

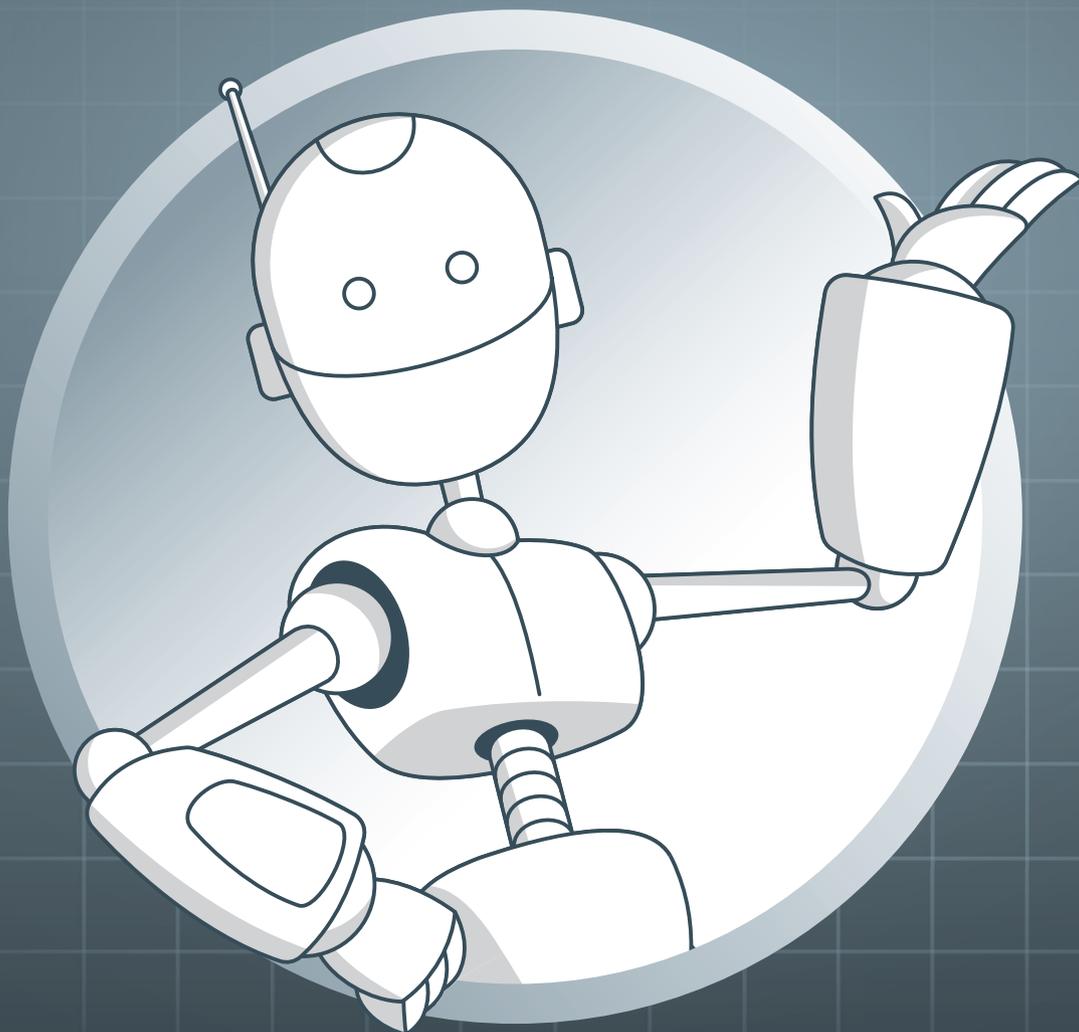
# DIGITAL Teens

2

# ROBOTICS

with

**Mindstorms EV3**



  
binarylogic

*Versión en Español*

## TAREA 1

# Operaciones lógicas

En una sentencia condicional, usamos operaciones relacionales (por ejemplo  $>$  o  $=$ ) para verificar si una condición es VERDADERA o FALSA. Sin embargo, si necesitamos verificar una condición que es más compleja, podemos usar Operaciones lógicas. También se llaman operaciones booleanas porque solo puede haber una de dos respuestas posibles en un caso determinado: VERDADERO o FALSO.

## Bloque Operaciones Lógicas

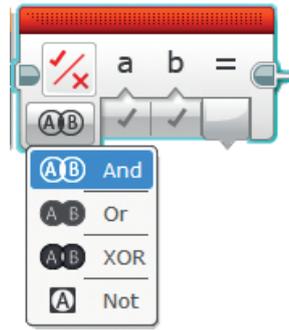
En la programación de Mindstorms EV3, puedes utilizar el bloque de Operaciones lógicas si quieres realizar una operación lógica. La Operación lógica toma entradas que son verdaderas o falsas, y produce una salida verdadera / falsa. Las Operaciones lógicas disponibles son AND, OR, XOR y NOT, y el bloque calculará el resultado de salida en función de las entradas.



Modo	Resultado del bloque de cálculo basado en las entradas
 And	Verdadero si A y B son Verdaderas, si no, Falso
 Or	Verdadero si A ó B (o ambas) son Verdaderas, Falso si A y B son Falsas
 XOR	Verdadero si solo A ó solo B son Verdaderas, Falso si A y B son Verdaderas, Falso si A y B son Falsas
 Not	Verdadero si A es Falsa, Falso si A es Verdadera

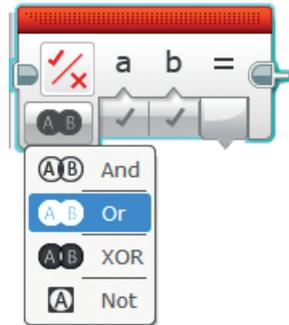
## AND

La Operación lógica **AND** permite al usuario conectar dos condiciones. Si las expresiones condicionales son verdaderas, devuelve verdadero. Si ambas no son verdaderas, devuelve falso.



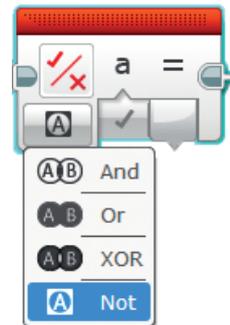
## OR

La Operación **OR** se utiliza para conectar dos o más condiciones. Si al menos una de ellas es verdadera, devuelve verdadero. Si ninguna de ellas es verdadera, devuelve falso.

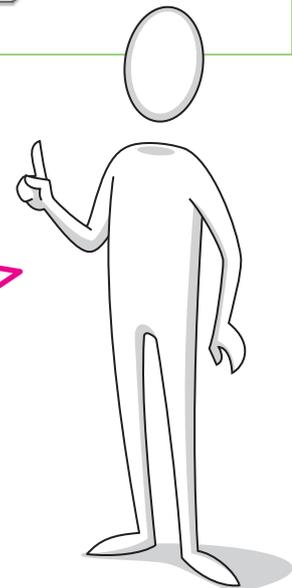


## NOT

La Operación lógica **NOT** permite al usuario expresar las condiciones que se expresan mejor de manera negativa. Si la expresión condicional es falsa, solo en esa ocasión devuelve verdadero. Si la expresión condicional es verdadera, devuelve falso.

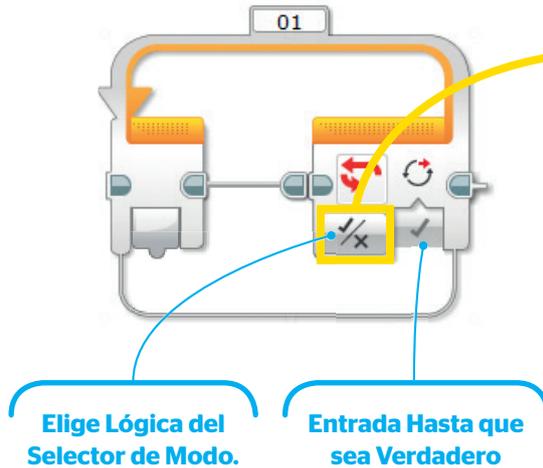


*Los robots pueden tomar decisiones solo cuando está muy claro qué opciones tienen y en qué circunstancias están. Por eso, sus decisiones deben basarse en respuestas de preguntas que tienen dos posibles y simples respuestas: verdadero o falso (significa sí o no). Sentencias como estas se llaman sentencias booleanas.*



## Bloque Bucle

Cuando el bloque Bucle está en modo lógico, el bucle se repetirá hasta que la entrada Hasta que sea verdadero sea Verdadero al final de la secuencia del bucle. Puedes usar el Cable de datos para conectar la entrada Hasta que sea Verdadero a una salida lógica de un bloque de programación que hay dentro del bucle.

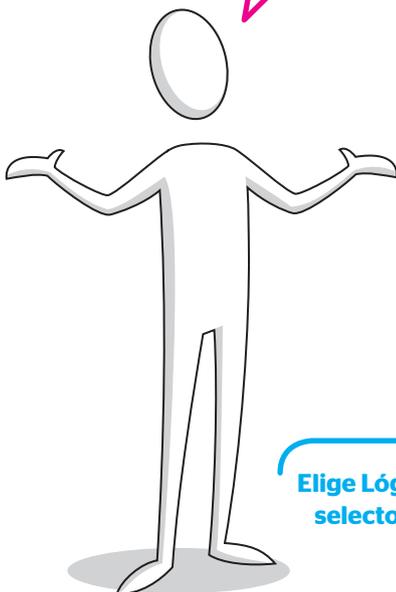


- Botones del Bloque EV3 ▶
- Sensor de color ▶
- Girosensor ▶
- Sensor infrarrojo ▶
- Rotación del motor ▶
- Sensor de temperatura ▶
- Temporizador ▶
- Sensor táctil ▶
- Sensor ultrasónico ▶
- Medidor de energía ▶
- Sensor de Sonido NXT ▶
- Mandar mensajes ▶
- Ilimitado
- # Conteo
- Lógica**
- ⌚ Tiempo

## Bloque Interruptor

Cuando el bloque Interruptor está en modo lógico, el Interruptor elige entre un caso Verdadero y un caso Falso basándose en el valor de la entrada lógica. Puedes usar el Cable de datos para conectar la entrada lógica a una salida lógica de cualquier bloque de programación.

*En la programación de EV3, a menudo usamos una combinación del bloque Comparar y el bloque Bucle o Interruptor en el modo lógico. El valor de salida del bloque Comparar se utiliza como un valor de entrada en el bloque Bucle o Interruptor.*

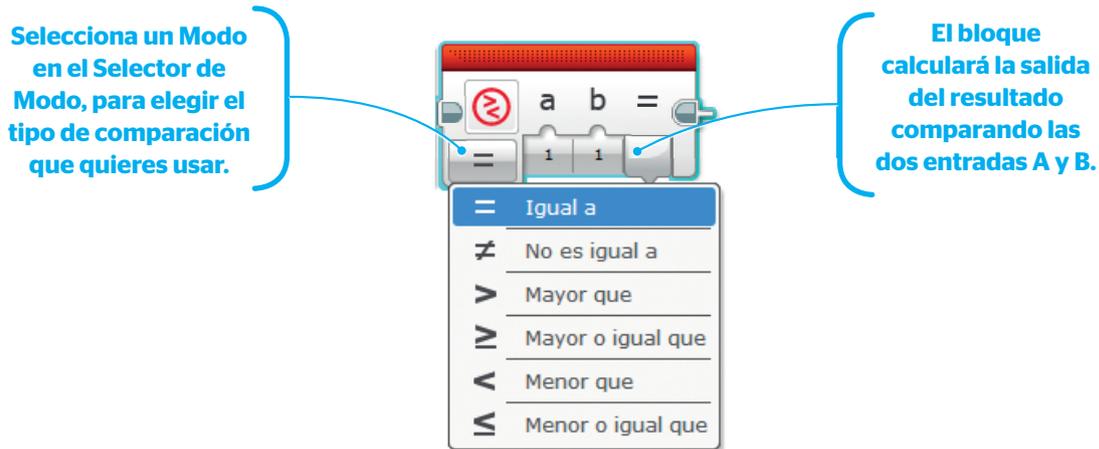


Elige Lógica desde el selector de modo.

- Botones del Bloque EV3 ▶
- Sensor de color ▶
- Girosensor ▶
- Sensor infrarrojo ▶
- Rotación del motor ▶
- Sensor de temperatura ▶
- Temporizador ▶
- Sensor táctil ▶
- Sensor ultrasónico ▶
- Medidor de energía ▶
- Sensor de Sonido NXT ▶
- Mandar mensajes ▶
- T Texto
- Lógica**
- # Numérico

## Bloque Comparar

En la programación de Mindstorms EV3, a veces usamos comparaciones entre números para programar nuestro robot y que realice acciones específicas, ya sea que el resultado de la comparación sea Verdadero o Falso (salida lógica). Por eso, podemos utilizar el **bloque Comparar**. El bloque Comparar compara dos números para averiguar si son iguales o no, qué número es mayor / menor o qué número es mayor / menor o igual que. Puedes elegir uno de estos seis Modos y el resultado de comparación será Verdadero o Falso.

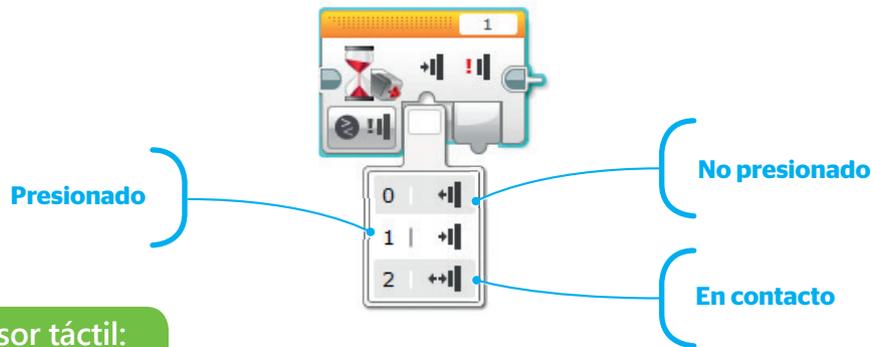
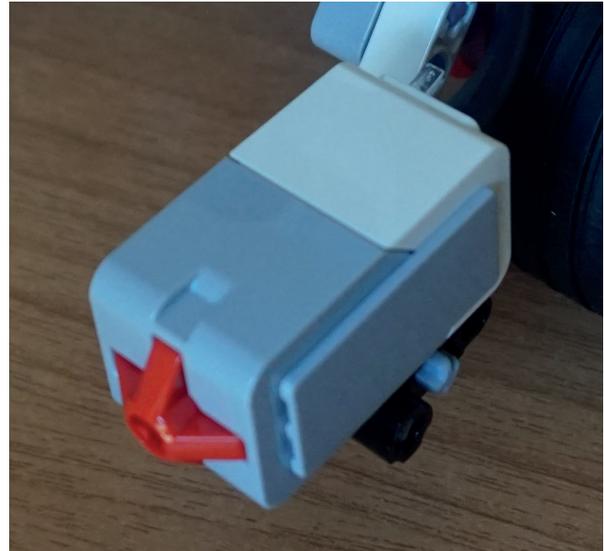


Comparar bloques			
Modo	Entradas	Salidas Lógicas	
	Igual a	A, B	Verdadero si $A = B$ , si no, es Falso
	No es igual a	A, B	Verdadero si $A \neq B$ , si no, es Falso
	Mayor que	A, B	Verdadero si $A > B$ , si no, es Falso
	Mayor o igual que	A, B	Verdadero si $A \geq B$ , si no, es Falso
	Menor que	A, B	Verdadero si $A < B$ , si no, es Falso
	Menor o igual que	A, B	Verdadero si $A \leq B$ , si no, es Falso

## Programar con el Sensor táctil

En la programación de Mindstorms EV3, el Sensor táctil es un sensor analógico que puede detectar si se presiona o se suelta el botón frontal. Además, es capaz de contar cuantas veces se presiona simples y múltiples. Se utiliza para detectar obstáculos o para activar una acción cuando el usuario lo presiona con el dedo.

El Sensor táctil muestra datos lógicos (Verdadero o Falso). La posición del Sensor táctil se llama Estado, y es Verdadero si está presionado y Falso si no lo está. También tiene tres estados más: Presionado, No presionado y En contacto.

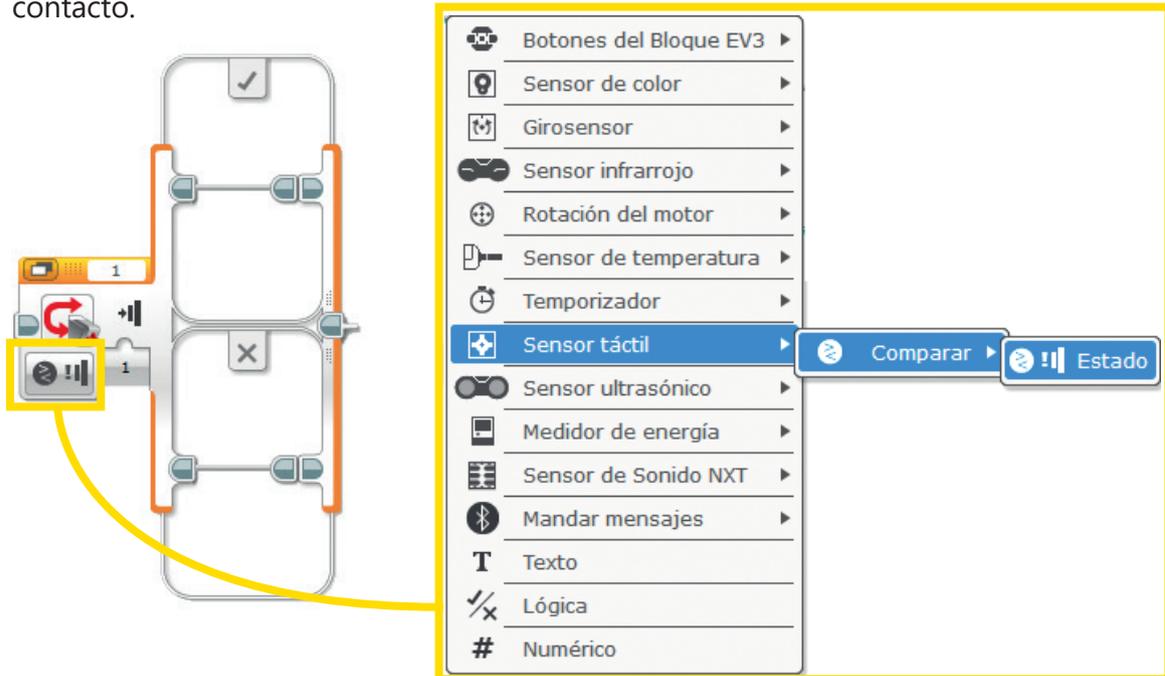


Datos del Sensor táctil:		
Datos	Tipo	Notas
<b>Estado</b>	Lógico	Verdadero si el botón está presionado, Falso si no lo está.
<b>Presionado</b>	Lógico	Verdadero si está presionado, Falso si no lo está. (igual que el Estado).
<b>No presionado</b>	Lógico	Falso si está presionado, Verdadero si no lo está. (el opuesto al Estado).
<b>En contacto</b>	Lógico	Verdadero si el botón se presionó y se soltó anteriormente. El siguiente acontecimiento En contacto requerirá que se vuelva a presionar y soltar.

Dependiendo de la situación, puedes programar el robot EV3 para que responda a cada estado del Sensor táctil, utilizando el bloque Esperar, el bloque Bucle o el bloque Interruptor.

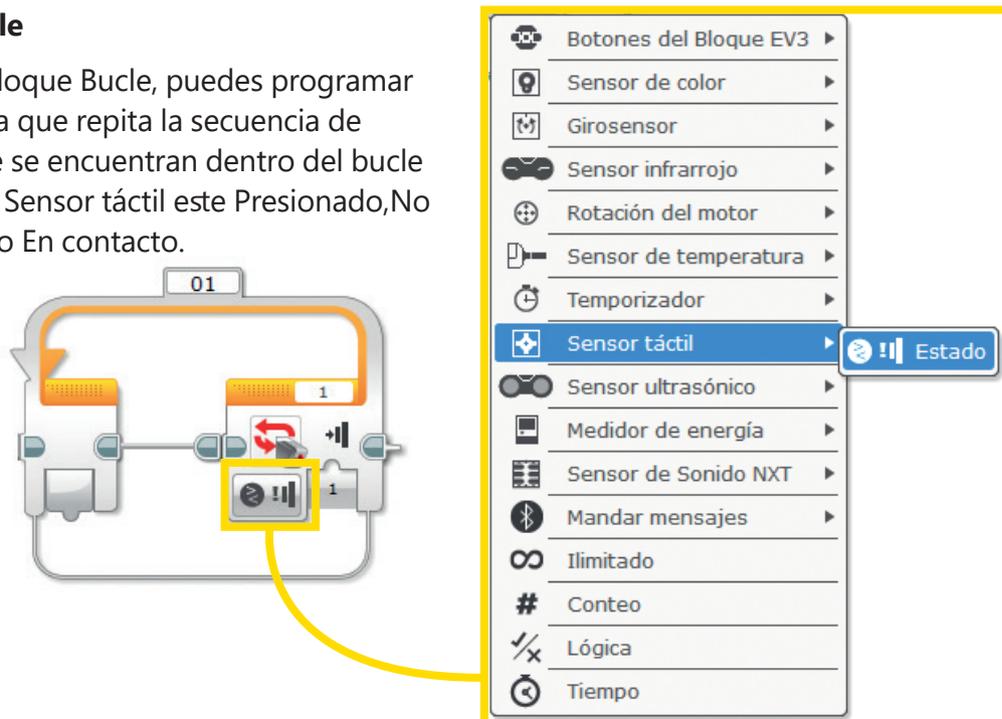
### El bloque Interruptor

Usando el bloque Interruptor, puedes elegir entre dos secuencias (Caso verdadero o Caso falso) dependiendo si el Sensor táctil está Presionado, No presionado, o En contacto.



### Bloque Bucle

Usando el bloque Bucle, puedes programar el robot para que repita la secuencia de bloques que se encuentran dentro del bucle hasta que el Sensor táctil este Presionado, No presionado o En contacto.



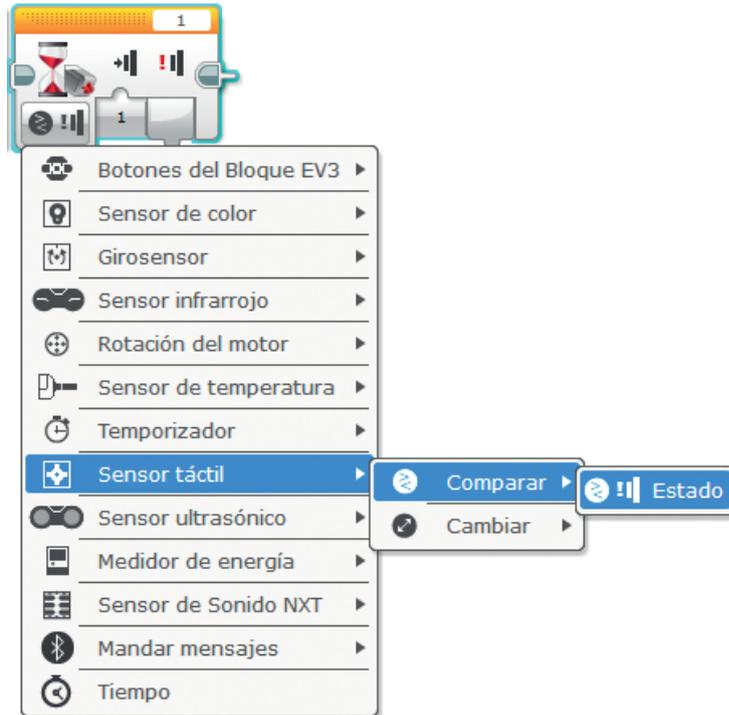
## Bloque Esperar

Usando el bloque Esperar, puedes programar que el robot espere que el Sensor táctil este en estado Presionado, No presionado o En contacto o simplemente esperar que cambie su estado.

### Modos del bloque Esperar :

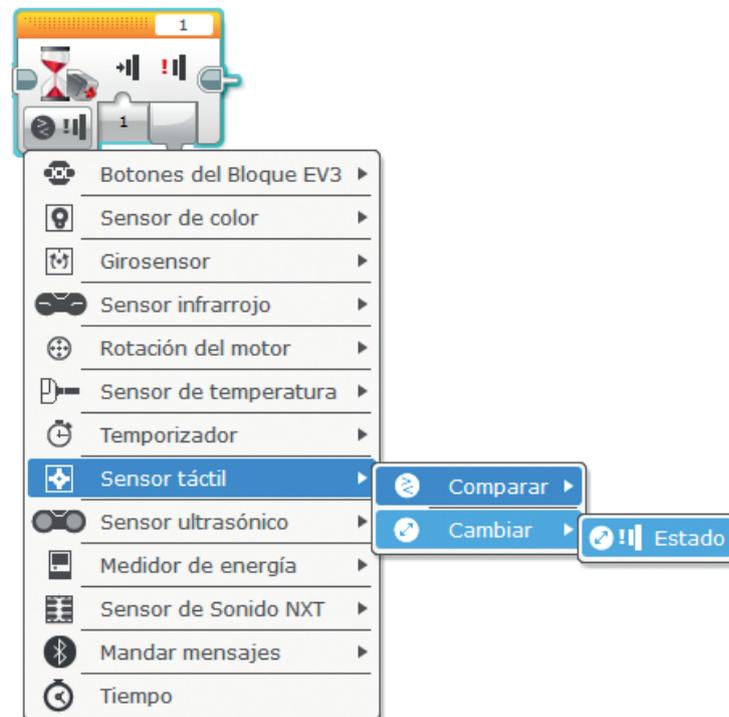
#### Comparar-Modo Estado

Esperar que el Sensor táctil sea Presionado, No presionado o En contacto.



#### Cambiar-Modo Estado

Esperar que el Estado del Sensor táctil cambie.



# ¡Manos a la obra!

Con los datos  $k=5$  y  $m=2$   
 llena las siguientes tablas.

CONDICIÓN	Devuelve
$k < 5 \text{ OR } m \geq 2$	-----
$k \geq 5 \text{ AND } m > 2$	-----
$k+2 > 5 \text{ OR } m+1 < 4$	-----

CONDICIÓN	Devuelve
$\text{NOT } k < 6$	-----
$\text{NOT } (k=5 \text{ or } m < 1)$	-----
$\text{NOT } k > 6$	-----

Haz las combinaciones  
 apropiadas de  $k$  y  $m$   
 para que tengan sentido las  
 siguientes combinaciones.

CONDICIÓN	Devuelve
$k \dots \text{ AND } m \dots$	Verdadero
$k \dots \text{ OR } m \dots$	Falso
$\text{NOT } (k = \dots \text{ OR } m = \dots)$	Falso