

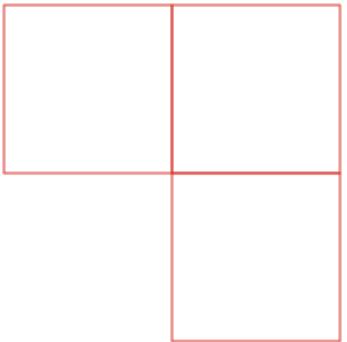
1

Expresiones condicionales



Ve más allá



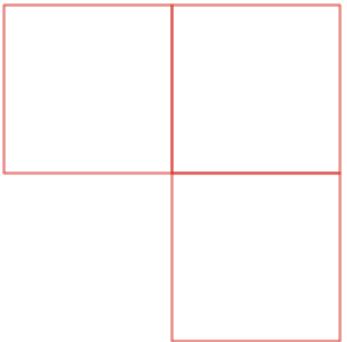


CONTENIDOS

- **Unidad 2: Sentencias de Control**
 1. **Expresiones condicionales**
 2. Sentencias de decisión
 3. Sentencias de repetición



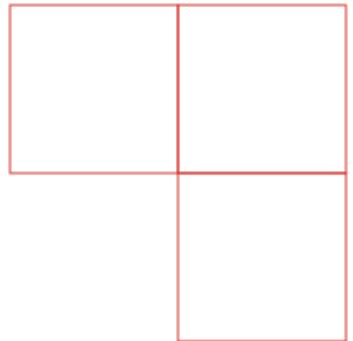
Imagen freepik.es



INDICE

- Expresiones condicionales
 - 1. Introducción
 - 2. Instrucciones secuenciales
 - 3. Álgebra de Boole
 - 4. Tablas de Verdad
 - 5. Tipos de sentencias

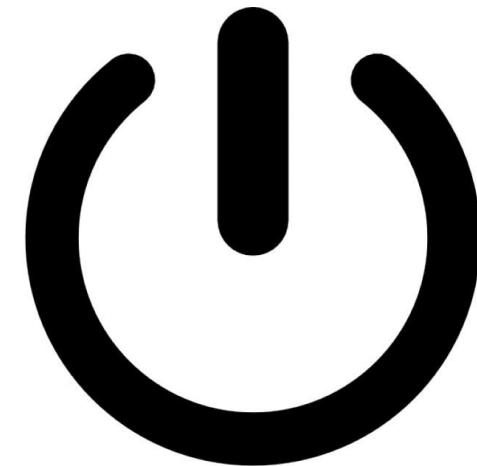


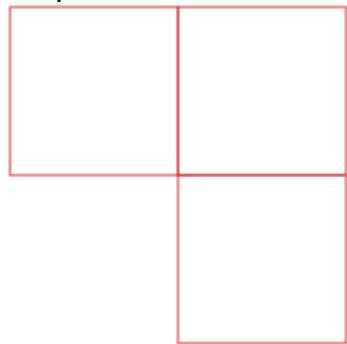


Introducción

En esta sección nos introducimos en la lógica booleana, que es la que utilizan los ordenadores internamente tanto a nivel físico o hardware (en sus circuitos) como a nivel lógico o software (en los programas).

Se basa en dos posibilidades que habitualmente conocemos como verdadero o falso y que internamente se refiere a los valores que puede tomar un bit: 0 o 1. En un circuito el 1 es que pasa corriente y el 0 que no pasa corriente.

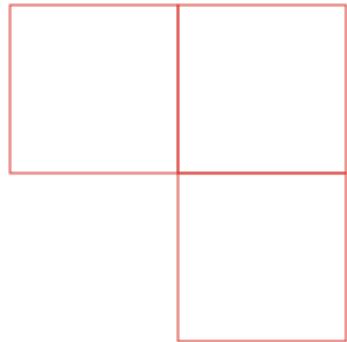




Instrucciones secuenciales

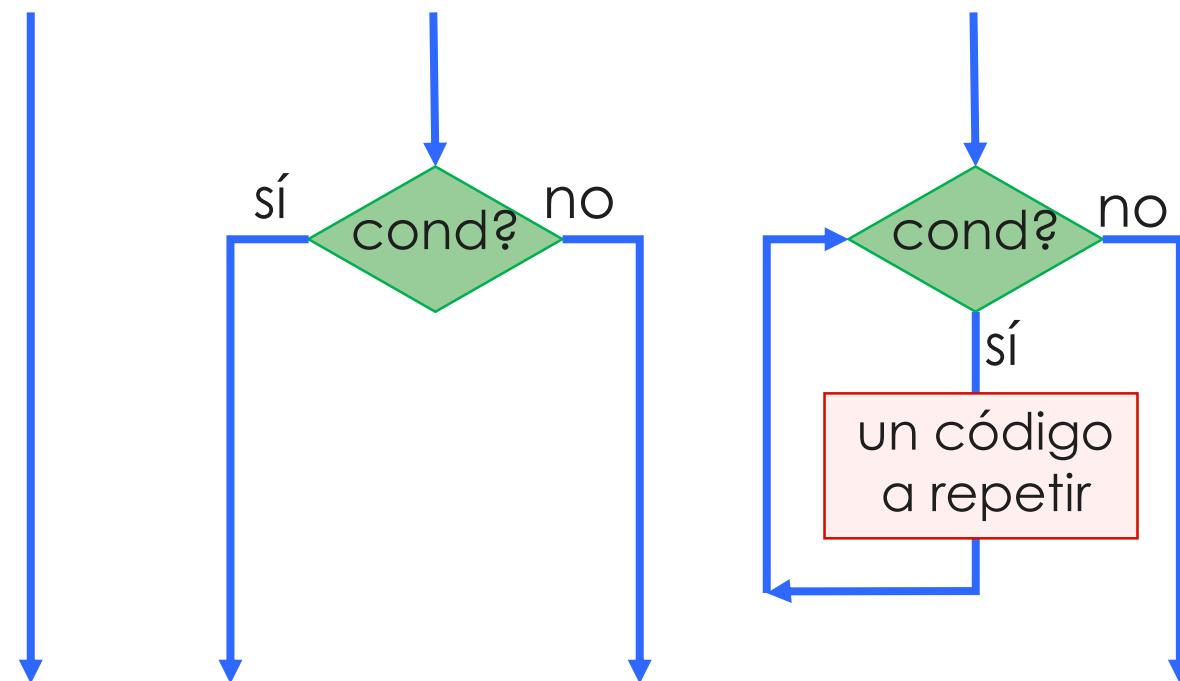
Hasta ahora, lo normal en un programa es que escribamos una instrucción detrás de otra para que el ordenador las ejecute en esa secuencia



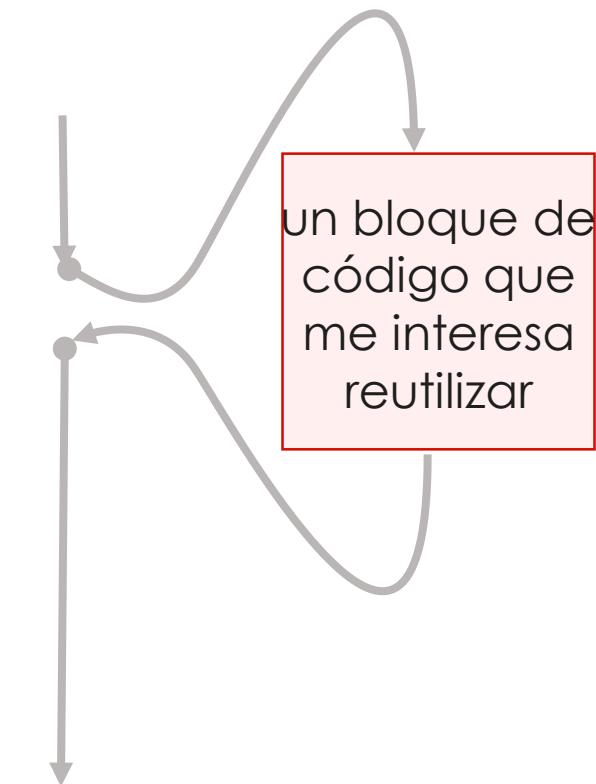


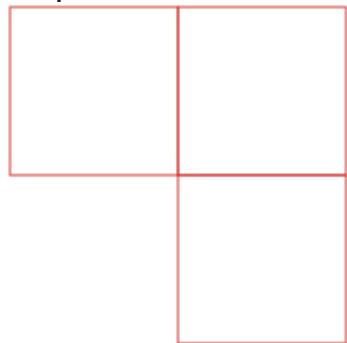
Instrucciones secuenciales

Pero no siempre nos interesa que sea así



Por lo que debemos estudiar cómo construir esas condiciones.





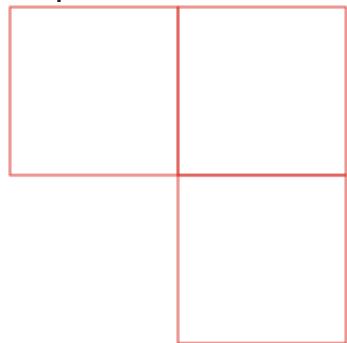
Álgebra de Boole

El álgebra de Boole debe su nombre a **George Boole** (1815-1864), matemático inglés que fue el primero en definirla como parte de un sistema lógico.

Es un sistema matemático que utiliza variables que pueden valer 1 o 0 es decir true o verdadero y false o falso.

Podemos verle en la siguiente ilustración:





Álgebra de Boole

Todos los operadores relacionales que ya vimos en la unidad anterior devuelven uno de esos dos valores. Por ejemplo, prueba a poner en un programa la siguiente instrucción:

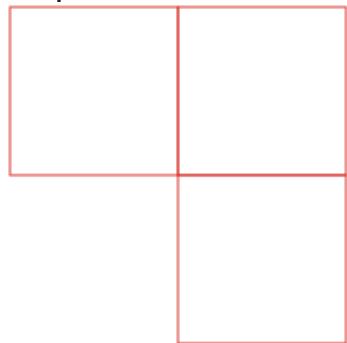
```
boolean condicion1 = 8 > 5;
```

¿Qué valor tendrá la variable condicion1?

- a) 0
- b) 1
- c) true
- d) false

Sin embargo, a veces nuestras condiciones pueden ser más complejas

estaEncendido() y nivelCombustible > 0



Álgebra de Boole

Llamamos expresiones condicionales o booleanas a las que combinan operadores relacionales y lógicos y tienen como resultado un valor booleano (true o false).

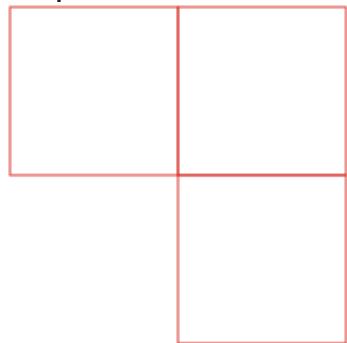
Por ejemplo, prueba a poner en un programa la siguiente instrucción:

```
boolean condicion2 = 8 > 5 && 6 < 3;
```

¿Qué valor tendrá la variable condicion2?

- a) 0
- b) 1
- c) true
- d) false





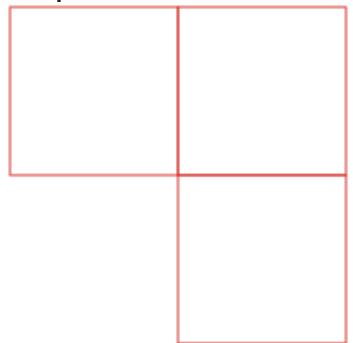
Tablas de verdad

Las tablas de verdad se utilizan para conocer el valor de un operador ante las posibles combinaciones de sus operadores.

Se utilizan en la lógica formal y se aplican igualmente a la lógica computacional. En consecuencia, las tablas de verdad constituyen un método de decisión para chequear si una expresión es o no cierta.

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Aunque en otros lenguajes sea equivalente usar 0 y 1, en java usamos boolean que admite solo los valores true y false

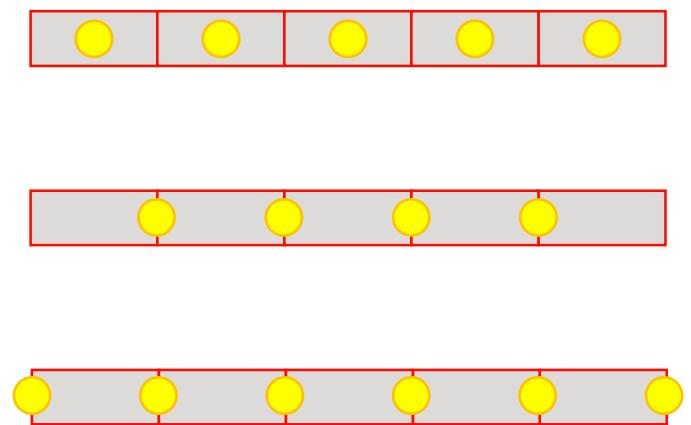


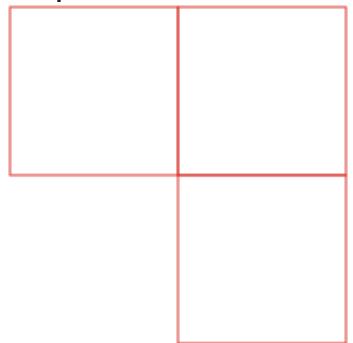
Tablas de verdad

Ojo con las condiciones de borde (debemos usar p.e. “<” o “<=”?)

Imagina que Tenemos una calle de 100m y queremos poner farolas cada 20m

¿Cuántas farolas necesitamos?





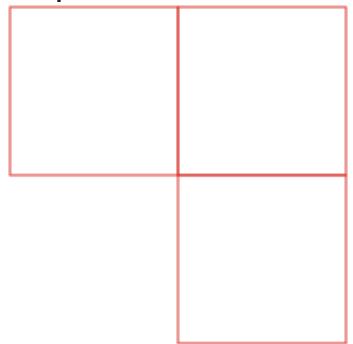
Ejercicio 1

Dadas las siguientes variables:

```
boolean v1 = true;  
boolean v2 = false;  
boolean v3 = true;
```

¿Qué valor tendrá la variable condicion3 = (v1 && v2) || (v1 && v3)?

- a) true
- b) false



Ejercicio 1

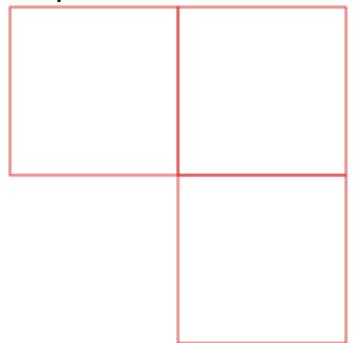
Para entender la respuesta veamos el siguiente desarrollo:

$(v1 \&\& v2) \mid\mid (v1 \&\& v3)$

$(\text{true} \&\& \text{false}) \mid\mid (\text{true} \&\& \text{true})$

false || true

true



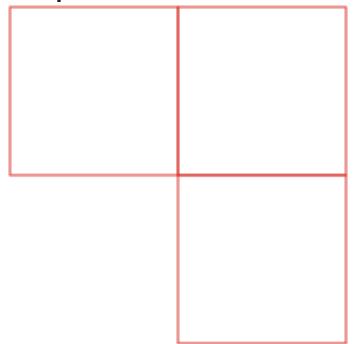
Ejercicio 2

Dadas las siguientes variables:

```
boolean v1 = true;  
boolean v2 = false;  
boolean v3 = true;
```

¿Qué valor tendrá la variable `condicion4 = (v1 || !v2) && (!v1 || v3)` ?

- a) true
- b) false



Ejercicio 2

Para entender la respuesta veamos el siguiente desarrollo:

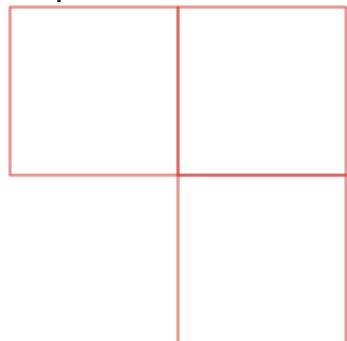
$(v1 \mid\mid !v2) \quad \&\& \quad (!v1 \mid\mid v3)$

$(\text{true} \mid\mid \text{!false}) \quad \&\& \quad (\text{!true} \mid\mid \text{true})$

$(\text{true} \mid\mid \text{true}) \quad \&\& \quad (\text{false} \mid\mid \text{true})$

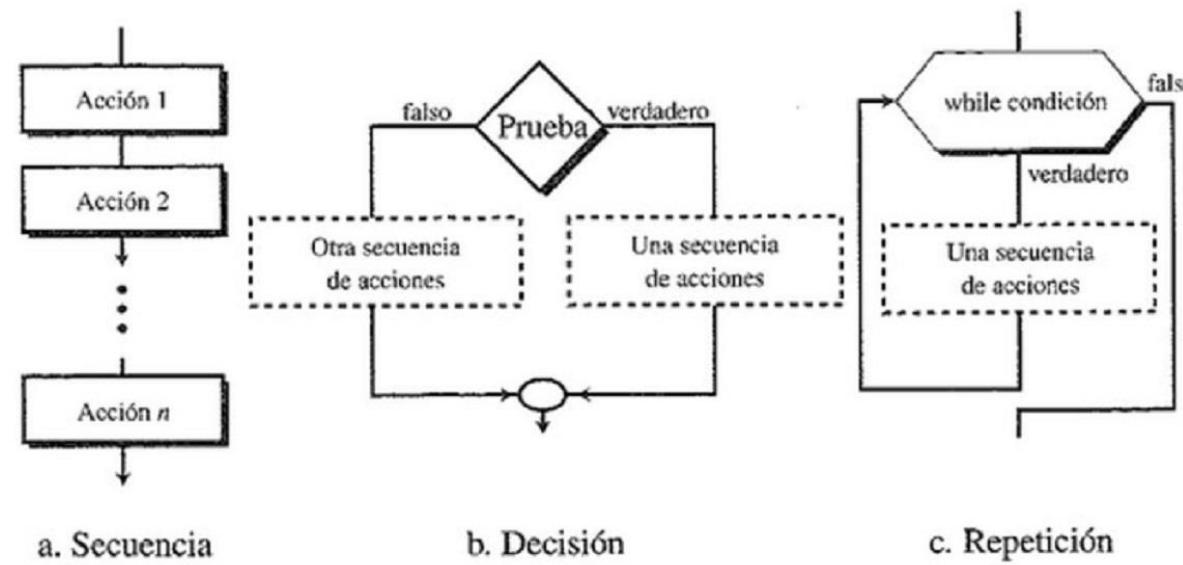
true $\&\&$ true

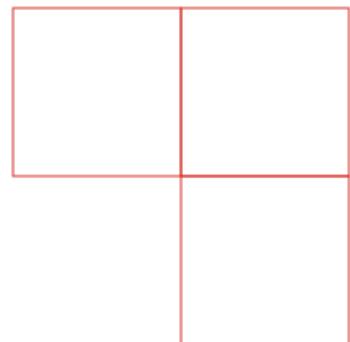
true



Tipos de sentencias

En este apartado vamos a hacer una mención especial a la que es considerada la primera programadora de la historia: **Ada Byron**. En 1843 pronosticó los tres tipos de sentencias que debería tener una máquina de propósito general (lo que ya sabemos que es un ordenador). Los tipos son: secuenciales, decisión y repetición, tal y como puede verse en la siguiente imagen:





APUNTE...



Tim Urban: En la mente de un maestro procrastinador | TED Talk ...
https://www.ted.com/talks/tim_urban_inside_the_mind_of_a_master_procrastinator?...es





**Universidad
Europea**

GRACIAS

Pedro J. Camacho

Universidadeuropea.com

Ve más allá