

# Programando con B4X

## Tema 8 – Subrutinas

Version 1.0, marzo 2021

Autor original: [Prokopis Leon](#)

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



Anywhere Software

## Tema 8 – Subrutinas

🕒 3h

Lo que los estudiantes aprenderán

- Qué es una subrutina (Sub)
- Declaración de una Subrutina
- Paso de Valores
- Devolución de Valores de una Subrutina

En programación, una subrutina es una secuencia de instrucciones que realizar una tarea concreta y que forman una unidad. Esta unidad puede usarse después en los programas dondequiera que esa tarea sea realizada.

Las subrutinas pueden definirse dentro de nuestro programa o dentro de bibliotecas que agrupan múltiples subrutinas para que puedan ser usadas por otros programas. Según el lenguaje de programación, a las subrutinas se les puede llamar funciones, subprogramas, métodos o procedimiento. Para crear una subrutina el programador debe tener en cuenta esto:

- Debe realiza una única tarea.
- Debe ser relativamente pequeña e, idealmente, no mayor que lo que cabría en una pantalla para que pueda leerse fácilmente.
- Debe tener un nombre que haga referencia a lo que hace.

### Crear una subrutina en B4J

Ya hemos visto la subrutina **Button1\_Click** en B4J cuando se crea un nuevo programa. Fíjate que los eventos de "Click" se gestionan mediante subrutinas.

#### Ejemplo 1

Imagínate una función que deba sumar dos números introducidos por el usuario. Se trata de un ejemplo muy sencillo, pero lo usaremos para entender el concepto de subrutina y ver cómo funciona.

Los dos enteros intA e intB se declaran en el programa, se les asigna un valor y después invocamos la subrutina MostrarSuma1.

```
8 Sub class_globals
9     Private Root As B4XView
10    Private xui As XUI
11 End Sub
12
13 Public Sub Initialize
14
15 End Sub
16
17 'This event will be called once, before the page becomes visible.
18 Private Sub B4XPage_Created (Root1 As B4XView)
19     Root = Root1
20     Root.LoadLayout("MainPage")
21 End Sub
22
23 'You can see the list of page related events in the B4XPagesManager class
24
25 Sub Button1_Click
26     xui.MsgboxAsync("¡Hola mundo!", "B4X")
27 End Sub
```

Imagen 1. Subrutinas



Invocar a una subrutina se hace escribiendo su nombre (el que queramos) y entre paréntesis ponemos las variables cuyos valores debe conocer para funcionar.

La subrutina se escribe antes o después del programa actual:

Si delante ponemos la palabra reservada **Private**, significará que esta subrutina sólo se conocerá dentro del código B4XmainPage; si ponemos **Public**, la subrutina se podrá usar en otras partes de nuestra aplicación.

La palabra "Sub" es una abreviatura de subrutina. Entre paréntesis escribimos el nombre de las variables cuyos valores se enviarán a la función.

Fíjate en la **Imagen 2** en que los datos se envían a la función en el orden en que están escritos al invocar a la subrutina. Por ej. el valor de "intA" se introduce en la variable "a" y el valor de "intB" en la variable "b".



#### Recuerda

Las variables usadas en las subrutinas cuando son invocadas se llaman **parámetros**.

La subrutina funcionará con los datos contenidos en los parámetros "a" y "b" y NO con las variables "intA" e "intB".

```
18 Private Sub B4XPage_Created (Root1 As B4XView)
19     Root = Root1
20     Root.LoadLayout("MainPage")
21     Private intA, intB As Int
22     intA = 10
23     intB = 30
24
25     MuestraSuma1(intA, intB)
26     Log(intSum)
27
28     Log(Suma(intA, intB))
29 End Sub
30
31 Private Sub MuestraSuma1(a As Int, b As Int)
32     intSum = a + b
33     Log(intSum)
34 End Sub
```

Imagen 2. Invocando una subrutina



## La memoria de una subrutina en B4X

Cada subrutina tiene su propio espacio de memoria donde almacenar variables. La única excepción es la subrutina **Class\_Globals** cuyas variables pueden ser consultados por todas las rutinas de B4XMainPage por sus nombres.

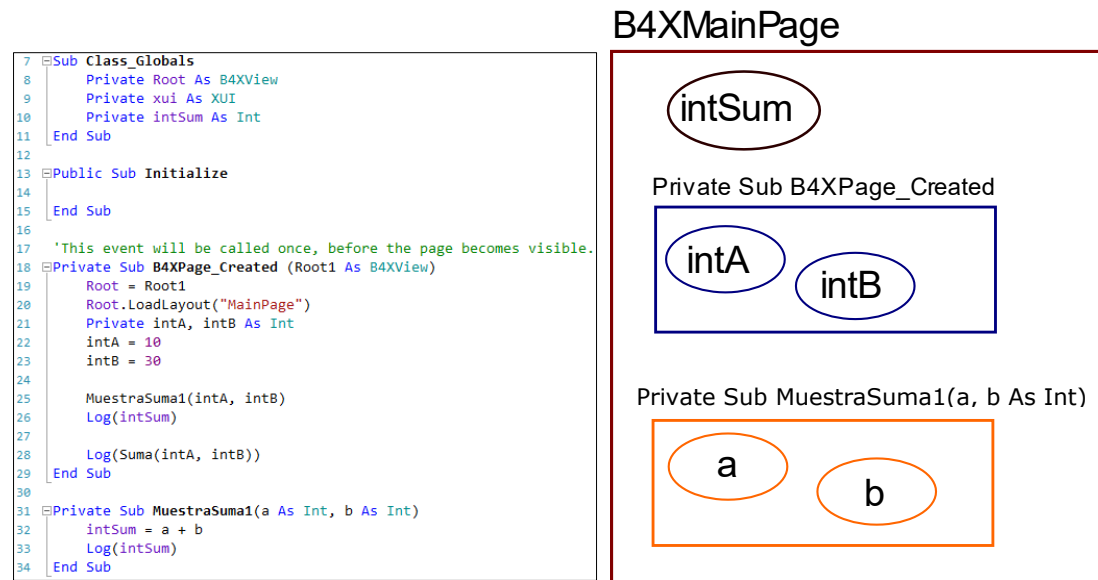


Imagen 3. La memoria

En la **Imagen 3** puedes ver que la variable **intSum** es accesible desde cualquier subrutina y se puede consultar escribiendo su nombre. En B4X estas variables se representan en un color diferente para que el programador pueda distinguirlas fácilmente. Por el contrario, las variables declaradas dentro de una subrutina sólo son accesible desde esa subrutina.

### Recuerda



Las variables declaradas dentro de **Class\_Globals** se pueden ver desde cualquier subrutina y se llaman **Globales**.

Las variables declaradas dentro de una subrutina sólo se pueden ver dentro de esa subrutina y se llaman **Locales**.



## Devolución de un valor desde una subrutina

Una subrutina puede devolver un valor al código que la invocó. Esto se hace de la siguiente forma:

```
18 Private Sub B4XPage_Created (Root1 As B4XView)
19     Root = Root1
20     Root.LoadLayout("MainPage")
21     Private intA, intB As Int
22     intA = 10
23     intB = 30
24
25     MuestraSuma1(intA, intB)
26     Log(intSum)
27
28     Log(Suma(intA, intB))
29 End Sub
30
31 Private Sub MuestraSuma1(a As Int, b As Int)
32     intSum = a + b
33     Log(intSum)
34 End Sub
35
36 Private Sub Suma(a As Int, b As Int) As Int
37     Return(a+b)
38 End Sub
```

Imagen 4. Devolver valores al programa

1. En la declaración de la subrutina tienes que incluir el tipo de variable que devolverá (**as Int** en la imagen).

2. Debes usar la sentencia **Return** dentro de la subrutina para devolver el valor que quieras.

3. El código que ha invocado la subrutina usará el valor devuelto por la subrutina como si fuera una variable.



### Recuerda

Normalmente, a las subrutinas que devuelven valores al código se les llama **Funciones**.

## Ejemplo 2

Escribe un programa que lea 3 enteros y devuelva el mayor.

```
5 Sub Class_Globals
6     Private Root As B4XView
7     Private xui As XUI
8 End Sub
9
10 Public Sub Initialize
11 End Sub
12
13 Private Sub B4XPage_Created (Root1 As B4XView)
14     Root = Root1
15     Root.LoadLayout("MainPage")
16     Private intA, intB, intC As Int
17     intA = 20
18     intB = 10
19     intC = 300
20
21     Log(CalcularMax(intA, intB, intC)) ' Muestra 300
22 End Sub
23
24 Private Sub CalcularMax(a As Int, b As Int, c As Int) As Int
25     Private intM As Int
26     intM = a
27     If b > intM Then
28         intM = b
29     End If
30     If c > intM Then
31         intM = c
32     End If
33
34     Return(intM)
35 End Sub
```

Imagen 5. Ejemplo 2

1. Declaramos 3 variables enteras (Inta, intB, intC) en la subrutina **B4XPage\_Created**.

2. Asignamos valores a las 3 variables.

3. Invocamos a la subrutina CalcularMax con las 3 variables como parámetros.

4. La subrutina aplica el algoritmo Máximo a las variables **a, b, c** y devuelve el resultado intM calculado

5. El valor mayor es mostrado en la ventana de Log.



### Recuerda

Nos encantan las subrutinas porque:

- Es aburrido escribir siempre el mismo código.
- Es más rápido usarlas que escribir de nuevo el código.
- Es útil no repetir los mismos fallos. ¡Ya los has corregido una vez!
- Los buenos programadores usan subrutinas.



## Ejercicios



### Consejo para el profesor

Anima a los estudiantes a solucionar y discutir sus soluciones en clase.  
Dedícale al menos una hora a explicarlos.

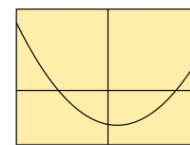
1. Escribe un programa que calcule el área de un círculo. El usuario debe introducir el radio del círculo en un `textField` y después usa una subrutina para calcular y devolver el área.
2. Escribe un programa que calcule la solución a la ecuación de segundo grado  $ax^2 + bx + c = 0$ .
  - a. El usuario introducirá los coeficientes **a**, **b** y **c** en `TextFields`.
  - b. El resultado aparecerá en uno o dos `textField` según el valor del discriminante.
  - c. El discriminante debe calcularse con una subrutina que devolverá el valor.
  - d. No se permitirá calcular el discriminante hasta que todos los coeficientes **a**, **b** y **c** se introduzcan.

Para mostrar mensajes de error puedes usar `xui.MsgboxAsync("Message","Title")`

Imagen 7. Esquema

Ecuación cuadrática

$$ax^2 + bx + c = 0$$



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Imagen 6. Origen: Wikipedia.org

3. Construye un programa que use la tortuga y que dibuje cuadrados de lado igual a un valor introducido por el usuario. Crea una subrutina que reciba la longitud del lado y que dibuje el cuadrado empezando por el lugar donde esté actualmente la tortuga y que se mueva en el sentido de las agujas del reloj.
4. Un teatro tiene 3 tipos de entradas: anfiteatro, galería y proscenio. Cada entrada vale 20, 30 y 40€ respectivamente. Construye un programa que:
  - a. Lea los números 1, 2 o 3 que representan las 3 categorías (1: anfiteatro; 2: galería y 3: proscenio).
  - b. Le el número de entradas que se quieren.
  - c. Calcule el valor total de las entradas con una subrutina que devuelva la cantidad y que se muestre el resultado en un `textField`.

