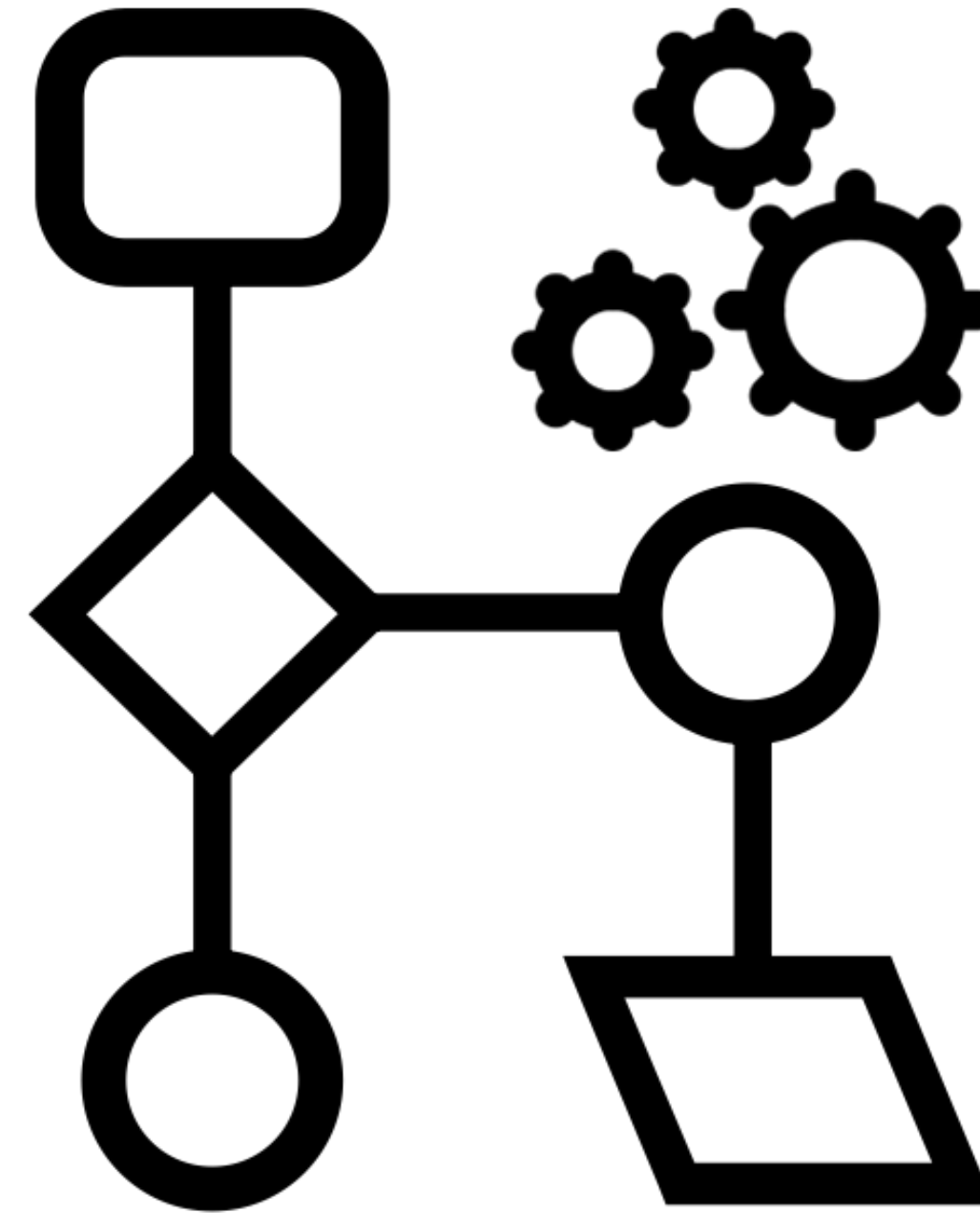


Lógica y condicionales





¿Qué es la lógica?



La lógica es la disciplina filosófica que tiene un carácter formal, ya que estudia la estructura o formas de pensamiento (tales como conceptos, proposiciones, razonamientos) con el objeto de establecer razonamientos o argumentos válidos o correctamente lógicos.(1)

$$\frac{p \quad p \rightarrow q}{\therefore q}$$

"Si P implica Q; y si P es verdad; entonces Q también es verdad."

¿Qué es la lógica?



La lógica es la disciplina filosófica que tiene un carácter formal, ya que estudia la estructura o formas de pensamiento (tales como conceptos, proposiciones, razonamientos) con el objeto de establecer razonamientos o argumentos válidos o correctamente lógicos.(1)

$$\frac{p \quad p \rightarrow q}{\therefore q}$$

"Si P implica Q; y si P es verdad; entonces Q también es verdad."

Modus ponendo ponens



El modus ponendo ponens (latín: "el modo que, al afirmar, afirma", también llamado modus ponens, eliminación de la implicación, regla de separación, afirmación del antecedente.(2)



Ejemplo

Si está lloviendo, te espero dentro del teatro.

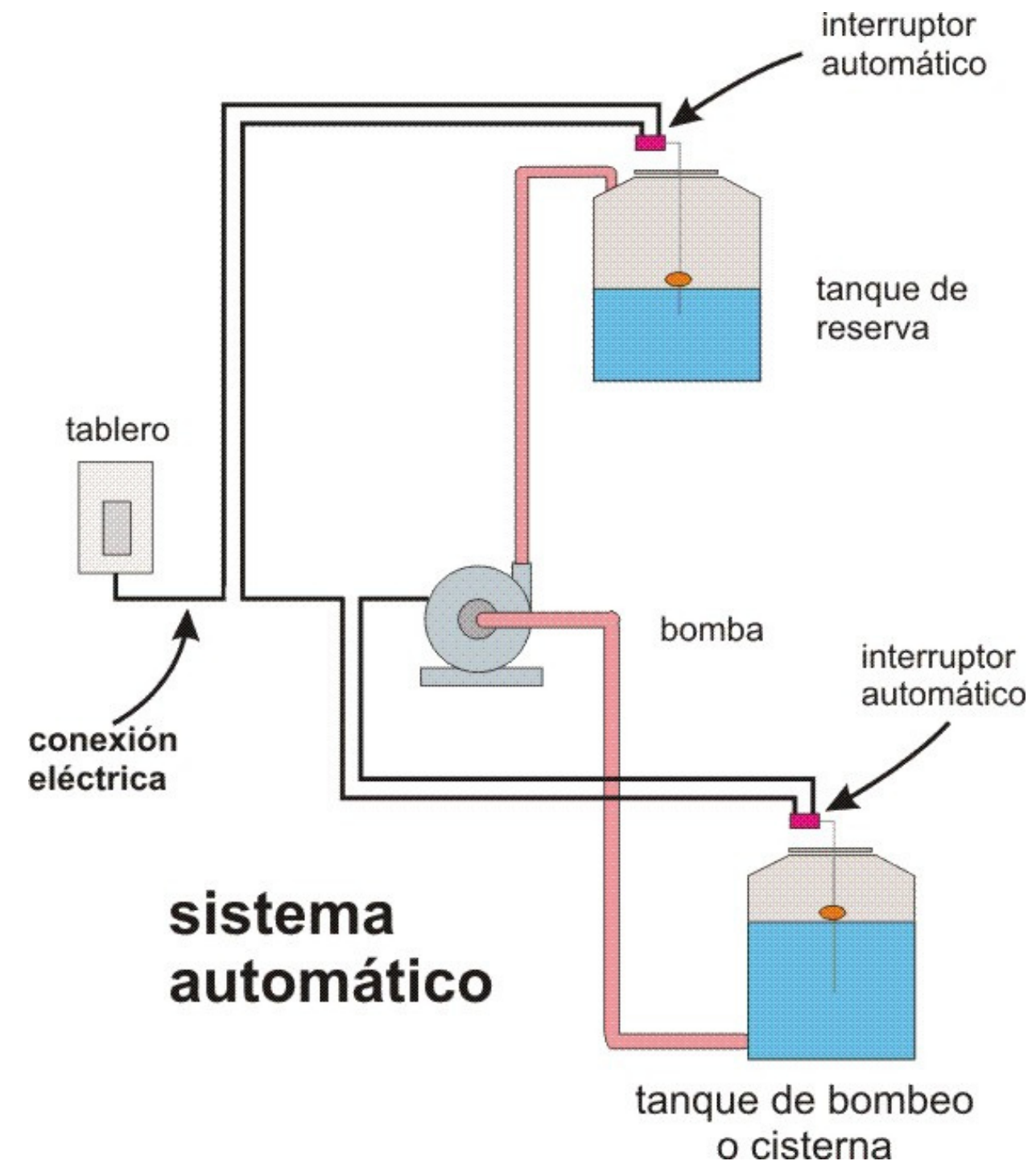
Está lloviendo.

Por lo tanto, te espero dentro del teatro.

Aplicación



1. Si se acaba el agua del tinaco, enciende la bomba de agua.
El agua del tinaco se agotó.
Por lo tanto, enciende la bomba.
2. Si hay agua en la cisterna, enciende la bomba de agua.
Hay agua en la cisterna.
Por lo tanto, enciende la bomba.



Álgebra Booleana



El álgebra de Boole fue desarrollada por George Boole en 1854. Es una estructura matemática que esquematiza las operaciones lógicas. Es decir que por medio de circuitos eléctricos, se interpretan operaciones lógicas.(3)

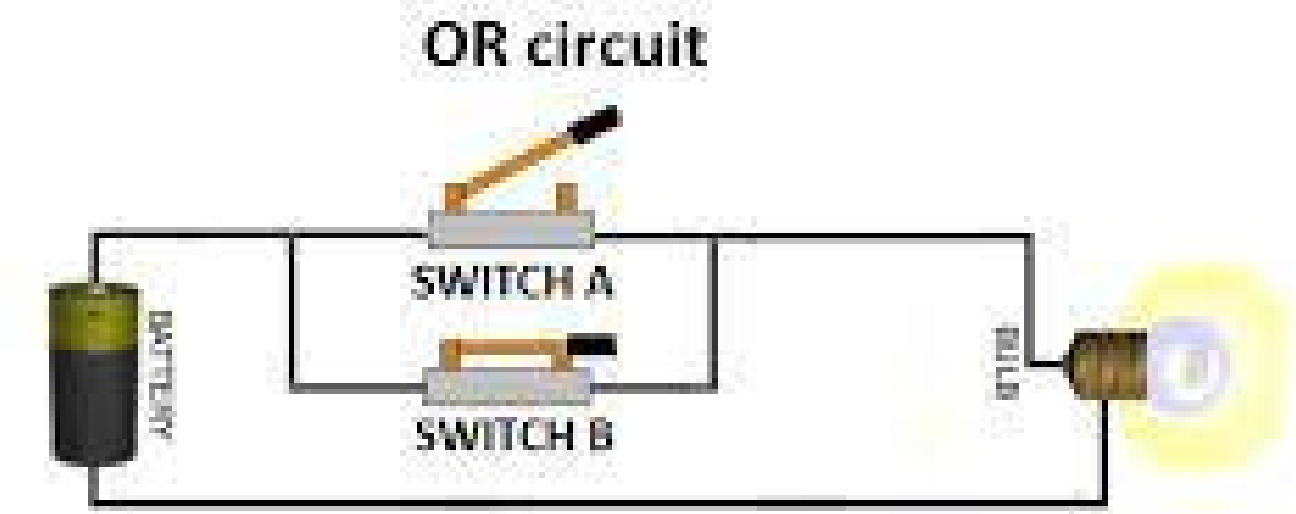
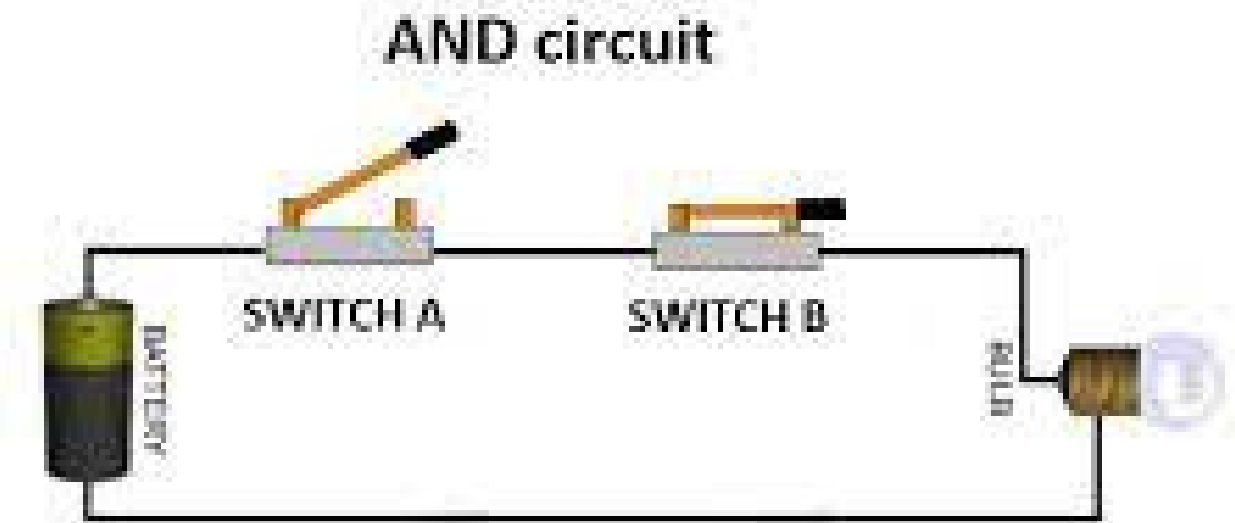


Tabla de verdad



p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

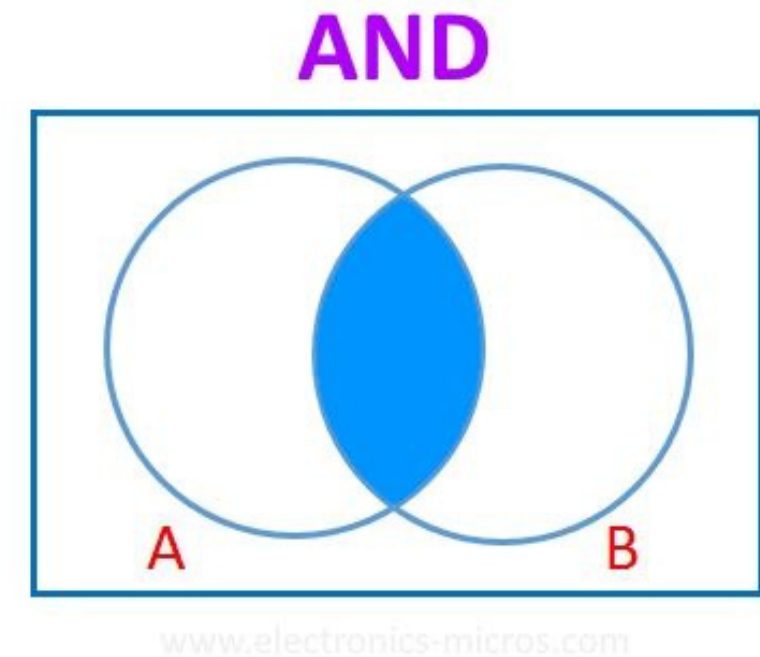
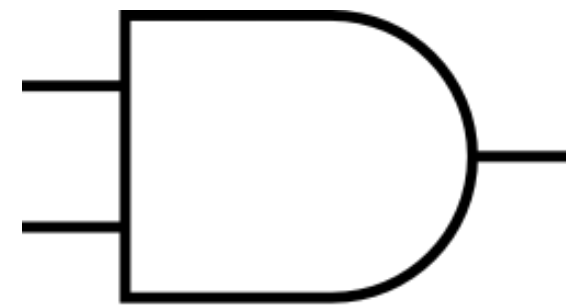
V y F quiere decir Verdadero y Falso, en inglés se interpreta como TRUE y FALSE.

Una tabla de verdad, o tabla de valores de verdades, es una tabla que muestra el valor de verdad de una proposición compuesta, para cada combinación de verdad que se pueda asignar.(4)

Operador AND

Es una puerta lógica digital que implementa la conjunción lógica, es decir, desde un enfoque matemático, obedece la lógica de una multiplicación.
(5)

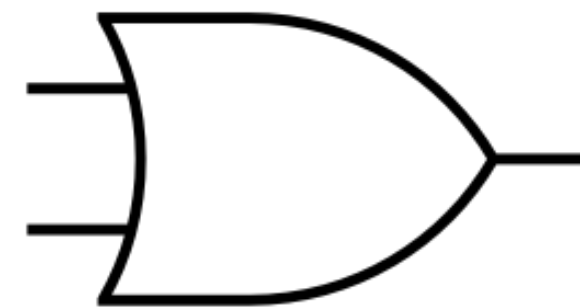
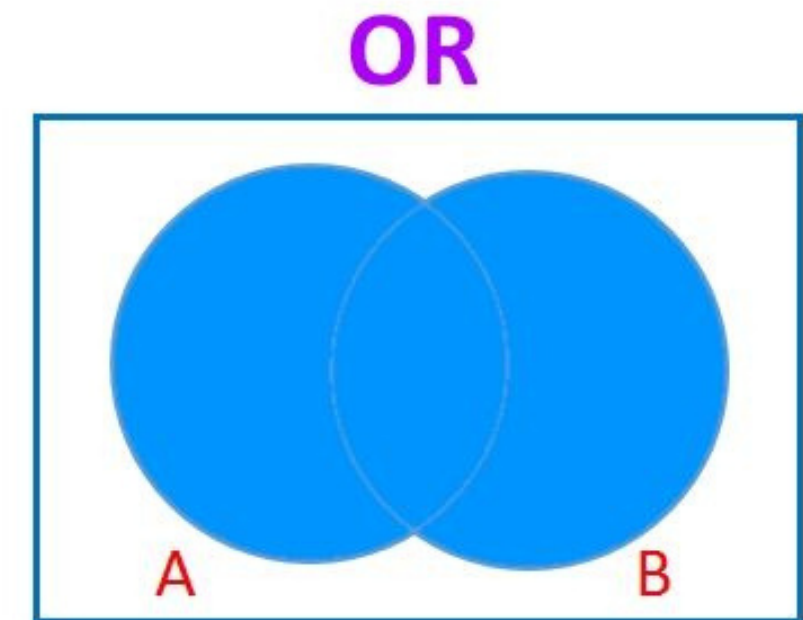
AND		
x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1



Operador OR

Es una puerta lógica digital que implementa la disyunción lógica, es decir, desde un enfoque matemático, obedece la lógica de una suma o adición.
(6)

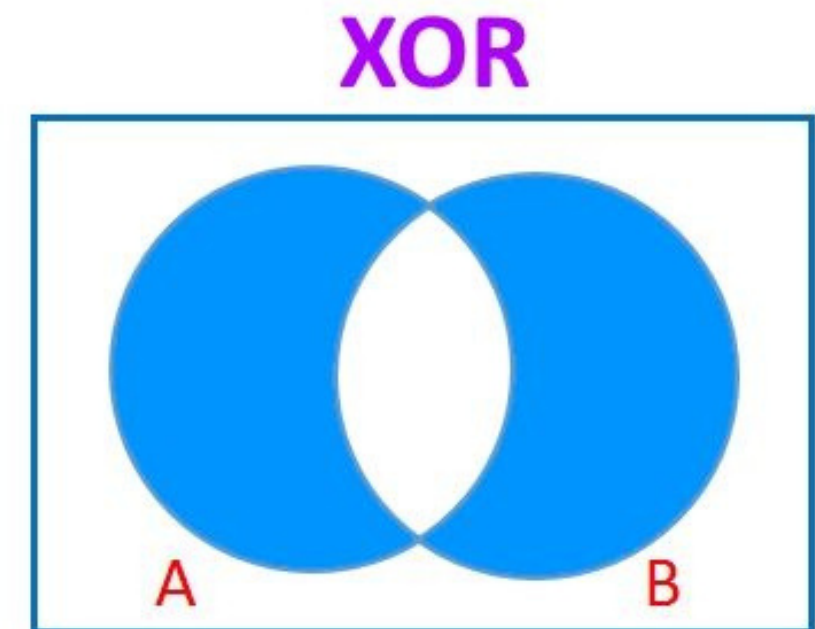
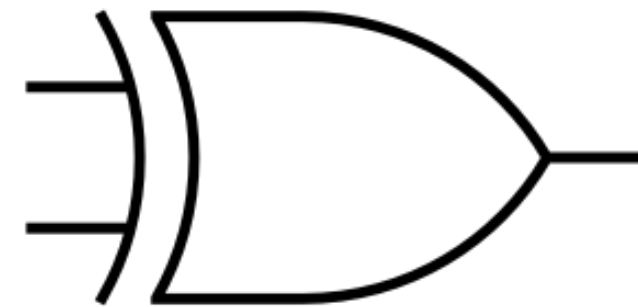
OR		
x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1



Operador XOR

Es una puerta lógica digital que implementa el o exclusivo, es decir, desde un enfoque matemático, obedece la lógica de una suma o adición sólo que en esta tabla de verdad, cuando ambos valores son verdaderos, la salida será falsa o cero.(7)

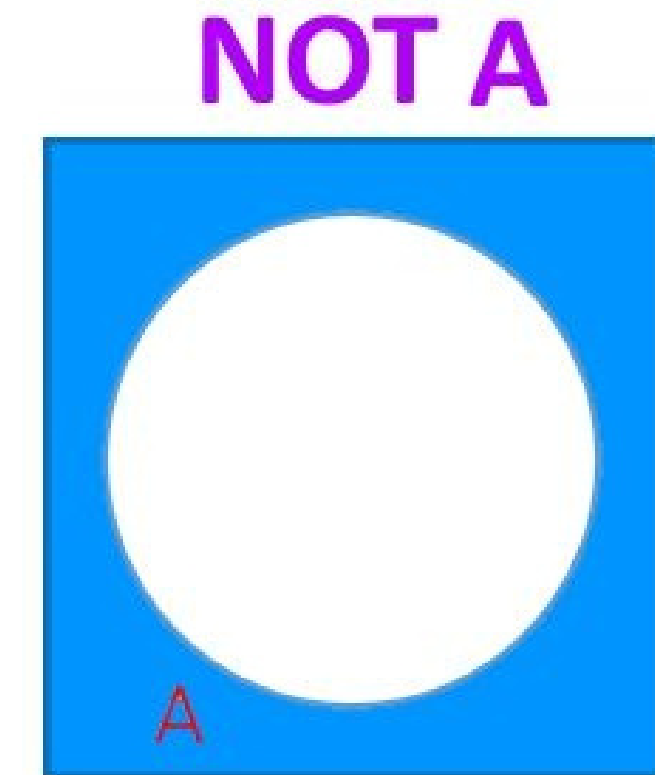
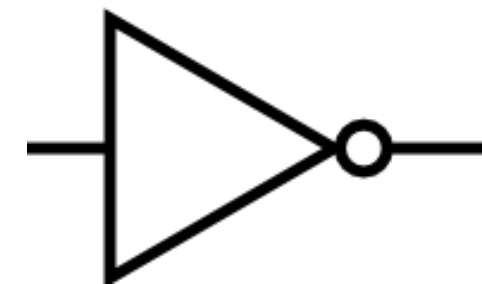
XOR		
x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



Operador NOT

Es una puerta lógica que implementa la negación lógica, es decir, esta compuerta "niega" o "invierte" el valor que tengamos a la entrada.(8)

<i>NOT</i>	
<i>x</i>	<i>x'</i>
0	1
1	0



Operadores lógicos en Arduino



Operadores Bit a Bit

Bitwise Operators

& (bitwise and)

<< (bitshift left)

>> (bitshift right)

^ (bitwise xor)

| (bitwise or)

~ (bitwise not)

Operadores Booleanos

Boolean Operators

! (logical not)

&& (logical and)

|| (logical or)

Operadores Bit a Bit



Se aplican directamente a los bits "físicos" de un dato, es decir, vamos a operar directamente sobre los 1s o 0s respetando la lógica de cada operador.

Ejemplo AND(9)

$$\begin{array}{r} 10011101 \\ \& \\ 10110011 \\ \hline 10010001 \end{array}$$

Operadores Bit a Bit



Ejemplo OR

	1	0	0	1	1	1	0	1
	1	0	1	1	0	0	1	1
<hr/>								
	1	0	1	1	1	1	1	1

Ejemplo NOT

	1	0	0	1	1	1	0	1
~	<hr/>							
	0	1	1	0	0	0	1	0

Operadores Bit a Bit



¿Cómo se usan en el código?

```
resultado = operador1 & operador2;
```

Variable donde
se guardará el
resultado.

Primer dato.

Operador lógico.

Segundo dato.

Operadores Booleanos



Ejemplo AND(10)

Este tipo de operador también es lógico, pero sólo funciona con datos tipo Bool, es decir, sólo funciona con datos TRUE o FALSE.

$(\text{TRUE} \ \&\& \ \text{TRUE}) = \text{TRUE}$

$(\text{TRUE} \ \&\& \ \text{FALSE}) = \text{FALSE}$

$(\text{FALSE} \ \&\& \ \text{TRUE}) = \text{FALSE}$

$(\text{FALSE} \ \&\& \ \text{FALSE}) = \text{FALSE}$

Operadores Booleanos



Ejemplo OR

$(\text{TRUE} \parallel \text{TRUE}) = \text{TRUE}$

$(\text{TRUE} \parallel \text{FALSE}) = \text{TRUE}$

$(\text{FALSE} \parallel \text{TRUE}) = \text{TRUE}$

$(\text{FALSE} \parallel \text{FALSE}) = \text{FALSE}$

Ejemplo NOT

$(!\text{TRUE}) = \text{FALSE}$

$(!\text{FALSE}) = \text{TRUE}$

Operadores Booleanos



Ejemplo OR

$(\text{TRUE} \parallel \text{TRUE}) = \text{TRUE}$

$(\text{TRUE} \parallel \text{FALSE}) = \text{TRUE}$

$(\text{FALSE} \parallel \text{TRUE}) = \text{TRUE}$

$(\text{FALSE} \parallel \text{FALSE}) = \text{FALSE}$

Ejemplo NOT

$(\text{!TRUE}) = \text{FALSE}$

$(\text{!FALSE}) = \text{TRUE}$

Operadores Booleanos



Ejemplo OR

$(\text{TRUE} \parallel \text{TRUE}) = \text{TRUE}$

$(\text{TRUE} \parallel \text{FALSE}) = \text{TRUE}$

$(\text{FALSE} \parallel \text{TRUE}) = \text{TRUE}$

$(\text{FALSE} \parallel \text{FALSE}) = \text{FALSE}$

Ejemplo NOT

$(\text{!TRUE}) = \text{FALSE}$

$(\text{!FALSE}) = \text{TRUE}$



Tiempo de descanso y de restablecer humanidad.(:

Operadores comparativos.



Este tipo de comparadores son de carácter cuantitativo, y evalúan las expresiones de manera "matemática", es importante recordar que para usar estas expresiones, los dos datos a comparar tendrán que ser del mismo tipo.

Comparison Operators

!= (not equal to)

< (less than)

<= (less than or equal to)

== (equal to)

> (greater than)

>= (greater than or equal to)

Operadores comparativos.



Comparison Operators

\neq (not equal to)

$<$ (less than)

\leq (less than or equal to)

$==$ (equal to)

$>$ (greater than)

\geq (greater than or equal to)

Ejemplos

$(5 \neq 23) = \text{TRUE}$

$(8 < 3) = \text{FALSE}$

$(45 \leq 45) = \text{TRUE}$

$(7 == 3) = \text{FALSE}$

$(5 < 99) = \text{TRUE}$

$(10 \geq 1) = \text{TRUE}$

Operadores comparativos.



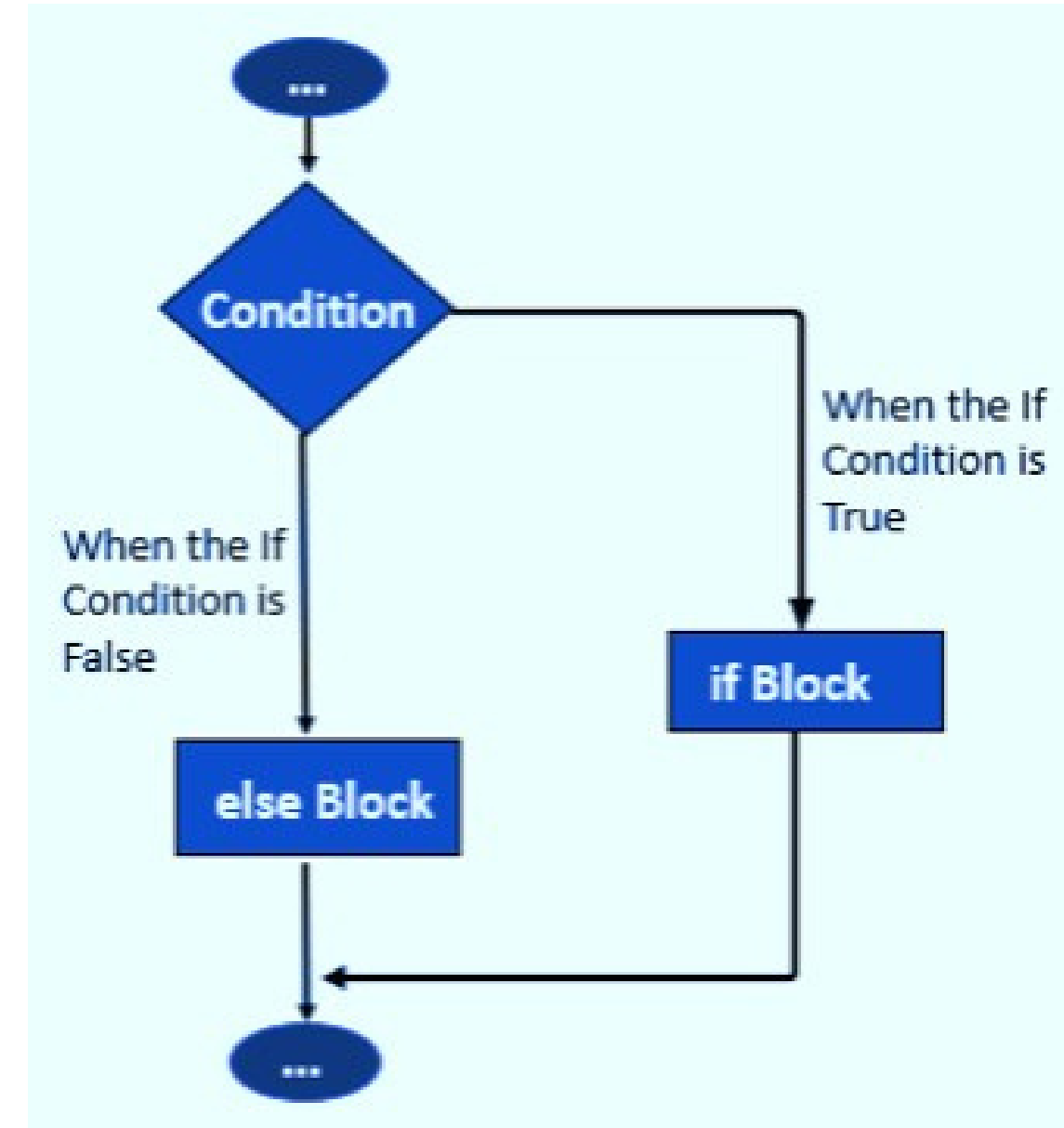
Es común que este tipo de comparadores se usen para monitorear variables de un sensor o la lectura de algún sistema.

```
(temperaturaDeseada == temperaturaSensor1)
```

Condicional IF



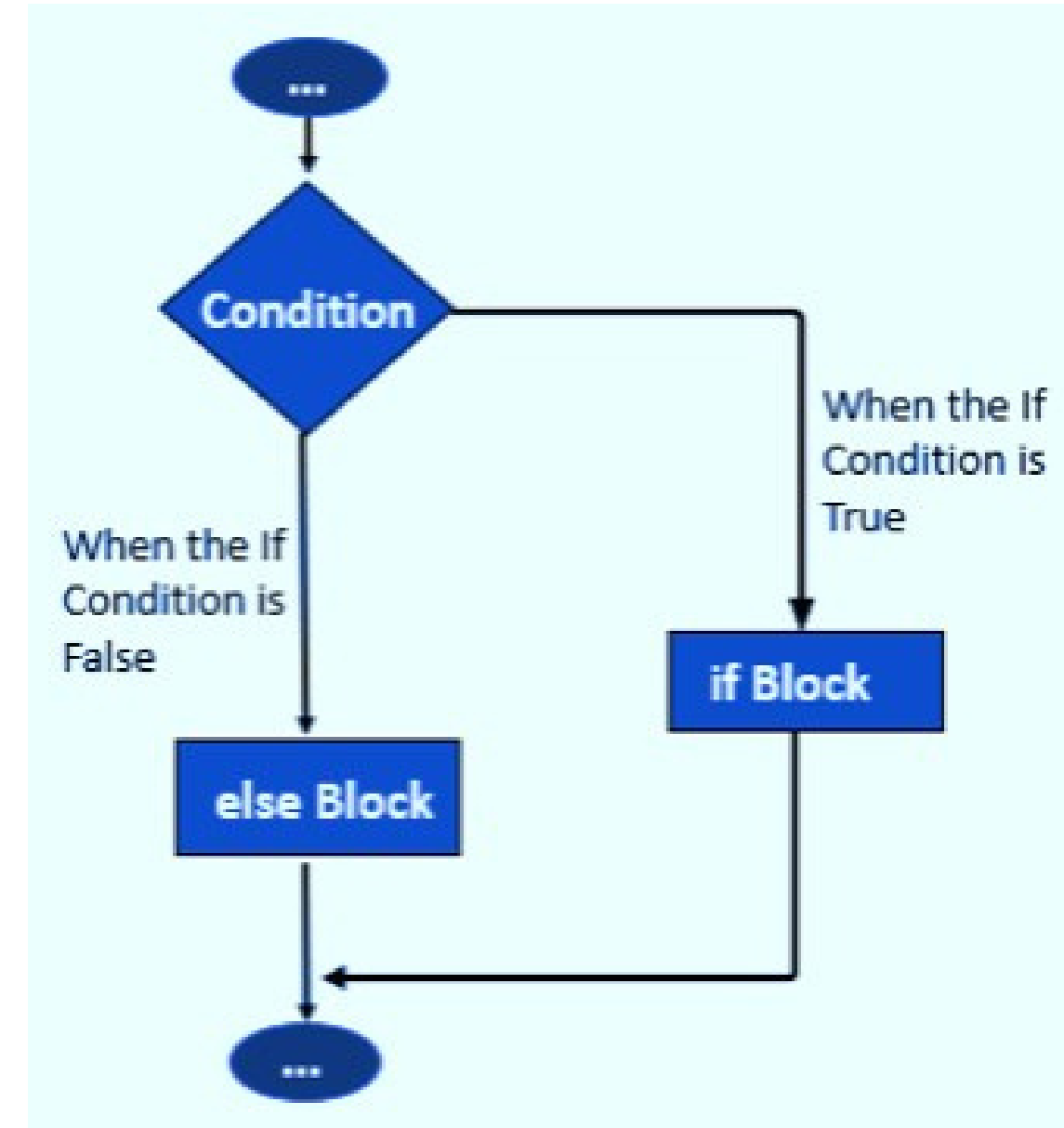
Esta estructura de control, evalúa una condición lógica, si esta condición se cumple, ejecuta las instrucciones programadas, de lo contrario no lo hace.
(11)



ELSE



Podemos agregar una serie de instrucciones en caso de que la condición no se cumpla, a esta parte se le conoce como bloque 'ELSE'.



¿Cómo se usa un IF en Arduino?



```
if(condición){  
    dentro de los corchetes  
    van las instrucciones que  
    queremos  
}
```

¿Cómo se usa un IF en Arduino?



```
if(condición){  
    dentro de los corchetes  
    van las instrucciones que  
    queremos  
}  
else{  
    instrucciones en caso de  
    que la condición no se  
    cumpla  
}
```

Ejemplo



En esta condicional "pregunto" si la variable temperatura es mayor a 50 grados, si es así, ejecuto una función que enciende un ventilador.

```
if(temperatura > 50){  
    enciendeVentilador();  
}
```



Ejemplo

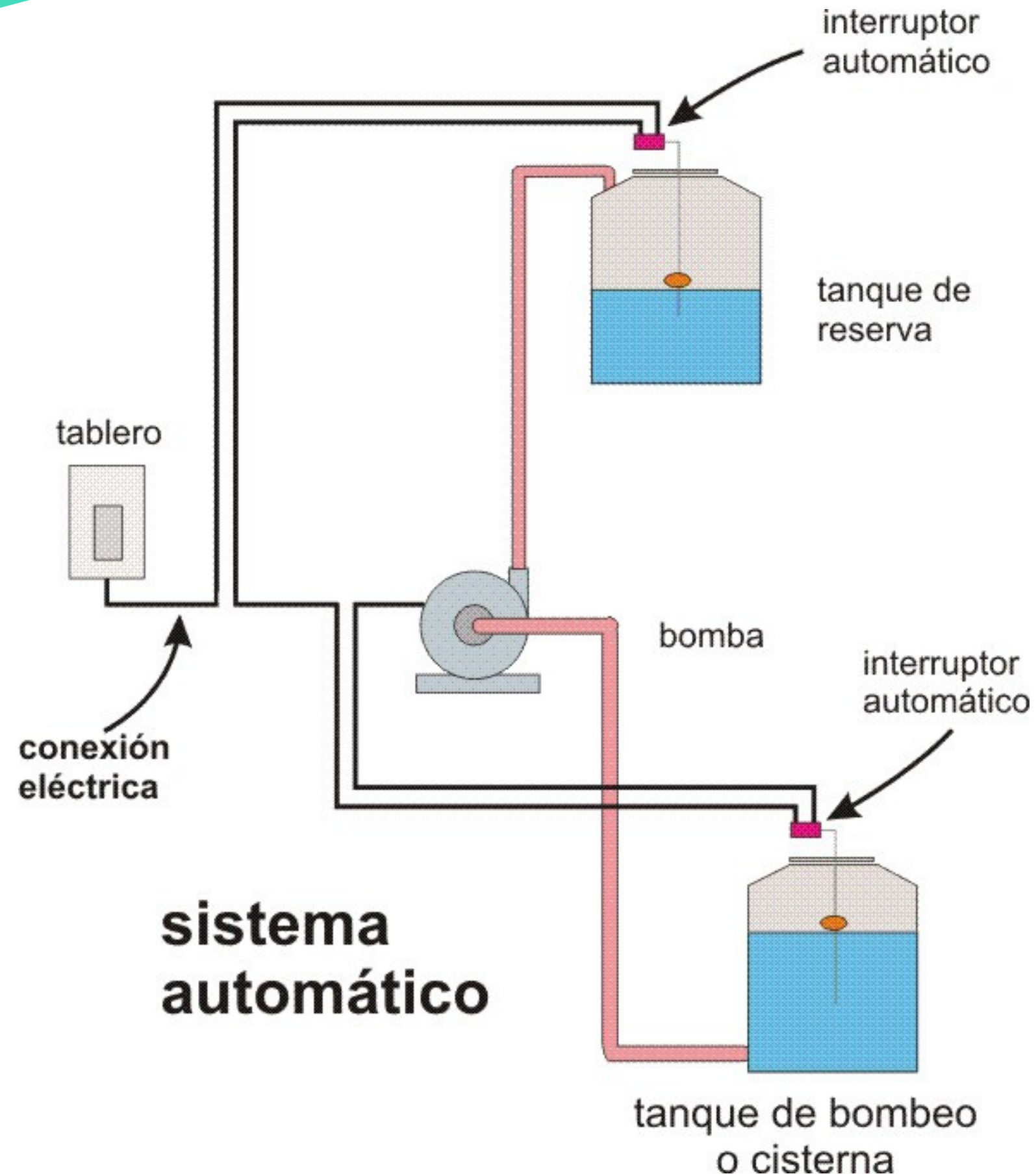


En esta condicional "pregunto" si la variable temperatura es igual a 50 grados ó si la humedad es mayor a 100, si se cumple cualquiera de las dos, ejecuto una función que enciende un ventilador.

```
if((temperatura > 50) || (humedad > 100)){  
    enciendeVentilador();  
}
```


Ejemplo

¿Cómo se
soluciona
este problema
usando
lógica de
programación?



Solución 1



```
if((nivelTinaco == HIGH) && (nivelCisterna == HIGH)){  
    enciendeBomba();  
}
```

Solución 2



```
if(nivelTinaco == HIGH){  
    if(nivelCisterna == HIGH){  
        enciendeBomba();  
    }  
}
```

"Estructura anidada", es decir, las condiciones son consecutivas, no simultaneas.

Conclusión



Las soluciones al problema anterior no son las únicas, existen infinidad de maneras de resolver este y muchos otros problemas, depende de nuestro pensamiento y capacidad lógica para poder llegar a una solución. En esencia, la programación consiste en "entrenar" a nuestro microcontrolador para ejecute las acciones que nosotros haríamos para resolver o actuar en determinadas situaciones y circunstancias.

Fuentes



1. <http://www.conocimientosfundamentales.unam.mx/vol1/filosofia/m01/t01/01t01s01.html>
2. https://es.wikipedia.org/wiki/Modus_ponendo_ponens
3. https://es.wikipedia.org/wiki/%C3%81lgebra_de_Boole
4. https://es.wikipedia.org/wiki/Tabla_de_verdad
5. https://es.wikipedia.org/wiki/Puerta_AND
6. https://es.wikipedia.org/wiki/Puerta_OR
7. https://es.wikipedia.org/wiki/Puerta_XOR
8. https://es.wikipedia.org/wiki/Puerta_NOT
9. <https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitwiseand/>
10. <https://www.arduino.cc/reference/en/language/structure/boolean-operators/logicaland/>
11. <https://www.arduino.cc/reference/en/language/structure/control-structure/if/>