**Programando con B4X**

Tema 6 – Del Diseñador al Código

Version 1.0, marzo 2021

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

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



# Tema 6 – Del Diseñador al Código

* Class\_Globals
* Variables y subrutinas (Subs)
* Paso de Valores al Código
* Eventos
* Atributos

Lo que los estudiantes aprenderán

**2h**

Diseñar la apariencia de la aplicación en el Diseñador es el primer paso al crear una aplicación. A menudo, los nuevos programadores recurren a él para rediseñar, corregir o agregar información individual.

Cuando la pantalla está lista, el desarrollador pasa a la siguiente etapa que consiste en programar las funciones. Es decir, hay que hacer que todos los elementos que se incluyeron en el diseño realicen sus funciones correctamente. Por ejemplo, los cuadros de texto deben poder leer datos, los botones deben lanzar funciones, las listas deben mostrar datos, etc.

## Class\_Globals

Al comienzo del código en la pestaña **B4XMainPage** hay un conjunto de declaraciones de variables entre **Sub Class\_Globals** y **End Sub**.

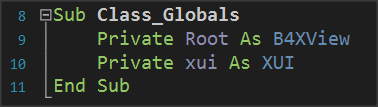
Como se vio en el tema 3, una Sub (subrutina) es un trozo de código que realiza una operación específica. Dentro de **Class\_Globals** se recogen las declaraciones de variables que queremos que se conozcan a lo largo del código de la pestaña **B4XMainPage**, es decir, en cada subrutina.

Imagen 1. Sub Class\_Globals

Además, si una sentencia de declaración de variables empieza con la palabra **Public**, entonces estará disponible para cualquier otra “pestaña”.

## Un estudio más profundo del uso de variables

En el siguiente código puedes ver 3 subrutinas:

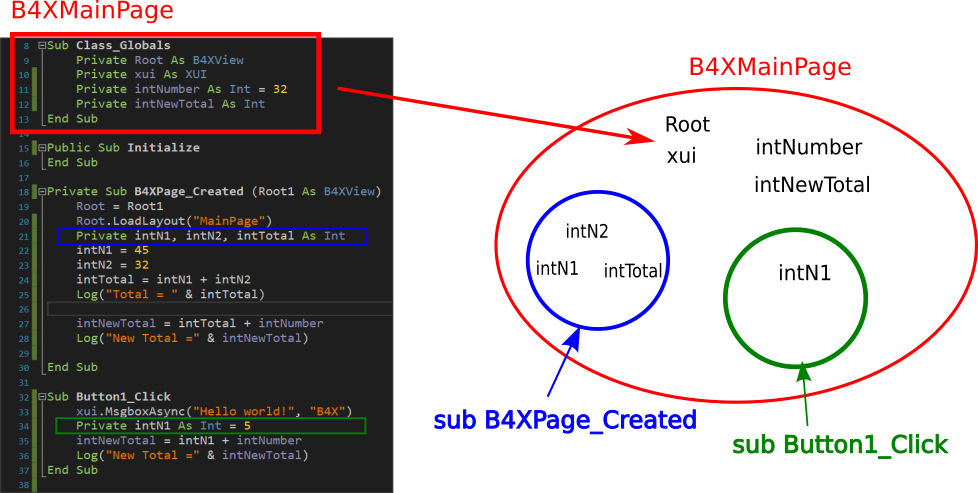


Imagen 2. Variables y ámbitos

Las variables **intNumber**, **intNewTotal**, **Root** y **xui** declaradas dentro de **Class\_Globals** "viven" dentro del módulo **B4XMainPage**. Se trata de variables de ámbito global que pueden ser accedidas desde cualquier subrutina dentro de la pestaña **B4XmainPage.**

Por el contrario, las variables **intN1**, **intN2**, **intTotal** (color azul) “viven” dentro de la subrutina **B4XPage\_Created** y ninguna otra subrutina podrá usarlas. Al mismo tiempo, la variable **intN1** declarada dentro del botón **Button1\_Click** **no es la misma** que la que hay dentro de **B4XPage\_Created**. Ambas tienen su propia zona de memoria y pueden tener el mismo nombre por estar en subrutinas distintas.

**Consejo para el profesor**

El uso y el ámbito de las variables son conceptos que confunden a los nuevos programadores. En función de tu alumnado, sería útil usar ejemplos que les resulten familiares.



## Paso de Valores al Código

En la pantalla que has diseñado en el tema 5 hay objetos (etiquetas, textos, botones, etc.) que hay que declarar dentro de **Class\_Globals** antes de usarlos en tu programa.

Algunos de los objetos de la pantalla no hay que incluirlos en el código porque no se van a usar. Por ejemplo, ninguna de las “etiquetas” va a ser gestionada en el código. Sin embargo, los botones y los campos de texto (TextFields) sí que van a ser modificados en el código.

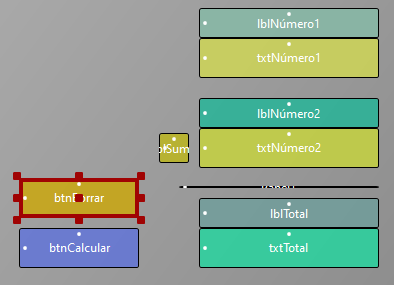


Imagen 3. Ejemplo 1. Diseñador Abstracto

Hay dos formas de insertar objetos en el código. La primera se hace fácilmente a través del **Diseñador** siguiendo estos pasos:

* Vamos a **Diseñador 🡪 Abrir Diseñadir interno**
* Elegimos dentro del menú **Herramientas** la opción **Generar Miembros.**

Dentro de la pantalla de **Generate Members** hacemos clic en los objetos:

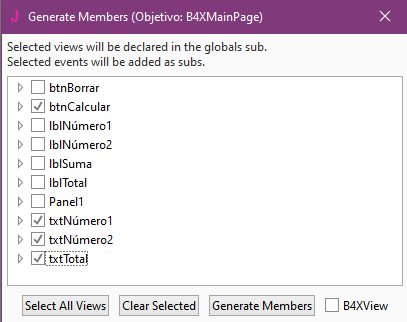


Imagen 4. Generate Members

* btnCalcular
* txtNúmero1
* txtNúmero2
* txtTotal

y pulsamos en **Generate Members**.

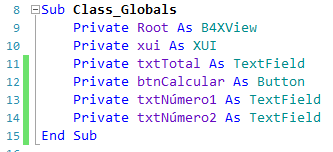


Imagen 5. Class\_Globals

Tu código en la subrutina **Class\_Globals** se actualizará automáticamente con las nuevas variables.

**Recuerda**

Cada objeto que importamos es de un tipo concreto que coincide con el tipo de las variables que se han creado



La segunda opción consiste en escribir las variables uno mismo, prestando atención a que los nombres de los objetos de la pantalla sean los mismos que los que escribes como variables (y también su tipo: Button, TextField, etc).

## Eventos

Tras declarar las variables para los objetos, el último paso es activas las funciones del formulario.

Esto dependerá de lo que haga tu aplicación. En el ejemplo que estamos viendo hay un botón llamado **Calcular** que lo que hará es sumar los dos números de la pantalla y mostrará el resultado en pantalla.

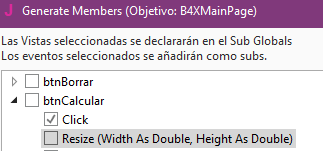
La operación de suma es disparada por un proceso llamado **Evento**. El programador debe detectar el evento de pulsar en el botón **Calcular** para realizar las operaciones necesarias para mostrar el resultado.

**Recuerda**

Hay miles de eventos que pueden dispararse en una aplicación; es tu responsabilidad como programador decidir cómo responder a ellos.



La detección de un evento es fácil y sólo hay que crear una nueva subrutina con el nombre del evento. Esto se puede hacer al mismo tiempo que se realizó la declaración de las variables a través del **Diseñador**.

* Vamos a **Diseñador 🡪 Abrir Diseñadir interno**
* Elegimos dentro del menú **Herramientas** la opción **Generar Miembros.**
* Dentro de la pantalla de **Generate Members** desplegamos la lista que hay en el botón **btnCalcular**.
* Marcamos el evento **Click** y de nuevo pulsamos en **Generate Members.**

Ahora aparecerá una **nueva subrutina** para el evento **btnCalcular\_Click** donde podrás escribir el código necesario:

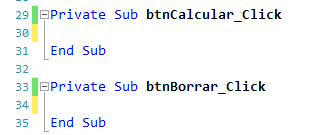


Imagen 6. El evento Click

## Escribiendo código en el Evento

Dentro de la subrutina de un Evento se escriben todas las acciones que deben lanzarse cuando se produce ese evento.



Imagen 7. Realizar cálculos con los TextFields

En la Imagen 9 estamos haciendo lo siguiente:

*El contenido del TextField llamado txtTotal será igual a la suma de los contenidos de los TextFields txtNúmero1 y txtNúmero2*

### Propiedades

Cada objeto que insertar en tu código tiene varias propiedades. Por ejemplo, puede tener un color, un tamaño, una posición en pantalla, un contenido, etc. Estas propiedades pueden leerse o modificarse dentro de tu código. Por ejemplo, la propiedad **txtNúmero1.Text** nos dice qué contiene el campo de texto **txtNúmero1**; pero también nos permite modificar su contenido dependiendo de cómo la usemos.

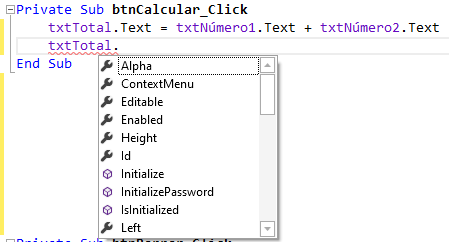
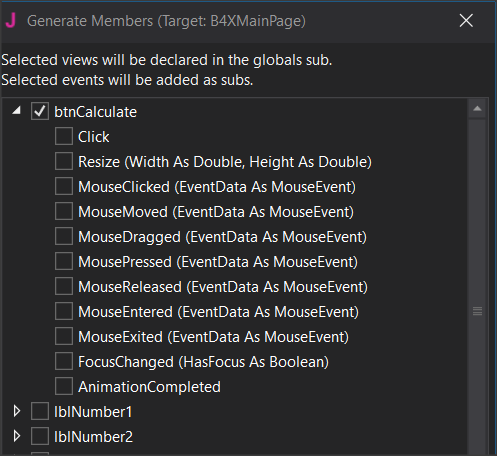


Imagen 8. Lista de propiedades de un objeto

Las propiedades de un objeto podemos averiguarlas simplemente escribiendo el nombre del objeto y poniendo un punto (.) a continuación del nombre. Entonces, nos aparecerá una lista de todas las propiedades (ver Imagen 10).

**Recuerda**

Puedes hacer operaciones matemáticas con cadenas de texto **sólo si contienen números.**



Imaginemos que queremos crear una nueva función en la calculadora que borre el formulario para escribir nuevos números. Las acciones que tienes que hacer son las siguientes:

* Abrir el **Diseñador** y añadir un nuevo botón llamado **btnBorrar**.
* Crear una variable dentro de **Class\_Global** para acceder a ella.
* Crear el evento **btnBorrar\_Click**.
* Dentro del evento, fija la propiedad **Text** de **txtNúmero1**, **txtNúmero2** y **txtTotal** al valor "" (cadena vacía).

Cada vez que pulses el botón **Borrar**, se borrará el formulario.

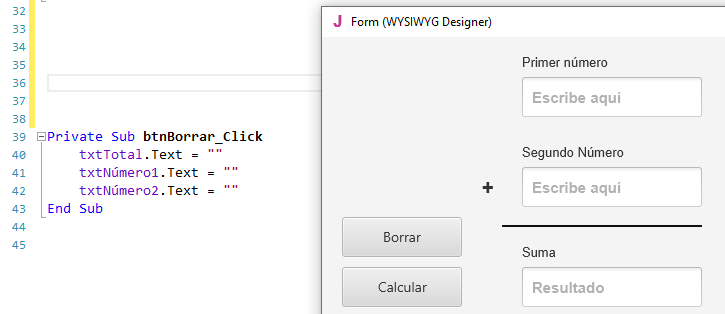


Imagen 9. Botón de Borrar y código asociado

## Ejercicios

1. Amplía el ejemplo visto en clase para que realice las 4 operaciones: suma, resta, multiplicación y división cuando se pulse el botón adecuado. Añade los objetos necesarios en el diseñador y completa el código.
2. En el ejercicio 2 del tema anterior, continúa y completa la aplicación de calcular notas medias para que el botón “Calcular” calcule la nota media de todas las materias.

Calculadora de nota media

Matemáticas

Calcular

Física y Quím.

Lengua

Inglés

Música

Informática

Ed. Física

Geo. e Historia

Nota media

Debes comprobar que las notas de cada materia están entre 0 y 10. Si no, deberías mostrar un mensaje de error.