

# USO Y EJEMPLOS PARA SIMPLIBOT V1.0

Autor:  
Héctor Alonso del Bosque





## Tabla de contenido

1. Introducción.....	3
2. Los lenguajes de programación de bloques.....	3
Evolución de los lenguajes de programación.....	3
COMENZANDO A PROGRAMAR.....	4
3. Componentes de Simplibot.....	7
1.1.1. Placa de control Simplibot.....	7
1.1.2. Arduino Nano.....	8
1.1.3. Servos de rotación continua.....	8
1.1.4. Sensores de infrarrojos.....	9
1.1.5. Servo SG90.....	9
1.1.6. Sensor HC-SR04.....	10
4. Programación de Simplibot con Arduino.....	10
Esquema de entradas y salidas de Simplibot.....	10
5. Retos y proyectos para el aula.....	12
Probando la conexión, “Hola mundo”.....	13
Nivel de iniciación.....	14
Ejercicio 1.....	14
Ejercicio 2.....	15
Ejercicio 3.....	17
Ejercicio 4.....	18
Ejercicio 5.....	20
Ejercicio 6.....	21
Ejercicio 7.....	22
Ejercicio 8.....	22
Nivel Medio.....	23
Ejercicio 9.....	23
Ejercicio 10 - Máquina Morse.....	25
Ejercicio 11 - Caja de música.....	26
Ejercicio 12 - Instrumento musical.....	27
Ejercicio 13 - Sensor aparcamiento.....	29
Ejercicio 14.....	31
Nivel avanzado.....	31
Ejercicio 15 - Esquiva obstáculos.....	32
Ejercicio 16 - Siguelineas.....	33



# 1. Introducción

En el siguiente documento encontrará todas las claves y nociones necesarias para poder acceder de una manera sencilla al mundo de la programación y robótica, usando para ello lenguajes sencillos y materiales de bajo coste como el robot diseñado para este efecto llamado Simplibot.

Para seguir el siguiente manual es recomendable tener disponible un ordenador con los siguientes programas instalados, cuyos procesos de instalación se recogen en el Anexo de instalación al final de este documento:

\* IDE de Arduino en la versión 1.6.5

\* Visualino en su versión 0.7.1

Y disponer de un robot Simplibot completo, aunque algunos de los ejemplos podrían ser ejecutados y son compatibles con otras versiones de Arduino como puede ser los Arduino UNO.

Con todo preparado es el momento de tomar contacto con el mundo de la programación.

# 2. Los lenguajes de programación de bloques

## *Evolución de los lenguajes de programación*

Como consecuencia de la invención de las primeras máquinas de cálculo como la inventada por Charles Babbage en el siglo XVII se han ido desarrollando diferentes lenguajes de programación que nos permiten comunicarnos con esas máquinas.

Esos lenguajes siempre terminan basándose en un código Binario de ceros y unos que es la manera en la que una máquina es capaz de tomar decisiones o ejecutar programas, mediante decisiones sencillas. Pero para el ser humano es complicado llegar a trabajar de esa manera, por eso se han ido desarrollando diferentes maneras de comunicarnos con las máquinas.

Tomaremos algunos lenguajes y ejemplos que nos servirán para ver la evolución de esos lenguajes de programación a lo largo de los años, partiendo desde las tarjetas perforadas que en base a si existía o no un agujero en ellas se podían mandar series de ceros y unos a una máquina y esta realizar diferentes cálculos.

### IMAGEN TARJETA PERFORADA

Como podemos ver este sistema hacía difícil la programación y sobre todo la modificación de los programas en caso de existir algún error en ellos, por lo tanto con la aparición de los sistemas y periféricos que nos permitan interactuar con las máquinas se pudo evolucionar la programación hacia una programación más directa en lo conocido como lenguaje máquina en el que se envían directamente esas series de ceros y unos al sistema.

### EJEMPLO LENGUAJE MÁQUINA

Este método de programación seguía teniendo unas limitaciones muy grandes ya que generar programas y poderlos interpretar posteriormente era muy complicado, además de ser muy sencillo el cometer errores en la programación de las máquinas, por ello surgen los lenguajes de bajo nivel como puede ser ensamblador donde se envían órdenes a las máquinas mediante comandos más cercanos al ser humano que posteriormente se convierten a lenguaje máquina.

### EJEMPLO DE ENSAMBLADOR

Aún con estas facilidades la curva de aprendizaje de estos lenguajes es muy pronunciada y el control sobre los programas muy limitado al tener que ejecutar muchas líneas de código para ejecutar ordenes muy sencillas. Por lo que surgen los lenguajes de alto nivel los cuales son más fácilmente entendibles por el ser humano de manera que podamos ejecutarlos simulando que “hablamos” con nuestro procesador.

```
/** Global variables */  
  
/** Function declaration */  
  
void setup()  
{  
  
    pinMode(13,OUTPUT);  
  
}  
  
void loop()  
{  
    digitalWrite(13,HIGH);  
  
}  
  
/** Function definition */
```

Por último llegamos a los sistemas de programación que nos servirán durante este manual y son los lenguajes de programación basados en bloques, los cuales se sirven de relacionar diferentes comandos con un bloque o elemento visual, habitualmente a modo de puzzle donde unas piezas solo encajan con otras, de modo que podamos hacer muy accesible la programación facilitando en gran medida su aprendizaje.






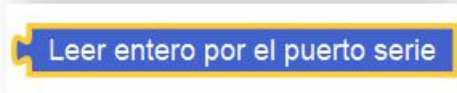
## COMENZANDO A PROGRAMAR

Un error muy común a la hora de empezar a programar es ejecutar directamente el entorno donde vayamos a programar y ponernos a arrastrar bloques o generar código sin un orden concreto o una planificación, lo que nos hará perder más tiempo a la hora de resolver futuros problemas y complicará el aprendizaje.

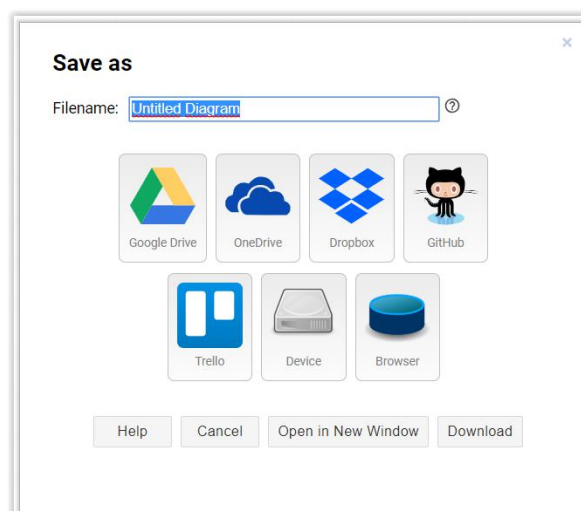
Por eso es muy importante comenzar aprendiendo a dividir o analizar los problemas que se nos presenten en decisiones sencillas donde únicamente respondamos Sí o No, para ello nos apoyaremos en el uso de los diagramas de flujo.

Un diagrama de flujo nos servirá para tener claro que bloques deberemos utilizar más adelante en nuestro programa, ya que la mayoría de símbolos o conectores que usemos tendrán su equivalente dentro de nuestro entorno de programación.

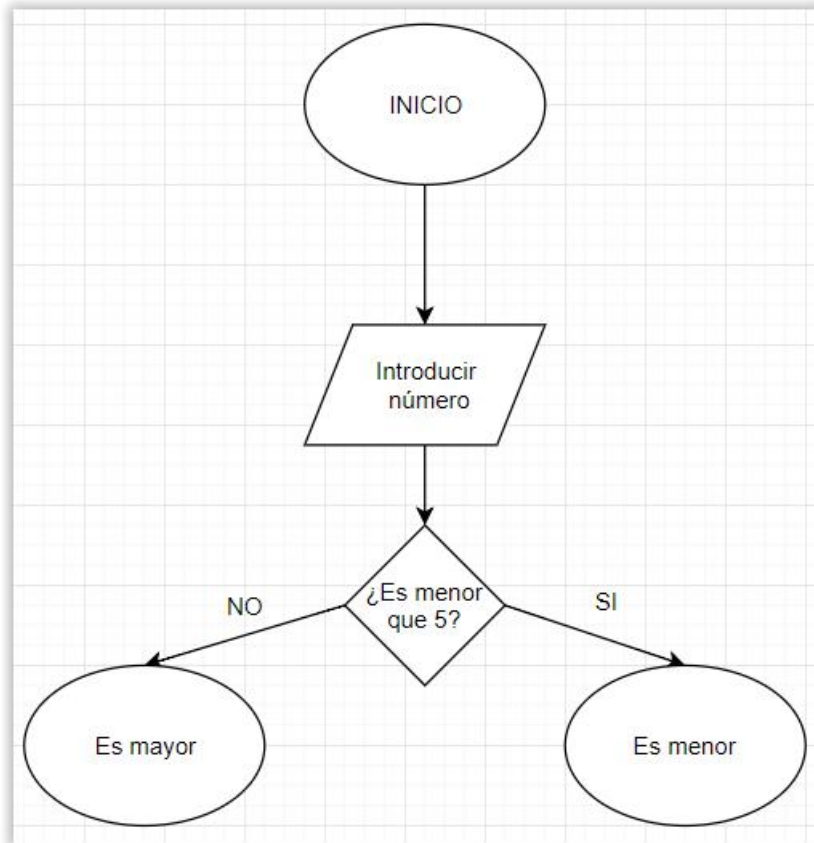
En la siguiente imagen podremos ver los conectores más comunes que usaremos para crear nuestros programas y su bloque equivalente dentro de Visualino que será el entorno que usaremos.

<b>Bloque diagrama</b>	<b>Bloque Visualino</b>
	
	
	
	

Para crear estos diagramas podemos utilizar la página web <https://www.draw.io/> la cual nos permite generar de manera gratuita diagramas simples y guardarlos en diferentes plataformas como podemos ver.



El uso de la herramienta es muy intuitivo y consta de mucha información en Internet por lo que no se entrará a detallar en este manual, pero si que se mostrará como debemos afrontar por un problema que se nos plantee como puede ser saber si un número que pasemos a nuestro programa es menor que 5.



Como podemos ver se ha seguido un flujo descomponiendo ese problema en decisiones sencillas y simples que podremos traducir a nuestro código de bloques:



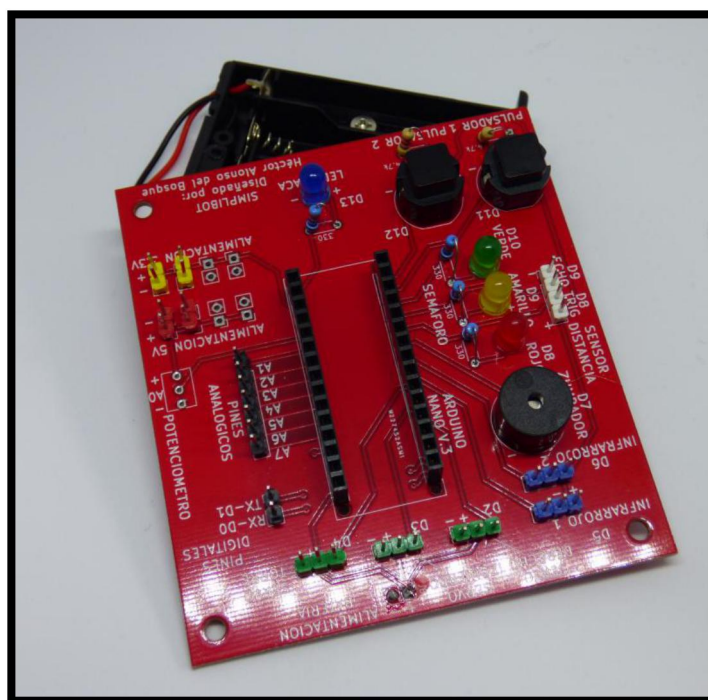
Esta parte a pesar de parecer más pesada o una pérdida de tiempo sobre todo al inicio es muy importante de trabajar ya que serán las bases sobre las que aprenderemos a programar y donde se adquiere la lógica de programación que nos facilitará posteriormente entender y crear programas en cualquier lenguaje.

### 3. Componentes de Simplibot

La electrónica diseñada y usada dentro del proyecto **Simplibot** es de bajo coste y código libre, ya que se persigue el acceso a la robótica y la programación a todas las personas sin que sea necesario un gran desembolso o conocimientos.

Haremos un repaso a todos los componentes de nuestro robot con la finalidad de conocerlos un poco más en profundidad y poder entender su funcionamiento de cara a saber como usarlos dentro de nuestros programas.

#### 1.1.1. Placa de control Simplibot

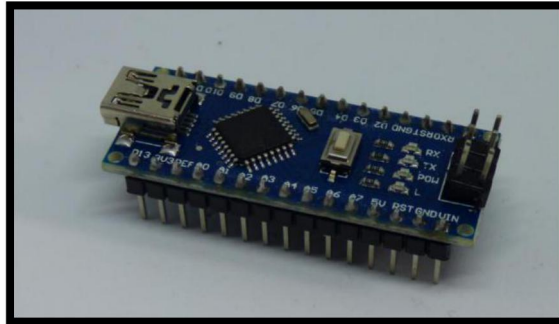


La placa de control sirve para interconectar los diferentes elementos y sensores entre sí, además de disponer de diferentes montajes electrónicos simples que permiten al usuario centrarse únicamente en la programación de la placa sin necesidad de conocimientos electrónicos.

Los ejercicios y funciones que ofrece la misma vendrán detallados al final de este documento, además de ofrecerse los archivos necesarios para la fabricación de la placa por parte de las personas o instituciones que lo necesiten sin depender de un único proveedor, siempre que no se obtenga beneficio económico de esa fabricación y se nombre al diseñador de la placa dentro del proyecto.

La placa dispone de una conexión para alimentación preparada para conectar un portapilas de 4 baterías AA que den una tensión de 6V al sistema. Se ha dejado preparado para ser soldado al sistema con el fin de evitar problemas durante el conexionado y que se pueda invertir la polaridad del mismo

### 1.1.2. Arduino Nano



Se trata del “cerebro” del robot, se ha elegido este micro-controlador por la facilidad de uso que tiene, y por la cantidad de documentación que existe al respecto. Además, el coste de este dispositivo es relativamente contenido (entre 3 y 5 Euros) por lo que es accesible a la mayoría de bolsillos.

El diseño nos permite quitar fácilmente el Arduino de la placa de control para programarlo más cómodamente, o para ser reemplazado en caso de fallo, ayudando así a la reparación y mantenimiento de **Simplibot**.

### 1.1.3. Servos de rotación continua

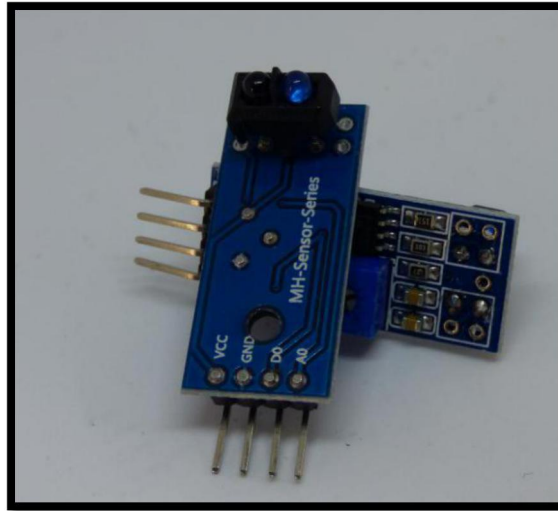


Para el movimiento del robot se utilizan dos servos de rotación continua modelo FS90R, que nos permiten girar 360° de manera indefinida con un buen par de arranque y una velocidad de 130 RPM, siendo alimentados a 6V.

El coste de estos servos es bastante reducido, pero cabe la posibilidad de usar servos SG90 de movimiento limitado siempre que se modifiquen para su uso como rotación continua, tal y como se puede ver en numerosos artículos de Internet



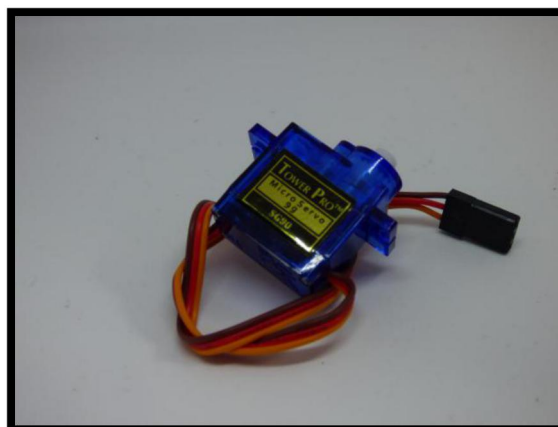
### 1.1.4. Sensores de infrarrojos



Los sensores de infrarrojos podemos considerarlos opcionales dentro del sistema, pero ofrecen una funcionalidad muy atractiva para los usuarios, y es la posibilidad de tener un robot que siga líneas de manera autónoma, distinguiendo entre los colores blanco y negro para ello.

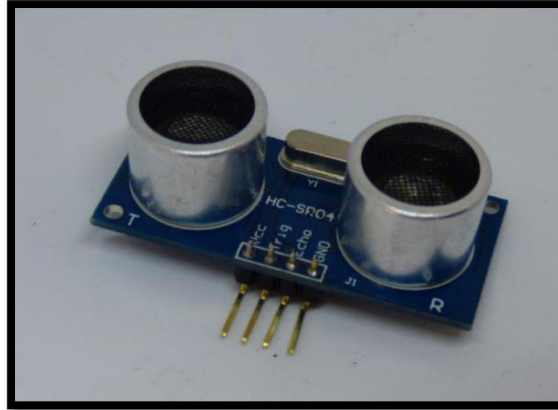
El modelo utilizado trae toda la electrónica necesaria para ser montado de manera sencilla en el robot, haciendo uso únicamente de 3 cables para su conexión y usando la salida Digital para saber si se encuentra sobre un color claro u oscuro.

### 1.1.5. Servo SG90



Además de los dos servos anteriores tenemos la posibilidad de montar un tercer servo de movimiento limitado (0° - 180°) modelo SG90 sobre el que montaremos nuestro sensor de ultrasonidos para poder ser utilizado como radar y detector de obstáculos.

### 1.1.6. Sensor HC-SR04

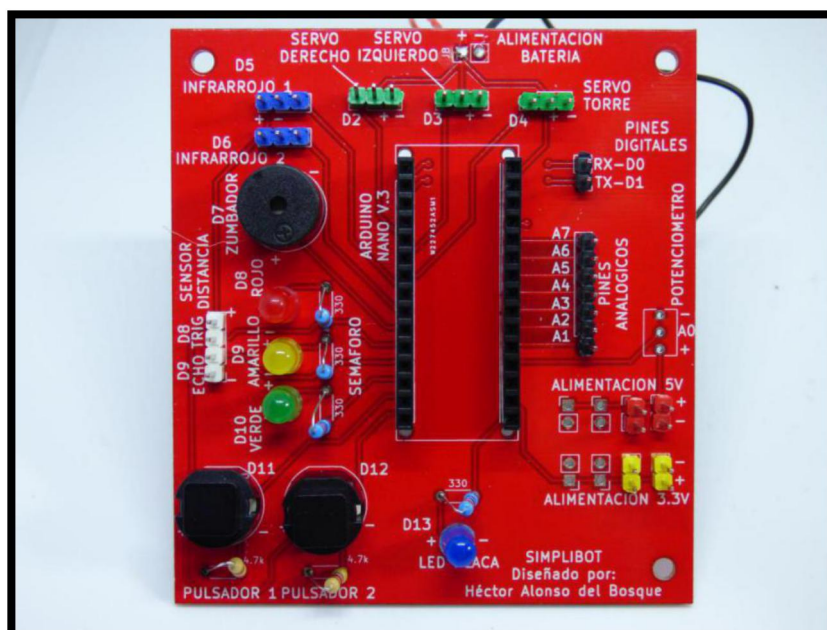


Sensor de ultrasonidos modelo HC-SR04, muy utilizado en Arduino y compatible con multitud de librerías y módulos. Es de bajo coste y nos permitirá medir distancias y conocer si tenemos algún obstáculo en la trayectoria de **Simplibot** actuando en consecuencia.

No es un elemento necesario para el funcionamiento del **Simplibot** pero nos dará la posibilidad de hacer ejemplos muy relacionados con el día a día, como puede ser un sistema de aparcamiento como el de un coche.

## 4. Programación de Simplibot con Arduino

### *Esquema de entradas y salidas de Simplibot*





En la imagen podemos ver la placa con todas sus conexiones, las cuales pasaremos a detallar para conocer las posibilidades que nos ofrece **Simplibot**. Todos los conectores traen marcado el pin al que corresponde en la placa de Arduino para facilitar la programación posterior.

### **Conectores Verdes**

Disponemos de 3 conexiones para Servos, alimentadas directamente desde los 6V del portapilas para evitar reinicios y problemas de alimentación. Al tratarse de un conector donde tendremos un elemento al que enviaremos información, lo consideraremos como una conexión de salida.

### **Conectores Azules**

Conexión para dos sensores de infrarrojos o dos sensores digitales, alimentados a 5V, que necesitemos. Estos conectores quedan disponibles para la conexión de elementos que envíen información a nuestro Arduino por lo que son pines que nos dan una conexión de entrada.

### **Conector blanco**

Pensado para la conexión de un sensor de ultrasonidos HC-SR04, comparte pines con el montaje de semáforo que se ve en la imagen por lo que no se pueden realizar a la vez ambos ejercicios.

### **Conectores Rojos**

Alimentación a 5V para sensores o actuadores que sirvan de expansión en diferentes proyectos.

### **Conectores Amarillos**

Pines para la alimentación de dispositivos extra que funcionen a 3.3V como pueden ser sensores o actuadores.

### **Semáforo**

Montaje de un semáforo con 3 leds para practicar el uso de las salidas digitales con Arduino y su programación. Además nos servirán para facilitar la comprensión de ciertos ejercicios ya que podremos recrear elementos fácilmente reconocibles en nuestro día a día de los que conocemos y vemos su funcionamiento de manera habitual.

### **Zumbador**

Zumbador capaz de emitir sonidos en base a lo que nosotros le programemos, es muy útil ya que estamos habituados a recibir señales de confirmación en la mayoría de máquinas que usamos diariamente y con este elemento podremos generarlas cuando necesitemos.



## Pulsadores

Pareja de pulsadores para practicar el uso de entradas digitales en Arduino, de manera que podamos interactuar con nuestra placa de manera sencilla. Nuestro robot no debe ser simplemente algo autónomo por lo que nos servirán para tener un modo con el que interactuar con el.

## Pines Analógicos

Extensiones para la conexión de sensores o elementos analógicos con los que trabajar. Una de las características de Simplibot es su capacidad de poder extenderse y hacer más proyectos por eso es importante disponer de estos pines donde podremos conectar extensiones como pueden ser:

- \* Resistencias LDR para detectar la cantidad de luz
- \* Sensores de temperatura
- \* Sensores de humedad
- \* Etc..

## Pines Serie

Pines de comunicación serie reservados para la instalación de un módulo Bluetooth por ejemplo para el manejo remoto del robot o el envío y recepción de datos por parte del robot.

## Potenciómetro

Conexión de potenciómetro para el manejo de valores analógicos dentro de Arduino. El manejo de datos analógicos dentro de Arduino es importante ya que existe una gran cantidad de sensores que funcionan mediante este sistema y nos permitirá dotar de nuevas funciones a nuestro robot.

## Led en placa

Led conectado al Pin13 de Arduino que se ilumina a la vez que el integrado en la placa de Arduino, se monta para poder tener el mismo tipo de Led en todos los ejercicios y facilitar la asociación de conceptos a los alumnos.

# 5. Retos y proyectos para el aula

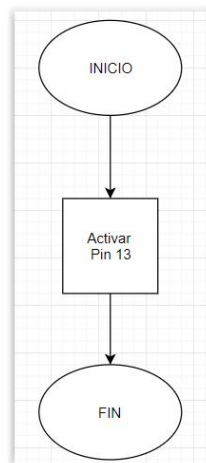
Una vez hemos detallado todos los conceptos anteriores ya estamos listos para poder empezar a programar nuestro robot, esta parte se divide en varios ejercicios ordenados por niveles de dificultad que irán aumentando una vez tengamos afianzados los conocimientos anteriores.

Es muy importante pensar que en esta guía se dan una serie de ideas a modo orientativo pero que es importante probar a modificar y a crear nuestros propios programas para coger soltura en el proceso de creación de un programa.

## Probando la conexión, “Hola mundo”

Lo primero que debemos hacer es probar la conexión con nuestro robot, en este caso además aprovecharemos la ocasión para generar nuestro primer programa en Visualino, lo que en programación se conoce como “Hola Mundo”.

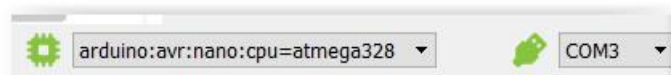
Un “Hola mundo” es el programa más sencillo que podamos ejecutar en un lenguaje de programación y en este caso será activar el led integrado en la placa, a pesar de ser un programa sencillo, deberemos acostumbrarnos a crear sus diagramas de flujo como vemos a continuación para facilitar la comprensión de los programas.



Y pasamos entonces a crear nuestro programa en Visualino como vemos a continuación:



Una vez creado podremos conectar nuestro Simplibot al ordenador y veremos que en la esquina inferior derecha de nuestro entorno de programación podremos seleccionar diferentes puertos y placas. Para que funcione correctamente la carga de nuestro programa deberemos dejarlo configurado como vemos en la siguiente imagen:



\* NOTA: El número de puerto COM será diferente en cada ordenador ya que cambia según el número de dispositivos que se tengan o hayan sido conectados mediante USB al mismo, por eso es importante siempre revisar cuantos puertos COM tenemos en la lista ANTES de conectar nuestro robot, y una vez lo conectemos seleccionaremos el puerto que nos haya aparecido nuevo en la lista que será nuestro robot.

Una vez configurada la conexión únicamente debemos pulsar en el botón de cargar:



Y esperar a que se cargue el programa en nuestro Arduino, sabremos que ha acabado el proceso porque además de que se encenderá nuestro led de la placa tendremos un mensaje en Visualino como el siguiente:

```
Compilando...
Loading configuration...

Initializing packages...

Preparing boards...

Verifying and uploading...

Sketch uses 846 bytes (2%) of program storage space. Maximum is 30.720 bytes.

Global variables use 9 bytes (0%) of dynamic memory, leaving 2.039 bytes for local variables. Maximum is 2.048 bytes.

Finalizó.
```

Ahora sabemos que todo funciona correctamente y además hemos creado nuestro primer programa en Visualino por lo que estamos listos para avanzar.

## ***Nivel de iniciación***

En esta serie de ejercicios haremos una primera toma de contacto de como funciona y como podemos utilizar cada uno de los elementos del robot de manera independiente unos de otros de manera que podamos conocer mejor su funcionamiento y nos sirva para poder crear programas más complejos en los siguientes niveles.

### **Ejercicio 1**

Modificaremos nuestro “Hola mundo” para hacer un programa que haga parpadear el led de la placa con una cadencia de 1 segundo, de esta manera añadiremos un bloque que se usa muy a menudo en programación y es el bloque de “Esperar”, este bloque debe completarse con un bloque numérico que está en el apartado de Matemáticas donde pondremos la cantidad de ms que queremos hacer esperar a nuestro Simplibot.

Hacerle esperar significa que se queda ejecutando la última orden recibida durante la cantidad de tiempo que le indiquemos.

## Diagrama de flujo



## Programa Visualino



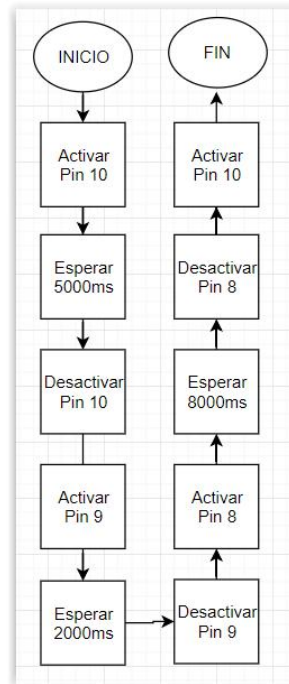
## Ejercicio 2

Una vez que controlamos nuestra primera salida de Arduino y la temporización de su encendido pasaremos a utilizar el semáforo de nuestra placa de manera que usaremos varias salidas diferentes en un mismo programa.

Para ello simularemos el comportamiento normal de un semáforo haciendo que cumpla los siguiente requisitos:

- 1- Se encienda el led verde durante 5 Segundos.
- 2- Pase a la fase de amarillo después y esté 2 segundos en esta fase
- 3- Pasemos a la fase de rojo y nos mantengamos en esta fase durante 8 segundos
- 4- Volvamos a la fase verde

## Diagrama de flujo

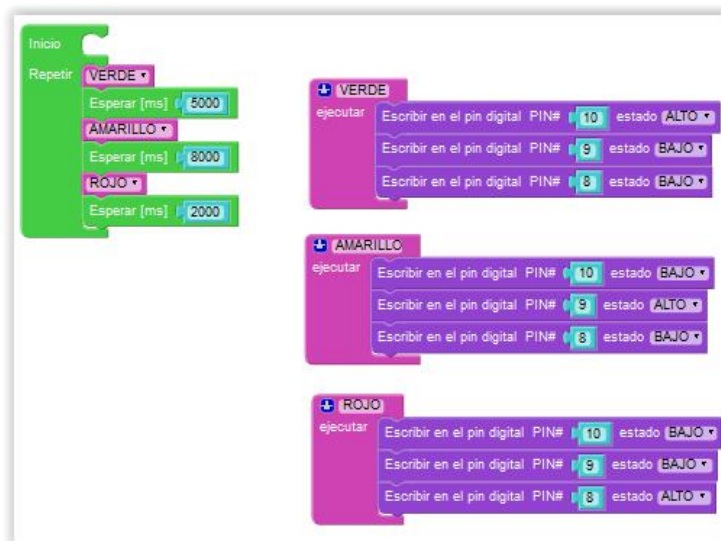


En este caso resolveremos el programa de dos maneras diferentes, la primera de ellas es la más sencilla y es ejecutar todas las ordenes seguidas dentro del programa como vemos a continuación:



La otra manera un poco más avanzada y que puede ser usada más adelante es mediante el uso de funciones. Una función es un sistema que nos permite agrupar ordenes para ser ejecutadas en conjunto cada vez que usemos esa función. En este caso veremos el tipo de función más sencilla ya que se trata de un elemento que puede llegar a ser muy complejo, pero que es interesante de conocer sobre todo cuando hagamos programas más complejos.





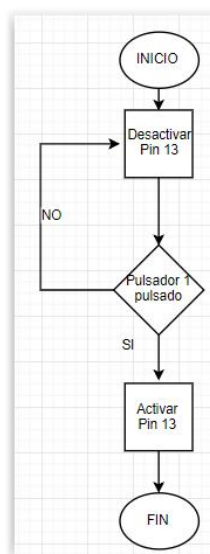
### Ejercicio 3

El siguiente componente a utilizar en nuestro Simplibot sería el pulsador ya que hasta ahora los ejemplos hechos solo han sido relacionados con salidas digitales, es decir enviar ordenes desde Arduino a ciertos componentes. Pero una parte muy importante de uso es el envío de señales a nuestro Arduino mediante diferentes componentes configurando nuestros pines como entradas digitales.

En este caso simplemente haremos un ejemplo sencillo en el que al accionar el pulsador 1 de nuestro Simplibot se encienda el led de la placa, para ello debemos seguir las siguientes instrucciones:

- 1- El led de la placa se apaga
- 2- Si el pulsador está pulsado se enciende
- 3- Si no está pulsado se mantiene apagado

Diagrama de flujo



El programa quedaría funcionando de la siguiente manera

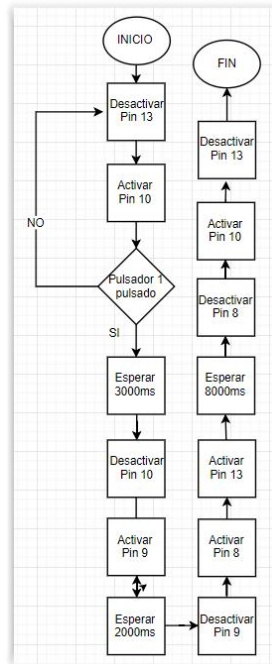


## Ejercicio 4

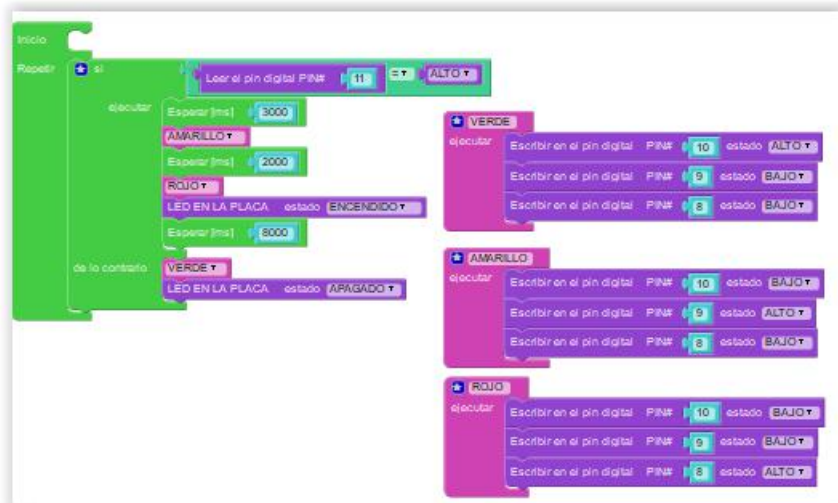
Evolucionaremos el ejemplo del semáforo anterior para introducir nuevos conceptos de programación vistos en el ejemplo anterior, en este caso la lectura de datos o del estado de un sensor con la finalidad de ejecutar ciertas ordenes. El ejemplo que usaremos para ello será la programación de nuestro semáforo para que se ejecute la siguiente secuencia que simula un semáforo con un pulsador para peatones.

- 1- El led de la placa (simula el semáforo verde de peatones) está apagado
- 2- El semáforo tiene el led verde encendido
- 3- Si el pulsador se encuentra sin pulsar se mantiene en verde
- 4- Si el pulsador está pulsado esperamos 3 segundos con el led verde encendido
- 5- Pasamos a la fase amarilla y esperamos 2 segundos
- 6- Pasamos a la fase roja y nos mantenemos en esta fase 8 segundos
- 7- Durante la fase roja se debe encender el led de la placa simulando el semáforo de peatones en verde
- 8- Volvemos a la fase verde.

Diagrama de flujo



El programa traducido al lenguaje de bloques quedaría de la siguiente manera:



## Ejercicio 5

Otro de los elementos incluidos dentro de Simplibot es un zumbador que nos servirá para poder enviar señales sonoras con nuestro robot, es un uso muy común para advertir al usuario de ciertos eventos que puedan ocurrir durante el uso del robot. El zumbador se debe tratar como una salida más, igual que usamos un led, por lo que su programación será muy sencilla y la haremos de la misma manera que con un led. Por lo que haremos sonar el zumbador de manera intermitente con una cadencia de 2 segundos.

Diagrama de flujo



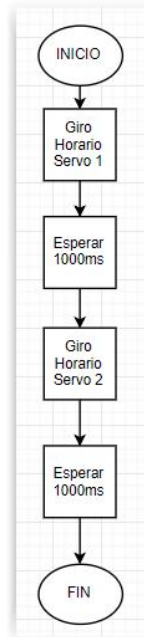
El programa es igual de sencillo que el “Hola mundo” de un led.



## Ejercicio 6

Una de las características principales de Simplibot es su capacidad de poder moverse de manera autónoma, para ello hace uso de dos servos de rotación continua tal y como se ha detallado anteriormente. Para manejarlos haremos uso de los bloques específicos que nos proporciona Visualino, en este caso simplemente veremos la manera en la que hacerlos girar de manera alterna y nos servirá para ver el sentido de giro de cada uno de ellos.

Diagrama de Flujo



Los bloques que usaremos serán los siguientes:

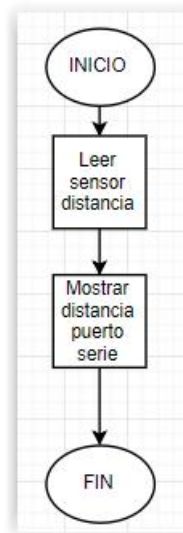


Tal y como podemos ver cada uno de los motores gira en un sentido a pesar de estar programados ambos para girar en sentido horario, esto es debido a la posición en la que están instalados en el robot, veremos como de esta manera podemos hacer que nuestro robot gire hacia la derecha.

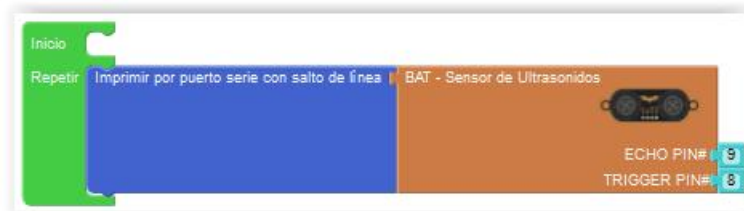
## Ejercicio 7

Otro elemento que nos dará muchas posibilidades en nuestro robot es el sensor de ultrasonidos, con este sensor además de medir distancias tomaremos contacto con otra manera nueva de mostrar datos dentro de Arduino y es mediante el uso del puerto serie y la consola de Visualino. Para ello haremos uso de nuevos bloques que nos facilitaran esta tarea. En el ejemplo siguiente simplemente leeremos la distancia a la que se encuentre nuestra mano del sensor de ultrasonidos y lo mostraremos a través de la consola de Visualino.

Diagrama de flujo



Y el programa quedaría de la siguiente manera:

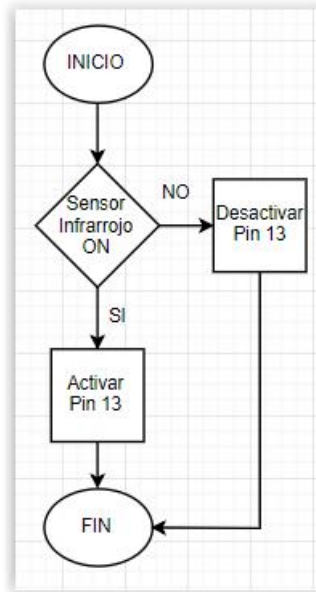


## Ejercicio 8

El último elemento que probaremos será el sensor de infrarrojos colocado en la parte inferior de Simplibot, este sensor está pensado para poder distinguir diferentes tonos de color, en este caso lo usaremos para diferenciar el color negro y blanco y que en los siguientes ejercicios nos permita hacer que nuestro robot siga una línea.

En este caso el ejercicio consistirá en encender el led de la placa cuando nuestro robot detecte una línea negra pintada en un folio blanco.

Diagrama de flujo



El programa de Visualino sería de al siguiente manera:



## Nivel Medio

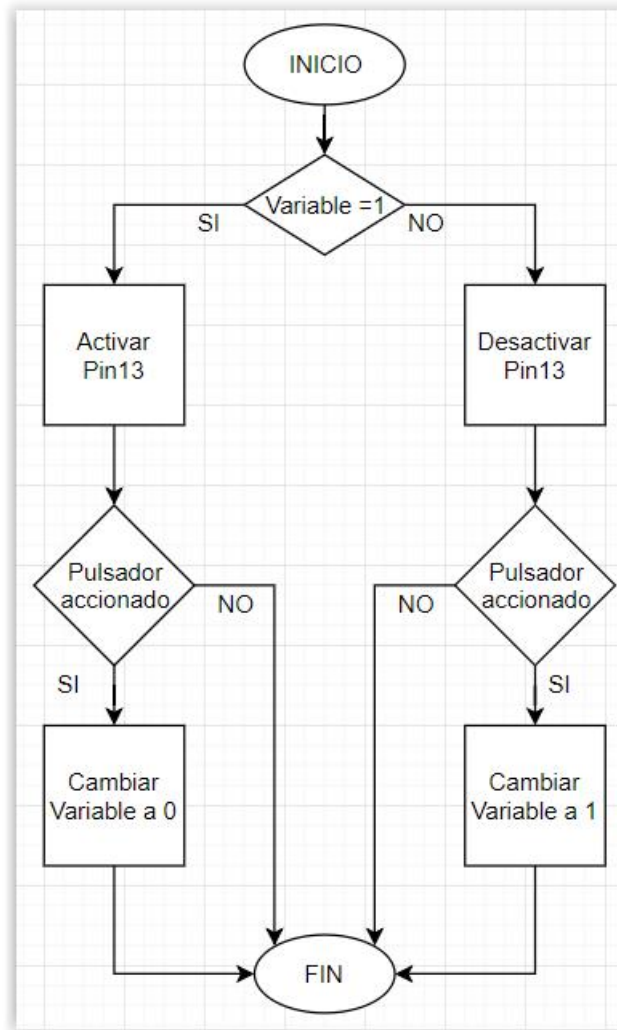
Los siguientes ejercicios harán uso de varios elementos del robot a la vez por eso se consideran de un nivel algo superior y serían la evolución normal de uso de nuestro Simplibot.

### Ejercicio 9

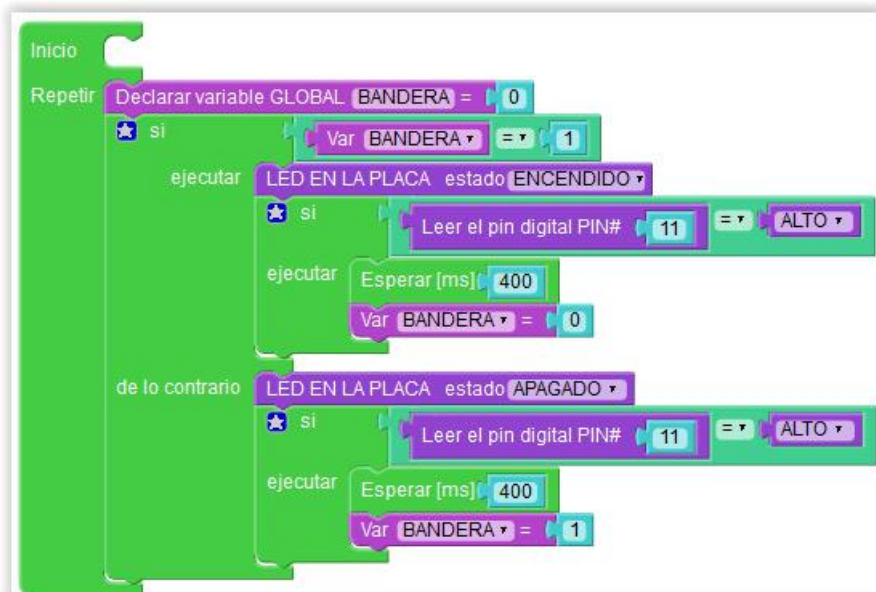
Empezaremos este apartado usando de nuevo un pulsador de la placa para poder aprender un nuevo concepto imprescindible dentro de la programación, y es el uso de variables. Una variable nos servirá para almacenar un valor que irá cambiando según necesitemos, ya sea mediante la lectura de un sensor o un valor que nosotros introduzcamos. En este caso el ejercicio consistirá en encender y apagar un led mediante el mismo pulsador.

- 1- El led empieza apagado
- 2- Accionamos el pulsador
- 3- El led se enciende y mantiene encendido
- 4- Accionamos el pulsador
- 5- El led se apaga y mantiene apagado

Para ello usaremos el siguiente diagrama de flujo:



Y su programa en Visualino sería como el siguiente:



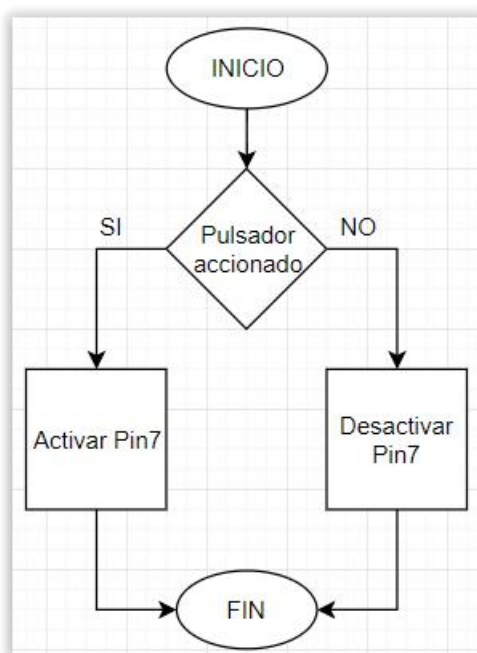


## Ejercicio 10 - Máquina Morse

Con todos los conceptos tratados en este manual anteriormente deberíamos estar en disposición de comenzar proyectos más complejos que incluyan varios de los sensores disponibles en Simplibot. En este caso nos interesa además hacer que dichos ejercicios puedan servir incluso de refuerzo para otras asignaturas de manera que facilite el aprendizaje por parte de los alumnos.

Con el siguiente ejercicio haremos una máquina sencilla de Morse, esta máquina puede servir de manera transversal en otras asignaturas para que puedan físicamente tocar y probar algunas de las tecnologías de las que se les hablan. Por ello simplemente lo que haremos será encender nuestro zumbador cuando se accione el pulsador.

El diagrama de flujo al igual que el programa es muy sencillo.



Nuestro programa de Visualino



En este caso es interesante recalcar sobre todo que a pesar de la sencillez del programa nos permitirá varios usos dentro del aula y facilitará la comprensión de ambas asignaturas a los alumnos, al ver un uso a la programación por bloques y además poder probar algo que se suele explicar de manera teórica.

## Ejercicio 11 - Caja de música

Otro de los usos gracias al zumbador incorporado, será crear una caja de música con nuestro Simplibot, esto nos servirá para que prueben y generen sus propias partituras de manera sencilla, para reforzar los conocimientos impartidos en las clases de música.

El ejemplo será muy sencillo y se basará en tocar la canción de Cumpleaños Feliz con nuestro Simplibot. Para ello debemos establecer una serie de relaciones y conocer como funciona el bloque del zumbador ya que no emitiremos un sonido plano y debemos darle duraciones.

<u>Nota musical</u>	<u>Tiempo</u>
Redonda	2000ms
Blanca	1000ms
Negra	500ms
Corchea	250ms
Semicorchea	125ms

Tomamos como referencia las blancas para asignarle un tiempo de 1 segundo, si queremos acelerar nuestras canciones solamente debemos dar un menor tiempo a esa nota y crear nuestra tabla de nuevo.

**Cumpleaños feliz** **Popular**

**Allegro**

Flauta

*ritardando*

Las notas musicales son las que se muestran anteriormente y en este caso no sería necesario crear un diagrama de flujo ya que nuestro programa se basa en las instrucciones de nuestra partitura.

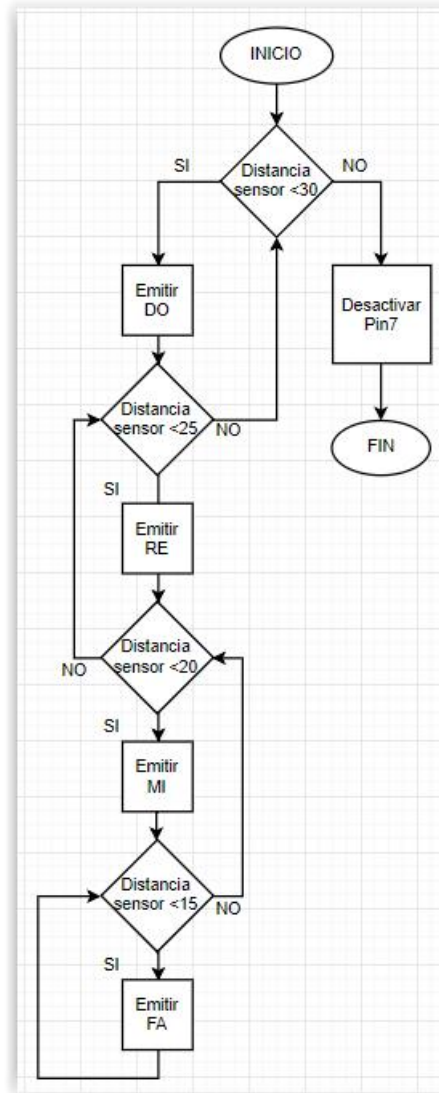
El programa de Visualino quedaría de la siguiente manera:



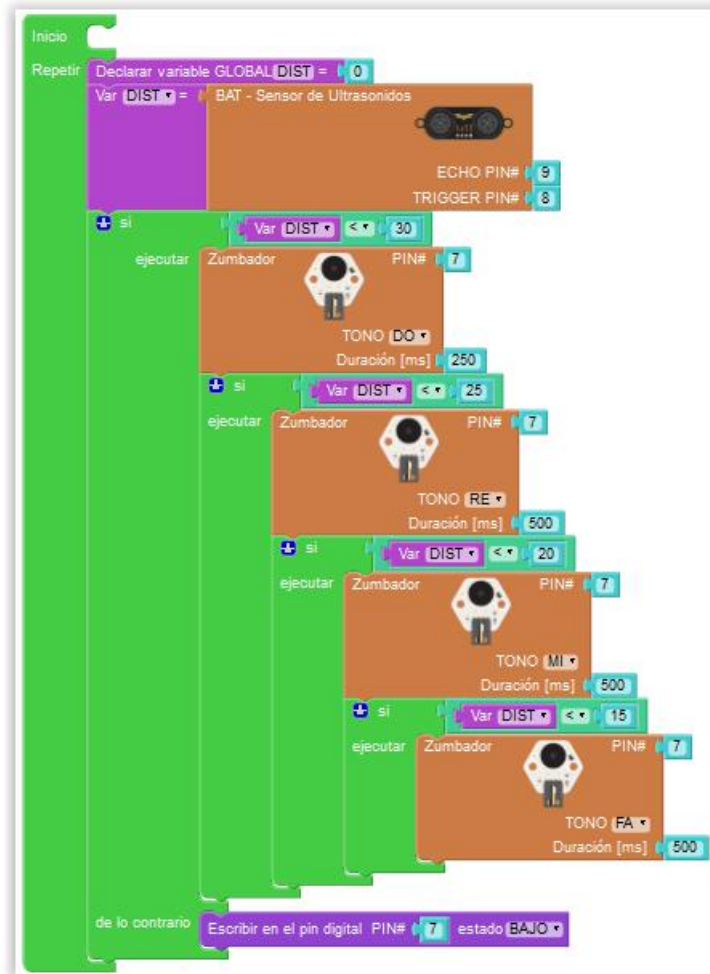
## **Ejercicio 12 - Instrumento musical**

Continuamos en este caso usando nuestro zumbador pero ahora de una manera algo diferente ya que crearemos nuestro propio instrumento musical, ese instrumento será un Theremin. Este instrumento se basa en leer la distancia a la que colocamos nuestras manos de nuestro sensor de ultrasonidos para emitir diferentes notas según varíe esa distancia.

El diagrama de flujo que seguiremos en la programación será el siguiente:



Vemos que parece algo más complejo que los anteriores pero es simplemente por que encontramos varios condicionales If anidados unos dentro de otros pero en la práctica el programa quedaría de la siguiente manera:

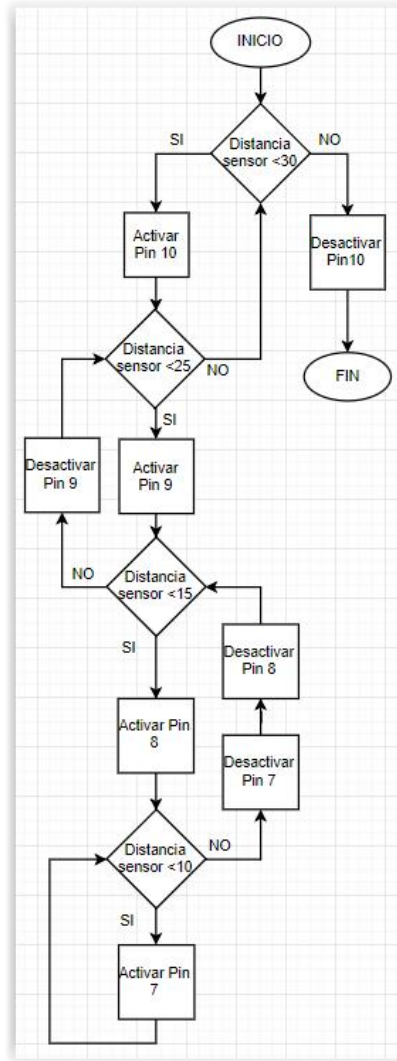


Es un programa largo pero que se puede hacer usando muchas veces el duplicar bloques por lo que es sencillo, a pesar de ayudarnos a manejar con más soltura los condicionales dentro de la programación.

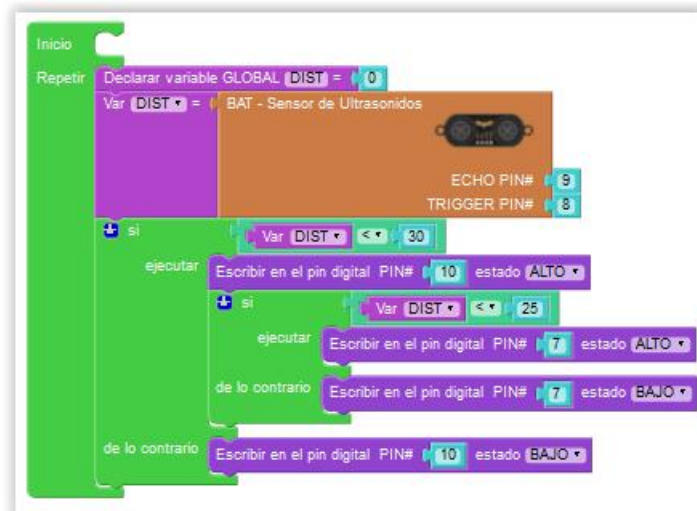
### **Ejercicio 13 - Sensor aparcamiento**

Con lo visto anteriormente estaremos ya en disposición de hacer nuestro primer ejercicio en el que incluiremos tres componente de Simplibot de manera simultanea. Para ello haremos un ejemplo que suele gustar mucho a los alumnos ya que programaran un sistema que ahora ven de manera cotidiana en su día a día, este sistema es un sensor de aparcamiento.

Para ello haremos uso del sensor de infrarrojos, del semáforo y del zumbador siguiendo el diagrama de flujo que podemos ver a continuación.



El programa como en el caso anterior llega a ser más sencillo que descomponer cada una de las acciones en el diagrama de flujo.



## Ejercicio 14

Este último ejercicio del nivel intermedio nos servirá para controlar el movimiento de nuestro robot y conocer como podremos hacer que se mueva en la dirección que nosotros deseemos, ya que debido a su construcción veremos que debemos tener en cuenta el sentido de giro de cada servo para lograr dirigir nuestro robot.

Para ello nos apoyaremos en otro de los conceptos vistos anteriormente y que nos ayudaran después a manejar de manera sencilla a Simplibot y es el uso de funciones, de manera que crearemos 4 funciones, una por cada dirección del robot (Delante, detrás, derecha, izquierda) y le haremos moverse un tiempo determinado en cada dirección.

Diagrama de flujo

DIAGRAMA DE FLUJO CON LOS SENTIDOS E GIRO DE LOS SERVOS

El programa final quedaría de la siguiente manera:



## Nivel avanzado

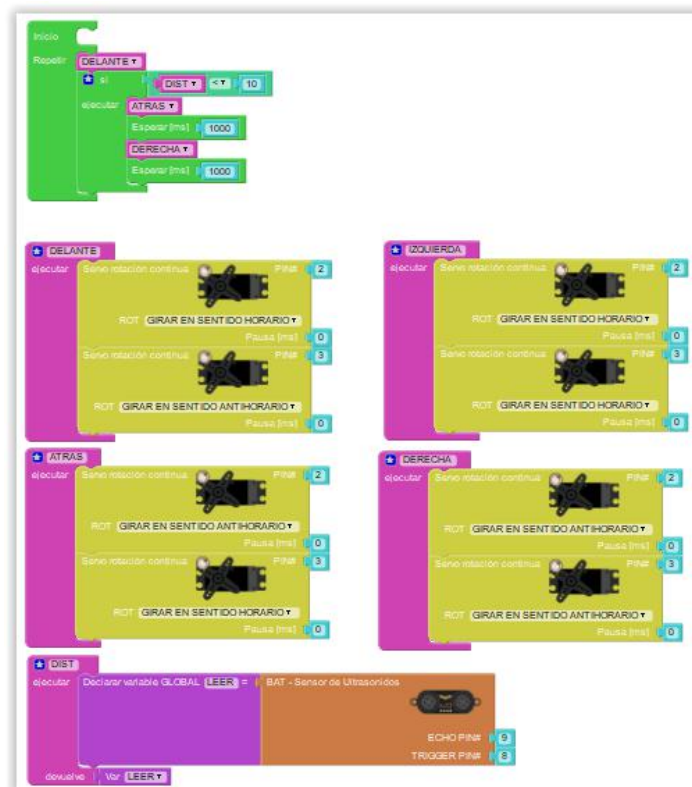
En este nivel haremos uso de diferentes sensores para una tarea más específica como es el movimiento del robot para tareas más concretas como puede ser salir de un laberinto esquivando obstáculos o seguir una línea a través de un circuito.

## Ejercicio 15 - Esquiva obstáculos

El primer ejemplo de este apartado usará la manera más sencilla de salir de un circuito de obstáculos y es mediante el uso del sensor de ultrasonidos y una serie de movimientos predefinidos. Para ello seguiremos el siguiente diagrama de flujo.



El programa quedaría de la siguiente manera:



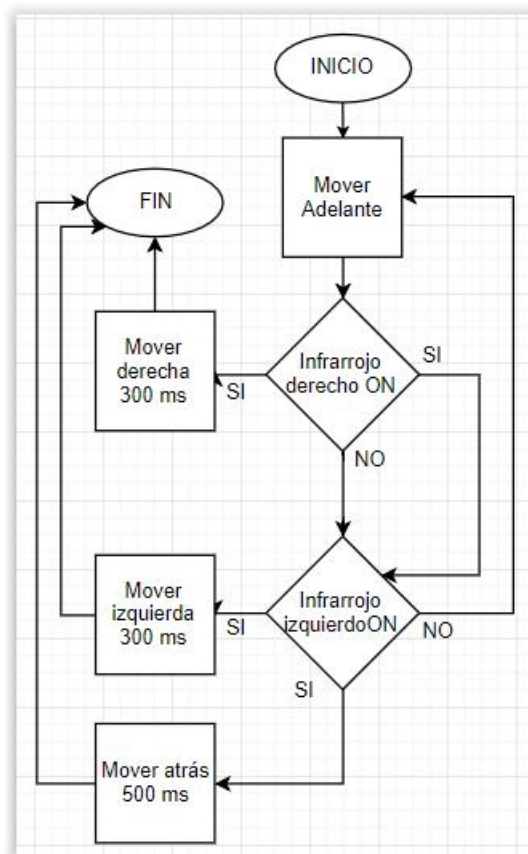


Este programa sería muy básico y necesitaríamos pensar el circuito a seguir de una manera sencilla, ya que sino podría darse el caso en el cual el robot se quedaría dando vueltas en un rincón sin salir de el. Una manera de poderlo solucionar es mediante el uso de contadores, haciendo que gire cada vez en un sentido y si debe ejecutar más de X veces el mismo giro en el siguiente giro cambie la dirección, pero eso se verá de manera más detallada en nuevos manuales y durante el curso donde usaremos el robot.

## Ejercicio 16 - Siguelineas

El último ejercicio se trata del uso más común de este tipo de robots educativos y es realizar un vehículo autónomo que pueda ser capaz de seguir una línea sin salirse del circuito. Para esta tarea usaremos los sensores de infrarrojos disponibles en la parte inferior del robot. Tendremos que controlar cual de los dos sensores es el que detecta el cambio de color para poder decidir hacia donde girar.

Si ambos sensores están detectando color blanco iremos en línea recta pero cuando el sensor derecho detecte color negro deberemos parar y realizar un giro a la derecha ya que habrá detectado la línea girando hacia la derecha. Todo esto se verá mejor al comenzar a trabajar con el robot y siguiendo el diagrama de flujo para su programación.



Como podemos ver es algo más “complejo” ya que tenemos que tener en cuenta que si ambos sensores de infrarrojos detectan el color negro deberemos pararnos e ir hacia atrás por que es posible que nos hayamos girado demasiado y se haya salido del circuito, de esta manera trataremos de retomar el mismo.