

# PROGRAMACIÓ DE MICROROBOTS POR BLOQUES

VNIVERSITAT (👤⚙️)  
DE VALÈNCIA  
Escola Tècnica Superior d'Enginyeria



**Sedi**  
Servei d'Informació  
i Dinamització

Julio 2016

**Julio Martos Torres**

**Adrián Suárez Zapata**

**Daniel García Costa**

**Pedro A. Martínez Delgado**

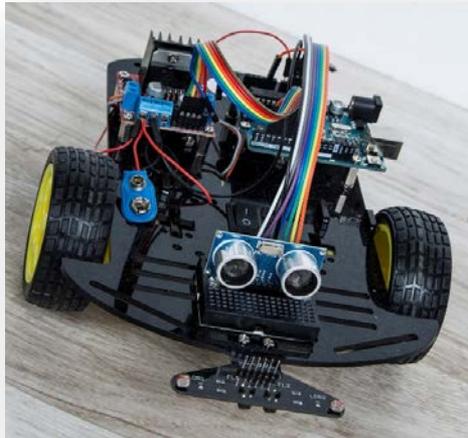
# PROGRAMACIÓ DE MICRORROBOTS POR BLOQUES

ADRIÁN SUÁREZ

PEDRO A. MARTÍNEZ

JULIO MARTOS

DANIEL GARCÍA



## Sedi

Servei d'Informació  
i Dinamització



VNIVERSITAT  
DE VALÈNCIA  Escola Tècnica Superior d'Enginyeria

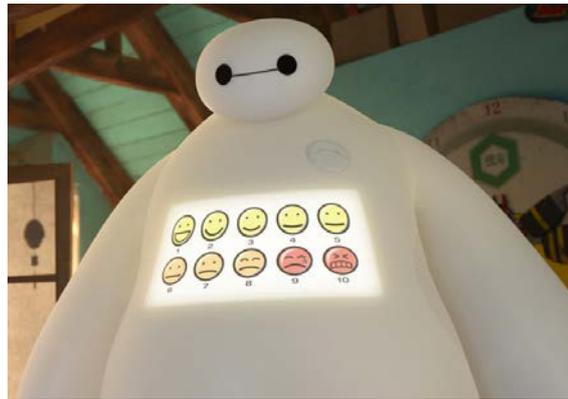
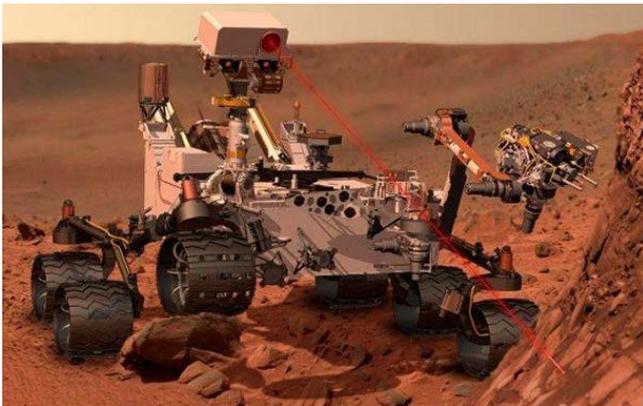


# ¿QUÉ ES UN ROBOT?

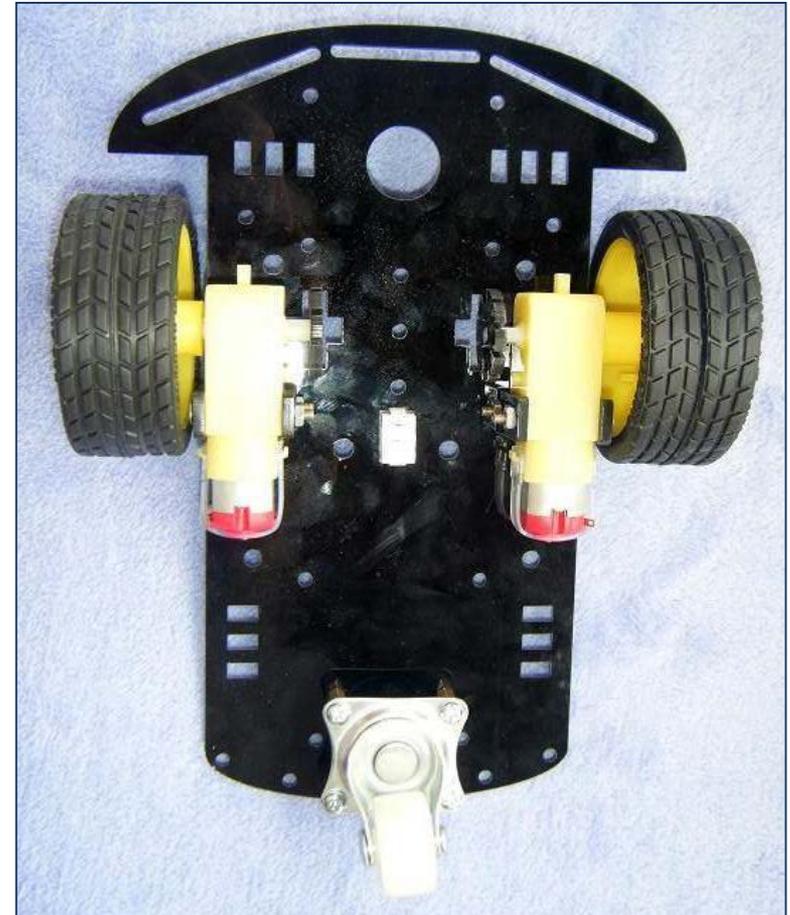
Máquina automática programable capaz de realizar tareas.

Sustituir a las personas en algunas tareas repetitivas o peligrosas.

Suelen tener sensores que le permiten entender su entorno.

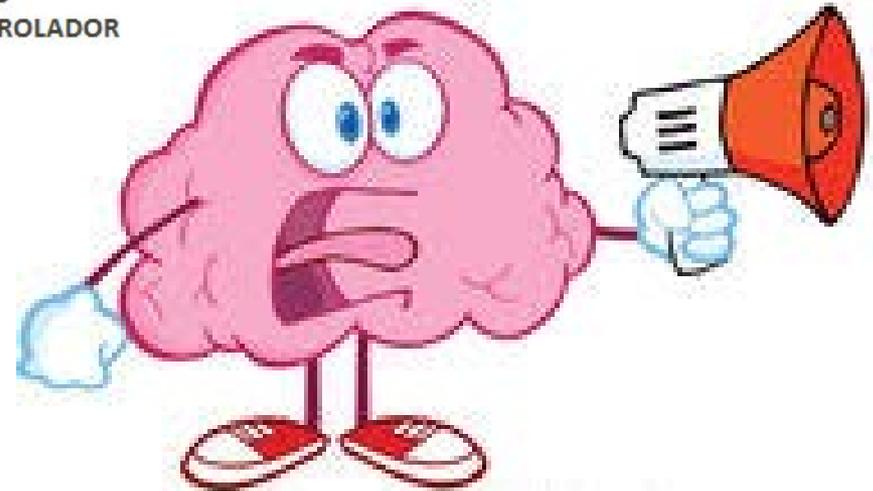


# ESTRUCTURA DEL ROBOT ARDUINO



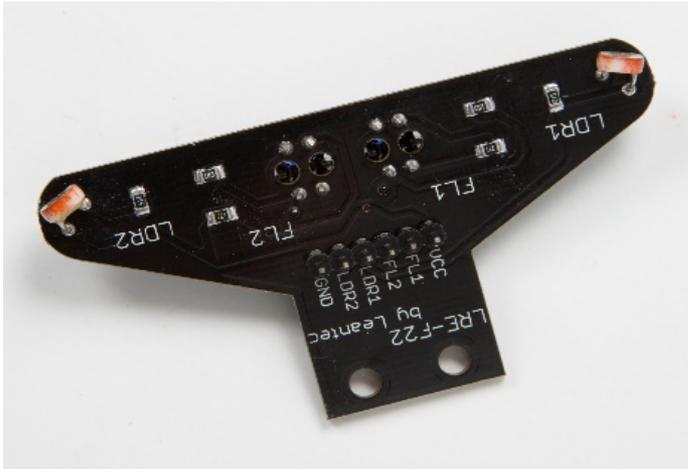
**¿Cómo conseguimos que gire?**

# PROCESADOR – PLACA ARDUINO

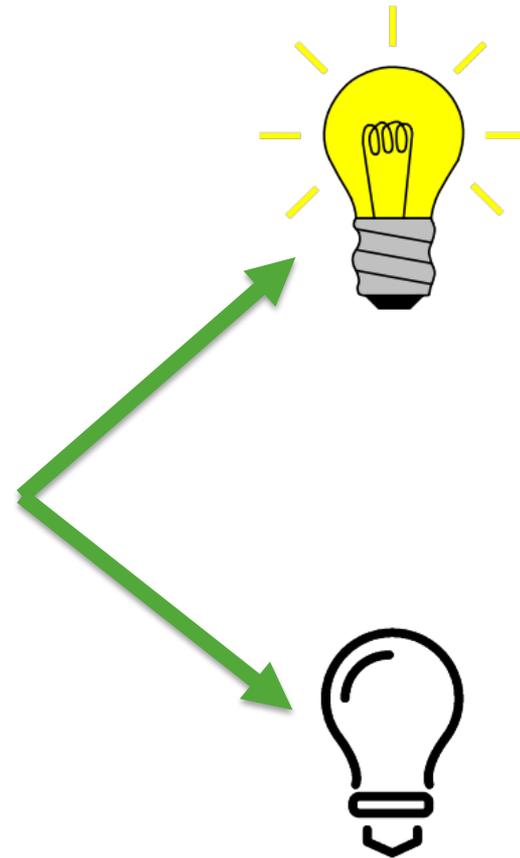


**CEREBRO**

# SENSORES



**Luz**



# SENSORES

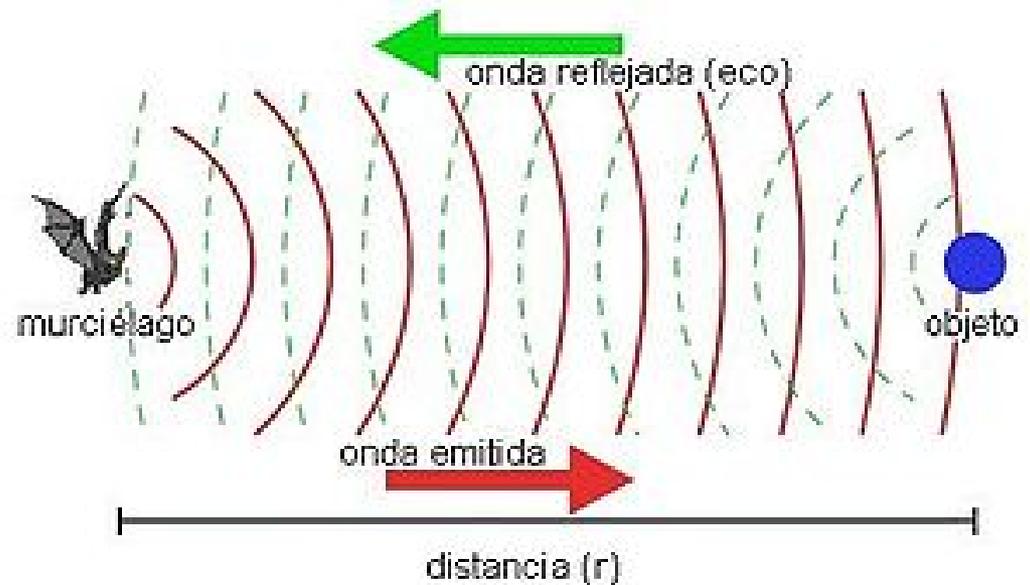


**Ultrasonidos**

# SENSORES



## Ultrasonidos



Sirve para medir distancia y detectar objetos

# ACTUADORES



Altavoz

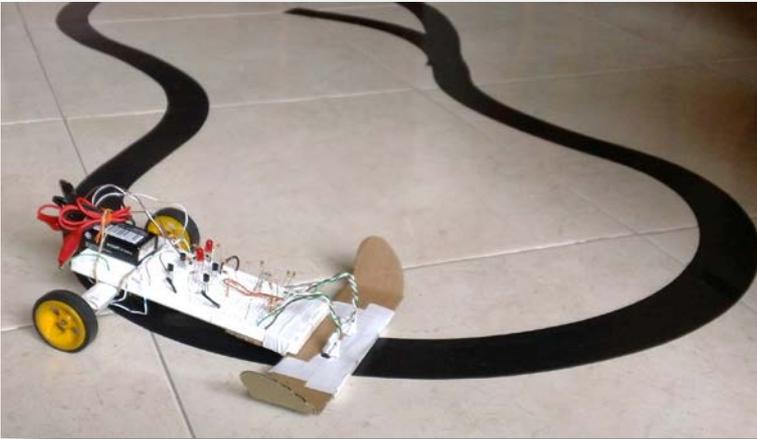


LED o Bombilla



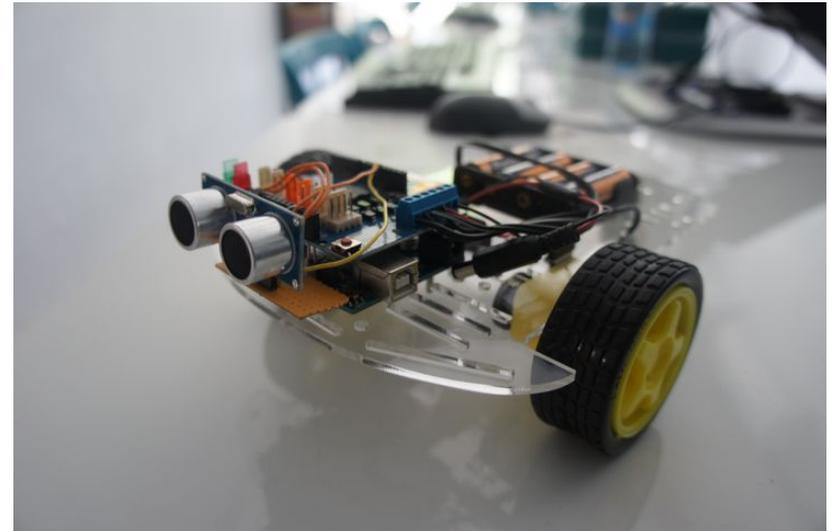
Motores

# EJEMPLOS DE APLICACIONES

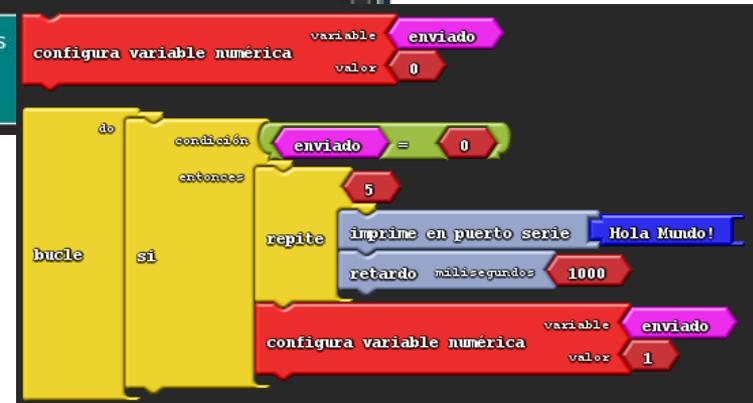
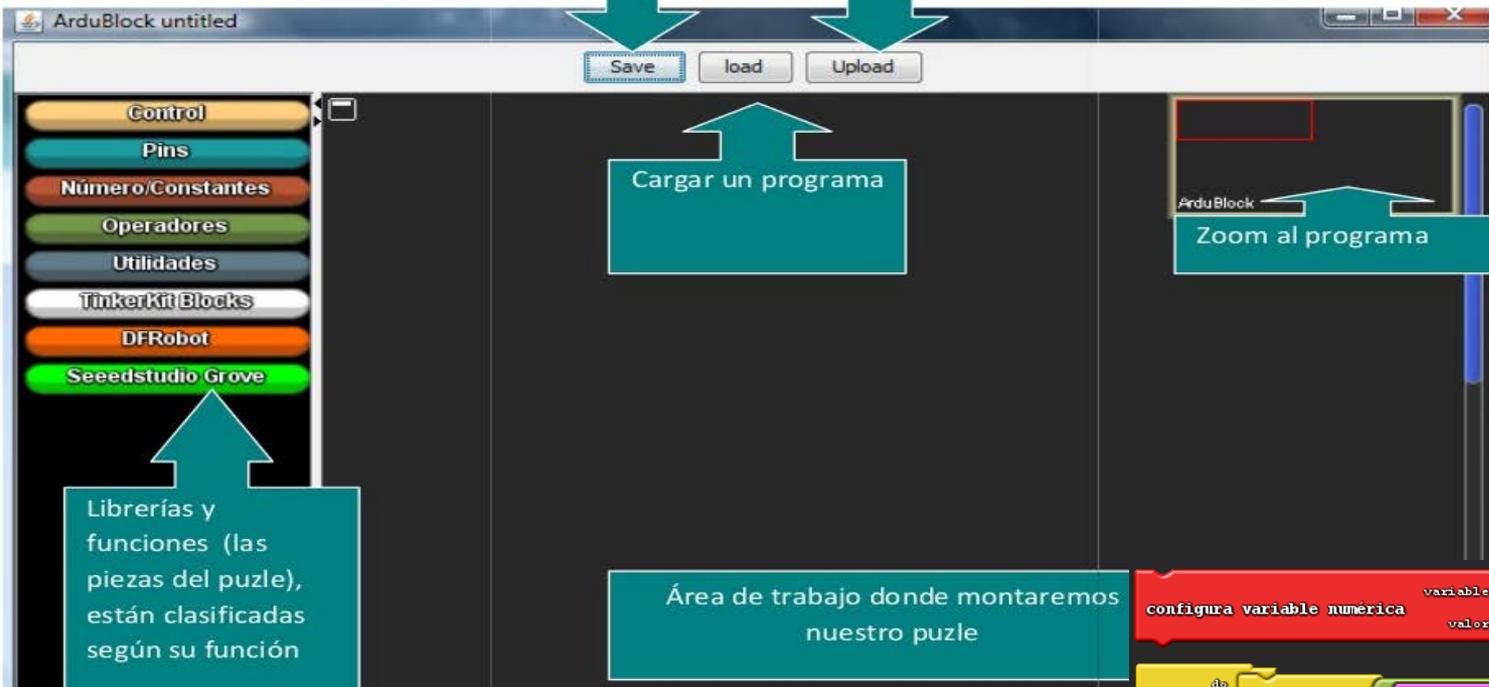
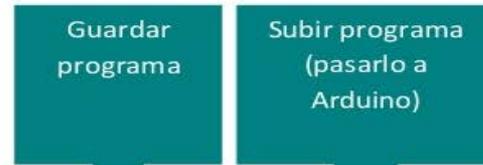


Seguidor de Línea

Esquiva Objetos



# PROGRAMACIÓN





!!! PASAMOS  
A LA ACCION !!!

!!! A PROGRAMAR !!!

# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.

# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



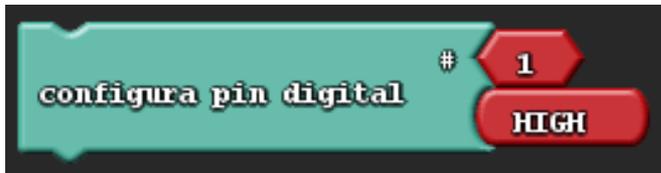
# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el **LED** conectado al pin 13 de la placa Arduino UNO se **encienda** intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



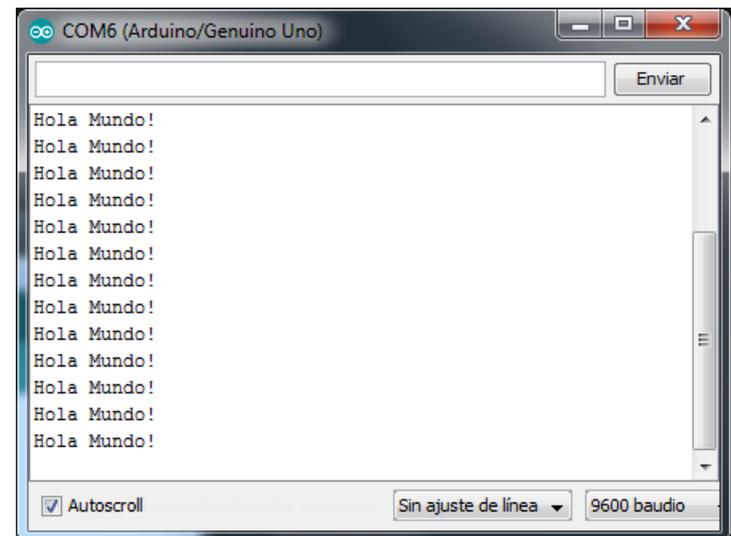
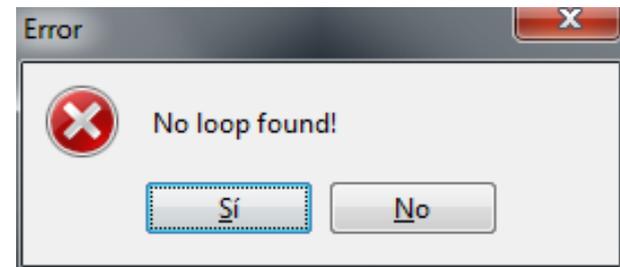
# PROGRAMACIÓN

**Primer Programa:** Crea un programa en ArduBlock que haga que el LED conectado al pin 13 de la placa Arduino UNO se encienda intermitentemente cada 3 segundos. El programa se repetirá de forma indefinida.



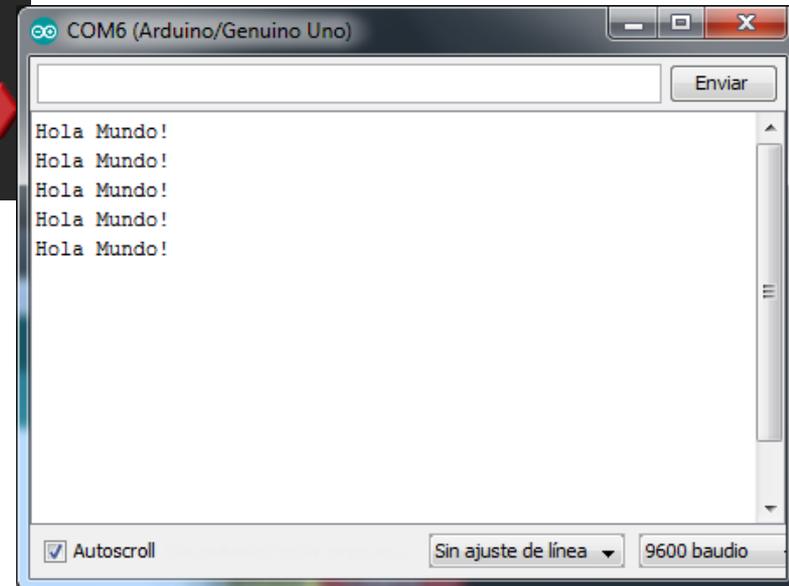
# PROGRAMACIÓN

**Segundo Programa:** Crea un programa en ArduBlock que envíe la frase "Hola Mundo!" 5 veces por puerto serie al PC.



# PROGRAMACIÓN

**Segundo Programa:** Crea un programa en ArduBlock que envíe la frase "Hola Mundo!" 5 veces por puerto serie al PC.

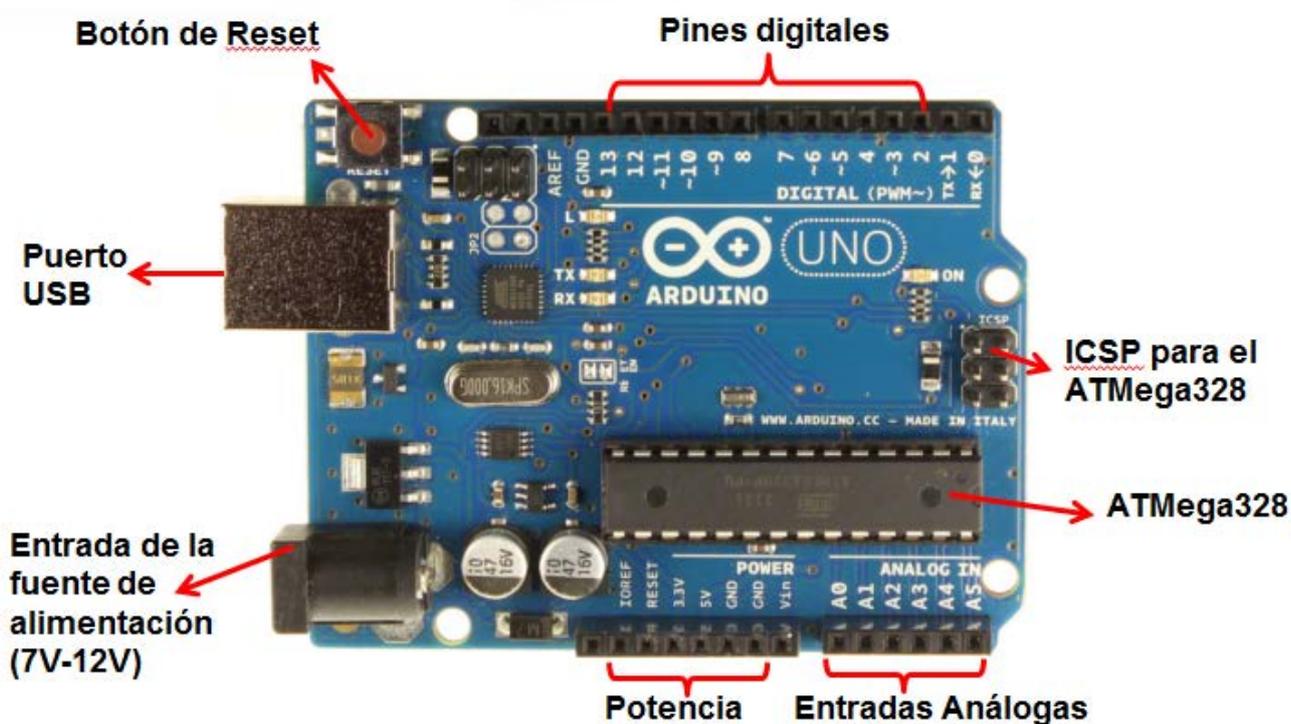


## ¿QUÉ ES ARDUBLOCK?

Es un **lenguaje gráfico de programación** que se distribuye como un complemento que se añade a las herramientas del **Entorno de Programación**, con lo que permite la elaboración de programas para las placas de desarrollo de Arduino. Se basa en el uso de una sencilla interfaz gráfica utilizando **bloques funcionales**, que simbolizan diferentes elementos de programación, por ejemplo: instrucciones, condiciones, variables, bucles...

## LA PLACA DE DESARROLLO ARDUINO UNO

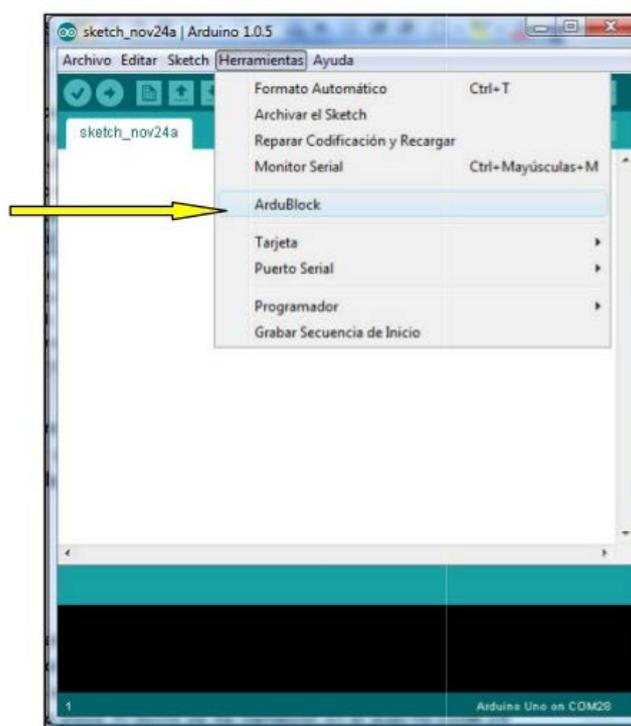
Es una tarjeta electrónica sencilla y de bajo precio, diseñada para el aprendizaje de electrónica y la creación de prototipos. Qué nos vamos a encontrar en esta tarjeta: pines digitales (para encender por ejemplo leds), pines analógicos (para leer el valor de los sensores), memoria flash (en la que se guardan los programas que realizamos), puerto USB (para programar el microcontrolador y comunicarse con la placa), botón de Reset (para reiniciar el dispositivo), indicadores LED (indican que la placa está encendida, un LED que se puede programar, indicadores de comunicación).



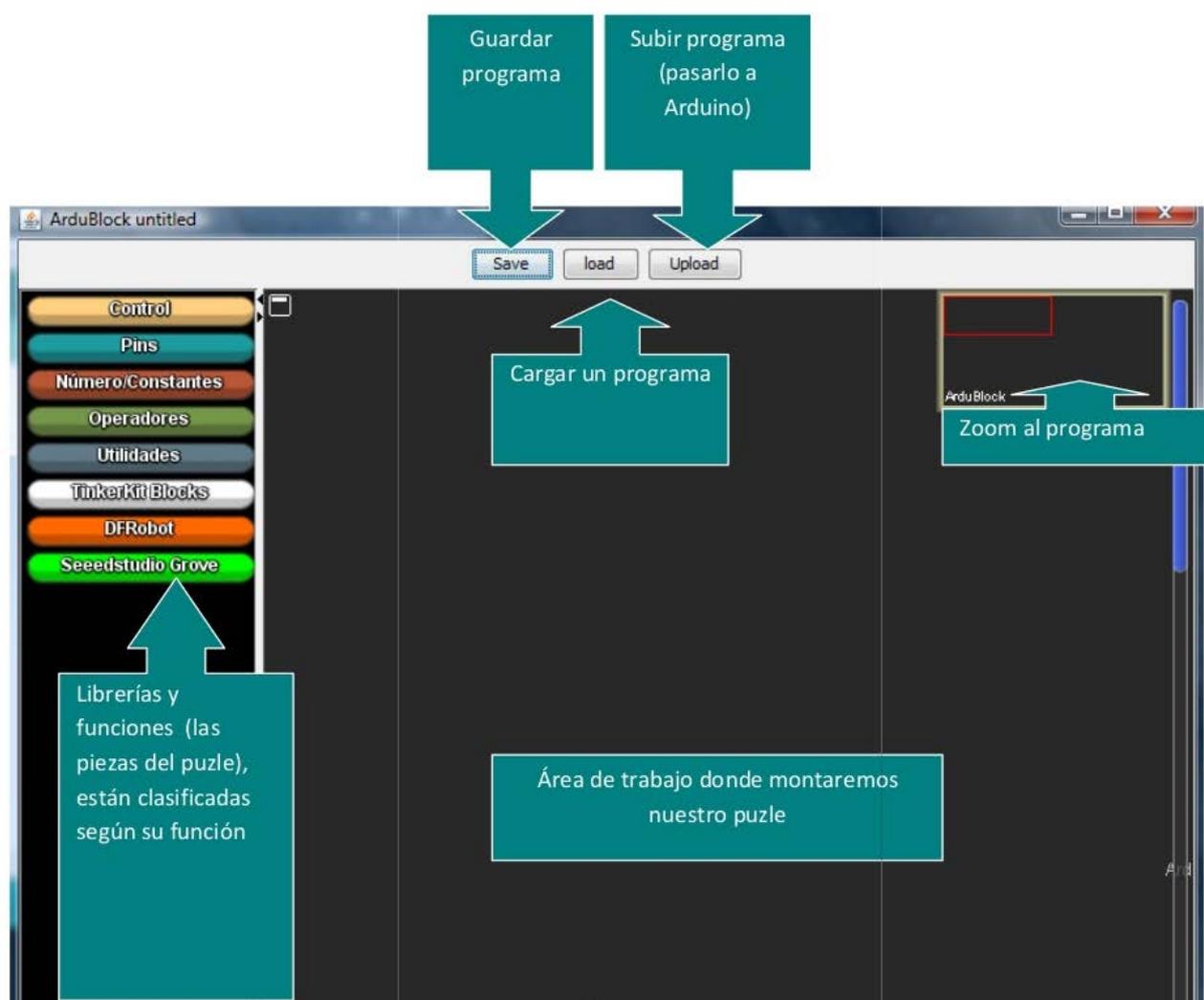
## ENTORNO DE PROGRAMACIÓN

Lo primero que vamos a hacer es abrir el Entorno de Programación Arduino IDE (si no está abierto).

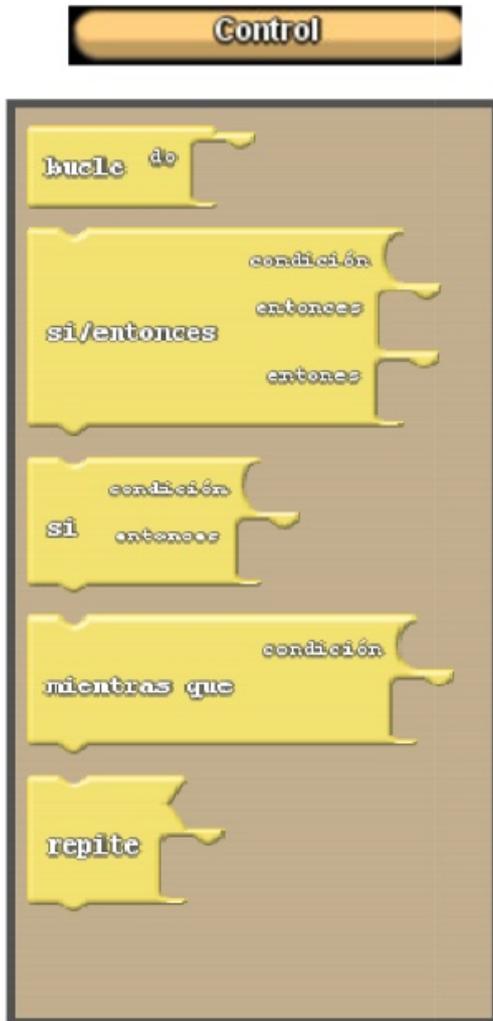
Seleccionamos Herramientas en la barra superior del programa y hacemos clic en ArduBlock para ejecutar el Entorno de Programación Gráfica.



A continuación se nos abrirá ArduBlock.



Ahora vamos a analizar de forma general las librerías de Ardublock. Como puedes ver, se encuentran clasificadas por colores según su funcionalidad y son muy intuitivas y sencillas de manejar. Como ya hemos dicho van encajándose unas en otras como las piezas de un puzle, si colocamos una función o pieza con otra incompatible, ocurre como en un puzle, es decir, no se podrán unir de acuerdo con las normas de programación.



Funciones de Control, estas piezas permiten realizar:

- Bucle: ejecuta la acción de forma indefinida.
- Si/entonces: ejecuta otras instrucciones de manera condicional (está tiene tres enganches) en condición, colocamos la instrucción que establece la condición, en el *primer entonces* la acción que se ejecutará si la condición es **verdadera**, y en el *segundo entonces* la que la acción que se ejecutará si la condición es **falsa**.
- Mientras que: al establecer una condición se mantiene la acción encajada en la parte inferior de la ficha del puzle, mientras la condición establecida siga siendo verdadera.
- Repite: En la parte superior aparecerá el número de veces que se repite la acción colocada en hueco inferior de la pieza.



Funciones de Pines, con ellas se lleva a cabo la gestión de las entradas y salidas tanto analógicas como digitales.

Por ejemplo:

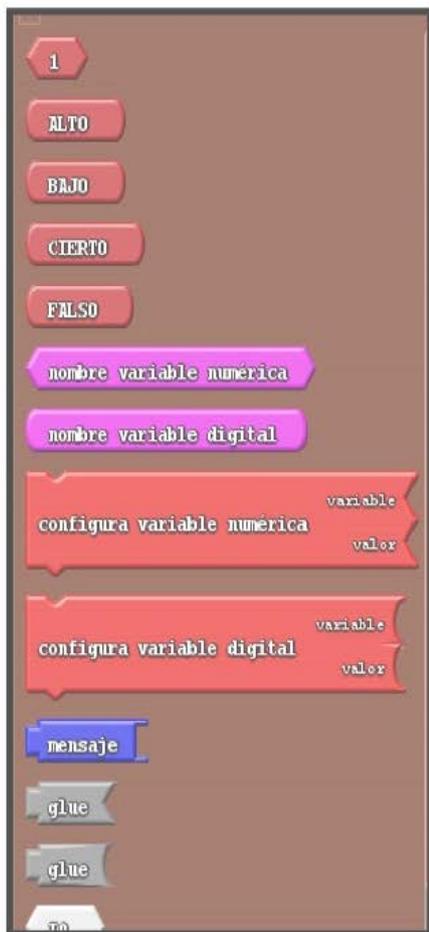
-  Permite leer el estado de una entrada digital, al colocarla en el área de trabajo nos aparecerá asociada al un número 1  al clicar sobre el número escogeremos el número del pin que queramos leer. Lo mismo para pin analógico.

-  Con esta función podemos darle el valor "alto" o "bajo" a la entrada o salida elegida.

-  con esta función gestionaremos los valores de los servos como el ángulo inicial.

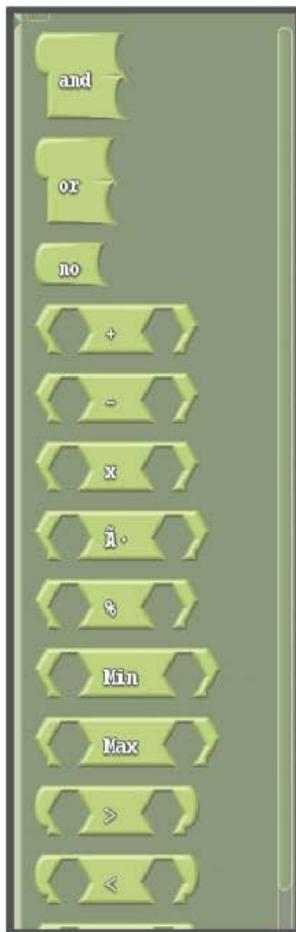
### Número/Constantes

Aquí encontramos fichas muy utilizadas en los proyectos para temporizar como por ejemplo la función retardo, que detiene el progreso del programa los milisegundos que indiquemos.



### Operadores

Estos operadores permiten realizar operaciones lógicas y aritméticas, como por ejemplo las funciones lógicas AND, NOT, OR...



### Utilidades

Con ellas podemos dar valores para operar con ellos (1,2,..) o constantes (alto, bajo, verdadero...) a las entradas y salidas. Fíjate que sus formas encajan con las funciones de pines vistas anteriormente.



## NUESTRA PRIMERA PRÁCTICA: LED INTERMITENTE

Nuestro primer ejercicio será el apagado y encendido del LED L de la tarjeta de experimentación ARDUINO UNO, este LED se corresponde con el pin 13 de la placa y se encuentra conectado internamente por lo que no hay que añadir ninguna conexión mediante cables.

Por lo tanto, en primer lugar incluiremos el bloque BUCLE para que el programa se repita infinitas veces. A continuación añadimos el bloque CONFIGURA PIN DIGITAL con el pin 13 correspondiente al LED y lo configuraremos a nivel ALTO (HIGH) para que se encienda. El siguiente bloque a incluir debajo de la configuración del pin es el de RETARDO MILISEGUNDOS y lo configuramos a 3000 para que el LED se encuentre encendido durante 3s (3000ms). El siguiente bloque a incluir es de nuevo CONFIGURA PIN DIGITAL con el mismo pin, el 13 y a nivel BAJO (LOW) para que se apague el LED. El último bloque a añadir es de nuevo el de RETARDO MILISEGUNDOS y en este caso lo configuramos con 1000 para que permanezca apagado durante 1s.

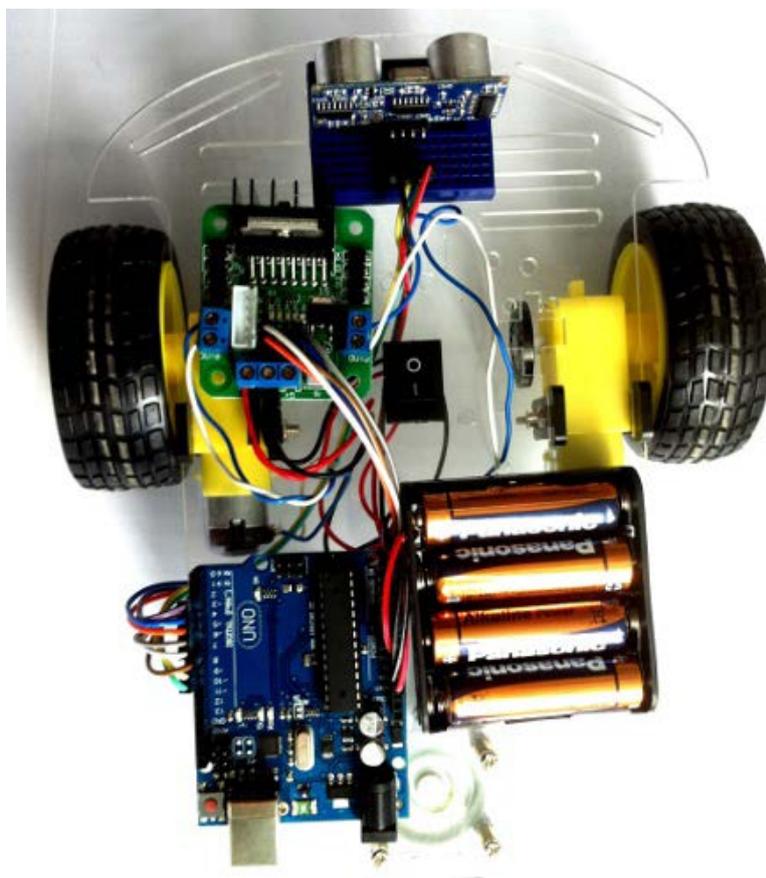
De esta forma ya tenemos el código listo para programar Arduino. Para enviar este código a la placa hay que hacer clic en el botón SUBIR (UPLOAD) y automáticamente nos aparecerá en la pantalla de Arduino IDE el mensaje "compilando el sketch" y en unos segundos aparecerá el mensaje "carga completada".

Si observamos la placa Arduino UNO, veremos que el LED L parpadea, iluminándose 3s y apagándose durante 1s.



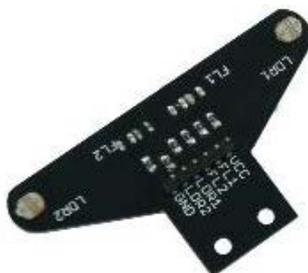
## PRÁCTICA 2: MOVEMOS EL ROBOT

En este ejercicio vamos a hacer que el Robot avance hacia delante durante 2s y a continuación se mantenga parado durante 5s. Por tanto, lo primero a tener en cuenta en este ejercicio es saber qué pines controlan los motores del Robot. Nuestro Robot posee dos motores de forma que cada uno de ellos controla cada rueda motriz, para mover la rueda izquierda hacia delante tenemos que configurar el pin 2 a nivel alto (5 voltios) y para hacer que avance la rueda derecha tenemos que configurar el pin 7 a nivel alto. Si queremos que las ruedas no avancen, es decir, se frenen, tenemos que poner los dos pines que controlan los motores de las ruedas a nivel bajo (pin 7 y 2 a 0 voltios).



### PRÁCTICA 3: ROBOT QUE DETECTA LUZ

En este ejercicio, el Robot se va a mover como en el ejercicio anterior, pero en este caso le vamos a proporcionar capacidad de decisión, es decir, sólo avanzará si iluminamos con una linterna la parte delantera del Robot. Si no lo iluminamos con una linterna, el Robot se mantendrá parado. Para ello, tenemos que tener instalado en el Robot, el sensor de luz que proporciona a Arduino (el cerebro del Robot) el valor de luz que mide. Cuando Arduino lee este valor decide si el Robot tiene que avanzar o permanecer parado.



Sensor LRE-F22 – Sensor de Luz basado en LDR

Para poder leer el valor del sensor, tenemos que emplear el bloque **CONFIGURA VARIABLE NUMÉRICA** (rojo) y cambiamos el nombre de la pieza rosa y lo llamamos luz. Para que lea el valor del sensor de luz que hay conectado a Arduino, también hay que indicar que el pin analógico del que se lee es el 1 (A1). En la siguiente figura se observa esta configuración:



Para que el Robot sea capaz de decidir si le estamos apuntando con un foco de luz directa, tenemos que incluir en nuestro código el bloque de control **SI/ENTONCES** que se encuentra en la librería de CONTROL.



En este bloque, se introduce en primer lugar un operador comparativo, que en nuestro caso será el signo menor que  $<$  y tenemos que comprobar si la variable luz que está midiendo el sensor es inferior a 600. Este bloque está dentro de la librería OPERADORES.



El parámetro 600 corresponde a la luz ambiente de forma que si iluminamos el Robot con una linterna el sensor leerá un valor superior a 600.

Por tanto, si luz es menor que 600 el Robot estará parado y, sino el Robot avanzará como en el ejercicio anterior. A partir de los consejos anteriores, completa el siguiente programa para que el Robot avance cuando detecte que lo estamos iluminando con una linterna.



## PRÁCTICA 4: ROBOT QUE DETECTA OBSTÁCULOS

Un complemento imprescindible para muchos Robots controlados a distancia es un sensor que nos permita saber la distancia libre de obstáculos para movernos. Si queremos movernos en áreas grandes y poder medir distancias en un rango de varios metros el complemento perfecto es un Sensor de Ultrasonidos ya que funciona exactamente igual que un radar. Emite un pulso de sonido a una frecuencia muy elevada y cronometra el tiempo que el sonido tarda en llegar a un obstáculo, rebotar y volver al sensor, es decir, detecta objetos de forma similar a los murciélagos.



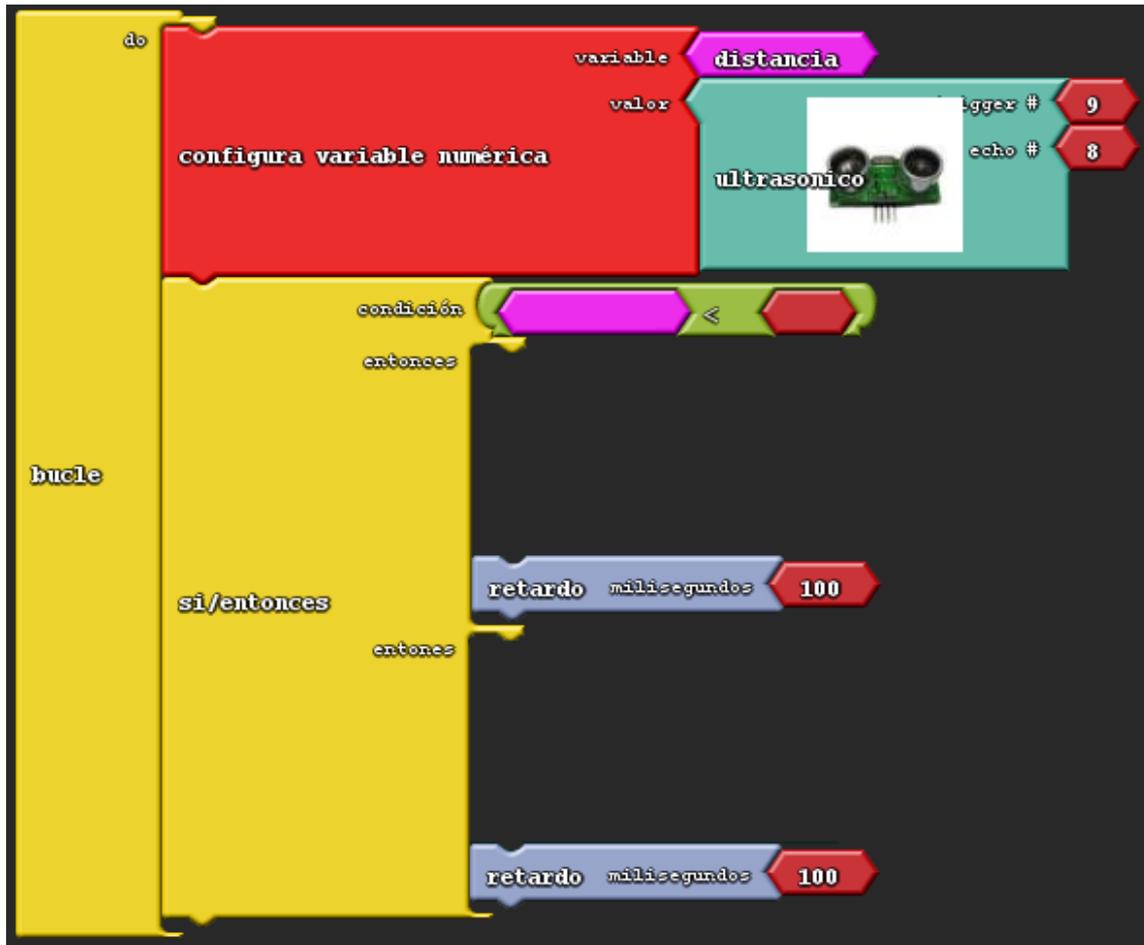
Sensor HC-SR04 Ultrasonidos – Medidor de Distancia

En primer lugar, desmontamos el sensor de luz.

A continuación, incluiremos en el BUCLE el bloque de CONFIGURA VARIABLE NUMÉRICA para leer el valor de distancia que nos proporciona el Sensor de Ultrasonidos. De este modo en variable (pieza rosa), modificamos el nombre y la renombramos como distancia en la que guardaremos el valor de los objetos que detecta el sensor. En el campo valor vamos a incluir el bloque ULTRASÓNICO que lo podemos encontrar en la librería Pins. En este bloque tenemos que incluir los pines 9 y 8 debido a que son los que proporcionan la medida realizada al microcontrolador de Arduino para que lo procese.



A continuación, incluimos igual que en el ejercicio anterior, el bloque SI/ENTONCES para que el Robot decida si debe avanzar o si debe parar dependiendo de lo cerca que se encuentre de algún objeto. Para ello, habrá que incluir un operador que compare si la variable distancia es menor que (<) 35 (centímetros). En el caso de que sea menor, el Robot deberá parar y si no, avanzará. De este modo, completa el siguiente código para que el Robot realice la funcionalidad que queremos:



¿Qué tendríamos que modificar en nuestro código para que, en lugar de parar cuando detecta un obstáculo a menos de 35cm, el Robot gire para evitar el obstáculo y continúe avanzando?