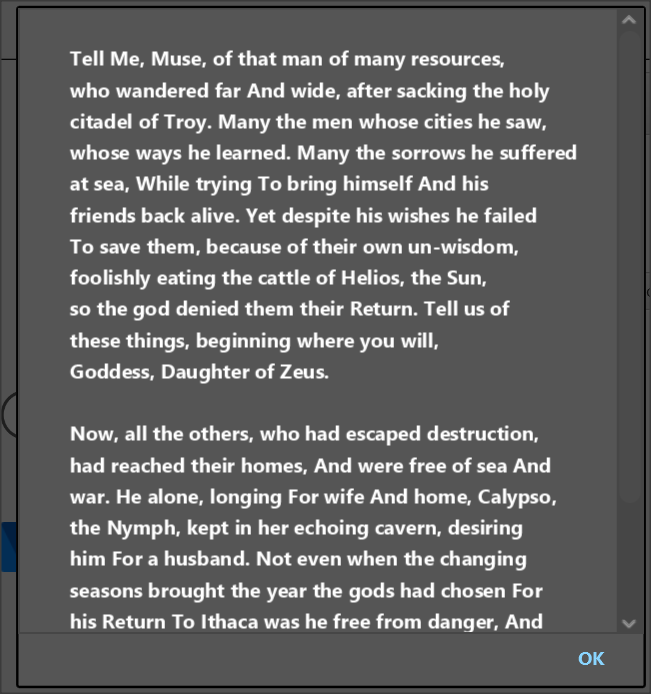


Imagen 16. Plantilla para crear color

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

### B4XLongTextTemplate

B4XLongTextTemplate displays a window on the screen to display long text. Also, it displays a navigation bar to scroll the text.



Picture 17 LongTextTemplate

As in previous templates, set a variable of type B4XLongTextTemplate, and then after you initialize it you can set a size for the window that you are going to create with the “Resize” command.

Finally, set the Text property to the text you want to display and call “ShowTemplate”.

## Exercises

1. Create a program that asks for two dates between 2000 and 2021 and shows the distance between them in days, months and years.

Tip:

DateUtils.PeriodBetween(date1, date2).Days

DateUtils.PeriodBetween(date1, date2).Months

DateUtils.PeriodBetween(date1, date2).Years

1. Create a program that reads the following customer information:
   1. Name
   2. Surname
   3. Phone
   4. Phone Account (in € )

Then create a button on the form that says payment and if pressed display the message "Do you want to pay?" If answered yes set 0 to the variable corresponding to the Phone Account.

1. Create a program that for 5 books displays a summary of them in an appropriate window. Books must be selected from an equal number of buttons with book’s images on them. The summaries and images of 5 books can be found in the supporting material of the exercise.
2. Create a form that contains the following items:
   1. Name
   2. Surname
   3. Phone

It also includes a switch that when the form is open has a light\_Gray background color with text fields in White while when closed it has dark gray background with light\_Gray text Fields. Also include a label to say day or night with black or white letters each.

*Tip: Use* ***JFX library*** *to set floatTextFields colors*

**Private** fx **As** JFX ‘ *in Class\_Globals*

lblDayNight.TextColor = fx.Colors.Black ‘ *Where you want to change color*

1. A water tank is 5 meters height and has a base of 3x3 meters. Make a program that,
2. continues reads the level height in meters and display a bar, represent the height. Text field will not allow any texts but only decimals values.
3. If water height is above 5 meters show message "Danger Water Leak!".
4. If the level Is less than 0.5 meters display a message. "Warning there is not enough water in the tank.
5. In every change show the total amount of water in cubic meters.