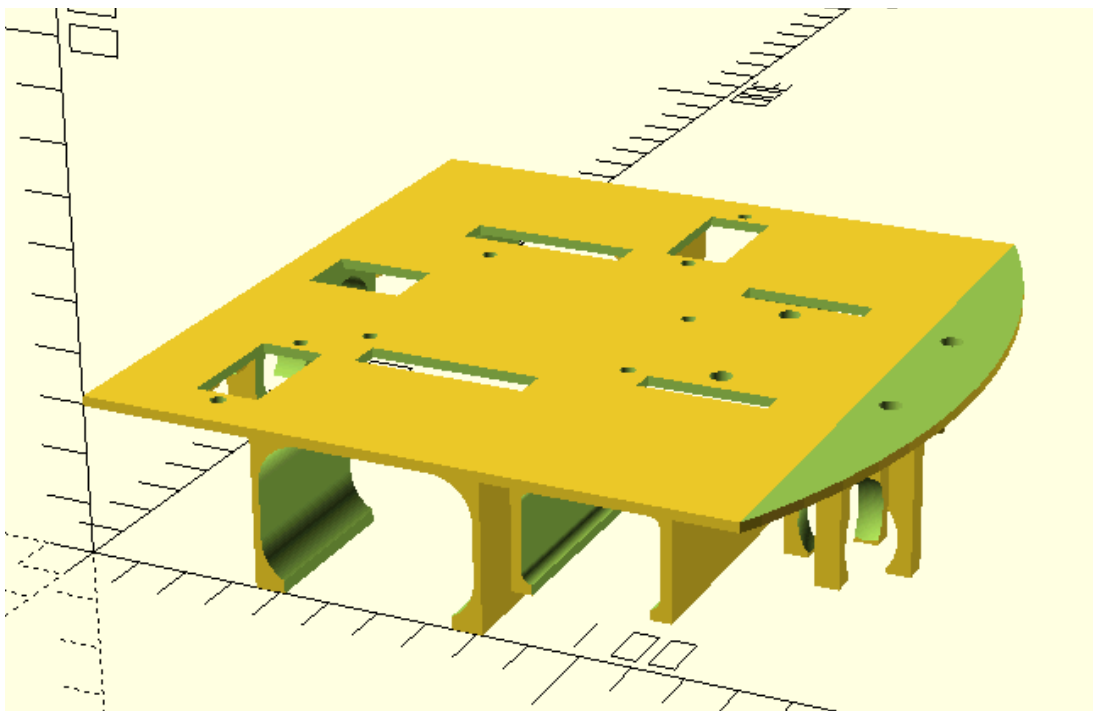
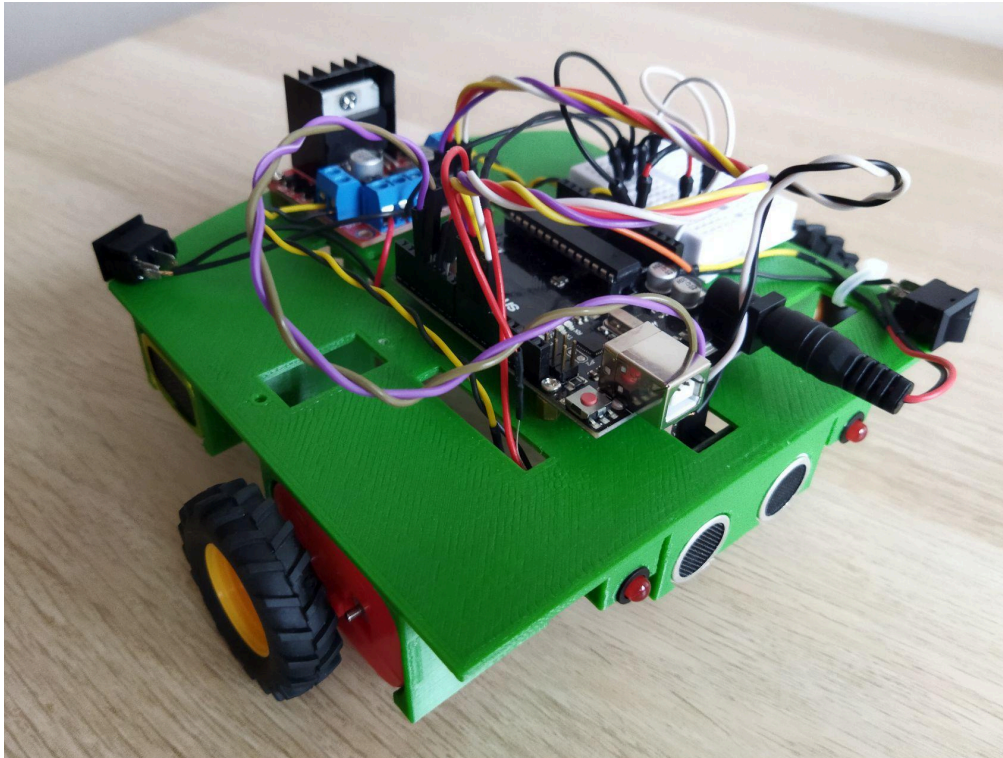


# ROBOT SALVA-OBSTÁCULOS CON SENSOR ULTRASÓNICO Y ARDUINO



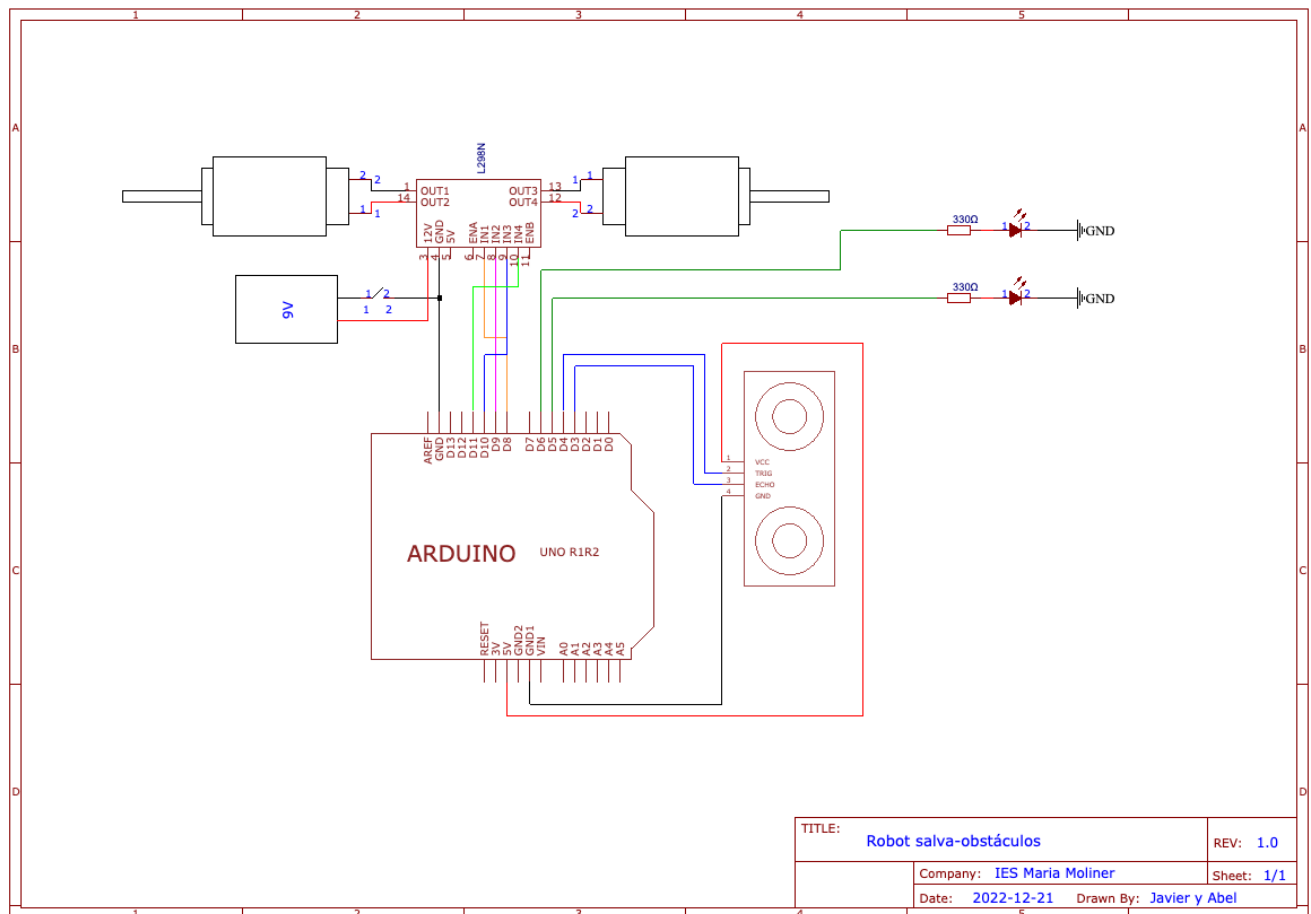
## Índice

<b>1. Descripción</b>	<b>2</b>
<b>2. Esquema eléctrico</b>	<b>2</b>
<b>3. Lista de componentes</b>	<b>2</b>
<b>4. Código arduino</b>	<b>3</b>
<b>5. Funcionamiento</b>	<b>5</b>
<b>6. Diseño del chasis con openScad</b>	<b>6</b>
Faro con dos leds	6
Arduino:	7
Alojamiento del motor:	7
Alojamiento del sensor ultrasonidos	8
Alojamiento de batería:	9
Servomotor	9
Bola:	10
Chasis completo:	11
<b>7. Multimedia</b>	<b>12</b>
<b>8. Anexo</b>	<b>13</b>
Funcionamiento sensor ultrasónico	13
Funcionamiento puente L298N	13
<b>9. Extra: Librería y programación orientada a objetos</b>	<b>15</b>

## 1. Descripción

Al detectar un obstáculo, el robot avanza hacia detrás y gira aleatoriamente a un lado (derecha o izquierda).

## 2. Esquema eléctrico



## 3. Lista de componentes

Para realizar este proyecto necesitaremos:

- ARDUINO UNO
- Puente L298N
- Dos baterías de 9V
- Sensor ultrasónico HC-SR04
- Dos diodos led
- Dos resistencias 330Ω
- Dos motores con reductora
- Protoboard

## 4. Código arduino

```

#define led1 5
#define led2 6
#define motor1_atras 8
#define motor1_delante 9
#define motor2_atras 10
#define motor2_delante 11

long tiempo;
int trig=4;
int eco=3;
float distancia;
int numero=0; //variable para ver si gira hacia un lado o hacia otro

void setup() {
  pinMode(trig, OUTPUT); //es salida ya que envía el pulso
  pinMode(eco, INPUT); //es entrada ya que recibe el pulso
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  Serial.begin(9600);
  randomSeed(analogRead(A0)); //cambia la semilla del random para que
  al empezar el programa no se repita siempre la misma secuencia de giro
}

void loop() {
  delante(); //el coche va hacia delante de forma predeterminada
  digitalWrite(trig, HIGH); //envia el pulso
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  tiempo = (pulseIn(eco, HIGH)/2); //calcula el tiempo del pulso
  distancia = float(tiempo*0.0343); //calcula la distancia

  if(distancia<15){ //cuando detecta un obstáculo
    atras(); //el coche va hacia detrás
    for(int i=0; i<8; i++){ //parpadean los leds 7 veces
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
      delay(150);
      digitalWrite(led1, HIGH);
      digitalWrite(led2, HIGH);
      delay(150);
    }
  }
}

```

```
parar(); //pequeña pausa antes de realizar el giro
delay(250);
numero=random(0,2); //elige un numero aleatorio entre 0 y 1

if(numero==0){ //si el número es 0 gira hacia la derecha
    derecha();
    for(int i=0; i<6; i++){ //parpadean los leds 5 veces
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        delay(150);
        digitalWrite(led1, HIGH);
        delay(150);
    }
    parar(); //pequeña pausa antes de que siga hacia delante
    delay(250);
}

if(numero==1){ //si el número es 1 gira hacia la izquierda
    izquierda();
    for(int i=0; i<6; i++){ //parpadean los leds 5 veces
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        delay(150);
        digitalWrite(led2, HIGH);
        delay(150);
    }
    parar(); //pequeña pausa antes de que siga hacia delante
    delay(250);
}
}

void delante(){ //ambos motores van hacia delante
    digitalWrite(led1, HIGH); //los dos leds están encendidos
    digitalWrite(led2, HIGH);
    digitalWrite(motor1_delante, HIGH);
    digitalWrite(motor1_atras, LOW);
    digitalWrite(motor2_delante, HIGH);
    digitalWrite(motor2_atras, LOW);
}

void atras(){ //ambos motores van hacia detrás
    digitalWrite(motor1_delante, LOW);
```

```
digitalWrite(motor1_atras, HIGH);
digitalWrite(motor2_delante, LOW);
digitalWrite(motor2_atras, HIGH);
}

void izquierda(){ //un motor va hacia delante y otro hacia atrás
digitalWrite(motor1_delante, HIGH);
digitalWrite(motor1_atras, LOW);
digitalWrite(motor2_delante, LOW);
digitalWrite(motor2_atras, HIGH);
}

void derecha(){ //un motor va hacia atrás y el otro hacia delante
digitalWrite(motor1_delante, LOW);
digitalWrite(motor1_atras, HIGH);
digitalWrite(motor2_delante, HIGH);
digitalWrite(motor2_atras, LOW);
}

void parar(){ //todos los motores se paran
digitalWrite(motor1_delante, LOW);
digitalWrite(motor1_atras, LOW);
digitalWrite(motor2_delante, LOW);
digitalWrite(motor2_atras, LOW);
}
```

## 5. Funcionamiento

Hemos creado cinco funciones que contienen los cinco posibles movimientos de los motores (hacia delante, hacia atrás, giro en ambas direcciones y quedarse parado).

Por defecto, nuestro robot se mueve hacia delante y ambos leds están encendidos. Para detectar un obstáculo, hemos utilizado el sensor ultrasónico (sensor que mide el tiempo de ida y vuelta de un sonido ultrasónico) y cuando la distancia sea menor a 15cm, ejecuta la función `atras()`; y parpadean ambos leds 7 veces.

Una vez realizado esto, se elige un número aleatorio entre 0 y 1. Dependiendo de ese número, el robot girará hacia la derecha o hacia la izquierda. Además, parpadea el led del lado hacia el que está girando, simulando un intermitente.

Para evitar que se siga la misma secuencia de giro, hemos implementado la función `randomSeed()`; lo que cambia el punto de la secuencia por el que se empieza.

Finalmente, hemos añadido dos interruptores a las baterías de 9V para poder apagar y encender el robot cuando lo deseemos sin necesidad de desconectar las baterías.

## 6. Diseño del chasis con openScad

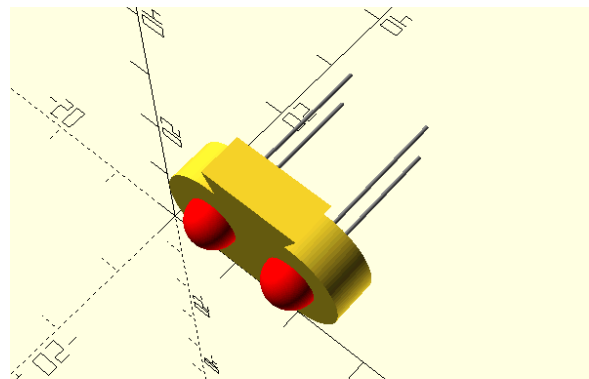
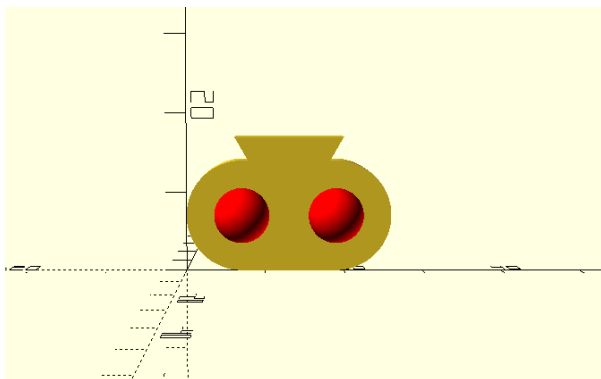
### Faro con dos leds

```

use<leds.scad>
use<enganche.scad>

module faro(leds=false,enganche=true){
    difference(){
        union(){
            hull(){
                translate([7,0,7]) rotate([270,0,0]) cylinder(r=7,h=6,$fn=100);
                translate([19,0,7]) rotate([270,0,0]) cylinder(r=7,h=6,$fn=100);
            }
        }
        leds();
    }
    if(enganche==true){
        translate([-10,-7,-12]) enganche();
    }
    if(leds==true){
        leds();
    }
}
faro(leds=true,enganche=true);

```



### Leds

```

module led1(){
    translate([7,0,7])color("red")sphere(d=7, $fn=100);
    translate([7,0,7]) color("red")rotate([270,0,0])cylinder(d=7, h=6.5, $fn=100);
    translate([6,6.5,7])rotate([270,0,0])color("grey") cylinder(d=0.5, h=20, $fn=100);
    translate([8,6.5,7])rotate([270,0,0])color("grey") cylinder(d=0.5, h=16, $fn=100);
}

module leds(){
    union(){
        led1();
        translate([12,0,0]) led1();
    }
}

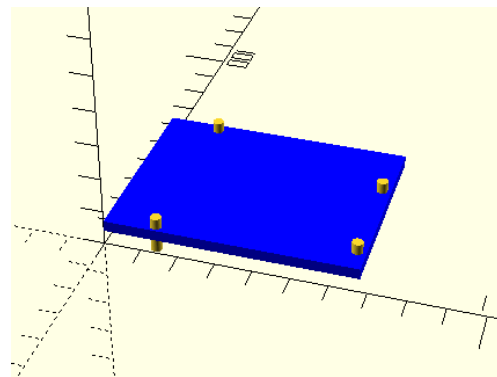
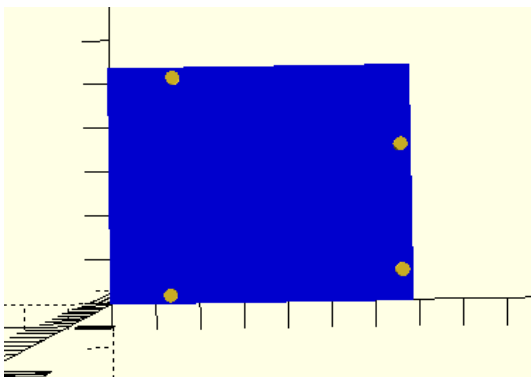
leds();

```

## Arduino

```
use<tornillos_arduino.scad>
```

```
module arduino(tornillos_arduino=false){
  difference(){
    translate([0,0,4])color("blue")cube([68,53,3]);
    tornillos_arduino();
  }
  if(tornillos_arduino==true){
    tornillos_arduino();
  }
}
arduino(true);
```



```
module tornillos_arduino(){
  translate([65.5,7,0]) cylinder(d=3,h=10,$fn=100);
  translate([65.5,35,0]) cylinder(d=3,h=10,$fn=100);
  translate([14.7,50.5,0]) cylinder(d=3,h=10,$fn=100);
  translate([13.7,2,0]) cylinder(d=3,h=10,$fn=100);
}

tornillos_arduino();
```

## Alojamiento del motor

```
module motor(){
  union(){
    hull(){
      translate ([10, 7, 10]) rotate([270, 0, 0]) cylinder (r=8.5, h=
33, $fn=100);
      translate ([10, 7, 21]) rotate([270, 0, 0]) cylinder (r=8.5, h=
33, $fn=100);
      translate ([36, 7, 10]) rotate([270, 0, 0]) cylinder (r=8.5, h=
33, $fn=100);
      translate ([36, 7, 21]) rotate([270, 0, 0]) cylinder (r=8.5, h=
33, $fn=100);
    }
    translate ([10, 0, 10]) rotate([270, 0, 0]) cylinder (d=2.1, h=47,
$fn=100);
  }
}
```

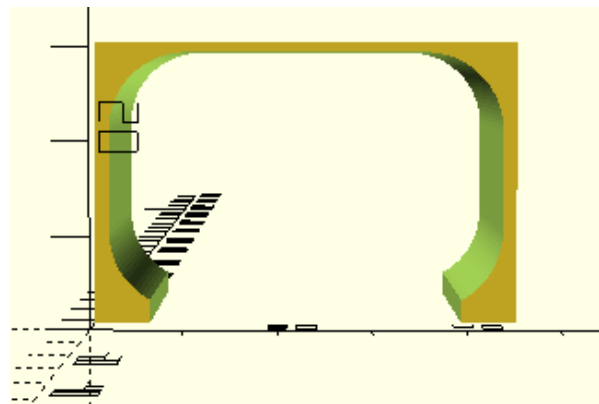
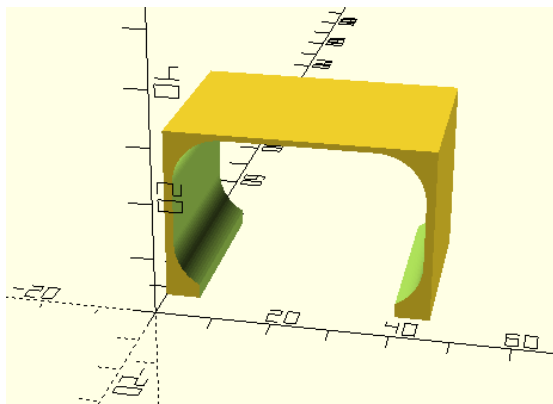


```

module alojamiento_motor(){
  difference(){
    union(){
      difference(){
        difference(){
          translate([0,7,0]) cube([46,33,30.5]);
          motor();
        }
        translate([6,7,0])cube([34,33,10]);
      }
    }
  }
}

alojamiento_motor();

```



### Alojamiento del sensor ultrasonidos

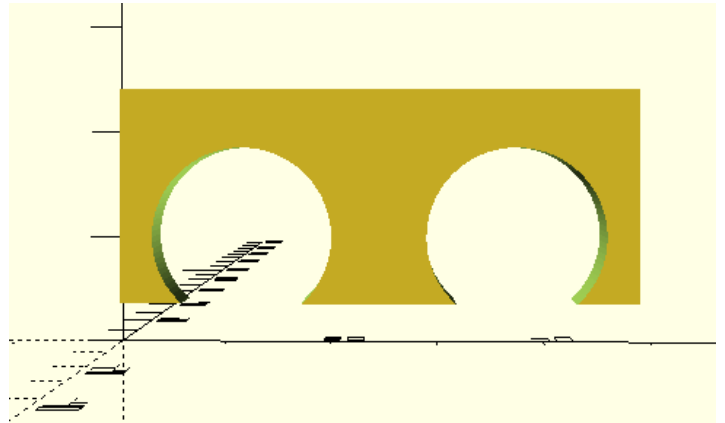
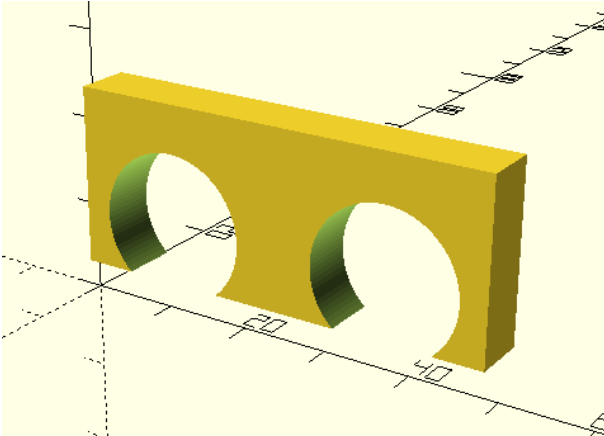
```

module sensor_ultrasonidos(){
  translate([0,3,0]) union(){
    translate([-1.5,2,0]) cube([50,24.6,23]);
    translate([10,-5,10]) rotate([270,0,0]) cylinder(d=17, h=12, $fn=100);
    translate([36,-5,10]) rotate([270,0,0]) cylinder(d=17, h=12, $fn=100);
    translate([17.25,13.6,17]) cube([10.5,12,10]);
  }
}

module alojamiento_sensor_ultrasonidos(sensor=false){
  difference(){
    difference(){
      difference(){
        translate([0,-1.5,0])cube([49,30,24]);
        translate([1.5,0,0]) sensor_ultrasonidos();
      }
      translate([0, 5, -1]) cube([58,25.6,26]);
    }
    translate([0,-2,0])cube([50,7.5,3.7]);
  }
  if(sensor==true){
    translate([2,0,0])sensor_ultrasonidos();
  }
}

alojamiento_sensor_ultrasonidos();

```



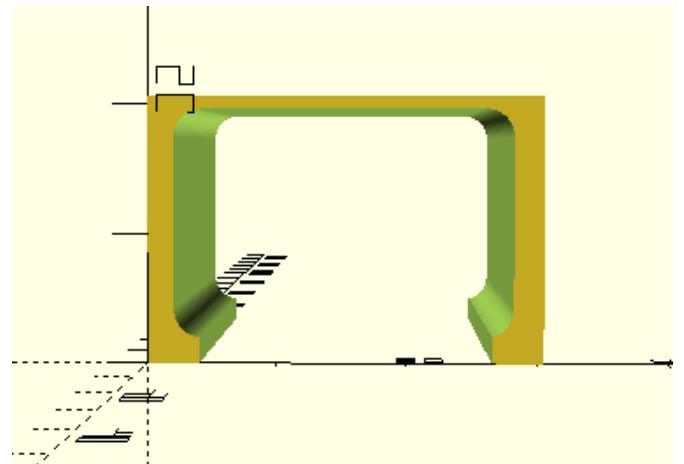
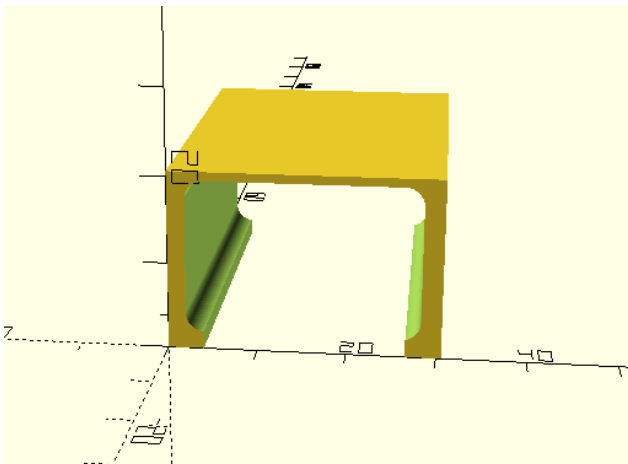
### Alojamiento de batería

```

module bateria(){
  hull(){
    translate([2.15,0,2.15]) rotate([270,0,0]) cylinder(d=4.3,h=45,$fn=
    100);
    translate([2.15,0,15.45]) rotate([270,0,0]) cylinder(d=4.3,h=45,$fn=
    100);
    translate([24.15,0,2.15]) rotate([270,0,0]) cylinder(d=4.3,h=45,$fn=
    100);
    translate([24.15,0,15.45]) rotate([270,0,0]) cylinder(d=4.3,h=45,$fn=
    100);
  }
}

module alojamiento_bateria(){
  difference(){
    difference(){
      cube([30.6,43,20.6]);
      translate([2,0,2])bateria();
    }
    translate([4,0,0])cube([22.6,50,3]);
  }
}

alojamiento_bateria();
    
```



## Servomotor

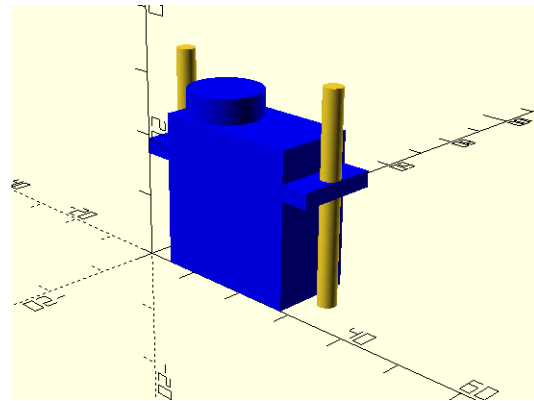
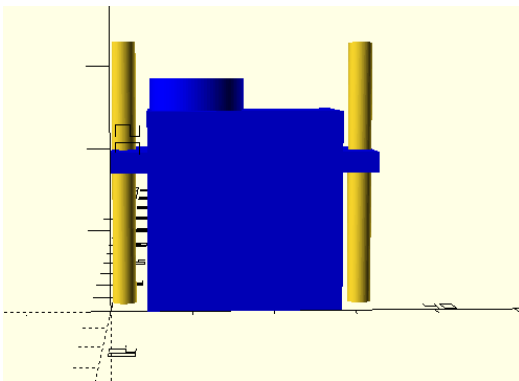
use<tornillos.scad>

```

module servo(tornillos=false){
  difference(){
    color("blue")union(){
      translate([4.5,0,0]) cube([24,13,24.5]);
      translate([0,0,17]) cube([4.5,13,2.5]);
      translate([28.5,0,17]) cube([4.5,13,2.5]);
      translate([10.7,6.2,0]) cylinder(d=11.9, h=28.5, $fn=100);
    }
    tornillos();
  }
  if(tornillos==true){
    tornillos();
  }
}

servo(true);

```



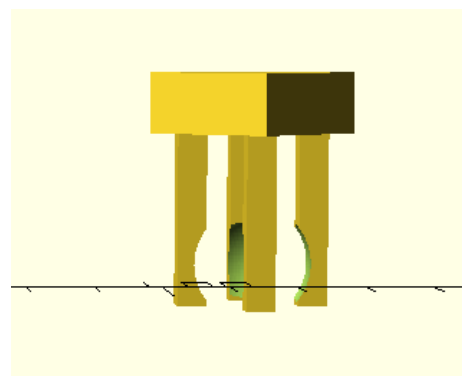
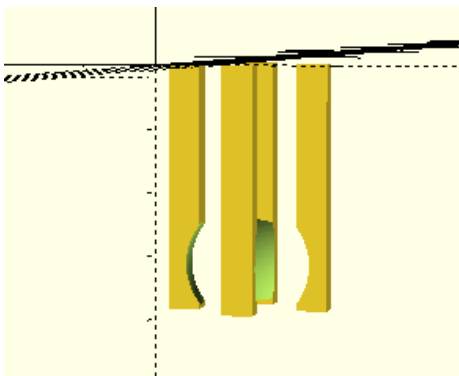
## Bola

```

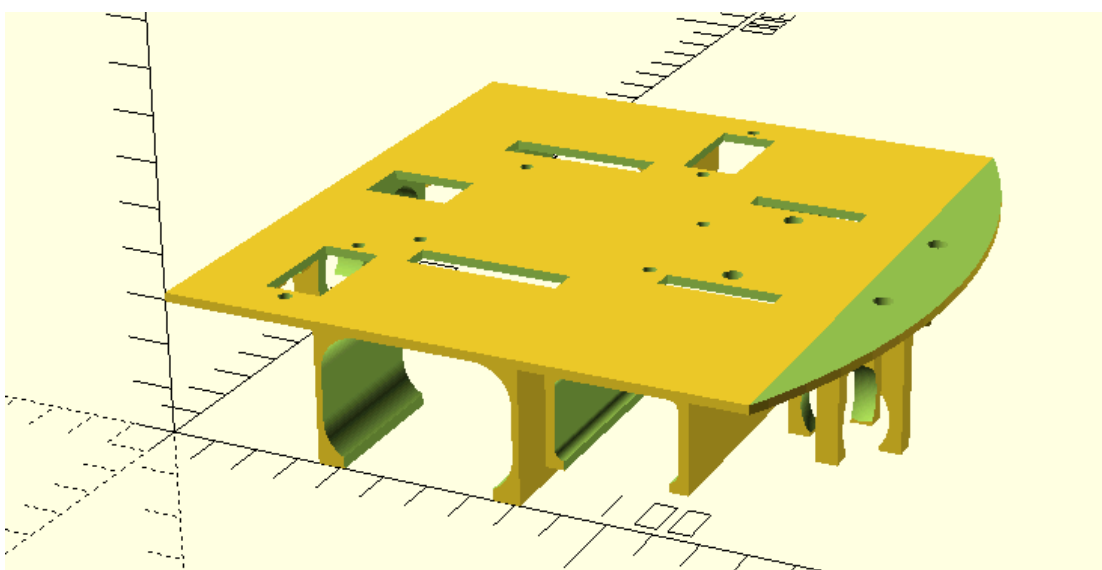
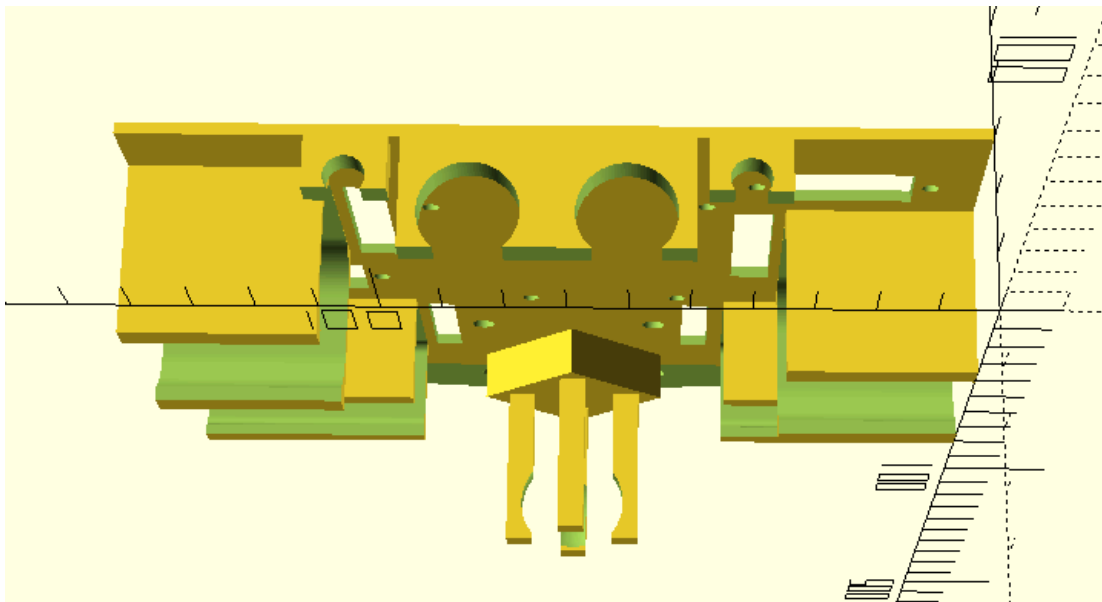
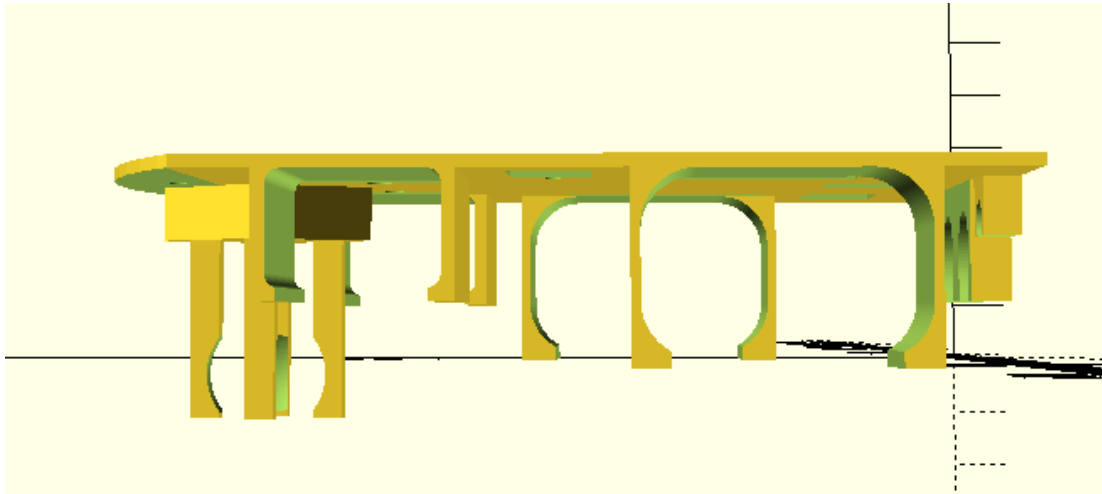
module bola(){
  difference(){

    union(){
      translate([10.5,0,0])cube([5,5,39.7]);
      translate([0,10.5,0])cube([5,5,39.7]);
      translate([21,10.5,0])cube([5,5,39.7]);
      translate([10.5,21,0])cube([5,5,39.7]);
    }
    translate([13,13,32.4]) sphere(d=20.8,$fn=150);
  }
}

```



**Chasis completo**



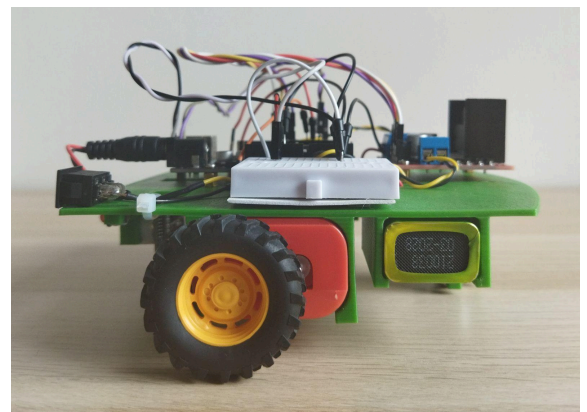
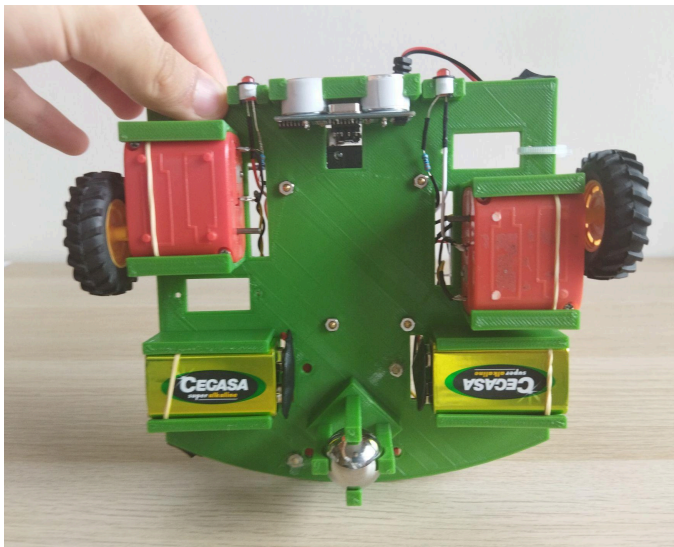
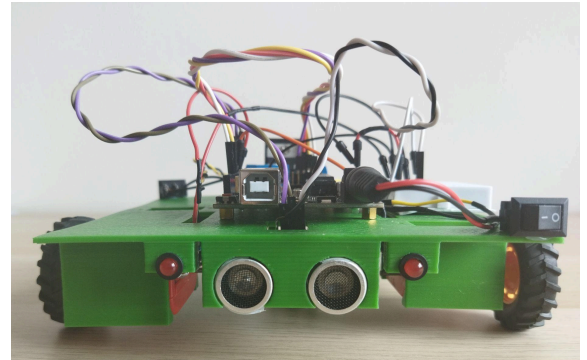
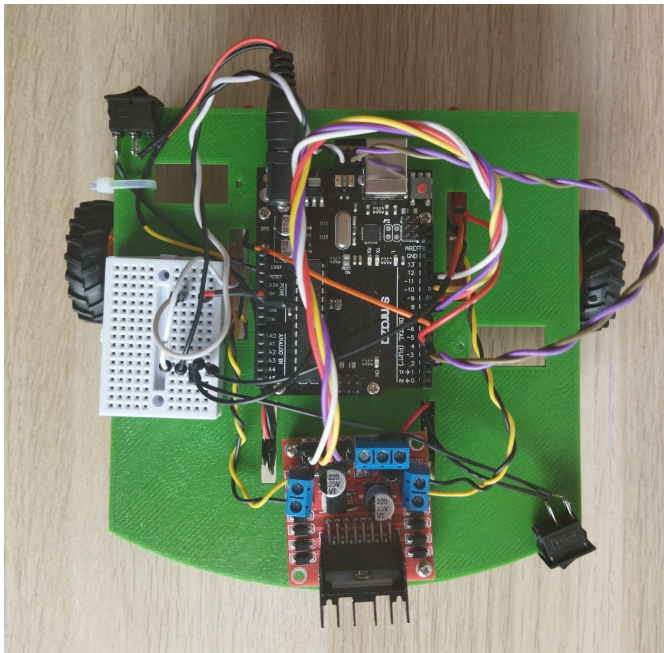
## 7. Multimedia

Vídeo: <https://www.instagram.com/reel/Cmbg3CehKVV/?igshid=YmMyMTA2M2Y=>

Archivos en Drive:

[https://drive.google.com/drive/folders/1SF1hgDt4iMV4UGvBKwBbX8JawIhpYPIR?usp=drive\\_link](https://drive.google.com/drive/folders/1SF1hgDt4iMV4UGvBKwBbX8JawIhpYPIR?usp=drive_link)

Fotos:

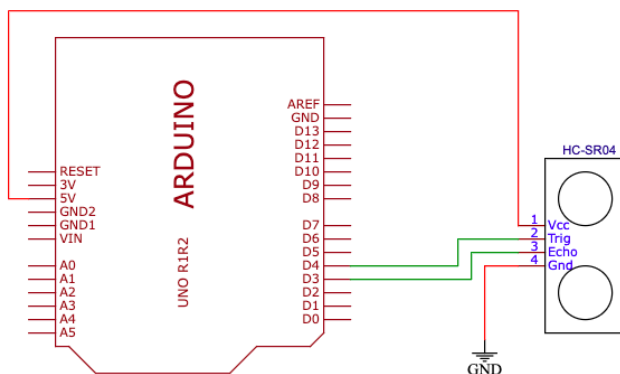
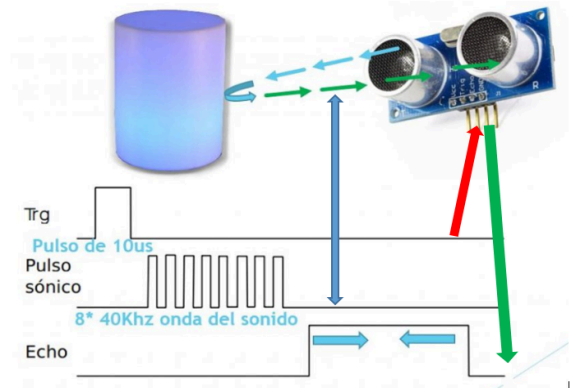


## 8. Anexo

### Funcionamiento sensor ultrasónico

Este sensor emite un **sonido ultrasónico** por uno de sus transductores y espera que el sonido rebote de algún objeto presente (un obstáculo). El eco es captado por el segundo transductor. La distancia es proporcional al tiempo que tarda en llegar el eco.

El rango de medición del sensor es de 2 a 400 cm, con una resolución de 0,3 cm.



El sensor ultrasónico tiene cuatro pines:

- **VCC**, conectado a 5V
- **GND**, conectado a tierra
- **Trigger**, que es el encargado de enviar los pulsos.
- **Echo**, que es el encargado de recibir los pulsos.

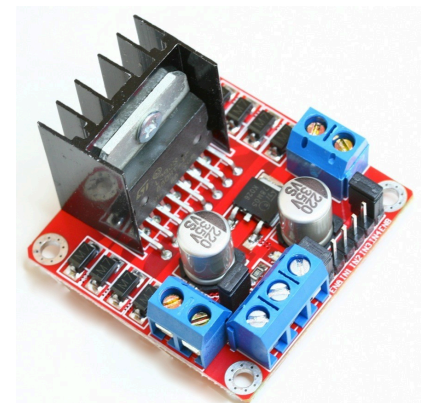
Para saber cuánto tiempo ha tardado el pulso en ir y volver, empleamos la función `pulseIn(pin, value)`. Una vez tenemos este tiempo, calcularemos la distancia empleando la velocidad del sonido (343 m/s) o su equivalente en cm/µs.

$$\text{Velocidad} = \frac{\text{distancia recorrida}}{\text{tiempo}} \quad 343 \frac{\text{m}}{\text{s}} \cdot 100 \frac{\text{cm}}{\text{m}} \cdot \frac{1}{1000000} \frac{\text{s}}{\mu\text{s}} = \frac{1}{29.2} \frac{\text{cm}}{\mu\text{s}}$$

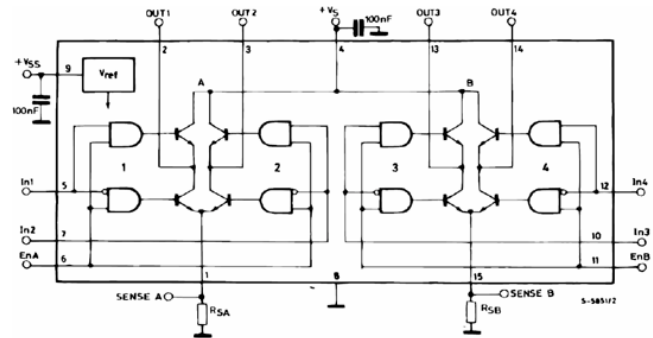
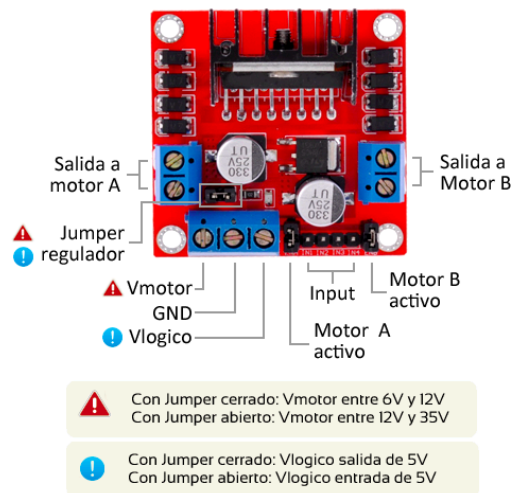
### Funcionamiento puente L298N

Un L298N consiste en dos puentes-H, uno para la salida A y otro para la salida B.

Un puente-H es un componente ampliamente utilizado en electrónica para alimentar una carga de forma que podemos invertir el sentido de la corriente que le atraviesa.





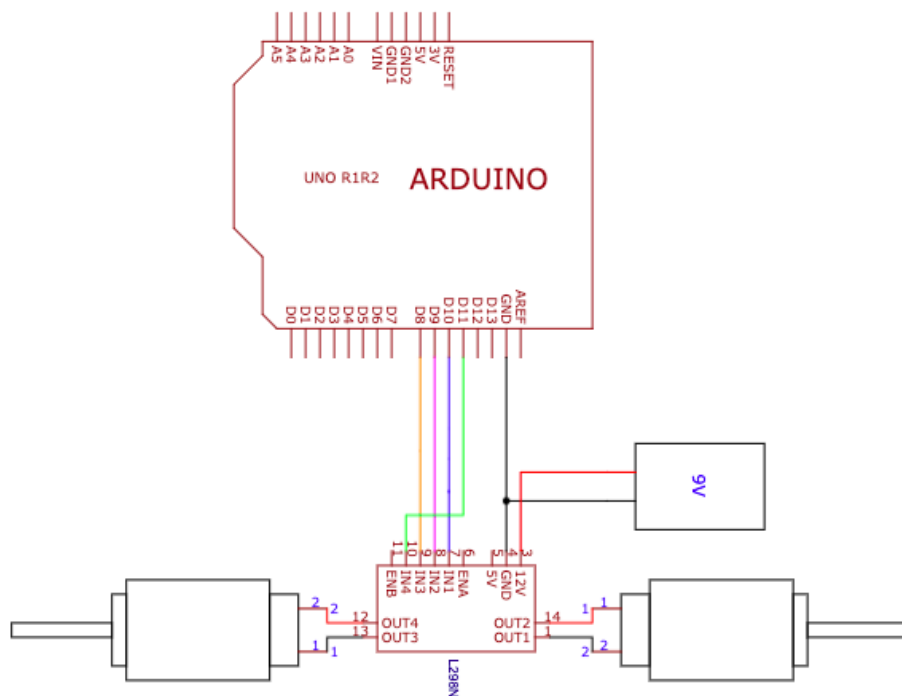


La entrada de tensión proporciona el voltaje a los motores. El rango de entrada admisible es de 3V a 35V (dependiendo si el jumper está abierto o cerrado). El pin GND deberemos conectarlo a tierra.

Por otro lado, tenemos los pines de entrada que controlan la dirección y velocidad de giro:

- Los pines IEA (), IN1 e IN2 controlan la salida A
- Los pines IEB (), IN3 e IN4 controlan la salida B

Los pines IEA y IEB desactivan la salida. Podemos conectarlos permanentemente mediante el uso de un jumper. También podemos controlar la velocidad de los motores conectándolos a una salida PWM y utilizando la función analogWrite().



## 9. Extra: Librería y programación orientada a objetos

- Archivo header:

```
#ifndef _COCHE_H
#define _COCHE_H

#include <Arduino.h>

class Coche{
private:
    int led1;
    int led2;
    int motor1_atras;
    int motor1_delante;
    int motor2_atras;
    int motor2_delante;
    int trig;
    int eco;

public:
    Coche(int _led1, int _led2, int _motor1_atras, int _motor1_delante,
int _motor2_atras, int _motor2_delante, int _trig, int _eco);
    init();
    void delante();
    void atras();
    void izquierda();
    void derecha();
    void parar();
    float calcularDistancia();
    void intermitenteAtras();
    void intermitenteDerecha();
    void intermitenteIzquierda();
};

#endif
```



- Archivo cpp:

```
#include "coche.h"

Coche::Coche(int _led1, int _led2, int _motor1_atras, int
_motor1_delante, int _motor2_atras, int _motor2_delante, int _trig, int
_eco) {
    this->led1 = _led1;
    this->led2 = _led2;
    this->motor1_atras = _motor1_atras;
    this->motor1_delante = _motor1_delante;
    this->motor2_atras = _motor2_atras;
    this->motor2_delante = _motor2_delante;
    this->trig = _trig;
    this->eco = _eco;
    init();
}

Coche::init() {
    pinMode(trig, OUTPUT);
    pinMode(ecu, INPUT);
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    randomSeed(analogRead(A0));
}

void Coche::delante() {
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(motor1_delante, HIGH);
    digitalWrite(motor1_atras, LOW);
    digitalWrite(motor2_delante, HIGH);
    digitalWrite(motor2_atras, LOW);
}

void Coche::atras() {
    digitalWrite(motor1_delante, LOW);
    digitalWrite(motor1_atras, HIGH);
    digitalWrite(motor2_delante, LOW);
```

```

    digitalWrite(motor2_atras, HIGH);
}

void Coche::izquierda() {
    digitalWrite(motor1_delante, HIGH);
    digitalWrite(motor1_atras, LOW);
    digitalWrite(motor2_delante, LOW);
    digitalWrite(motor2_atras, HIGH);
}

void Coche::derecha() {
    digitalWrite(motor1_delante, LOW);
    digitalWrite(motor1_atras, HIGH);
    digitalWrite(motor2_delante, HIGH);
    digitalWrite(motor2_atras, LOW);
}

void Coche::parar() {
    digitalWrite(motor1_delante, LOW);
    digitalWrite(motor1_atras, LOW);
    digitalWrite(motor2_delante, LOW);
    digitalWrite(motor2_atras, LOW);
}

float Coche::calcularDistancia() {
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    long tiempo = (pulseIn(eco, HIGH)/2);
    float distancia = float(tiempo*0.0343);
    return distancia;
}

void Coche::intermitenteAtras() {
    for(int i=0; i<8; i++){
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        delay(150);
        digitalWrite(led1, HIGH);
    }
}

```

```
        digitalWrite(led2, HIGH);
        delay(150);
    }
}

void Coche::intermitenteDerecha() {
    for(int i=0; i<6; i++){
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        delay(150);
        digitalWrite(led1, HIGH);
        delay(150);
    }
}

void Coche::intermitenteIzquierda() {
    for(int i=0; i<6; i++){
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        delay(150);
        digitalWrite(led2, HIGH);
        delay(150);
    }
}
```

- Archivo ino:

```
#include "coche.h"

float dis;
int numero=0;

Coche cupra(5, 6, 8, 9, 10, 11, 4, 3);

void setup() {
    Serial.begin(9600);
}

void loop() {
    cupra.delante();
}
```

```
dis = cupra.calcularDistancia();
if(dis<15){
    cupra.atras();
    cupra.intermitenteAtras();
    cupra.parar();
    delay(250);
    numero = random(0,2);
    if(numero==0){
        cupra.derecha();
        cupra.intermitenteDerecha();
        cupra.parar();
        delay(250);
    }
    if(numero==1){
        cupra.izquierda();
        cupra.intermitenteIzquierda();
        cupra.parar();
        delay(250);
    }
}
}
```