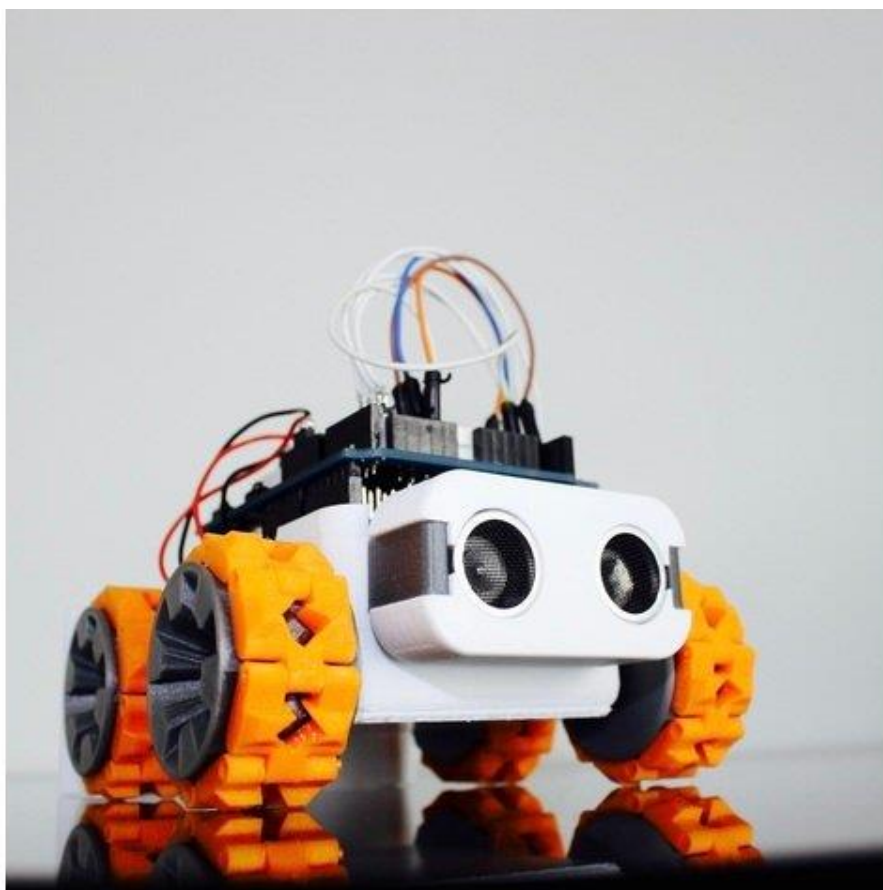




5 DE JUNIO DE 2019



## PROYECTO FINAL

IMPLEMENTACIÓN DE ROBOT MODULAR TIPO SMARS A TRAVÉS DE ARDUINO

**ARELLANO HERNÁNDEZ ERICK DANIEL**

**PÉREZ LÓPEZ LUIS AXEL**

CIRCUITOS DIGITALES

GRUPO 2805

## Tabla de contenido

Objetivo general.....	3
Introducción .....	3
Marco Teórico .....	4
Desarrollo del proyecto.....	14
Pruebas y Resultados.....	21
Conclusiones .....	22
Anexos .....	23
Referencias.....	35

## Objetivo general

Aplicar los conocimientos adquiridos a lo largo del curso de Circuitos Digitales, así como de los conocimientos ya obtenidos en otros cursos previos.

## Introducción

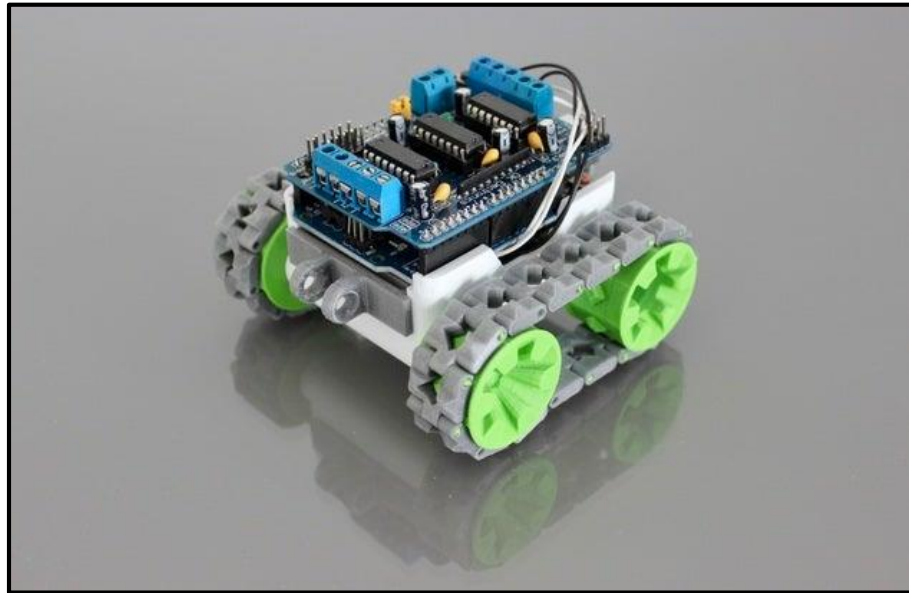
En el presente proyecto, se realizará el armado de un robot tipo SMARS, obtenido a través de la plataforma *Thingiverse*, un sitio web dedicado a compartir archivos de diseño digital creados por otros usuarios para su aplicación de forma libre. Para este proyecto específicamente, además del armado del robot se realizó la programación del mismo a través de la placa de desarrollo llamada *Arduino*.

Dicho programa realiza 5 funciones diferentes, organizadas en dos tipos diferentes de control: automático y manual. Como se mencionó antes, la placa utilizada fue el Arduino Uno, programada por medio de su propio IDE. Para la parte del robot, las partes plásticas del robot se imprimieron mediante una impresora 3D; de igual manera, para la parte electrónica se utilizaron los materiales propuestos en el modelo de Thingiverse, además de los componentes electrónicos necesarios para cada una de las funciones que va a realizar el robot. Los materiales base del modelo son los siguientes:

- 2 motorreductores tipo Pololu de 150 RPM
- 1 batería de 9V (de preferencia recargable)
- 1 conector para la batería de 9V
- 1 Arduino Uno
- 1 controlador para motores Adafruit

Además del robot, el proyecto también consta de una aplicación para dispositivos móviles, la cual fue programada por medio de la plataforma *App inventor*. Esta aplicación otorgará la posibilidad de conectar un dispositivo móvil al robot por medio de bluetooth para poder controlarlo de manera remota. Esta aplicación

se encargará de seleccionar la función que realizará el robot, enviando la señal correspondiente al Arduino.



**Fig. 1**

Modelo armado, mostrado en la plataforma de *Thingiverse*.

## Marco Teórico

### 1. Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Para su programación, utiliza el lenguaje de programación Arduino (basado en Wiring ) y el software Arduino (IDE) , basado en el procesamiento.

A lo largo de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta instrumentos científicos complejos. Una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores y profesionales) se ha reunido en torno a esta plataforma de código abierto, sus contribuciones han sumado una cantidad increíble de conocimientos accesibles que pueden ser de gran ayuda para principiantes y expertos por igual.

Algunas de las ventajas que tiene utilizar Arduino son las siguientes:

- **Barato:** las tarjetas Arduino son relativamente económicas en comparación con otros microcontroladores. La versión más económica del módulo Arduino se puede ensamblar a mano, e incluso los módulos Arduino premontados tienen un costo bajo.
- **Multiplataforma:** el software de Arduino (IDE) se ejecuta en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontroladores están limitados a Windows.
- **Entorno de programación simple y claro:** el software Arduino (IDE) es fácil de usar para los principiantes, pero es lo suficientemente flexible como para que los usuarios avanzados también lo aprovechen.
- **Software de código abierto y extensible:** el software Arduino se publica como herramienta de código abierto, disponible para la extensión por programadores experimentados. El lenguaje se puede expandir a través de las bibliotecas de C ++, y las personas que deseen comprender los detalles técnicos pueden dar el salto de Arduino al lenguaje de programación AVR C en el que se basa.
- **Fuente abierta y hardware extensible:** los planes de las tarjetas Arduino se publican bajo una licencia de Creative Commons, por lo que los diseñadores de circuitos experimentados pueden crear su propia versión del módulo, ampliarlo y mejorarlo.

La placa utilizada para este proyecto es Arduino Uno, cuyas características técnicas son las siguientes:

Microcontrolador	ATmega328P
Tensión de funcionamiento	5 V
Voltaje de entrada (recomendado)	7 – 12 V
Voltaje de entrada (límite)	6 – 20 V
Pines digitales I/O	14
Pines digitales I/O PWM	6
Pines de entrada analógica	6
Corriente DC por pin I/O	20 mA

Corriente DC para el pin de 3.3 V	60 mA
Memoria flash	32 Kb
SRAM	2 Kb
EEPROM	1 Kb
Velocidad de reloj	16 MHz

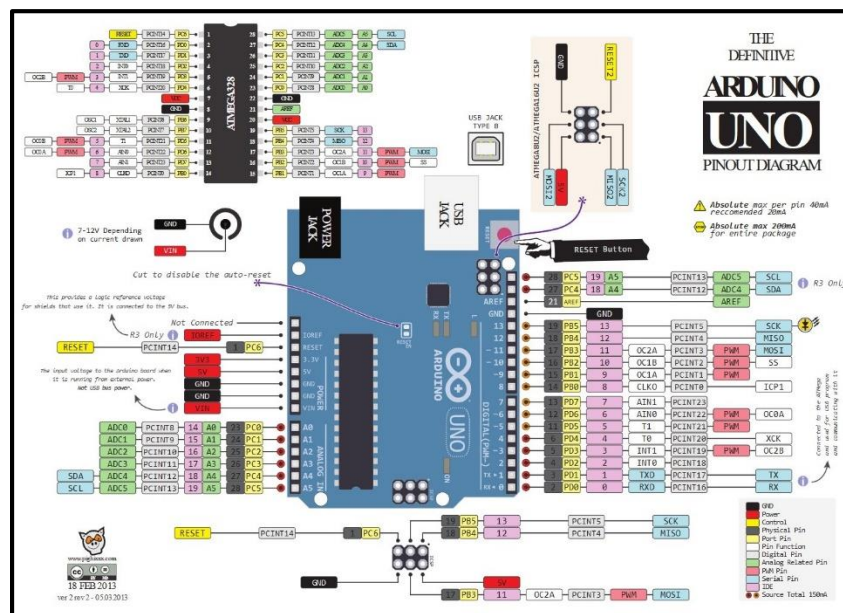


Fig. 2

Distribución de pines del Arduino Uno.

## 2. App Inventor 2

App Inventor 2 (AI2) es la versión mejorada de una herramienta de programación creada por el MIT (Instituto Tecnológico de Massachusetts) y que fue adoptada por Google para sus usuarios como solución para crear de una forma sencilla aplicaciones para dispositivos Android. El proceso de creación consta de 3 pasos:

1. **Diseñador.** Muestra el display de un móvil y se utiliza para el diseño de las pantallas de la aplicación donde se situarán los distintos componentes: imágenes, botones audios, textos, etc configurando sus propiedades (aspecto gráfico, comportamiento, etc).

2. **Editor de bloques.** Permite programar de una forma visual e intuitiva el flujo de funcionamiento del programa utilizando bloques.
3. **Generador de la aplicación.** Una vez terminada la aplicación se puede generar el instalador APK obteniéndose un código QR para su descarga desde el móvil o bien el propio archivo APK para descargar y enviar.

AI2 proporciona una herramienta en línea accesible a través de un navegador web si se dispone de una cuenta de usuario en Google. El equipo recomendado es un ordenador PC (Windows, Mac o Linux) – no una tableta o un móvil- y el navegador recomendado es la última versión de Google Chrome o Mozilla Firefox (Internet Explorer no está soportado). No es necesario tener instalado en el equipo Java ni ningún otro programa.

Gracias a esta plataforma, es posible crear una gran cantidad de aplicaciones móviles simples de una manera sencilla. También, con los elementos que posee se pueden establecer conexiones inalámbricas, reproducir multimedia, anexas imágenes, etc.



Fig. 3

Logotipo de la plataforma *App Inventor 2*.

### 3. Sensor ultrasónico

Otro de los dispositivos utilizados en el proyecto es el sensor ultrasónico HC-SR04, el cual funciona de la siguiente manera:

1. El altavoz emite un sonido de frecuencia ultrasónica (imperceptible al oído humano).
2. Se pone un cronómetro en marcha.
3. El micrófono recibe la señal que había sido emitida por el altavoz.
4. Se para el cronómetro que se había puesto en marcha al emitir el sonido
5. Se calcula, sabiendo la velocidad del sonido, la distancia a la que está el objeto sobre el cual ha rebotado la pulsación ultrasónica.

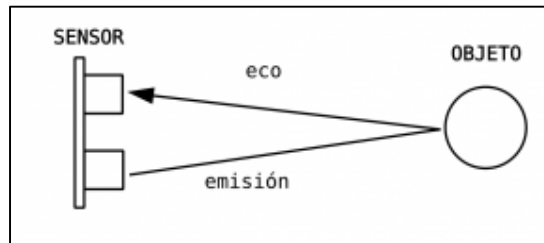


Fig. 4

Funcionamiento del sensor ultrasónico.

Para los cálculos realizados por este sensor, se utilizan las siguientes fórmulas:

$$\text{Tiempo} = 2 \left( \frac{\text{Distancia}}{\text{Velocidad}} \right)$$

Por lo que:

$$\text{Distancia} = \left( \frac{\text{Tiempo} * \text{Velocidad}}{2} \right)$$

Este sensor consta de 4 pines, alimentación (Vcc), un disparador (Trig), el receptor (Echo) y masa (GND). Para el correcto funcionamiento del sensor es necesario el conectar estos 4 pines. Las características básicas de este sensor y que se deben de tener en cuenta al trabajar con él se muestran en la Fig. 5:



Características	
Alimentación	+5v DC
Frecuencia de trabajo	40 KHz
Consumo (suspendido)	< 2mA
Consumo (trabajando)	15mA
Ángulo efectivo	< 15°
Distancia	2cm a 400cm *
Resolución	0.3 cm

\*A partir de 250cm la resolución no es buena



Fig. 5

Tabla de características del sensor ultrasónico (izquierda).

Fotografía del sensor utilizado (derecha).

Su principio de operación se basa en la emisión, por medio de una bocina, de trenes de pulsos a una frecuencia de 40 khz, y la detección a través del micrófono de la señal acústica rebotada por el objeto. Este sensor es utilizado para realizar la acción de *Evasor de Obstáculos*.

## 4. Sensor infrarrojo

Los sensores infrarrojos son unos componentes electrónicos compuestos normalmente de un LED infrarrojo y un fototransistor colocados uno al lado del otro, de forma que el LED actúa como emisor y el fototransistor como receptor. El LED infrarrojo emite luz infrarroja, o sea, de mayor longitud de onda (o menor frecuencia) que la podemos ver los humanos, así que para nosotros es invisible. Si esta luz choca contra una superficie blanca se reflejará y llegará al fototransistor. Si por el contrario golpea en una superficie negra, el material absorberá la mayoría de la luz y no llegará al fotoreceptor.

El módulo utilizado tanto para la función de seguidor de línea como para la función de quedarse dentro de un perímetro se muestra en la Fig. 6. Este modelo de sensor es el *Módulo De Sensor De Línea De Obstáculo De Infrarrojos De 4 Vías YL-70*. Este módulo está diseñado para dispositivos mecánicos automatizados, como autos inteligentes y robots, para proporcionar una solución de sistema de detección infrarroja multifuncional.

El módulo está compuesto por una tarjeta que incorpora el circuito integrado LM339 (YL-70), y 4 sensores que poseen emisor y receptor LED Infrarrojo (YL-73), los cuales permiten detectar variaciones de color entre blanco y negro. El módulo tiene varias capacidades de detección para satisfacer una gran variedad de aplicaciones de sistemas pequeños inteligentes e automatizadas.

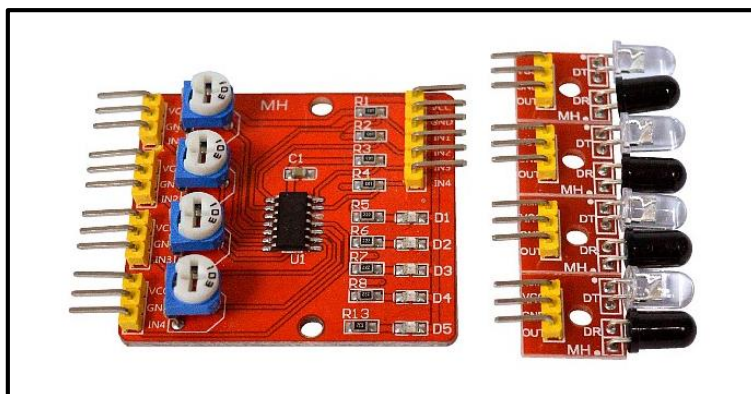


Fig. 6

Módulo utilizado para el seguidor de línea y para mantener el robot dentro de un perímetro.

Hay dos pruebas para comprobar el funcionamiento de este tipo de módulo:

- **Para la sonda:**

Retire todos los objetos frente a la sonda y no permita que la sonda apunte hacia la dirección del sol. Utilice un multímetro para medir el voltaje de salida después de alimentar la placa de la sonda, el valor debe ser de alrededor de 1v. Coloque papel blanco frente a la sonda, el voltaje de salida debe estar cerca del voltaje de la fuente de alimentación.

- **Para el módulo:**

Retire todos los objetos en frente de la sonda y no permita que la sonda apunte hacia la dirección del sol. Luego, conecte la sonda probada a la tarjeta, use un multímetro para medir el voltaje de salida después de encender la tarjeta de control, el valor debe estar cerca del voltaje de la fuente de alimentación. Coloque papel blanco delante de la sonda, el voltaje de salida debe estar alrededor de 0v.

Las características de este módulo son las presentadas a continuación:

Tensión de alimentación	3.3 – 5 V
Corriente de funcionamiento	150 mA (aproximadamente)
Temperatura de funcionamiento	-10 ~ +50 °C
Rango de detección LED IR	1 – 50 mm (ajustable)
Señal de salida	TTL
Tipo de salida	Digital ON - OFF

## 5. Sensor fotoeléctrico

El LDR, por sus siglas en inglés (Light Dependent Resistor), o fotoresistor es una resistencia que varía su valor en función de la cantidad de luz que incide sobre su superficie. Cuanto mayor sea la intensidad de luz que incide en la superficie del LDR o fotoresistor menor será su resistencia y en cuanto menor sea la luz que incida sobre éste mayor será su resistencia.

Cuando el LDR(fotoreistor) no está expuesto a radiaciones luminosas, los electrones están firmemente unidos en los átomos que lo conforman, pero cuando sobre él inciden radiaciones luminosas, esta energía libera electrones con lo cual el material se hace más conductor, y de esta manera disminuye su resistencia. Las resistencias LDR solamente reducen su resistencia con una radiación luminosa situada dentro de una determinada banda de longitudes de onda.

El módulo que se utilizó en el proyecto se muestra en la Fig. 7. Este módulo posee 2 salidas, una analógica que se debe conectar a una entrada analógica, utilizando el conversor ADC. La salida digital posee solo 2 estados: activo/apagado, el cambio de un estado a otro depende del umbral que se fije con el potenciómetro del módulo. La salida digital puede utilizarse para controlar un relay y así realizar una acción dependiente de la intensidad de luz.

Las características técnicas de este tipo de sensor son las siguientes:

- Voltaje de Operación: 5V DC
- Conexión de 4 cables: VCC, GND, DO, AO
- Salida analógica y digital(comparador)

- Opamp en modo comparador (LM393)
- Potenciómetro para ajuste de comparador
- Led rojo de encendido y verde de salida digital

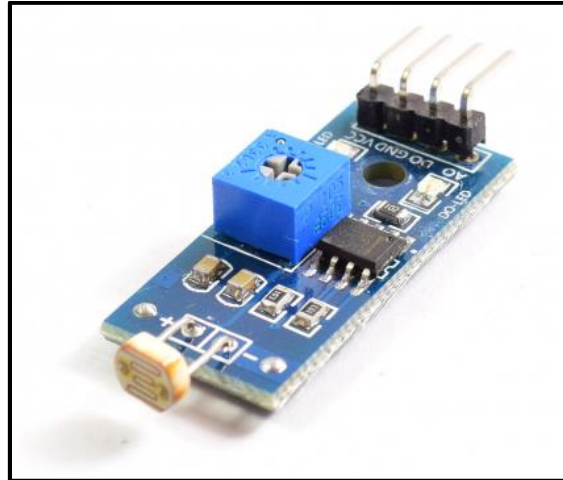


Fig. 7

Módulo LDR utilizado para el seguidor de luz.

## 6. Módulo Bluetooth

El módulo HC-06 es un módulo esclavo. No puede conectarse a otros dispositivos por sí mismo. Para eso, necesitará un módulo maestro como el módulo Bluetooth HC-05. Con este módulo se puede conectar el Arduino a otro sistema (teléfono inteligente, computadora u otros microcontroladores), para enviar y recibir datos. La comunicación Bluetooth permite controlar el robot a través de la aplicación para Smartphone.

El módulo Bluetooth HC-06 tiene 4 pins para establecer la conexión:

- **Alimentación VCC.** Normalmente conectado al pin 5V del Arduino.
- **Masa GND.** Normalmente conectado al pin GND del Arduino.
- **Pin de recepción RX.** Normalmente conectado al pin de transmisión Arduino (TX).
- **Pin de transmisión TX.** Normalmente conectado al pin de recepción Arduino (RX)

Este módulo se muestra en la Fig. 8. Sus características técnicas se muestran a continuación:

- Funciona como dispositivo esclavo bluetooth
- Configurable mediante comandos AT
- Bluetooth V2.0+EDR
- Frecuencia de operación: 2.4 GHz Banda ISM
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Potencia de transmisión:  $\leq 4\text{dBm}$ , Class 2
- Seguridad: Autenticación y Encriptación
- Perfiles Bluetooth: Puerto serie bluetooth.

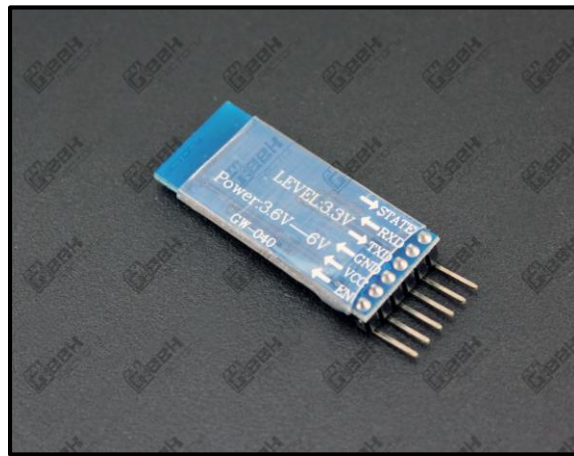


Fig. 8

Módulo bluetooth HC-06.

## 7. Driver para motores

El módulo que utilizamos en el proyecto fue el puente H *MX1616*, este módulo es una tarjeta para el control de motores de corriente directa, motores a pasos, solenoides y, en general, cualquier otra carga inductiva, se usa para controlar dos motores de corriente continua al mismo tiempo o un motor paso a paso, rotando de forma horaria o anti-horaria (ambos sentidos).

Es ideal para controlar motores DC en aplicaciones de robótica, permite controlar el sentido de funcionamiento de motores a una corriente de salida por cada canal de hasta 1.5A (por motor DC); cuenta con un sistema que detecta el sobrecarga.

lentamiento y apaga; presenta un rendimiento anti-interferencia excepcional; incorpora un circuito de protección térmica con efectos de histéresis (TSD), para que no haya que preocuparse por la parada del motor.

Las características técnicas de este controlador son las siguientes:

- Integrado de superficie L298/ Puente H Dual.
- Tensión de accionamiento 2V-10V.
- Corriente máxima: 1.5A por canal.
- Voltaje de control: de 1.8V-7V.
- Corriente de espera menor de: 0.1  $\mu$ A
- Temperatura de operación: -20°C a 135°C

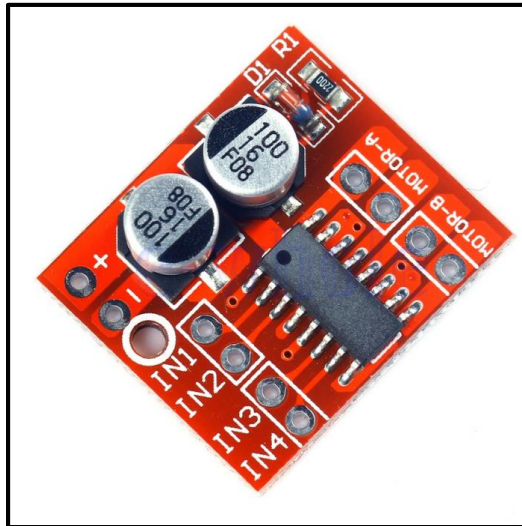


Fig. 9

Puente H utilizado para mover a los motores.

## Desarrollo del proyecto

Para el desarrollo de este proyecto, los materiales necesarios son los siguientes:

- Robot SMARS, impreso mediante una impresora 3D) conseguido a través de la plataforma *Thingiverse*).
- 1 Arduino Uno

- 2 motorreductores tipo Pololu de 150 RPM
- 1 batería de 9V (de preferencia recargable)
- 1 conector para la batería de 9V
- 1 controlador para motores Adafruit
- 2 módulos LDR
- 2 sensores infrarrojos
- 1 sensor ultrasónico
- 1 módulo bluetooth
- 1 dispositivo móvil compatible con la aplicación realizada

### Módulo Seguidor de Luz

Para la parte del seguidor de luz, utilizamos un sensor LDR en cada lado del robot. Cada uno de estos módulos se encarga de captar la luz y enviar la señal al Arduino. Dependiendo de la señal que envíe, el Arduino, mediante programación, se encargará de darle a los motores la energía necesaria para que pueda moverse.

En el caso de que uno de los sensores capte más luz que el otro, el Arduino se encarga de regular la velocidad de los motores, para que ambos motores funcionen de una manera similar. El diagrama de flujo de este módulo se muestra en la Fig. 11.

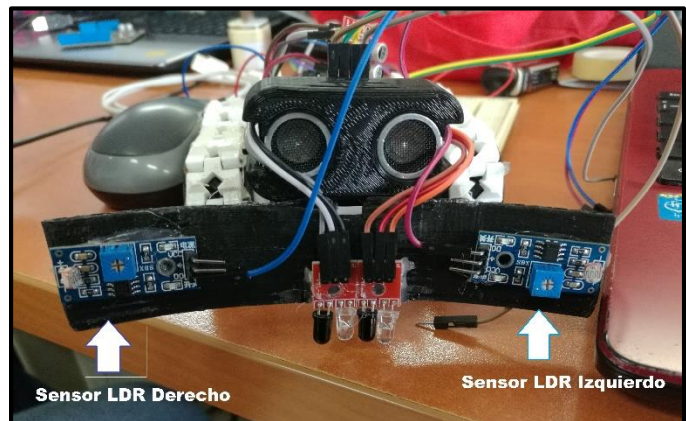
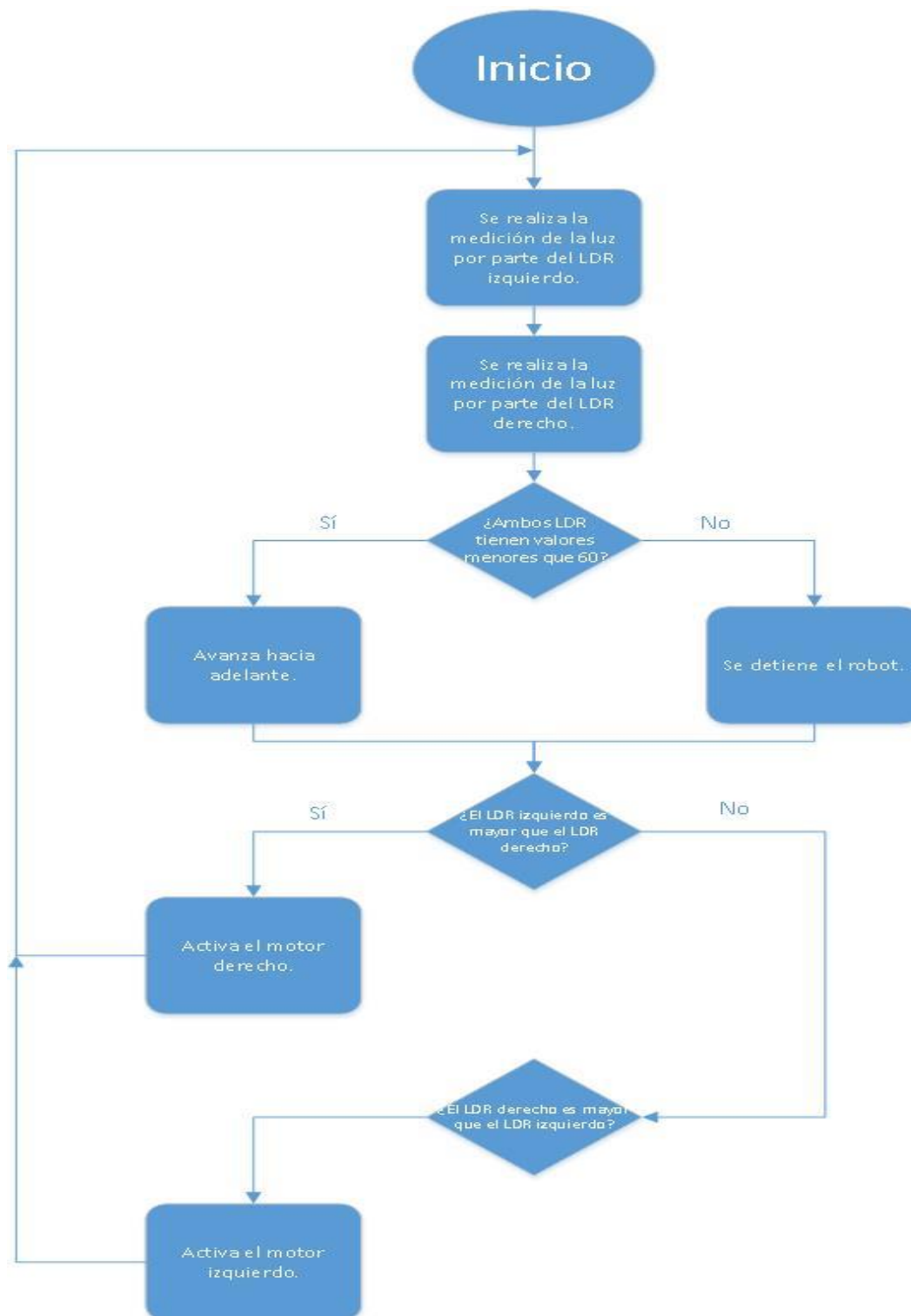


Fig. 10

Imagen de los sensores montados para el funcionamiento como seguidor de luz.





**Fig. 11**

Diagrama de flujo del módulo seguidor de luz.



## Módulo Seguidor de línea

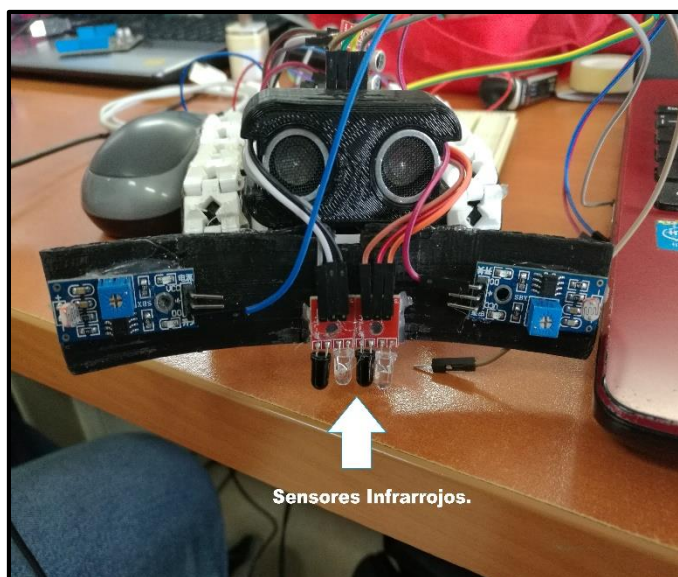


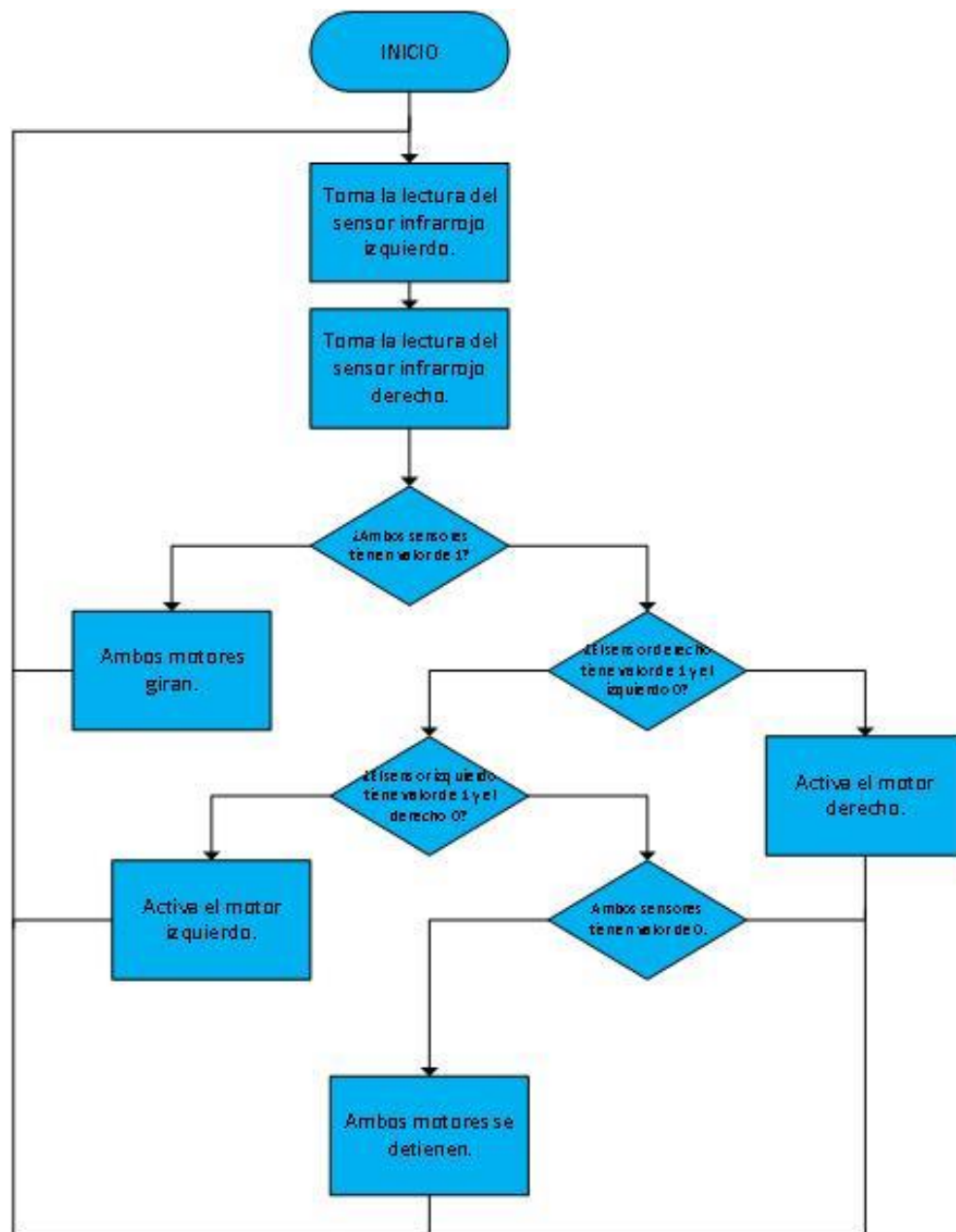
Fig. 12

Sensores infrarrojos instalados para el seguidor de línea.

Para este módulo, se utilizaron los sensores infrarrojos, los cuales permiten obtener una señal digital acerca de lo que obtienen. Para nuestro caso, el 1 lógico significa que los sensores no detectan un objeto sobre el cual chocar la señal, lo que implica que se encuentra fuera de la línea blanca. En caso contrario, enviará un 0 cuando se encuentren sobre la línea blanca.

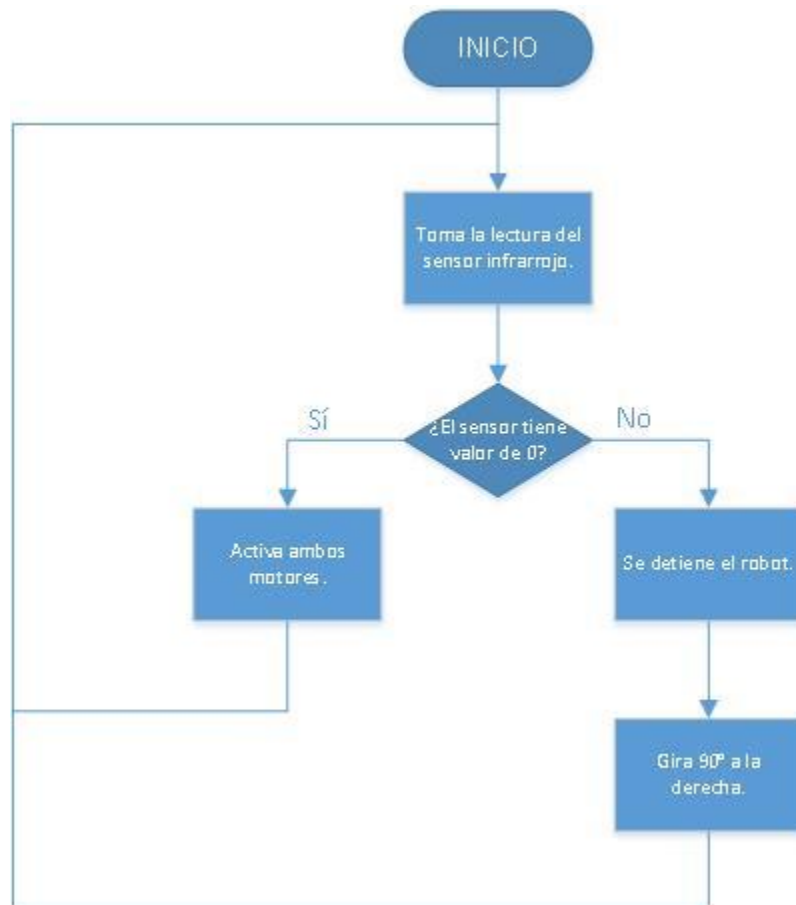
Teniendo esto en cuenta, cuando el sensor infrarrojo derecho envíe un 0 y el sensor infrarrojo izquierdo mande un 0, el motor derecho se parará, mientras que el izquierdo se moverá. En caso de que el sensor derecho mande un 1 y el izquierdo envíe un 0, el motor derecho se moverá y el izquierdo se detendrá.

Este módulo también se utilizó para realizar una función que mantenga el carrito dentro de un perímetro establecido. En este caso, al enviar la señal, el carro girará 90 grados y realizará una nueva lectura. Los diagramas de flujo para estos módulos se muestran en la Fig. 13 y en la Fig. 14.



**Fig. 13**

Diagrama de flujo del módulo seguidor de línea.



**Fig. 14**

Diagrama de flujo para la función de mantenerse en el perímetro.

## Módulo Evasor de Objetos

Para este módulo, utilizamos el sensor ultrasónico HC-SR04. Este módulo, detecta la distancia que hay hacia un objeto. Para este proyecto, la distancia que se tomó es de 15 cm.

Al detectar que la distancia es igual o menor a estos 15 cm, el robot se detiene y gira hacia la izquierda. Después, vuelve a repetir esta acción hasta que la distancia sea mayor a estos 15 cm, lo cual significaría que no hay obstáculo cercano al robot.

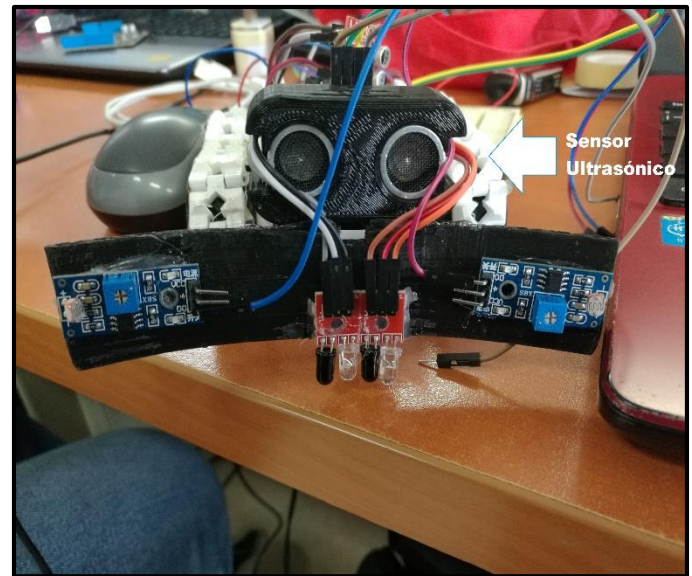


Fig. 15

Sensor ultrasónico instalado en el robot.

El diagrama de flujo para esta etapa se muestra en la Fig. 16.

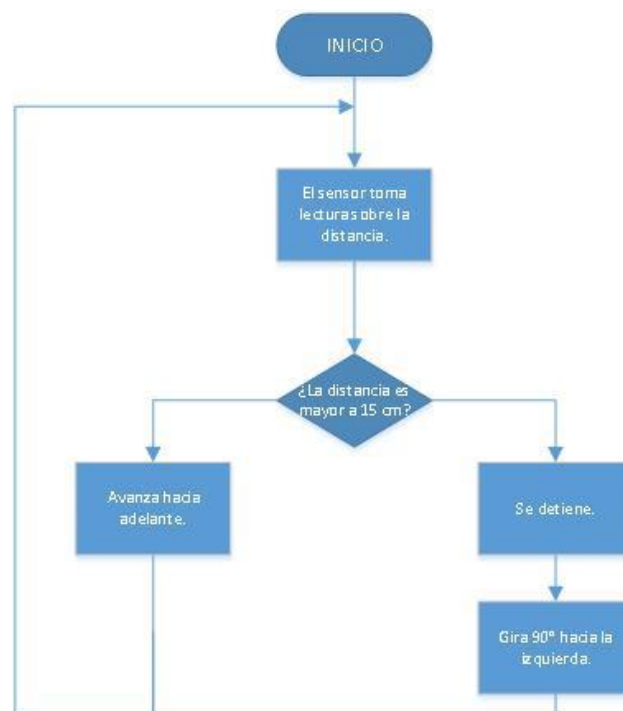


Fig. 16

Diagrama de bloques del módulo evasor de objetos.

## Pruebas y Resultados

Para este proyecto, es necesario descargar la aplicación creada en App Inventor. En esta aplicación en primera instancia, aparecerá el botón para conectar con el módulo bluetooth. Una vez conectado, aparecerán los botones para seleccionar el modo que se va a utilizar: automático o manual.

Dentro de los programas automáticos, aparecerá el módulo a elegir: seguidor de luz, seguidor de línea, evasor de obstáculos y permanencia en perímetro. En el modo manual, aparecerán unas flechas indicando la dirección hacia la cual se quiere mover al carrito, así como un botón de stop, que detendrá al robot. Las imágenes acerca de esta aplicación, se muestran en la Fig. 17.

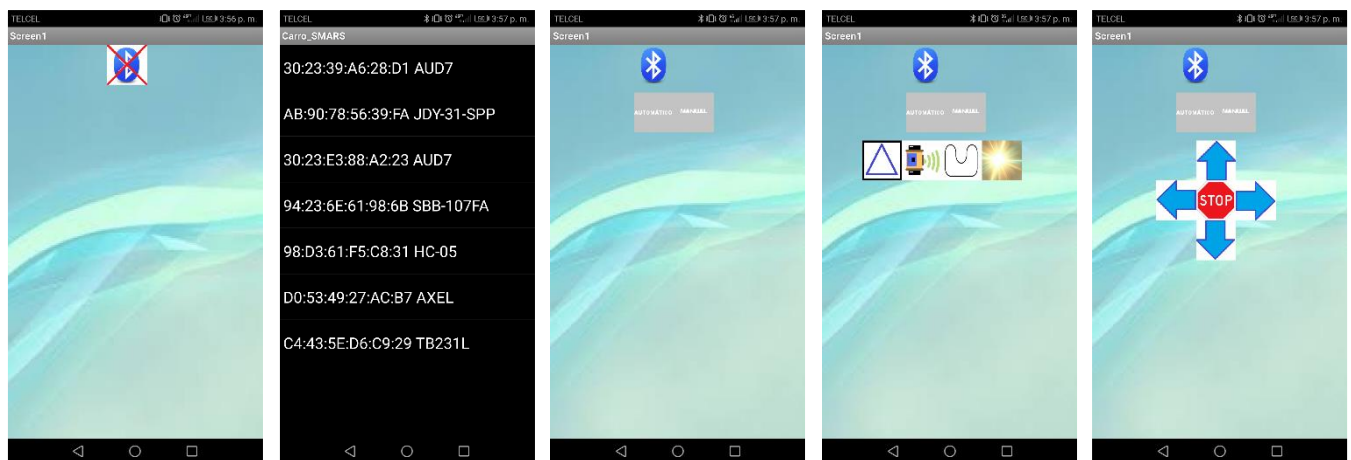


Fig. 17

Capturas, por partes, de la aplicación móvil.

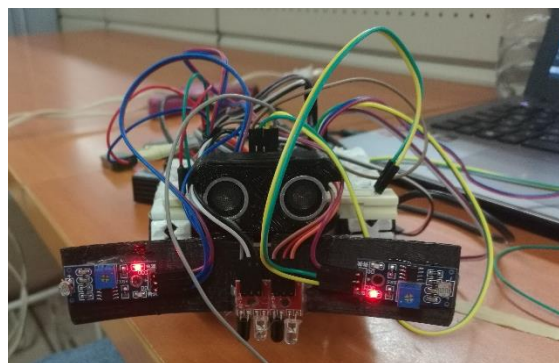


Fig. 18

Robot ensamblado y encendido.

## Conclusiones

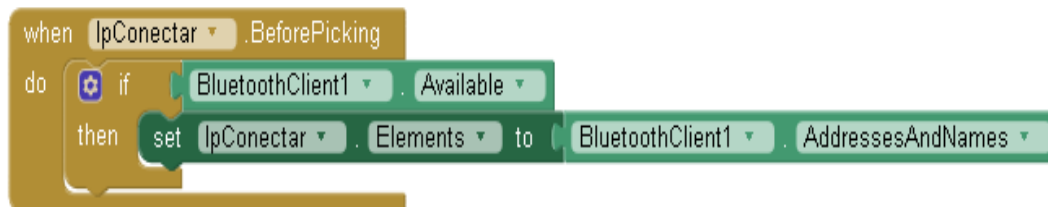
Este proyecto nos sirvió para aplicar los conocimientos sobre electrónica y programación que hemos adquirido a lo largo de la carrera de Ingeniería Eléctrica y Electrónica, en especial, en la materia de Circuitos Digitales. Pudimos darnos cuenta de que existen plataformas en Internet que pueden proporcionar diseños o proyectos de manera abierta, como en este caso con el robot SMARS. Sin embargo, a pesar de que el proyecto ya está diseñado, notamos que es más complejo que solamente imprimirlo y armarlo.

Aunque nos basamos en el proyecto ya mencionado, la parte del software la implementamos de diferente forma, lo que lo volvió aún más complicado. De igual forma, lo sencillo no fue programar las funciones por separado, ya que de esa manera los programas funcionaban de manera sencilla y, en caso de notar algún error, era fácil de identificar y corregir. Sin embargo, el reto para nosotros, vino al momento de juntar todos los programas en un solo, ya que se necesitaba hacer que los programas concordaran en cuanto a la declaración de variables y, cuando había un error, era más complicado ubicarlo.

Algo que nos fue de gran ayuda fue el hecho de que todos los materiales necesarios para el funcionamiento del robot nos fueran proporcionados por el Dr. Ismael, ya que esto redujo el costo del robot por parte de nosotros y evitó el tener que conseguir los materiales.

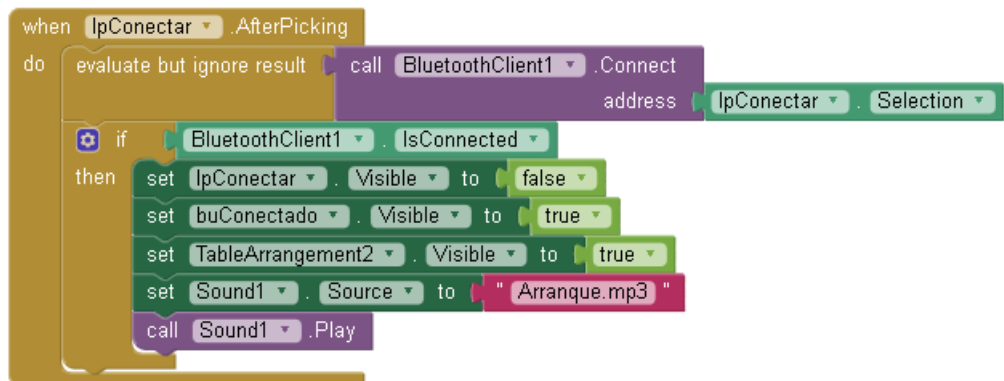
## Anexos

### I. Bloques de la programación en App Inventor



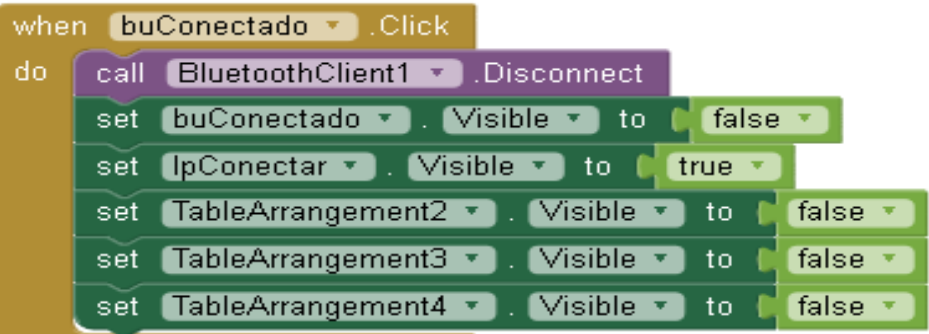
```
when IpConectar .BeforePicking
do
  if BluetoothClient1 . Available
  then
    set IpConectar . Elements to BluetoothClient1 . AddressesAndNames
```

Scratch code block for the `IpConectar .BeforePicking` event. The code contains an `if` block checking if `BluetoothClient1 . Available`. If true, it sets `IpConectar . Elements` to `BluetoothClient1 . AddressesAndNames`.



```
when IpConectar .AfterPicking
do
  evaluate but ignore result call BluetoothClient1 .Connect
  address IpConectar . Selection
  if BluetoothClient1 . IsConnected
  then
    set IpConectar . Visible to false
    set buConectado . Visible to true
    set TableArrangement2 . Visible to true
    set Sound1 . Source to "Arranque.mp3"
    call Sound1 .Play
```

Scratch code block for the `IpConectar .AfterPicking` event. The code first calls `BluetoothClient1 .Connect` with the address `IpConectar . Selection`. Then, it checks if `BluetoothClient1 . IsConnected`. If true, it sets `IpConectar . Visible` to `false`, `buConectado . Visible` to `true`, `TableArrangement2 . Visible` to `true`, sets `Sound1 . Source` to `"Arranque.mp3"`, and calls `Sound1 .Play`.



```
when buConectado .Click
do
  call BluetoothClient1 .Disconnect
  set buConectado . Visible to false
  set IpConectar . Visible to true
  set TableArrangement2 . Visible to false
  set TableArrangement3 . Visible to false
  set TableArrangement4 . Visible to false
```

Scratch code block for the `buConectado .Click` event. The code calls `BluetoothClient1 .Disconnect`, sets `buConectado . Visible` to `false`, `IpConectar . Visible` to `true`, and sets `TableArrangement2 . Visible`, `TableArrangement3 . Visible`, and `TableArrangement4 . Visible` all to `false`.

```
when buAutomatico .Click
do
  set TableArrangement3 . Visible to true
  set TableArrangement4 . Visible to false
```

```
when buManual .Click
do
  set TableArrangement3 . Visible to false
  set TableArrangement4 . Visible to true
```

```
when buObstaculos .Click
do
  call BluetoothClient1 .SendText
  text " a "
```

```
when buPerimetro .Click
do
  call BluetoothClient1 .SendText
  text " b "
```

```
when buLinea .Click
do
  call BluetoothClient1 .SendText
  text " c "
```



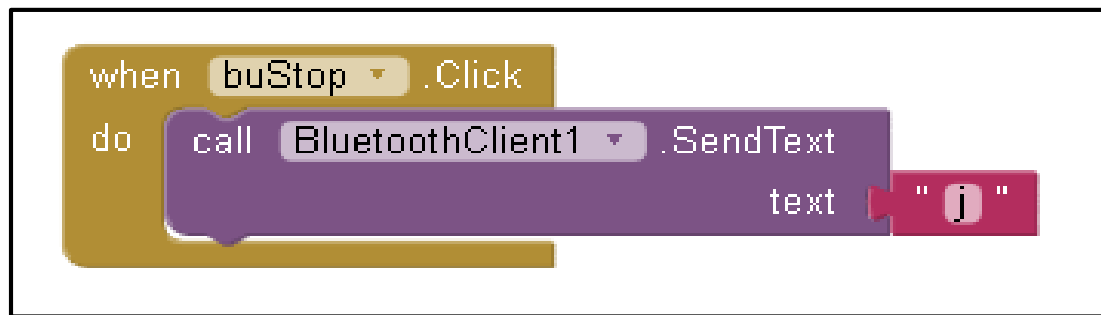
```
when buLuz ▾ .Click  
do call BluetoothClient1 ▾ .SendText  
text " d "
```

```
when buArriba ▾ .Click  
do call BluetoothClient1 ▾ .SendText  
text " e "
```

```
when buAbajo ▾ .Click  
do call BluetoothClient1 ▾ .SendText  
text " f "
```

```
when buDerecha ▾ .Click  
do call BluetoothClient1 ▾ .SendText  
text " g "
```

```
when bulzquierda ▾ .Click  
do call BluetoothClient1 ▾ .SendText  
text " i "
```



## II. Código de Arduino utilizado para el robot

```
////////Perimetro////////
int Sens = 0;      //
////////Sonico////////
const int EchoPin = 3; //
const int TriggerPin = 2;//
////////Linea////////
int sDerIR = 0;
int slzqIR = 0;

char dato;
void setup() {
  /////LDR/////
  pinMode(A5,0);
  pinMode(A4,0);
  /////Perimetro y linea///
  pinMode(5,0);
  pinMode(4,0);
  /////Sonico/////
  pinMode(TriggerPin, 1);
  pinMode(EchoPin, 0);
  /////Motores////////
  pinMode(13,1);
  pinMode(12,1);
  pinMode(8,1);
  pinMode(7,1);
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  if (Serial.available() >0) {
```

```
    dato = Serial.read();
```

```
    Serial.print(dato);
```

```
  }
```

```
  switch (dato) {
```

```
    case 'a':
```

```
      sSonic();
```

```
      break;
```

```
    case 'b':
```

```
      perimetro();
```

```
      break;
```

```
    case 'c':
```

```
      Sline();
```

```
      break;
```

```
    case 'd':
```

```
      Sluz();
```

```
      break;
```

```
    case 'e':
```

```
      front();
```

```
      break;
```

```
    case 'f':
```

```
        back();  
        break;  
  
        case 'g':  
            right();  
            break;  
  
        case 'i':  
            left();  
            break;  
  
        case 'j':  
            stopp();  
            break;  
    }  
}  
  
int dis() {  
    long duration, distanceCm;  
  
    digitalWrite(TriplePin, 0);  
    delayMicroseconds(4);  
    digitalWrite(TriplePin, 1);  
    delayMicroseconds(10);  
    digitalWrite(TriplePin, 0);  
  
    duration = pulseIn(EchoPin, 1);  
  
    distanceCm = duration * 10 / 292 / 2;  
    return distanceCm;  
}
```

```

}

void sSonic(){
    int d=dis;
    if (d <=15){
        stopp();
        left();
        delay(600);
        stopp();
        front();
    }
    else{
        front();
    }
}

void Sluz() {
    boolean Dato = digitalRead(2);
    boolean Dato1 = digitalRead(4);
    if (Dato == HIGH && Dato1 == HIGH) {
        stopp();
    }
    else if (Dato == LOW && Dato1 == HIGH) {
        left();
    }
    else if (Dato == HIGH && Dato1 == LOW) {
        right();
    }
    else if (Dato == LOW && Dato1 == LOW) {
        front();
    }
}

```

```

    }
}

void Sline() {
    Serial.print("\t Digital Reading=");
    sDerIR = digitalRead(4);
    slzqIR = digitalRead(5);
    if (sDerIR == 0 && slzqIR == 0) {
        stopp();
    }
    if (sDerIR == 1 && slzqIR == 0) {
        front();
    }
    if (sDerIR == 0 && slzqIR == 1) {
        left();
    }
    if (sDerIR == 1 && slzqIR == 1) {
        front();
    }
}

```

```

void perimetro() {
    Serial.print("\t Digital Reading=");
    sDerIR = digitalRead(4);
    slzqIR = digitalRead(5);

    if (sDerIR == 1 && slzqIR == 1) {
        stopp();
    }
}

```

```

    right();
}
else if (sDerlR == 0 && slzqlR == 0) {
    front();
}
else if (sDerlR == 0 && slzqlR == 1) {
    stopp();
    back();
    right();
}
else if (sDerlR == 1 && slzqlR == 0) {
    stopp();
    back();
    left();
}
}

void front(){
    //Motor 1
    digitalWrite(13,1);
    digitalWrite(12,0);
    //Motor2
    digitalWrite(8,1);
    digitalWrite(7,0);
    delay(200);
}

void back(){
    //Motor 1
    digitalWrite(13,0);
    digitalWrite(12,1);

```



```
//Motor2
digitalWrite(8,0);
digitalWrite(7,1);
delay(200);

}

void right(){
  //Motor 1
  digitalWrite(13,1);
  digitalWrite(12,0);
  //Motor2
  digitalWrite(8,0);
  digitalWrite(7,1);
  delay(200);
}

void left(){
  //Motor 1
  digitalWrite(13,0);
  digitalWrite(12,1);
  //Motor2
  digitalWrite(8,1);
  digitalWrite(7,0);
  delay(200);
}

void stopp(){
  //Motor 1
  digitalWrite(13,0);
  digitalWrite(12,0);
  //Motor2
```

```
digitalWrite(8,0);  
digitalWrite(7,0);  
delay(200);  
}
```

## Referencias

Arduino Uno R3. (s.f.). Extraído el 4 de junio de 2019 desde <https://www.infoo-tec.net/arduino/>

Diosdado, R. (s.f.). Sensor de ultrasonidos HC-SR04. Extraído el 22 de mayo de 2019 desde <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>

Diseño de apps con MIT App Inventor 2. (2015). Extraído el 4 de junio de 2019 desde <http://repositorio.puce.edu.ec/bitstream/handle/22000/13113/Anexo%203%20MIT%20App%20Inventor%202.pdf?sequence=4&isAllowed=y>

<http://www.maxelectronica.cl/luz-color/374-modulo-seguidor-de-linea-ajustable-de-4-canales-modelos-yl-70-e-yl-73.html>

<https://einstronic.com/store/sensor/infrared/yl-70-4-way-infrared-obstacle-line-sensor-module/>

<https://naylampmechatronics.com/sensores-luz-y-sonido/135-modulo-ldr.html>

[https://www.mecatronicalatam.com/resistencia/fotoresistor#que\\_es\\_la\\_fotoresistencia\\_o\\_ldr=](https://www.mecatronicalatam.com/resistencia/fotoresistor#que_es_la_fotoresistencia_o_ldr=)

Tristomietitoredeituit. (2017, 22 de noviembre). SMARS modular robot. Extraído el 4 de junio de 2019 desde <https://www.thingiverse.com/thing:2662828>

Xukyo. (2018, 12 de octubre). Comunicación con Arduino y el módulo HC-06. Extraído el 4 de junio de 2019 desde <https://www.aranacorp.com/es/comunicacion-con-arduino-y-el-modulo-hc-06/>