

# edubrick

[www.edubrick.cl](http://www.edubrick.cl)

## Guía de Robótica LEGO® MINDSTORMS® NXT



**edubrick** una empresa dedicada a la educación entretenida.  
Contáctenos al teléfono 263 5080 o al correo electrónico  
[contacto@edubrick.cl](mailto:contacto@edubrick.cl)  
Representante exclusivo de LEGO Education para Chile



# Guía de Robótica LEGO® MINDSTORMS® NXT

LEGO® MINDSTORMS® Son marcas registradas de la Compañía LEGO y esta guía de robótica educativa no cuenta ni con el auspicio ni la autorización de la compañía.

Guía de Robótica LEGO® MINDSTORMS® los derechos de autor son propiedad de Gustavo Brinck, la reproducción de este material por cualquier medio requiere la autorización por escrito del autor.

## Contenido

Introducción.....	1
Metodología en la sala de clases:.....	1
El cerebro de nuestro robot.....	2
Esquema .....	2
Características técnicas: .....	3
Operaciones básicas.....	3
Visualización .....	3
Menú principal.....	4
Puertas predeterminadas .....	4
Sensores.....	5
Sensores Opcionales .....	6
Sensores del Modelo RCX.....	7
Motores y Lámparas .....	7
Programación .....	9
Nuestro primer programa.....	9
Desafío 1 .....	11
Desafío 2:.....	12
Desafío 3.....	14
Ayuda.....	15
Sensor de Tacto .....	16
Desafío 4 .....	16
Loops.....	17
Sensor de Luz.....	18
Desafío 5.....	19
Sensor de Sonido .....	21
Sensor de Ultrasonido .....	22
Evaluando Condiciones.....	23
Visualización y Sonido.....	24
Desafío 6.....	25
Administración de Memoria .....	26
Botones .....	27
Case.....	28
Data Hub .....	30
Variables .....	31
Desafío 7 .....	32
Cronómetros.....	33
Desafío 8.....	33
Desafío 9.....	34
Multiprogramación.....	35
Comunicaciones.....	36

## Introducción

La realización de proyectos de robótica otorga al estudiante la capacidad de aprender importantes conocimientos a través de la construcción, programación y prueba de los robots. Durante este proceso el estudiante se encontrará con conceptos claves que se relacionan con las ciencias de la computación, matemáticas aplicadas, ciencias en general, trabajo en equipo y comunicación. Al mismo tiempo el estudiante aprende el proceso de explorar, planificar y resolver problemas. También se familiariza con el principio de dividir un proyecto en pequeñas partes y así lograr una solución metódica y más abordable.

### Metodología en la sala de clases:

Se presenta el problema que los estudiantes deben resolver, se discute las reglas, las probables soluciones, tiempo máximo del proyecto, limitación de materiales, cómo se debe presentar la solución del problema.

Durante la ejecución del proyecto el instructor se convierte en un facilitador, es decir a solicitud del estudiante actúa como consejero, para ayudar que el estudiante descubra e investigue por sí mismo. Permitiendo que el estudiante actúe con la máxima libertad posible.

Cada grupo presenta su proyecto de acuerdo a las bases de cada proyecto, se revisa la documentación, se realizan las pruebas y todos opinan de proceso de solución y solución misma.

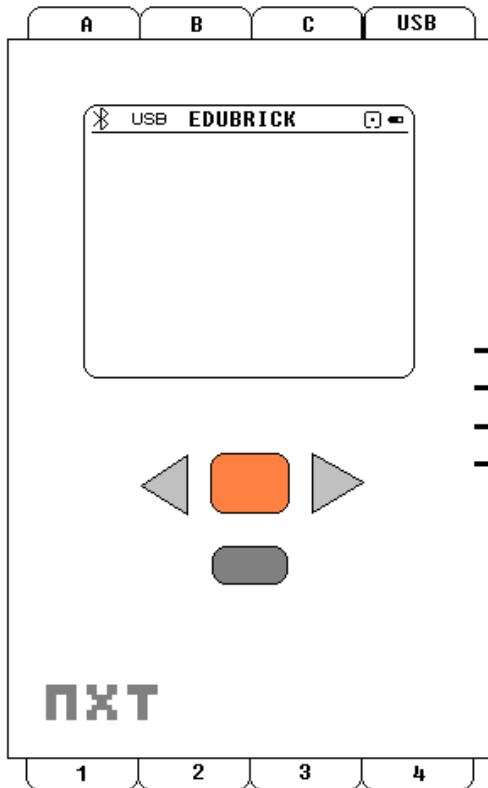
En ocasiones se pedirá a los estudiantes que realicen una mejora de la solución propuesta, con lo aprendido y discutido con el total de la clase, y así permitir una mejora continua.

## El cerebro de nuestro robot

El ladrillo inteligente LEGO NXT, es el cerebro de nuestro robot, básicamente es un microprocesador, con puertas de entradas, de salida y memoria para almacenar nuestros programas, se comunica con el computador a través de una puerta USB o Bluetooth.

### Esquema

- Tres puertas de salida, señaladas como A; B y C
- Una puerta de comunicación USB



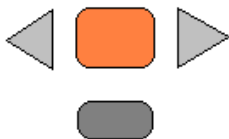
- Pantalla de cristal líquido.
- Mini parlante
- Botones de encendido apagado y navegación.

- Cuatro puertas de entrada, señaladas como 1; 2; 3 y 4.

## Características técnicas:

- Microcontrolador de 32 bits ARM7
  - Memoria FLASH de 256 Kbytes
  - Memoria RAM de 64 Kbytes
- Microcontrolador de 8 bit AVR
  - Memoria FLASH de 4Kbytes
  - Memoria RAM de 512 Bytes
- Comunicación Inalámbrica Bluetooth (Bluetooth Class II V2.0)
- Puerta de alta velocidad USB (12 Mbit/s)
- Cuatro puertas de entrada de seis contactos, plataforma digital
- Tres puertas de salida de seis contactos, plataforma digital
- Pantalla gráfica de cristal líquido de 64 x 100 puntos
- Parlante, calidad de sonido 8KHz
- Fuente de poder, 6 baterías AA

## Operaciones básicas



### Botón Naranja

- Encendido y Enter

### Botones Gris Claro

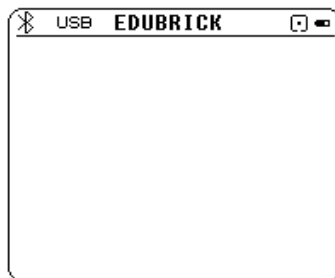
- Para moverse en el menú izquierda y derecha

### Botón Gris Oscuro

- Para retroceder en la selección del menú

Para apagar el NXT, presione el botón gris oscuro hasta que aparezca la pantalla “Turn off?”, luego presione el botón naranja.

## Visualización



En la parte superior de la pantalla del NXT, podemos ver el tipo de conexión que estamos usando (Bluetooth y/o USB), el nombre de nuestro robot, luego el símbolo que indica que está en operación y finalmente el estado de la batería.

## Indicadores

- ⌘ Bluetooth encendido
- ⌘< Bluetooth visible para otros dispositivos
- ⌘◇ Bluetooth conectado a otro dispositivo

- USB USB conectado y trabajando bien
  - ⌘ USB conectado, con problemas
  - ☑ NXT, operando correctamente
  - ⏻ Batería baja
  - Bateria bien
- EDUBRICK** Nombre del NXT, veremos como cambiar el nombre más adelante

### Menú principal

My Files	Software Files	Programas que usted a bajado del computador
	NXT Files	Programas que usted ha hecho directamente en el NXT
	Sound Files	Archivos de sonido
NXT Programs	En este sub-menú, usted puede programar directamente el NXT	
Try Me	Programas precargados para probar motores y sensores, debe usar las puertas predeterminadas	
View	Le permite visualizar el estado de los motores y sensores en tiempo real	
Setting	Puede ajustar el volumen del sonido del NXT, ajustar el modo "Sleep" para apagar automáticamente el NXT y también borrar programas	
Bluetooth	Le permite ajustar los distintos parámetros de la comunicación inalámbrica	

### Puertas predeterminadas

Puertas de Entrada		Puertas de Salida	
1	Sensor de Tacto	A	Motor para función extra.
2	Sensor de Sonido	B	Motor izquierdo
3	Sensor de Luz	C	Motor derecho
4	Sensor de Ultrasonido		

## Sensores



### Sensor de Tacto

Le proporciona al robot el sentido del tacto, tanto cuando se presiona como cuando se suelta. También podemos utilizarlo para ciclos completos de presionar y soltar.

### Sensor de Sonido

Le permite al robot escuchar.



El sensor de sonido puede detectar tanto decibeles (dB), como decibeles ajustados (dBA). Decibel es una medida de la presión de sonido.

dBA: La sensibilidad del sensor es ajustada al oído humano.

db: La sensibilidad del sensor, no está ajustada y puede “escuchar” sonidos por debajo o por encima de la capacidad del oído humano.

El sensor puede medir sonido hasta de 90 dB. La lectura del sonido se visualiza en porcentaje de acuerdo a la siguiente tabla:

Porcentaje	Descripción
4%-5%	Habitación en silencio
5%-10%	Personas hablando lejos
10%-30%	Conversación normal
30%-100%	Personas gritando y música a alto volumen

### Sensor de Luz

Le proporciona al robot el sentido de la visión.

Le permite al robot distinguir entre luz y oscuridad.

El sensor es monocromático, es decir puede distinguir entre el blanco y el negro pasando por una gama de grises, la lectura la entrega en porcentaje.







### Sensor Ultrasonico

También le proporciona al robot un sentido de visión.

El sensor emite un sonido y mide el tiempo que la señal tarda en regresar, para luego calcular la distancia, a la cual se encuentra el objeto u obstaculo.

Es el mismo principio utilizado por los murciélagos y el sonar de las naves, tiene una rango de 0 a 255 cm con una precisión de +/- 3 cm.

## Sensores Opcionales

### Sensor Brújula



Le permite al robot saber hacia donde esta apuntando.

Este ser sensor es capas de medir el campo magnético terrestre y calcular donde está el norte magnético.

Se visualiza la lectura en grados 0° a 359°

### Sensor de Color

Le permite al robot “ver” colores.

No sólo blanco y negro, este es un sensor capas de distinguir colores reales.



### Sensor de Inclinación y Aceleración



Le permite al robot saber que inclinación tiene.

Dispone de un acelerómetro en los tres ejes (X; Y y Z) con un rango entre -2g y +2g, lo cual le permite calcular la inclinación.

## Sensores del Modelo RCX

Los sensores del robot RCX, también pueden ser utilizados en el nuevo robot NXT, utilizando el cable adaptador.

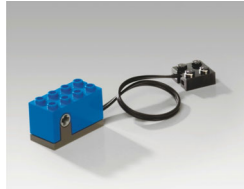
Sensor de Tacto



Sensor de Luz



Sensor de Rotación



Sensor de Temperatura



## Motores y Lámparas

Los servomotores, le darán movimiento a nuestro robot.



Los motores incluyen un sensor de rotación, con una precisión de +/- 1 grado, también podemos montar el motor en un eje y utilizarlo como sensor de rotación.

Para el movimiento de un modelo motorizado el firmware (el sistema operativo interno del NXT), dispone de un sofisticado algoritmo PID, el cual nos permite que nuestro modelo se desplace con precisión.

También podemos utilizar los motores, estándar LEGO, mediante el cable adaptador.



Las lámparas podemos utilizarlas como indicadores, también mediante el uso del cable adaptador

**Importante:**

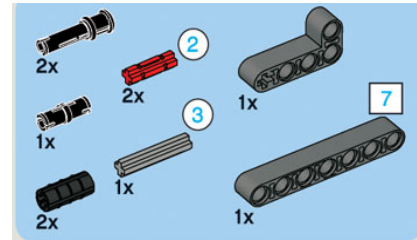


En el próximo capítulo comenzaremos con la programación, para ello necesitamos que arme el TriBot, comenzando en la página 8 hasta la 22, del manual de educación del conjunto 9797.

Note que en cada instrucción, se señala en un recuadro las piezas que ocupará en ese paso, es importante que seleccione

las piezas, antes de construir, así sabrá que no le sobra ni le falta ninguna.

Por ejemplo en el diagrama adjunto se indica que necesitamos 2 ejes color rojo, el número encerrado en el círculo indica que es de largo 2. El eje gris se indica que es de largo 3, y la viga con el recuadro 7 indica que tiene 7 hoyos o 7M.



## Programación

Para programar el NXT, utilizaremos el programa NXT-G, lenguaje gráfico basado en LabView de National Instrument, también es posible programarlo en RoboLab 2.9 y directamente en LabView.

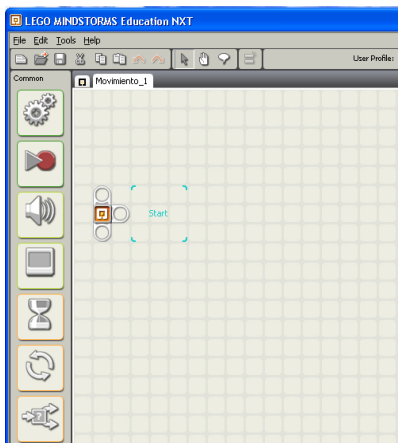
También hay desarrollos independientes que permiten programar el NXT en lenguaje C (NXC y RobotC), JAVA y Lua.

Debemos distinguir entre los programas que permiten que el robot opere en forma autónoma y los programas que pueden controlar el robot en forma remota, para esta última forma hay también gran número de programas como Microsoft Robotics Studio, C# ,Perl, JAVA, Ruby, etc. También muchos de los desarrollos independientes, citados anteriormente pueden operar al robot en forma remota.

### Nuestro primer programa

Debemos desplazar nuestro robot, cuatro segundos hacia adelante.

Inicie el programa NXT-G y continúe hasta llegar a la pantalla más abajo indicada.



En esta pantalla elegiremos el bloque que tiene los engranajes (Move Block) y lo desplazaremos al área de trabajo en la posición de inicio (Start).









Nos quedará conectado de la forma que se muestra en la figura.

En la parte inferior de la pantalla se encuentra el panel de configuración, el cual depende del bloque seleccionado de nuestro programa, en este caso sólo hay un bloque por lo que el panel de configuración muestra los parámetros del bloque "Move".

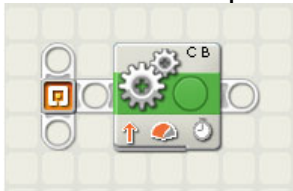


Panel de configuración del bloque "Move"

Detalle de los parámetros:

	Puerta de salida, en este caso tenemos dos motores en B y en C.
	Sentido de Avance, hacia adelante.
	Volante, derecho
	Potencia, 75%
	Duración, debemos elegir 4 segundos
	Siguiente acción, freno

Finalmente nuestro programa debe quedar como muestra la figura, note que el bloque ahora muestra un reloj en lugar de un indicador de rotación, ya que hicimos el cambio en el parámetro de duración a 4 segundos.



Ya tenemos nuestro programa y debemos bajarlo al NXT.

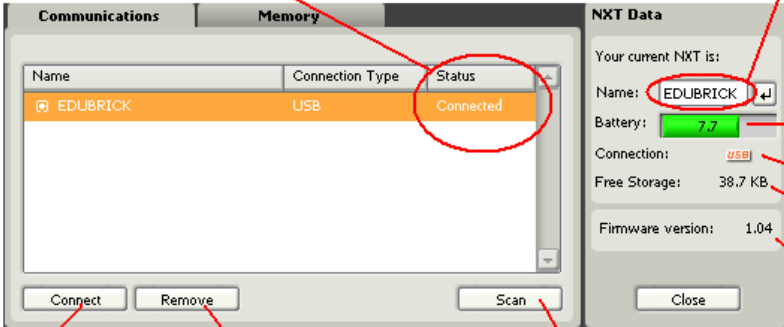
En el controlador, vea figura de la izquierda, con el robot NXT encendido y conectado al computador, presionamos el botón “NXT

Window” 



Para programar un NXT su estado debe ser conectado

Aquí puede poner el nombre de su NXT



Conecta un NXT disponible

Retira un NXT de la lista

Para buscar los NXT Disponibles

Nivel de batería

Tipo de conexión

Memoria disponible

Versión del programa de control

Ahora en el controlador presionamos el botón de “Download”, sacamos el cable USB, ponemos el robot en el suelo y presionamos el botón naranja en el NXT.



### Desafío 1

Intente hacer que su robot avance durante 4 segundos y luego retroceda 4 segundos.

La solución está en la página siguiente, trate de hacerlo sin ver la solución.

Notas:


Solución:

Para realizar este desafío debemos, agregar otro bloque de movimiento a nuestro programa y en el panel de configuración, cambiar la dirección hacia atrás, también es necesario que la duración sea 4 segundos.



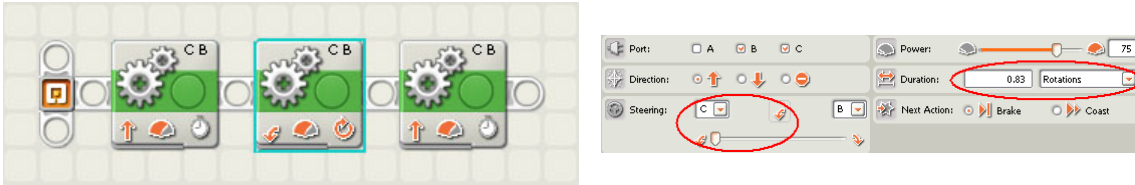
## Desafío 2:

Bien, ahora vamos a avanzar con nuestro robot, girar y regresar.

Notas:


Solución:

En este caso tenemos tres movimientos: avanzar por 4 segundos, girar un tiempo o una cantidad de rotaciones y regresar, que en realidad es avanzar nuevamente ya que el robot ha girado.



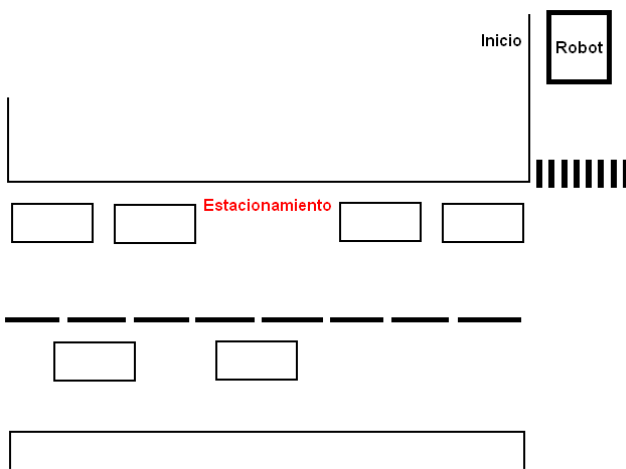
En el bloque central, para el giro, hemos movido el “volante” para realizarlo, en este caso (dadas las condiciones de la superficie y estado de la batería) el parámetro de la duración se estableció en 0.83 rotaciones, puede que usted requiera modificarlo.

En lugar de especificar la duración en rotaciones inténtelo con tiempo.

Notas:




### Desafío 3

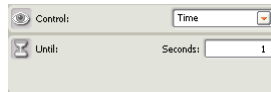


El robot debe estacionarse en el lugar indicado, trate no pasar el eje de la calzada, ya que hay vehículos en sentido contrario circulando.

**Bonus:**

Hay un paso de cebra, intente detenerse por dos segundos, antes de pasar.

**Ayuda:** Para esperar por un tiempo determinado debemos utilizar el bloque de espera por tiempo:



En nuestro caso el parámetro de control es el tiempo y las unidades debemos cambiarlas a 2 segundos.

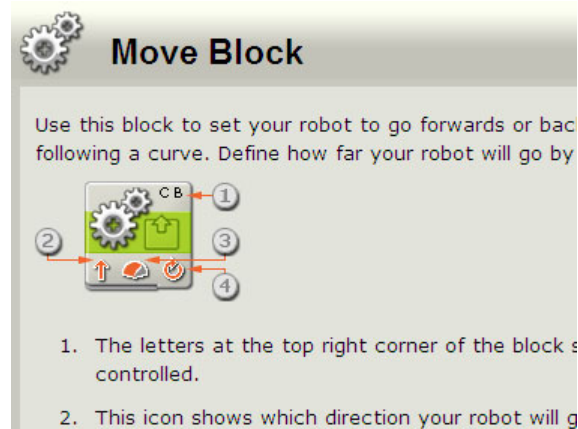
**Notas:**

## Ayuda

La opción de ayuda, esta disponible en el programa, cada vez que seleccionamos un bloque, en la parte inferior derecha aparece una breve descripción de su función.



Si necesitamos de más ayuda podemos presionar “More Help” y obtendremos una completa descripción del bloque.



Hasta ahora hemos controlado el robot con parámetros de tiempo y distancia a través de la rotación de los motores. Para ver de que otras opciones disponemos, vamos a instalarle un sensor de tacto.

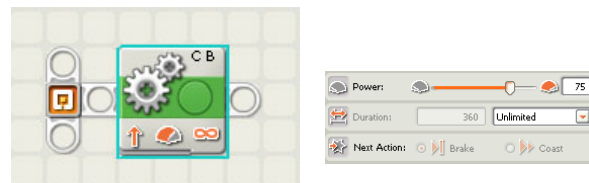
Vaya a la página 40 paso 26 del manual y construya el soporte del sensor de tacto, hasta la página 43,

**Importante:** Instálelo en la parte frontal del robot no en la parte posterior como indica el manual.

El cable del sensor de tacto, instálelo en la puerta de entrada 1.

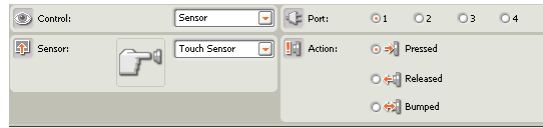
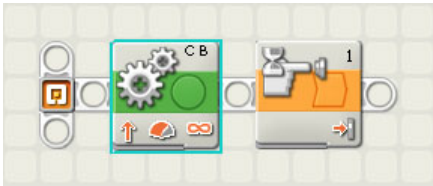
### Programa: Avance hasta chocar

A diferencia del programa anterior, ahora nuestro bloque de control de movimiento, debe hacer que el robot se desplace sin límites, por lo que en el parámetro duración elegiremos “Unlimited”.



Bien, ahora tenemos que detener el robot cuando choque, para lo cual utilizaremos el bloque de espera, pero en lugar de especificar tiempo como en el programa anterior, elegiremos sensor de tacto.

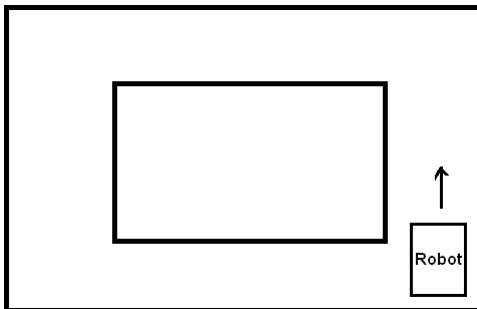
## Sensor de Tacto



Note que hemos seleccionado en el panel de configuración del bloque de espera, los siguientes parámetros:

Control	Sensor
Sensor	Touch Sensor (sensor de tacto)
Port	1, puerta a la cual conectamos el sensor
Action	Pressed, actuará al presionar el sensor

### Desafío 4

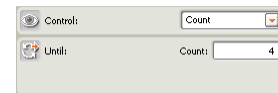
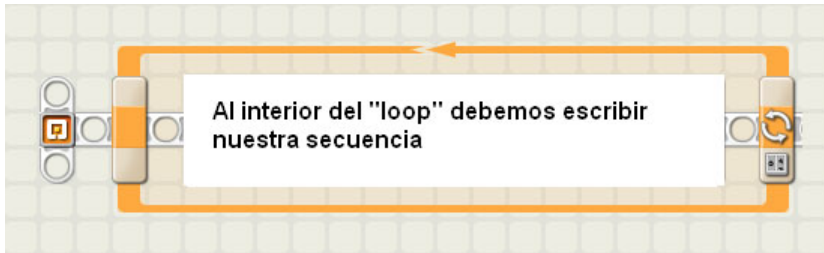


El robot debe recorrer el circuito de la figura, los lados del rectángulo son desconocidos, el rectángulo exterior es una pared sólida.

Notas:


## Loops

En la resolución del problema, nos daremos cuenta que la misma secuencia se repite cuatro veces, para lado del rectángulo, por lo que vamos a introducir un nuevo bloque llamado "loop", que nos permitirá en este caso ejecutar en este caso cuatro veces nuestra secuencia.



En el panel de configuración del bloque "loop", debemos especificar los siguientes parámetros: Control→Count (Cuenta) y Until→4 (4 veces).

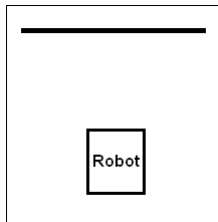
### Notas:


Ahora que contamos con más herramientas, podemos realizar programas más entretenidos.

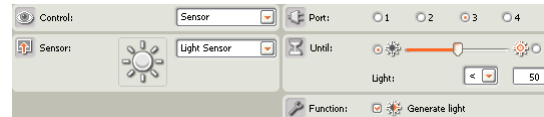
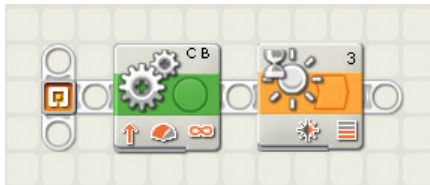
## Sensor de Luz

Vamos a instalar el sensor de luz en la parte frontal de nuestro robot, primero debemos sacar el sensor de tacto.

Construya el soporte para el sensor de luz comenzado en la página 32 paso 22 de nuestro manual, hasta el paso 23.



Al igual que en el caso del sensor de tacto, queremos que el robot avance hasta encontrar la línea negra, en realidad es exactamente el mismo programa, sólo debemos cambiar los parámetros del panel de configuración del bloque de espera.



Control	Sensor
Sensor	Light Sensor (sensor de luz)
Port	3, puerta a la cual conectamos el sensor
Until	< 50, Luz menor que 50
Funtion	Generate Light (luz reflejada)

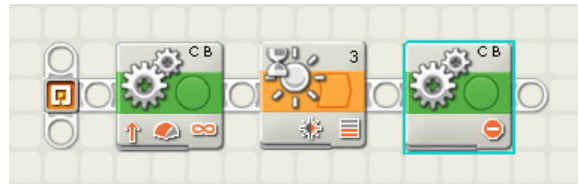
Para calibrar el sensor debemos medir la luz, en la superficie clara y también en la línea negra, para ello utilizaremos la función "View" directamente en el NXT

En el menú principal, **View** → **Reflectec light** → **Port 3** si la medición sobre el color claro es 60 y sobre la línea negra es 40, entonces elegimos 50, la idea es usar el punto medio.

En realidad el robot se detiene porque el programa ha finalizado, pero si nuestro programa tuviera más bloques, el robot seguiría avanzando (esto aplica también al ejemplo con el sensor de tacto, por lo tanto debemos agregarle un bloque de parada a nuestro robot.

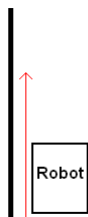


Nuestro último bloque de movimiento debe estar configurado para detenerse.

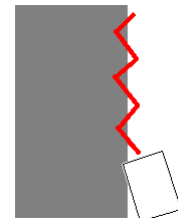


### Desafío 5

Veamos una primera versión del conocido “Siga la línea”



Para seguir la línea negra, ocuparemos un algoritmo que nos permitirá desplazarnos por el borde de la línea, la idea es que cuando el robot esté sobre la superficie clara, gire a la izquierda y cuando esté sobre la línea negra gire a la derecha. Mediante este movimiento de zig-zag, lograremos nuestro objetivo.



Escribamos nuestro programa en un pseudo lenguaje, es decir con palabras comunes y luego lo trasladaremos al lenguaje NXT-G

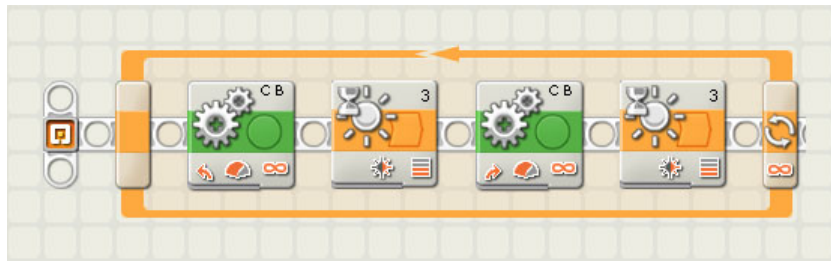
Pseudo lenguaje	Diagrama de flujo
<ul style="list-style-type: none"> <li>• Gire a la izquierda</li> <li>• Espere por color negro</li> <li>• Gire a la derecha</li> <li>• Espere por color blanco</li> <li>• Vuelva a la primera instrucción</li> </ul>	<pre> graph TD     A([Comienzo]) --&gt; B[Gire a la izquierda]     B --&gt; C[Espere por color negro]     C --&gt; D[Gire a la derecha]     D --&gt; E[Espere por color blanco]     E --&gt; B     E --&gt; F([Fin])         </pre>

Ciertamente nuestro programa no termina nunca, más adelante solucionaremos eso problema.

Notas:

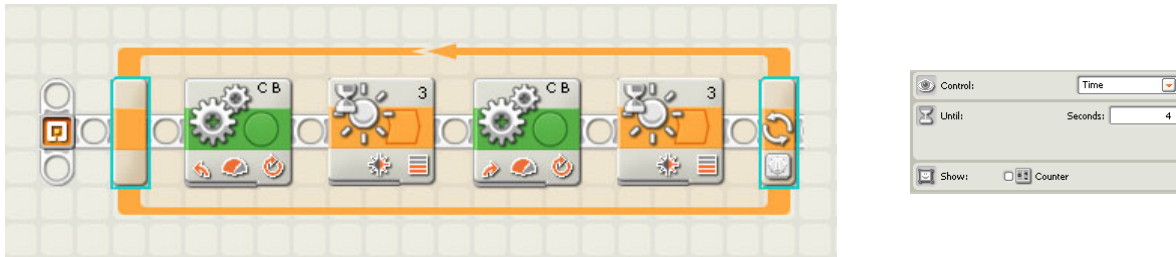

Ya sabemos como programar los giros y la espera por color negro, para la espera por color blanco sólo tenemos que cambiar el signo menor “<” por mayor “>”, en el panel de configuración del bloque de espera.

Y para que nuestra rutina se repita vamos a usar un “loop” infinito, lo cual definimos en el panel de configuración del bloque “loop”.



Algunas ideas para salir del “loop” infinito

Podemos cambiar, en el panel de configuración del bloque de “loop”, el parámetro de control, por tiempo y determinar cuantos segundos necesitamos para terminar el circuito. (por ejemplo 4 segundos)

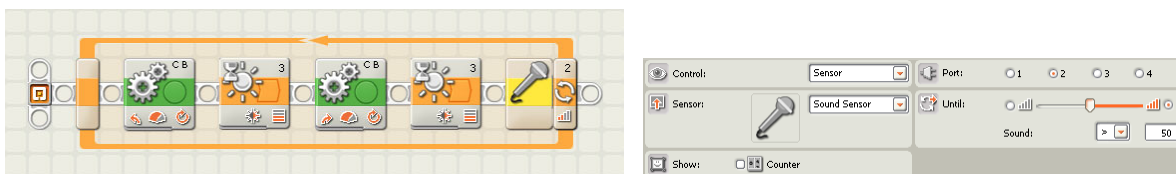


Notas:


## Sensor de Sonido

Otro método más interactivo, es usar el sensor de sonido, para lo cual construiremos su soporte, a partir de la página 24 paso 18.

Entonces en el panel de configuración del bloque “loop”, el parámetro de control será el sensor de sonido, debe ajustar la sensibilidad para que opere cuando de una orden en voz alta.



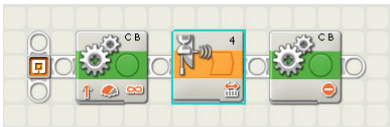
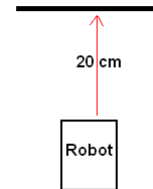


## Sensor de Ultrasonido

Utilizaremos el sensor de ultrasonido para medir la distancia a la cual nos encontramos de un objeto, el objeto debe ser capaz de reflejar el sonido, superficies rugosas o telas no reflejan muy bien el sonido.

De la misma forma que en otros ejemplos, debemos realizar un programa, para que nuestro robot avance hasta encontrarse a 20 cm de la pared.

Entonces comenzamos con el bloque de movimiento y esperamos a que la distancia sea menor que 20 cm, luego nos detenemos.

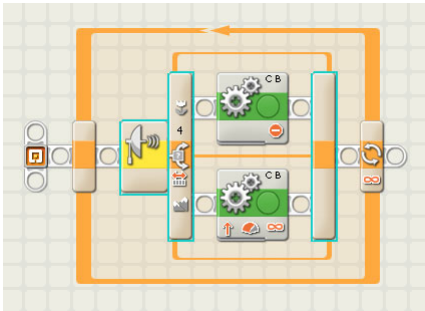


### Notas:


## Evaluando Condiciones

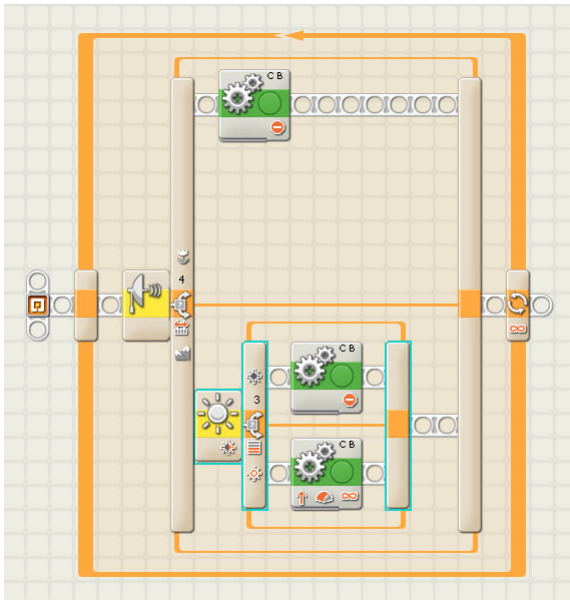
El problema del bloque “espere por”, es que si la condición de espera no se da nunca, el programa no pasa al siguiente bloque, para evitar esta situación utilizaremos la instrucción “IF”, que en el caso del programa NXT-G se llama Switch.

Vamos a realizar el mismo programa anterior de la siguiente forma, preguntamos por la condición del sensor de ultrasonido, si es menor que 20 cm, nos detenemos de lo contrario seguimos avanzando.



Note que si tomamos el robot y lo alejamos, seguirá avanzando hasta que la distancia sea menos a 20 cm.

Para ver más claramente la diferencia con el bloque “espere por”, demos la siguiente condición, el robot debe avanzar hasta 20 cm de la pared o hasta encontrar la línea negra.



Lo que hemos hecho es primero evaluar la condición de distancia y luego la condición de luz, cualquiera de las dos que sea verdadera detendrá el robot.

Es importante notar que podemos anidar bloques “Switch”, decir poner uno dentro de otro, también podemos anidar “loops”.

Este programa no es posible realizarlo con los bloques “espere por” ya que quedamos a la espera de una condición eternamente.

## Visualización y Sonido

El bloque “Display”, nos permite visualizar en pantalla texto, imágenes y también dibujar nuestros propios diseños.

Veamos el bloque “Display”:



En el panel de configuración del bloque, tenemos varios parámetros que permitirán la visualización en la pantalla:



Action: Texto, Imagen o Dibujo

Display: Borrar

File: Lo que queremos mostrar en la pantalla

Position: en que posición

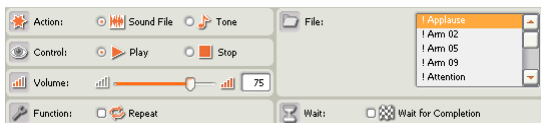
Por ejemplo para ver “ojos mirando a la izquierda”, debemos realizar el siguiente programa, hemos añadido un bloque de espere por un segundo, para poder ver la imagen. (no olvide poner los parámetros en el panel de configuración del bloque display.



El bloque de Sonido, nos permite emitir voces, efectos y también tonos.



En el panel de configuración del bloque, tenemos varios parámetros determinarán que sonido se emitirá.



Action: Podemos elegir un sonido pre-grabado o un tono musical

Control: Emitimos un sonido o lo terminamos

Volumen: Nivel del volumen

Función: Podemos repetir un sonido

File: Elección del sonido o tono si el parámetro “Action” está en “tone”.

Wait: Espera que se complete el sonido, o continúa con el siguiente bloque.



**Desafío 6**

Haga un programa que ejecute una escala musical y al final un aplauso de cierre.

Notas:


## Administración de Memoria

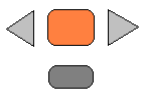
Los sonidos y los gráficos ocupan bastante memoria, por lo que no debemos abusar de ellos, los archivos quedan grabados en la memoria del NXT igual que los programas, para hacer una limpieza debemos presionar en el Controlador el botón NXT windows y luego seleccionar Memory.



En este panel podemos hacer mantención a la memoria del NXT






Notas:

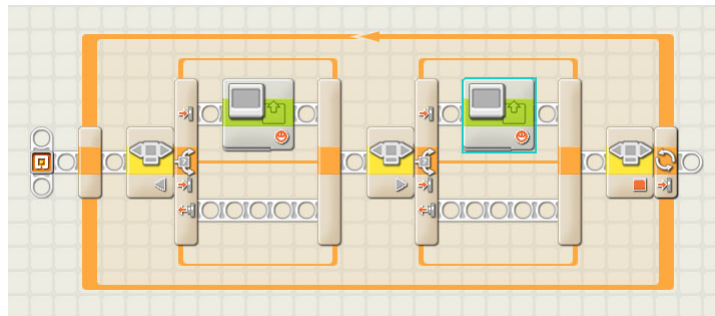

## Botones



Los botones del NXT también pueden ser utilizados como controles de entrada para nuestro programa, con el ejemplo siguiente vamos a ver en plenitud su utilidad.

El programa debe mostrar en la pantalla unos ojos mirando a la izquierda o a la derecha cuando presionemos los botones grises claros (Triangulares) cuando presionemos el botón anaranjado, el programa finaliza.

Símbolo	Descripción	Inglés	Acción
	Botón de Entrada	Enter Button	Salir
	Botón Izquierdo	Left Button	
	Botón Derecho	Righ Button	



Intente definir el panel de configuración de cada bloque.

### Notas:


## Case

Hasta ahora, los Switchs nos han permitido elegir entre dos alternativas, veremos como configurarlos para seleccionar mas posibilidades

Veamos algo sencillo, elegiremos un número al azar entre 1, 2 y 3 y de acuerdo a estas alternativas lo desplegaremos en la pantalla.

Número	Pantalla
1	Número 1
2	Número 2
3	Número 3



Para elegir un número al azar utilizaremos el bloque "Random", y definiremos el rango entre 1 y 3



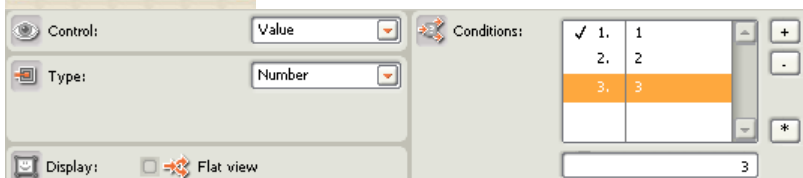
Para configurar el Switch, debemos elegir los siguientes parámetros:

Control → Value(Valor)

Type → Number (Númerico)

Display → Desactivar Flat View(representación plana)

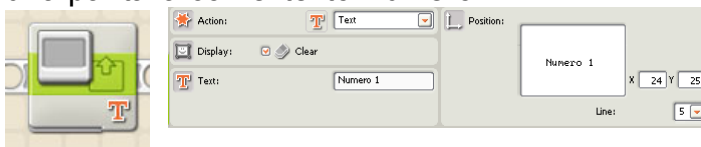
Utilizando los símbolos de "+" para agregar y "-" para borrar



debemos ingresar los datos, para que quede como en la figura.

El asterisco (\*) se ocupa para definir la selección por defecto, es decir si al evaluar el número no es 1, 2 o 3, tomará ese camino.

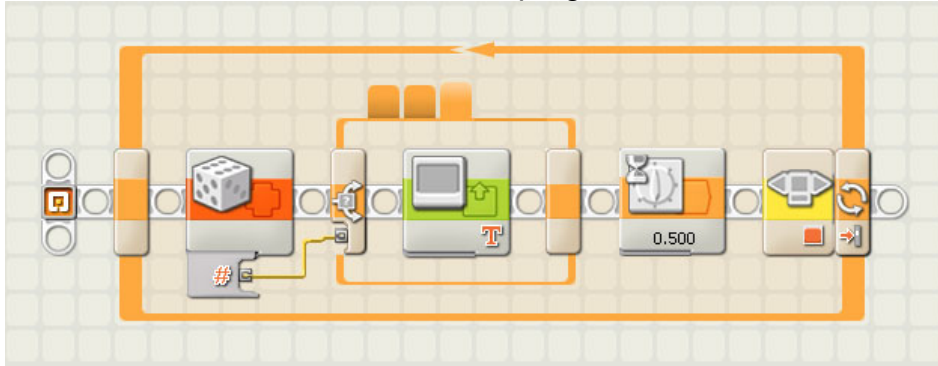
Note que ahora el bloque Switch tiene tres pestañas, al seleccionar la primera debemos insertar la acción que deseamos para esa alternativa, en nuestro caso una pantalla con el texto Numero 1.



luego seleccionamos la segunda pestaña, ponemos otro pantalla, con el texto Numero 2.

y repetimos para tercera pestaña.

Para finalizar el control de salida de nuestro programa será el botón anaranjado.



Trate de incorporar más alternativas y sonido

Notas:

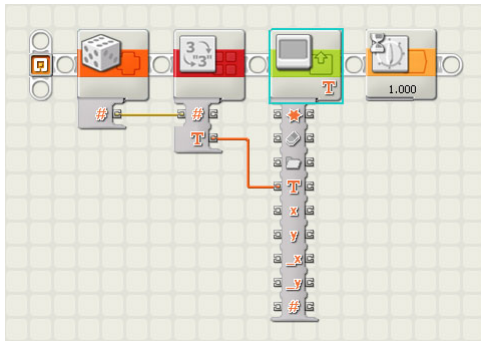



## Data Hub



Cada bloque posee un controlador de los parámetros que se superponen al panel de configuración, este controlador de datos (Data Hub) se expande posicionando el cursor en la parte inferior del bloque. Cada uno de los conectores pueden llevar información entre bloques, que puede ser numérica, lógica y de texto.

Supongamos que queremos mostrar en la pantalla un número al azar entre 1 y 6, al revisar el concentrador del bloque pantalla, nos daremos cuenta que no tiene una entrada numérica, por lo que tenemos que transformar el número a texto, lo bueno es que si hay un bloque para este propósito.



Para instalar los cables, sólo ponga el cursor en el conector, aparecerá un carrito haga un click en el mouse y vaya al destino del cable y haga nuevamente click.

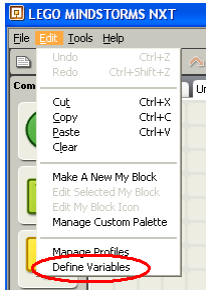
El color de los cables determina en tipo de información que viaja por ellos, ver la ayuda en el programa para más información.

Puede automatizar este programa, para visualizar un números al azar cada vez que se presione el botón derecho y finalizar el programa con el botón “enter”.  
¡Intentelo!

Notas:

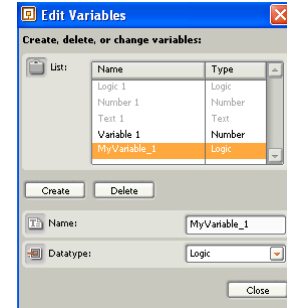

## Variables

Ocuparemos las variables para almacenar datos y hacer uso de ellos durante el programa.

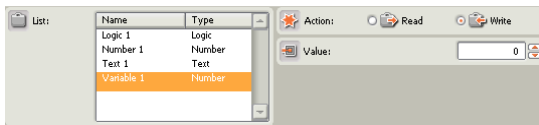


Para definir una variable, tenemos que seleccionar *Edit* → *Define Variable*, en el menú principal.

Luego, le podemos dar un nombre y definir el tipo de variable, numérica, lógica o texto.



Luego en el programa o en el panel de configuración debemos definir el valor que tomará, esta operación se llama grabar “write”, luego para utilizarla emplearemos un leer “Read”



Panel de configuración del bloque “Variable”.

El siguiente programa, elige un número al azar entre 0 y 100, el cual almacena en la variable Number 1, luego elige un segundo número al azar y revisa si son iguales, si lo son el programa termina.

El bloque de comparación nos permite en este caso determinar si los números son iguales

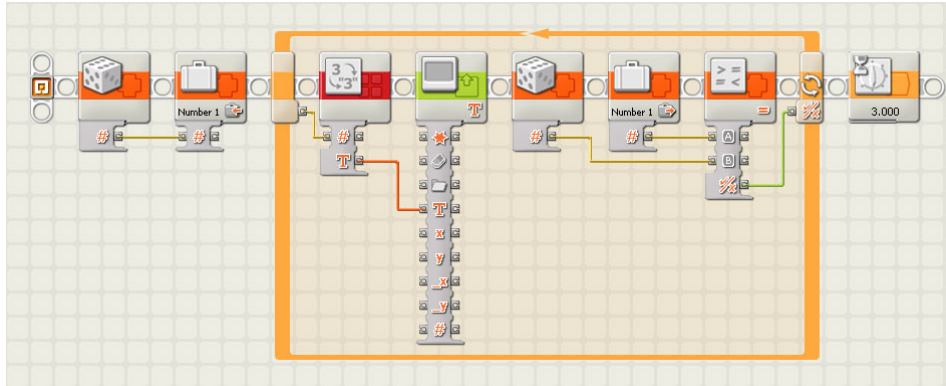


Notas:


### Desafío 7

¿Qué muestra en la pantalla durante 3 segundos este programa?

Si no entiende algún bloque utilice la ayuda en línea



Notas:


## Cronómetros

Medir tiempos es muy importante en algunas programas, para ello utilizaremos el bloque “Timer”.



Hay tres cronómetros disponibles, y básicamente podemos ponerlos en cero (reset) y leerlos (read).



También es posible comparar el valor del cronómetro con un valor y disparar una alarma.

### Desafío 8

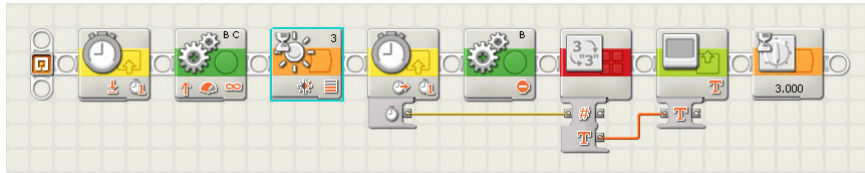
Necesitamos medir, cuanto se demora el robot desde una posición determinada hasta una línea negra.

#### Programa

- Comenzar a medir el tiempo
- Poner en marcha el robot
- Esperar por encontrar la línea negra
- Detener el robot
- Mostrar el valor del tiempo medido en pantalla
- Esperar 3 segundos, para visualizar el tiempo medido

#### Notas:


Solución:



El tiempo se mide en milésimas de segundo

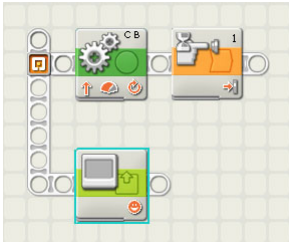
### Desafío 9

Modifique el programa anterior para que muestre en pantalla segundos y no milésimas de segundos

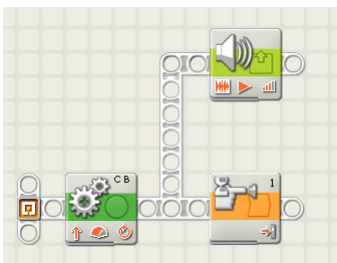
Notas:


## Multiprogramación

Algunas veces necesitamos que el robot realice más de una función simultáneamente, veamos como implementarla



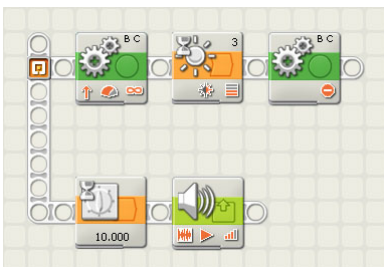
Debemos posicionar el mouse en cualquiera de las ramas(arriba o abajo) de la viga de secuencia y con el botón izquierdo del mouse dibujamos una nueva viga de secuencia.



También es posible generar una viga de secuencia en medio de un programa, en este caso posicione el mouse entre los bloques y presione la tecla shift.

Es importante tener la precaución de evitar los conflictos, por ejemplo en una rama puedo programar que el robot avance y en la otra que retroceda, lo cual obviamente causará un problema.

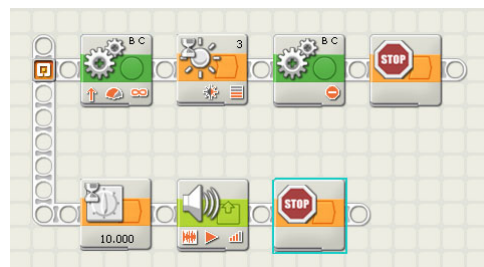
Veamos un ejemplo simple, el robot debe avanzar hasta encontrar la línea negra, si no la encuentra en diez segundos una alarma sonará



La secuencia superior, permite que el robot avance hasta la línea negra y luego se detenga.  
La secuencia inferior, espera por diez segundos y emite una alarma.

Tenemos un problema, ya que si el robot encuentra la línea negra, antes de los diez segundos, igual sonará la alarma y por otro lado si se cumplen los diez segundos la alarma sonará pero el robot seguirá buscando la línea negra.

Para solucionar este problema pondremos al final de cada secuencia un bloque de parada "Stop" que termina la ejecución de ambas secuencias, por lo tanto la que se cumpla primero detendrá la otra.



## Comunicaciones

El robot se puede comunicar vía Bluetooth con otros robot, previamente deben autorizarse y conectarse mediante el siguiente procedimiento.

En el NXT Maestro que se comunicará con los otros NXT, seleccione en el menú principal *Bluetooth* → *Search*, cuando la búsqueda finalice seleccione el NXT con el cual se quiere comunicar y luego escoja el número de la conexión (1,2 o 3).

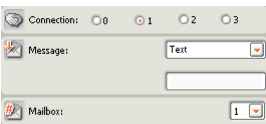
La primera vez que lo haga, se le pedirá que ingrese la clave en ambos NXT.

Un NXT maestro puede conectarse con 3 NXT esclavos.

Revisemos los bloques para enviar y recibir información



### Bloque para enviar información "Send Message"



**Connection:** Indica a quién le enviaré el mensaje. si es del Maestro puedo enviarle a los Esclavos 1,2 y 3. Si es de algún Esclavo debo usar la conexión 0.

**Message:** Puede ser numérico, lógico o texto, se puede incluir el mensaje en el panel de configuración del bloque o en el concentrador

**Mailbox:** Hay diez casillas disponibles y cada casilla puede almacenar hasta cinco mensajes, siempre se guarda los cinco últimos



### Bloque para recibir información "Receive Message"





**Message:** Puede ser numérico, lógico o texto, se puede incluir un mensaje para disparar una alerta.

**Mailbox:** Se debe especificar que casilla deseo leer.

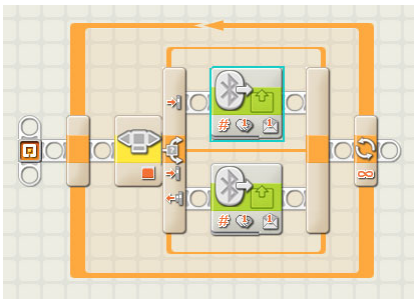
Veamos un ejemplo básico:

El NXT maestro le envía un mensaje a un esclavo, de acuerdo a si el botón anaranjado este presionado(10) o no(10).

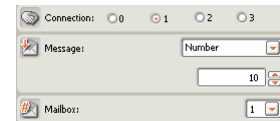
El NXT esclavo recibe el mensaje y lo interpreta de la siguiente forma:

Mensaje Recibido	Pantalla
0	
10	

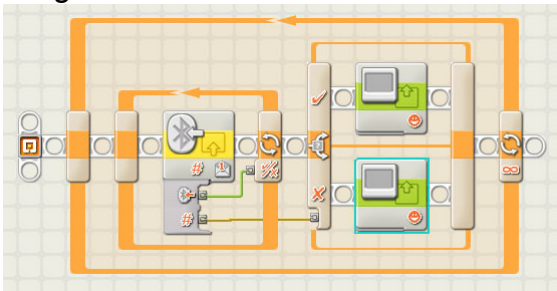
Programa de NXT maestro



Dentro del loop infinito se ha instalado un switch comandado por el botón anaranjado, cuando este sea presionado enviará un número 10 y un cero cuando no lo sea. Se muestra como se ha configurado el panel del bloque enviar para enviar el número 10, de la misma forma se debe configurar el bloque inferior,



Programa de NXT esclavo



El loop interno monitorea la llegada de un mensaje cuando esto ocurre, se sale del loop y el switch determina de acuerdo al número recibido si muestra una bomba o el boom.

Programe los NXT y haga variaciones y pruebas.