**Programando con B4X**

Tema 7 – Sentencias Condicionales

Version 1.0, marzo 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 7 – Sentencias Condicionales

* Variables Booleanas (o lógicas)
* Operadores relacionales
* Operadores lógicos
* Sentencia If
* Sentencia If-Else
* Sentencia If-Else-Else If
* Algoritmos para calcular el Máximo

Lo que los estudiantes aprenderán

**4h**

## Variables Lógicas o Booleanas

Anteriormente hemos visto tres tipos de variables: entero, decimal y cadena. Las variables lógicas son un tipo de dato muy sencillo, ya que sólo admiten dos valores posibles: True o False (Verdadero o Falso). Internamente el ordenador las representa como un 1 o un 0 (si hay corriente eléctrica o no).

La declaración de una variable lógica (Booleana) se hace igual que con los otros tipos de datos ya vistos:

**Private** intDistancia = 100, intTotalViaje = 0 As **Int**

**Private** blnIndicador  **As**  **Boolean** = False

**Private** blnHecho As **Boolean**

## Operadores relacionales o de comparación

Los operadores relacionales se usan para realizar comparaciones entre valores. Son los mismos que en matemáticas, aunque en informática se escriben de una forma un poco diferente:

|  |  |  |
| --- | --- | --- |
| **Símbolo matemático** | **B4X** | **Significado** |
| = | = | Igual a |
| ≤ | <= | Menor o igual a |
| ≥ | >= | Mayor o igual a |
| ≠ | <> | Distinto |
| < | < | Menor que |
| > | > | Mayor que |

En general, para hacer una comparación debes **comparar variables del mismo tipo**. Por ejemplo, enteros con enteros, decimales con decimales, cadenas con cadenas, etc. Sin embargo, en B4X puedes comparar también enteros con decimales y cadenas con números porque internamente se realiza la conversión de cadenas a números.

**Recuerda**

El resultado de una comparación es siempre un valor lógico (True o False)



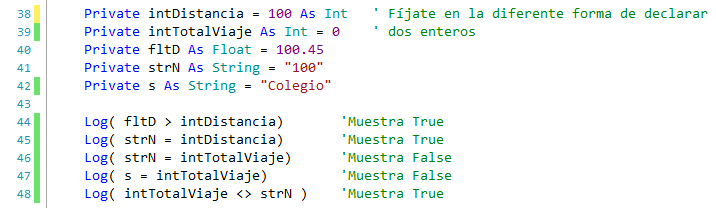


Imagen 1. Comparación de Variables

## Operadores Lógicos

Piensa en una afirmación como “Voy al colegio ahora”. ¿Es cierta o falsa?

El matemático George Boole creó un álgebra basada en sentencias lógicas.

En el álgebra booleana los valores de las variables son verdadero o falso. Normalmente se representan por un 1 o un 0, respectivamente. Al contrario que con el álgebra a la que estás acostumbrado, en el álgebra booleana hay 3 operaciones posibles: Y, O y Negación (en inglés, **AND**, **OR** y **NOT**).

**Recuerda**

**AND** (conjunción), indicado como **x AND y**. El resultado será 1 si **x** e **y** valen los dos 1. En otro caso, el resultado será 0.

**OR** (disyunción), indicado como **x OR y**. El resultado será 0 si **x** e **y** valen 0. En cualquier otro caso, el resultado es 1.

**NOT** (negación), indicado como **NOT x**. El resultado será 0 si **x** valía 1, o bien 1 si **x** valía 0.



Ejemplo:

**1ª Afirmación**: *Está diluviando*, **2ª Afirmación**: *Voy al colegio ahora*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Está diluviando**  **(P1)** | ***Voy al colegio ahora***  **(P2)** | **P1 AND P2** | **P1 OR P2** | **NOT P1** |
| True | True | True | True | False |
| True | False | False | True | False |
| False | True | False | True | True |
| False | False | False | False | True |

En la tabla podemos ver:

* Dos afirmaciones que al unirse con el operador **AND** dan como resultado un valor cierto sólo si las dos eran ciertas.
* Dos afirmaciones que al unirse con el operador **OR** dan como resultado un valor cierto si alguna de las dos eran ciertas.
* El operador NOT que invierte la veracidad o falsedad de una afirmación.

### Operadores Lógicos en programación

Los operadores lógicos se usan en programación para crear expresiones de comparación complejas. Esto permite al programador optimizar su código con menos líneas y con un código más sencillo.

En B4X las variables lógicas se usan así:

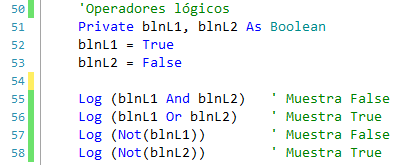


Imagen 2. Ejemplos de uso de variables lógicas

Fíjate que los operadores lógicos emplean variables lógicas o expresiones lógicas como veremos más adelante.

### Ejemplos de evaluación de sentencias lógicas

Supongamos que las siguientes variables tienen estos valores:

**Private** intA = 10, intB = 20, = 30 **As** **int**

**Private** strNombre1 = “Jorge”, strNombre2 = “Málaga”  **As**  **String**

**Private** blnA = True, blnB = False blnC = False **As** **Boolean**

Calcula el valor de las siguientes expresiones lógicas:

1.

|  |  |  |
| --- | --- | --- |
| blnA | **AND** | blnB |
| True |  | False |
|  | **False** |  |
|  |  |  |

2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Inta | **>** | intC | **AND** | blnA |  |
| 10 |  | 30 |  | True |  |
|  | False |  |  |  |  |
|  |  |  | **False** |  |  |

3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| intA + intB | **>=** | intC | **AND** | (blnA | **OR** | blnB) |
| 10 + 20 |  | 30 |  | True |  | False |
|  | True |  |  |  | True |  |
|  |  |  | **True** |  |  |  |

4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| intA + intB | **>=** | intC | **OR** | (blnA | **AND** | blnB) |
| 10 + 20 |  | 30 |  | True |  | False |
|  | True |  |  |  | False |  |
|  |  |  | **True** |  |  |  |

5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| strNombre1 | **=** | “Jorge” | **OR** | strName2 | **=** | “Juan” |
| Jorge |  | Jorge |  | Málaga |  | Juan |
|  | True |  |  |  | False |  |
|  |  |  | **True** |  |  |  |

## Sentencia If

Al igual que la vida real nos hacemos preguntas, en programación también hace falta comprar valores o cambiar el orden de ejecución del programa para que haga diferentes cosas.

La **sentencia If** se usa para realizar esas preguntas.

Su forma básica es al siguiente:

**If** ( condición ) **Then**

Instrucciones

**End If**

Donde **condición** debe contener una comparación o una expresión lógica de las vistas antes.

El significado es: **SI** la condición es **TRUE**, ejecuta las instrucciones que hay entre **Then** y **End** **If**.

Ejemplos:

**Private** intA = 10, intB = 20 **As Int**

**Private** fltA **As Float**

**If** intA > 0 **Then**

**Log**(intA & " es un número positivo")

**End If**

**If** intA > 10 **Or** intB > 10  **Then**

**Log**("Uno o los dos números son mayores que 10")

**End If**

**If** intA **Mod** 2 = 0 **Then**

**Log**(intA & " es un número par ")

**End If**

## If – Else

El comando **Else** permite ejecutar instrucciones cuando la condición del **If** es falsa:

Su forma básica es la siguiente:

**If** ( condición ) **Then**

Instrucciones\_Verdadero

**Else**

Instrucciones\_Falso

**End If**

El significado es: Si la **condición** es **True**, entonces ejecutar las **Instrucciones**\_**Verdadero**. Si la condición es **False**, ejecutar las **Instrucciones\_Falso**.

Ejemplos:

**Private** intA = 10, intB = 20 **As**  **Int**

**Private** fltA **As Float**

**If** intA > 0 **Then**

**Log**(intA & " es un número positivo ")

**Else**

**Log**(intA & " no es un número positivo ")

**End If**

**If** intA > 10 **Or** intB > 10 **Then**

**Log**("Uno o los dos números son mayores que 10")

**Else**

**Log**("Ninguno de los números es mayor que 10")

**End If**

**If** intA **Mod** 2 = 0 **Then**

**Log**(intA & " es un número par")

**Else**

**Log**(intA & " es un número impar")

**End If**

## If – else - else if

Podemos usar varios **If** con varias condiciones para extender la funcionalidad de un único comando **if** (se llama “anidar” comandos **If**).

Cómo lo construimos:

**If** ( condición1 ) **Then**

Instrucciones1

**Else If** ( condición2 )  **Then**

Instrucciones2

**Else If (** condición3 ) **Then**

Instrucciones3

**Else If (** condición4 ) **Then**

Instrucciones4

**...**

**Else**

Instrucciones\_Todo\_Falso

**End If**

El funcionamiento de los mútliples **If** anidados es el siguiente:

1. Se comprueba la primera condición (**condición1**). Si es **True**, ejecutamos el código **Instrucciones1** y el **If** finaliza (no se ejecuta nada más).
2. Si la primera condición del primer **If** (**condición1**) es **False**, entonces se comprueba la condición del segundo **If** (**condición2**) y, si es cierta, se ejecutan las **Instrucciones2** (y no se ejecuta nada más).
3. Para el resto de **If**, se comprueban sus condiciones si las anteriores son falsas.
4. Si **ninguna** de las condiciones de los **If** es cierta, se ejecutan las **Instrucciones\_Todo\_Falso**. El **Else** no es obligatorio usarlo.

**Ejemplo 3**

Un restaurante de comida rápida ofrece estas comidas:

|  |  |
| --- | --- |
| Comida | Precio |
| Hamburguesa | 5 € |
| Pizza | 3 € |
| Salchicha | 1,5 € |

Crea un programa que:

Lea el tipo de comida que el cliente quiere. Imprimar el coste de la comida. Por ejemplo: Entrada: “Salchicha”. Salida: “Salchicha 1,50 €”

Solución

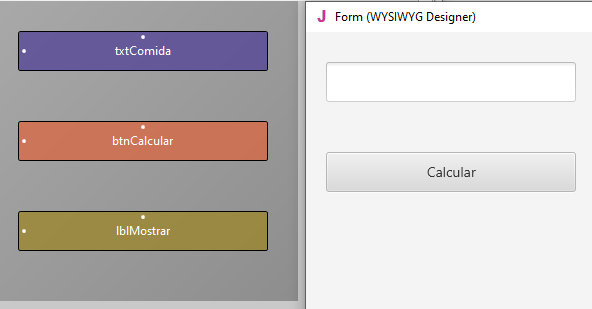


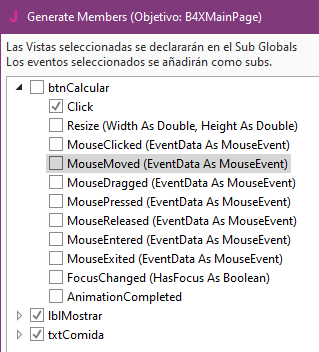
Imagen 3. Pantalla de la aplicación

**Paso 1**

Crea un nuevo proyecto de tamaño 300x300.

**Paso 2**

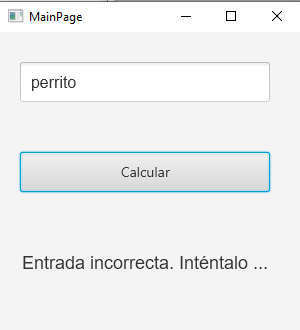
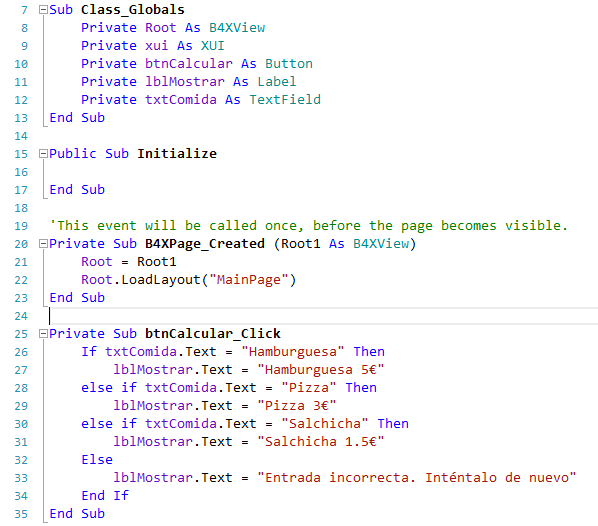
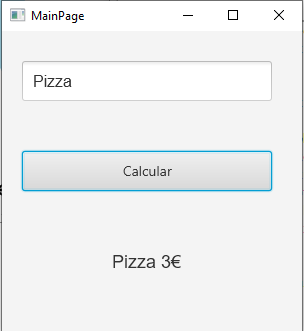
En el diseñador, crea la pantalla de la aplicación

**Paso 3**

Genera miembros para **txtComida**, **btnCalcular**, **lblMostrar** y **btnCalcular\_Click**.

**Paso 4**

El código que tienes que escribir en el **btnCalcular\_Click** es:



Fíjate que el texto que introducimos distingue entre mayúsculas y minúsculas, con lo que si escribimos “perrito” con la letra “p” en minúscula, no lo reconoce como válido porque en el código hemos dicho que tiene ponerse “Perrito”, con la “P” en mayúscula.

## Algoritmos con If

Vamos a ver cómo encontrar el máximo de un grupo de números.

Supongamos que leemos 3 números enteros y queremos encontrar el mayor de los 3. Veamos 3 formas diferentes de hacerlo:

Método 3 – Algoritmo Máx

**Máx = inta**

**If** intB > Máx  **then**

Máx = intB

**End If**

**If** intC > Máx  **then**

Máx = intC

**End If**

**Log(**Máx**)**

Método 2 –If anidado

**If** Inta > intB **then**

**If** Inta > intC **then**

**Log**(Inta)

**End If**

**End If**

**If** IntB > IntA  **then**

**If** IntB > intC  **then**

**Log**(IntB)

**End If**

**End**

**If** IntC > IntA  **then**

**If** IntC > IntA  **then**

**Log**(IntC)

**End If**

**End If**

Método 1 – Sentencia If simple

**If** Inta > intB  **AND** Inta > intC  **then**

**Log**(Inta)

**End If**

**If** intB > Inta  **AND** intB > intC  **then**

**Log**(intB)

**End If**

**If** intC > Inta  **AND** IntC > intB  **then**

**Log**(intC)

**End If**

## Ejercicios

1. Escribe las siguientes condiciones como expresiones lógicas:
   1. **A** pertenece al intervalo [-5, 6)
   2. **A** es menor que 3 o mayor que 15
   3. **A** es igual a **B** y **C**
   4. **A** no tiene el valor de 3
   5. **A** es menor que 2 o **B** es mayor de 78
   6. **A** y **B** son verdad y **C** es falsa
   7. **A** es verdad y o bien **B** o **C** son verdad
2. Calcula el resultado (True o False) de las siguientes expresiones lógicas suponiendo los siguientes valores de las variables:

A = 10, B = 2, C = -4, D = 9 and E = 1

* 1. (A>B) or (D=10)
  2. (D >= B) and (E <> C)
  3. not (E<=C) or (D<=C)
  4. not ((B<=C) and (D<2))
  5. not (not (B<=E) or not (C<=B))
  6. ((E<=A) and (E>=C)) and not (C>=A)
  7. not (not (A >= 2) and (C <>9))

1. Un restaurante de comida rápida ofrece estas comidas:

| **Comida** | **Precio** |
| --- | --- |
| Hamburguesa | 5 € |
| Pizza | 3 € |
| Salchicha | 1,5 € |

Crea un programa que:

Primero lea la comida que el cliente quiere y después cuántos productos quiere. Muestre el coste total de la comida.

Ejemplo de entrada: "Salchicha", 2

Salida: " 2 x Salchicha 3 €"

1. Has gastado una cantidad de X megabits en la Wikipedia y una cantidad Y de megabits en memes. El coste de visitar la Wikipedia es de 0’1€ por megabit y el de ver memes es de 0’05 € por megabit. Si el consumo total es mayor de 100 €, debes mostrar el mensaje “Consumo excesivo”. Si gastas más en ver memes que en visitar la Wikipedia, debes imprimir el mensaje “Demasiados memes”. Crea un programa que:
   1. Lea el valor de X (consumo de megabits de Wikipedia) y el de Y (consumo de megabits por ver memes).
   2. Calcule el consumo total.
   3. Si el consumo total es mayor de 100€, debe imprimir el mensaje adecuado. Si el consumo en ver memes es mayor que el de visitar la Wikipedia, se imprime el mensaje correcto.
2. Un cibercafé tiene 2 formas de cobrar. Si el usuario es miembro, pagaría 2€ por hora, si no, paga 5€ por hora. Averigua si alguien es miembro o no y calcula el precio que pagaría en función de las horas que emplee. Si el usuario es miembro, el impuesto es del 10%, si no, es del 20%. Crea un programa que:
   1. Lea cuántas horas ha usado el cliente.
   2. Pregunte si es miembro o no.
   3. Añada el impuesto adecuado.
   4. Imprima lo que tiene que pagar el usuario mostrando el texto: "El usuario es miembro y ha estado 2 horas por 2€/hora más el 10% de impuesto, que hace un total de 4’4€".
3. Quiere comprar algo en Amazon. El vendedor cobra diferentes gastos de envío según tu ubicación. Para EEUU son 5€, para Europa 7€, por Canadá 3€ y para el resto 9€. Crea un programa que:
   1. Lea el coste del producto.
   2. Lea tu ubicación.
   3. Imprima los gastos de envío.
   4. La salida debe ser: "Tienes que pagar 23€, 20€ por el producto y 3€ por los gastos de envío".
4. Una empresa vende un producto por 0’70€ la unidad si se piden hasta 200 unidades, o bien 0’50€ si se piden más de 200 unidades. Lee el número de piezas que se piden y calcula su valor.
5. Una empresa de telefonía tiene las siguientes tarifas:

|  | **Coste fijo 25€** |
| --- | --- |
| Duración de la llamada (en segundos) | Coste (€/por segundo) |
| 1-500 | 0,01 |
| 501-800 | 0,008 |
| 801+ | 0,005 |

Crea un programa que:

* 1. Lea el número de segundos que todas las llamadas.
  2. Calcule la factura mensual para el cliente.
  3. Imprima la cantidad total.
  4. Salida: “Importe total: 48€”.
  5. Fíjate que el cargo de los primeros 500 segundos es 0’01€ por segundo, para los segundos 501 a 800 es de 0’008€ y a partir de ahí 0’005€.

1. En la fase de clasificación en salto de longitud de los juegos olímpicos, un atleta realiza 3 intentos iniciales y, si logra una distancia mayor a 7’50 metros, entonces pasa de ronda y puede realizar otros 3 intentos más. Escribe un programa que lea los primeros 3 intentos de un atleta e imprima un mensaje diciendo si puede pasar de ronda o no y que también muestre la mayor distancia que ha saltado.

**Puedes usar el método Visible = True or False para ocultar o mostrar Labels, TextFields y Buttons.**

1. En una ciudad hay aparcamientos de pago. El coste de aparcar se calcula de la siguiente forma:

|  |  |
| --- | --- |
| **Duración aparcamiento** | **Coste por hora** |
| Hasta 1 hora 3.50 € | |
| Las siguientes 2 horas | 8.00 € |
| Las siguientes 2 horas | 12.00€ |
| Más de 5 horas | 15.00 € |

Escribe un programa que lea el tiempo que alguien dejó su coche en el aparcamiento y que calcule el coste total.