

# NAVEGACIÓN AUTÓNOMA BASADA EN RECONOCIMIENTO DE SEÑALES DE TRÁFICO

DEPARTAMENTO: INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**AUTOR:** FRANCISCO MÁRQUEZ CHAVES      **EMAIL:** FRAN29400@GMAIL.COM

ESTUDIANTE INGENIERÍA ELECTRÓNICA ROBÓTICA Y MECATRÓNICA  
INTENSIFICACIÓN EN ROBÓTICA Y AUTOMÁTICA

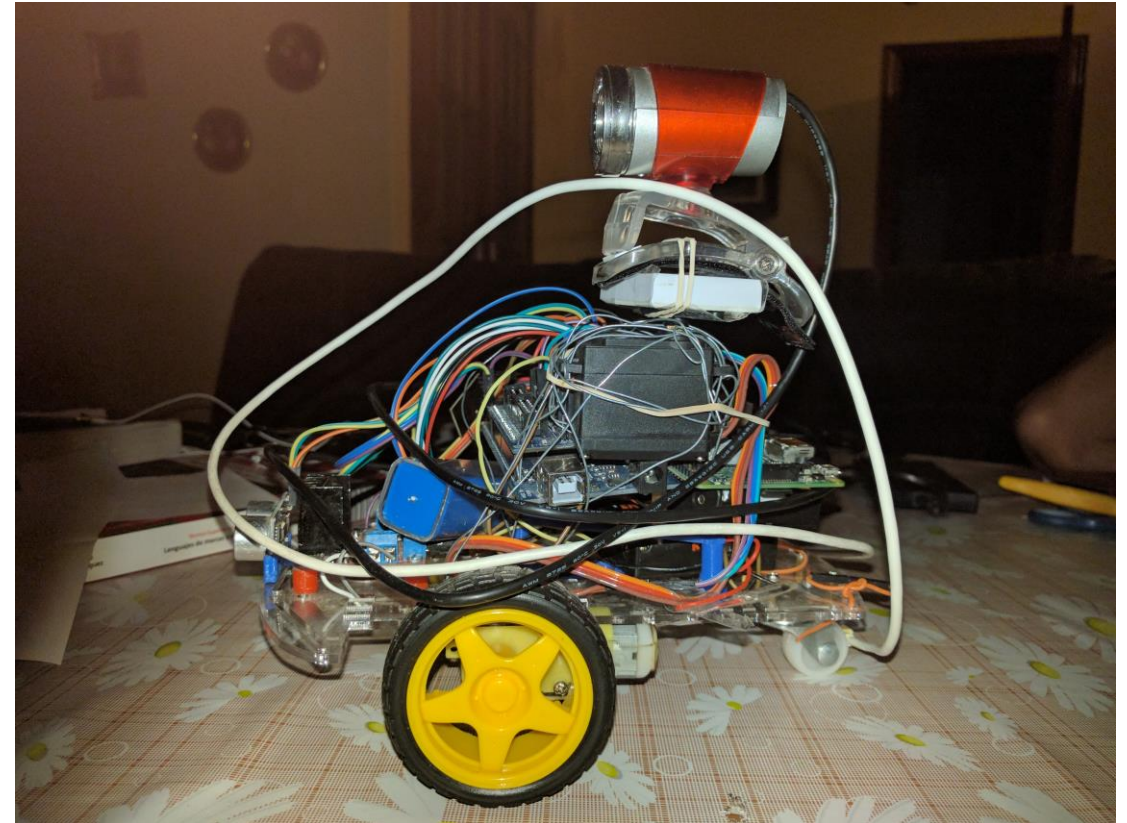
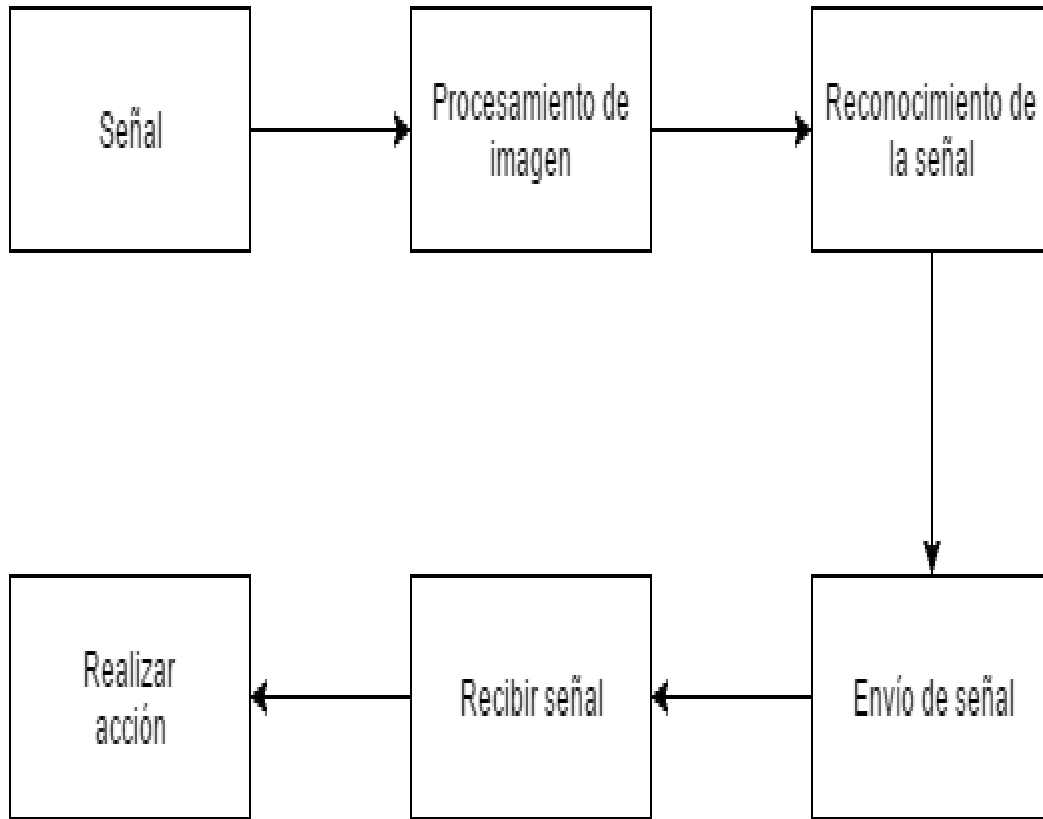
**TUTOR:** FEDERICO CUESTA ROJO      **EMAIL:** FCUESTA@US.ES

PROFESOR TITULAR EN LA ESCUELA SUPERIOR DE INGENIERÍA DE LA UNIVERSIDAD DE SEVILLA

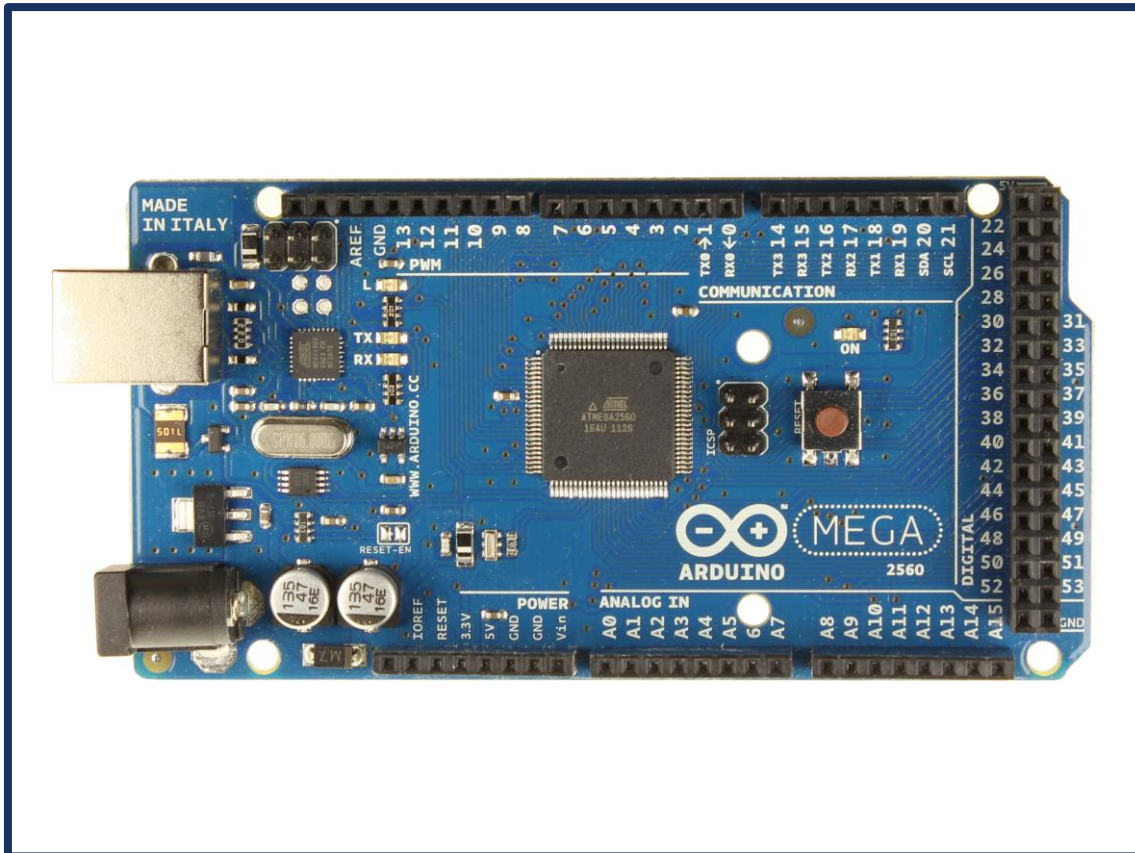
# ÍNDICE

- Descripción General.
- Material usado.
- Esquema de Control.
- Procesamiento de imágenes.
- Resultados experimentales.
- Conclusiones.

# DESCRIPCIÓN GENERAL



# MATERIAL USADO

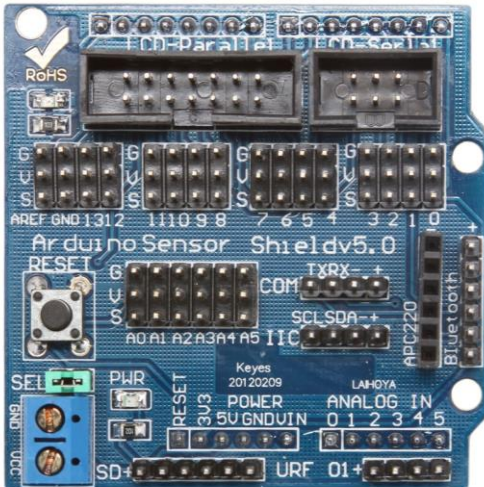


## Arduino MEGA 2560

- Tiene 54 entradas/salidas digitales.
- 15 de ellas pueden ser usadas como PWM.
- 16 entradas analógicas.
- 4 UARTs.

# MATERIAL USADO

## Arduino Sensor Shield



- Placa de expansión para Arduino UNO.
- Compatible con Arduino MEGA.
- Para cada pin, se tiene alimentación y tierra.
- Muy útil si se utilizan muchos sensores.



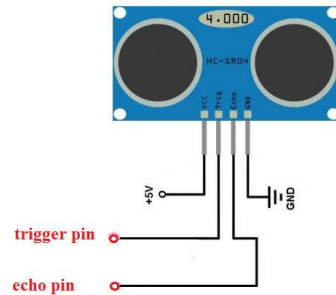
## MATERIAL USADO

### HC-SR04



- Sensor de Ultrasonido.
- Consta de 4 Pines.
- VCC: Alimentación.
- Trigger: Realiza un envío de una señal ultrasonido.
- Echo: Recibe una señal ultrasonido.
- GND: Tierra.

# IMPLEMENTACIÓN DE COMPONENTE



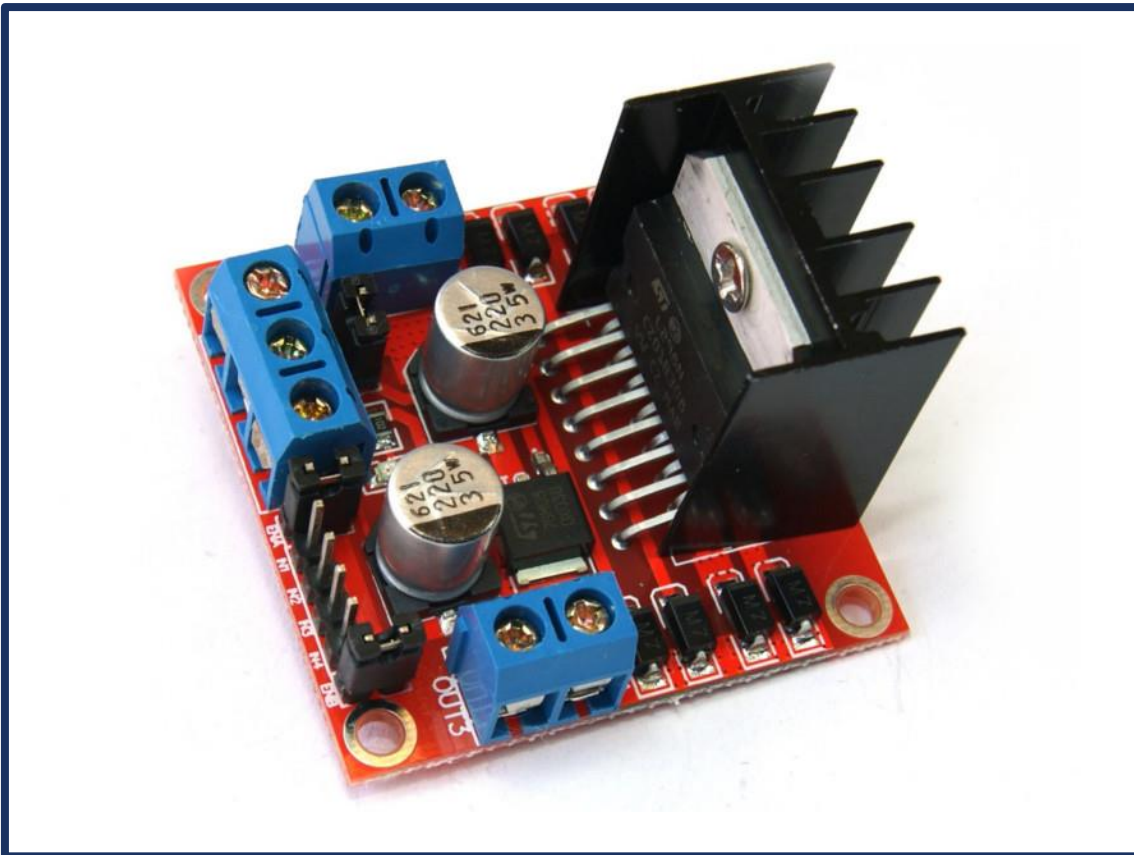
Características	
Alimentación	+5v DC
Frecuencia de trabajo	40 KHz
Consumo (suspendido)	< 2mA
Consumo (trabajando)	15mA
Ángulo efectivo	< 15°
Distancia	2cm a 400cm *
Resolución	0.3 cm

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

## HC-SR04

- Librería de uso libre NewPing.
- Se conecta el pin de alimentación a 5V. El pin GND a tierra.
- Trigger y Echo se conectan a cualquier pin digital.
- Se hace un disparo de ultrasonido activando el pin Trigger, y se mide el tiempo que se tarda en recibirse la señal mediante el pin Echo.
- $Distancia\ (cm) = \frac{340 * 100}{1000 * 2} * tiempo(ms) = 17 * tiempo(ms)$

# MATERIAL USADO

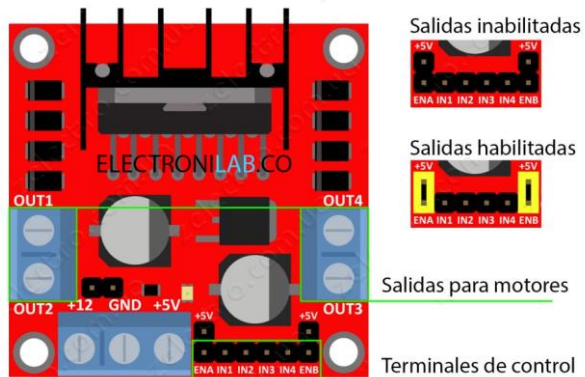


## Puente H L298N

- Tiene 4 salidas para controlar motores.
- 2 para cada motor.
- Podemos alimentar los motores a una tensión y sacar 5V estables para alimentar el Arduino.
- Tenemos 6 terminales de control.
- 2 para habilitar las salidas.
- Las otras 4 sirven para controlar el sentido de giro.



# IMPLEMENTACIÓN DE COMPONENTE



Terminal Activación	Terminal Giro 1	Terminal Giro 2	Función
Alto	Bajo	Alto	Giro derecha
Alto	Alto	Bajo	Giro Izquierda
Bajo	No importa	No importa	Dentención

## Puente H L298N

- Los motores se alimentan a 9,6 V. (8 pilas de 1,2 V)
- Se usa la salida de 5 V para alimentar al Arduino.
- En el terminal de habilitación es donde se recibe la señal para la velocidad del motor.
- Con los terminales de control se elige el sentido de giro de cada motor.

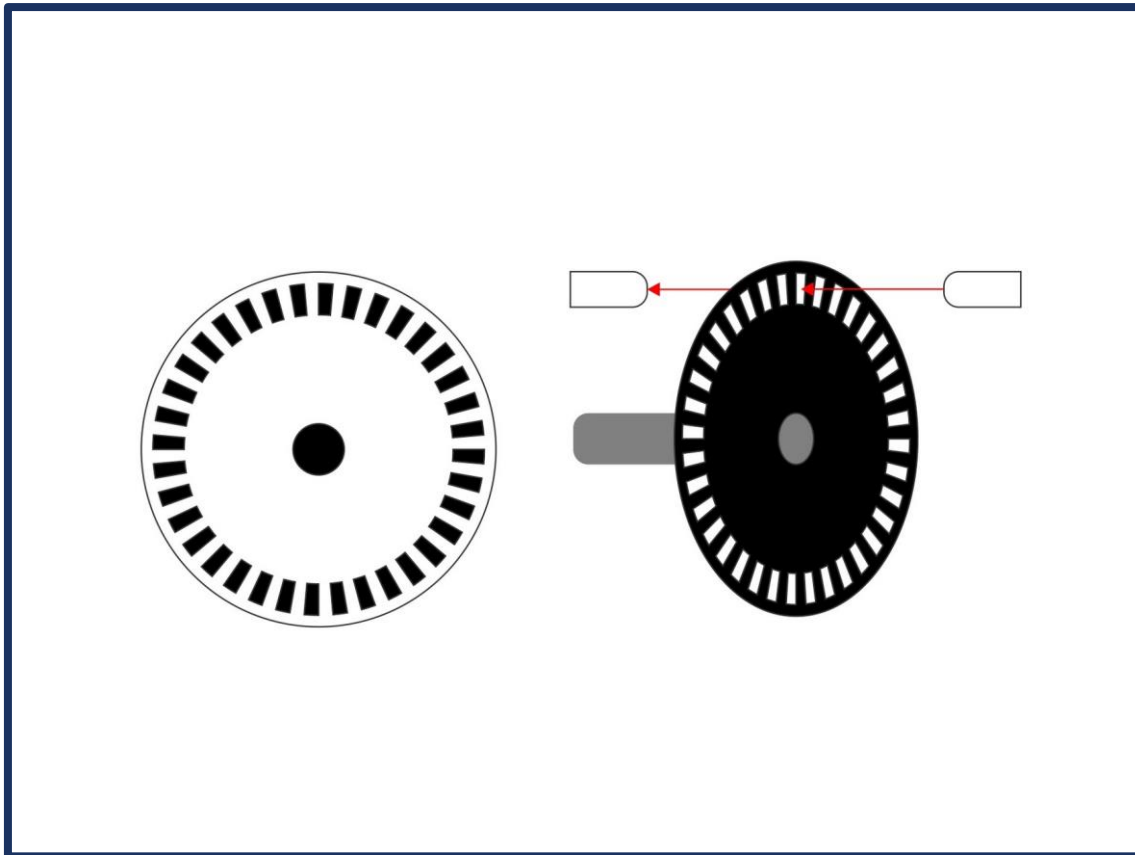
# MATERIAL USADO

## Kit de plataforma móvil



- Base de metacrilato.
- 2 ruedas.
- 2 motores con reductoras.
- 1 Rueda Loca.
- 2 encoders digitales.
- Tornillos y cables.

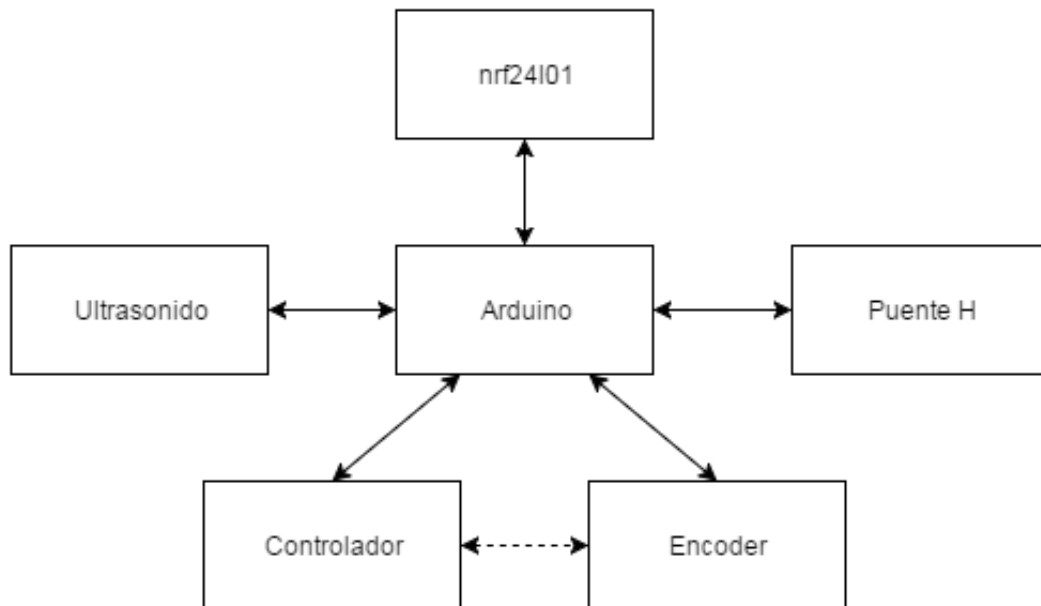
# IMPLEMENTACIÓN DE COMPONENTE



## Encoder digital

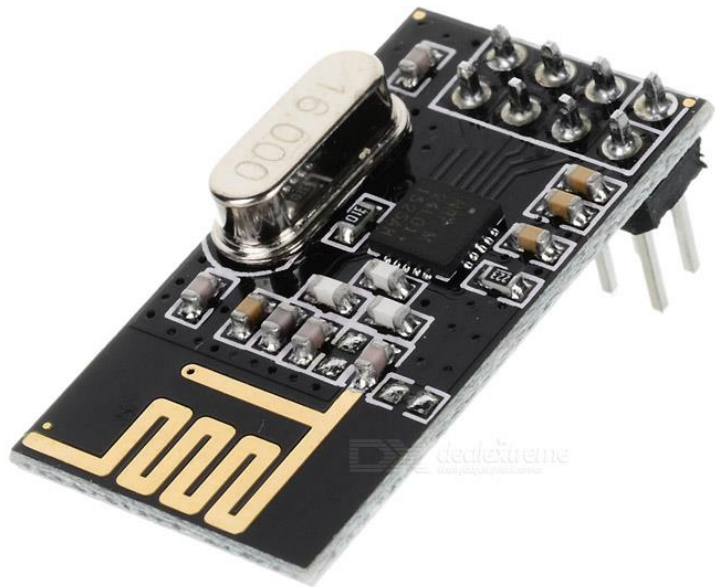
- Lecturas mediante interrupciones.
- Se han limitado los huecos.
- Cada vez que llega una interrupción de lectura, se contabiliza por el Arduino.
- La interrupción debe ser configurada previamente.

# ESQUEMA DE CONTROL



- Ultrasonido: Se usa para mantener al robot en una distancia de seguridad y para realizar el procesamiento desde una distancia adecuada.
- Nrf24l01: Módulo a través del cual nos comunicamos con la Raspberry.
- Controlador: Controlador tipo PI.
- Encóder: Permite llevar la cuenta de la distancia recorrida por cada rueda.
- Puente H: Se encarga de transmitir las señales de control desde el Arduino a cada motor.

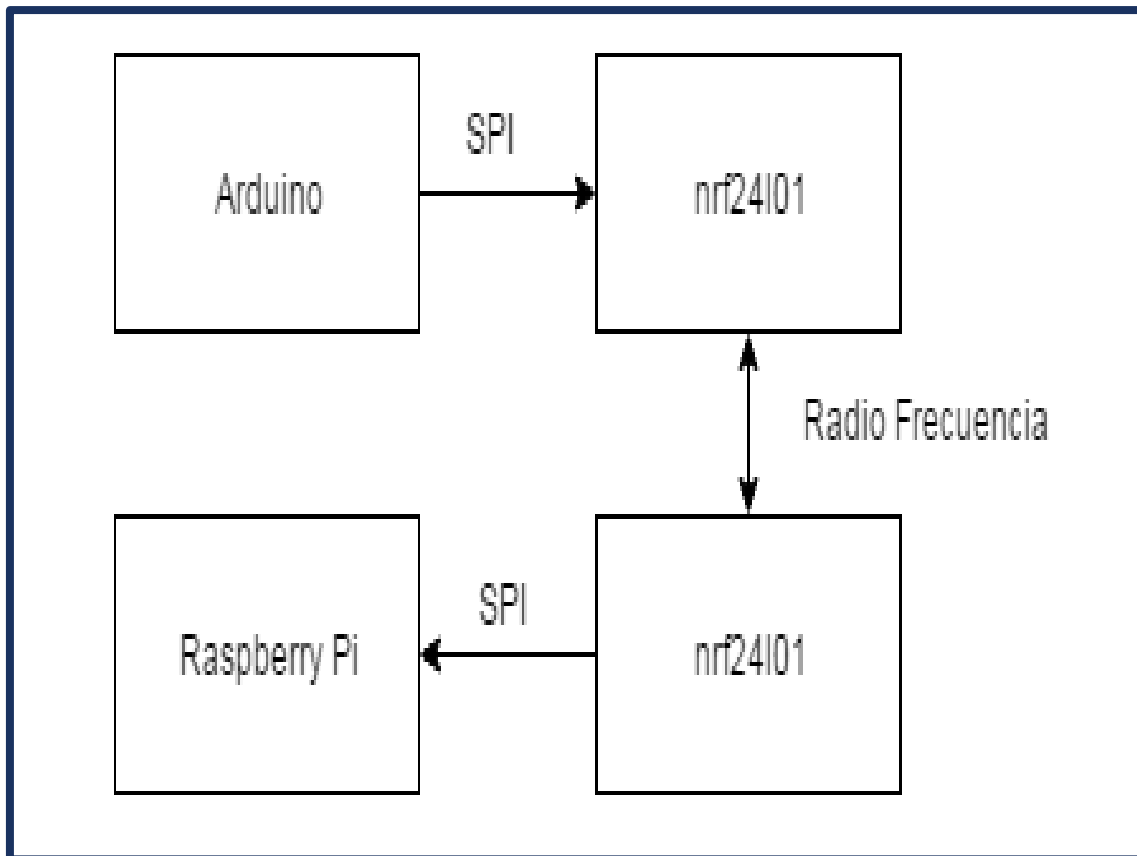
# MATERIAL USADO



## NRF24L01

- Módulo transceptor.
- Banda 2.4 GHz.
- Distancia funcionamiento de 10 a 25 metros.
- Bajo consumo y bajo coste.
- Fácil de usar.
- Mucha documentación.

# IMPLEMENTACIÓN DE COMPONENTE



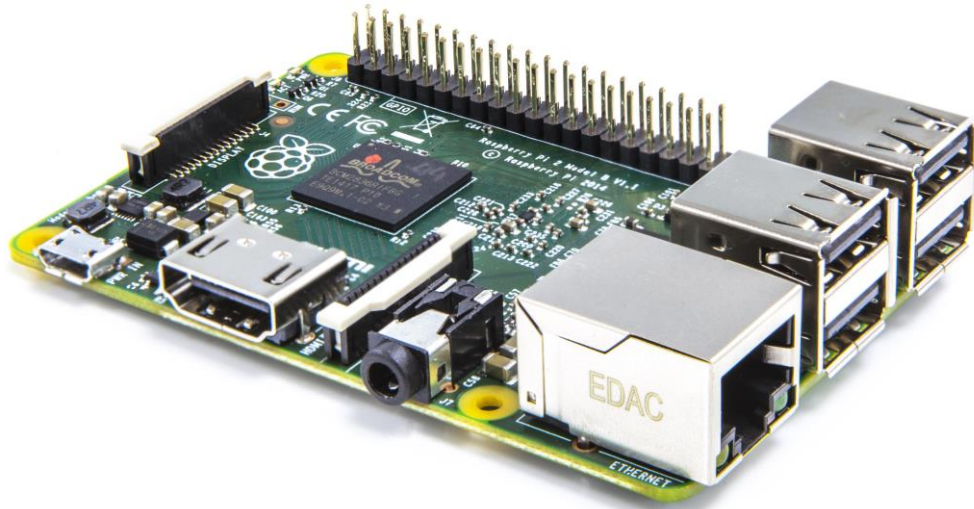
## NRF24L01

- Librería libre llamada: TMRH20.
- Raspberry Pi envía código de la señal.
- Arduino recibe la información.
- Arduino envía confirmación de llegada.
- Raspberry Pi recibe confirmación del Arduino.



# MATERIAL USADO

## Raspberry Pi 2 Model B



- Procesador ARM Cortex-A7 900MHz.
- 1 GB de RAM.
- GPIO de 40 pines.
- 4 Puertos USB.
- Sistema operativo Debian.
- Gran comunidad.

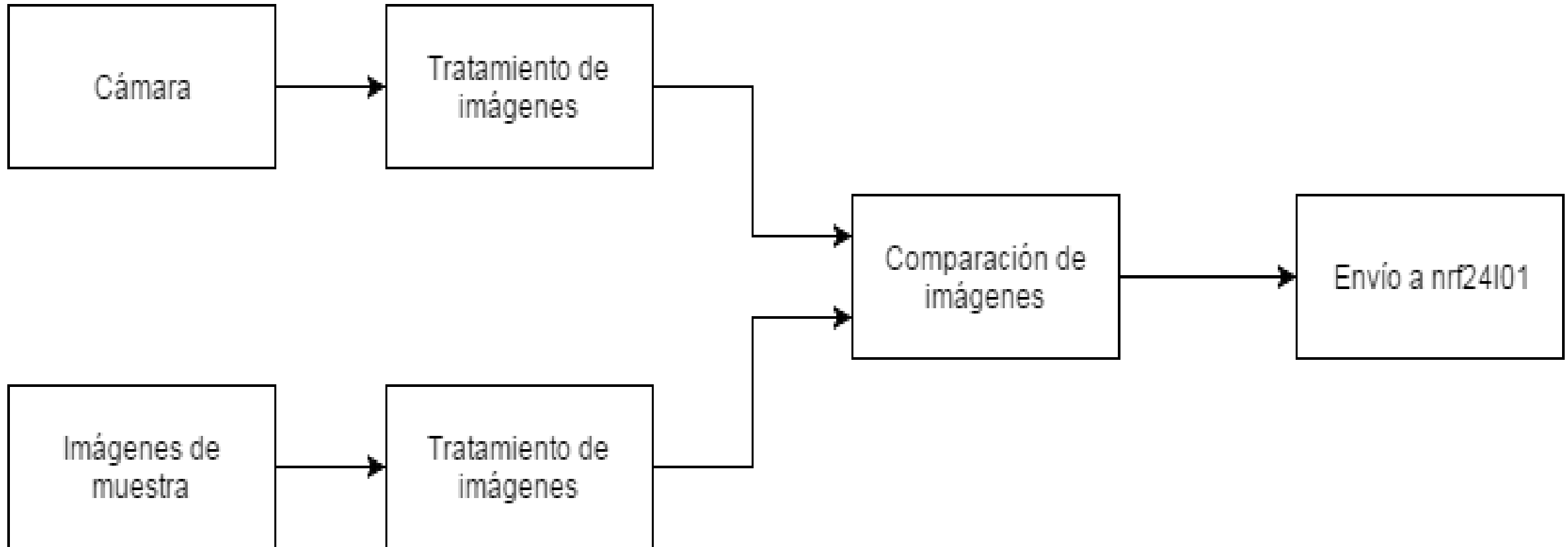
## MATERIAL USADO



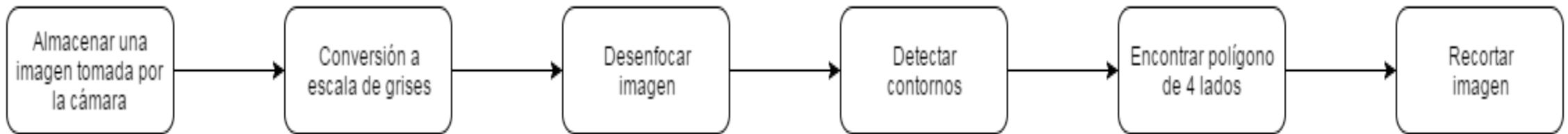
### Cámara USB

- 3.0 Megapíxeles.
- Plug and Play.
- Resolución: 640 x 380.
- Sin necesidad de alimentación externa.
- Enfoque manual.

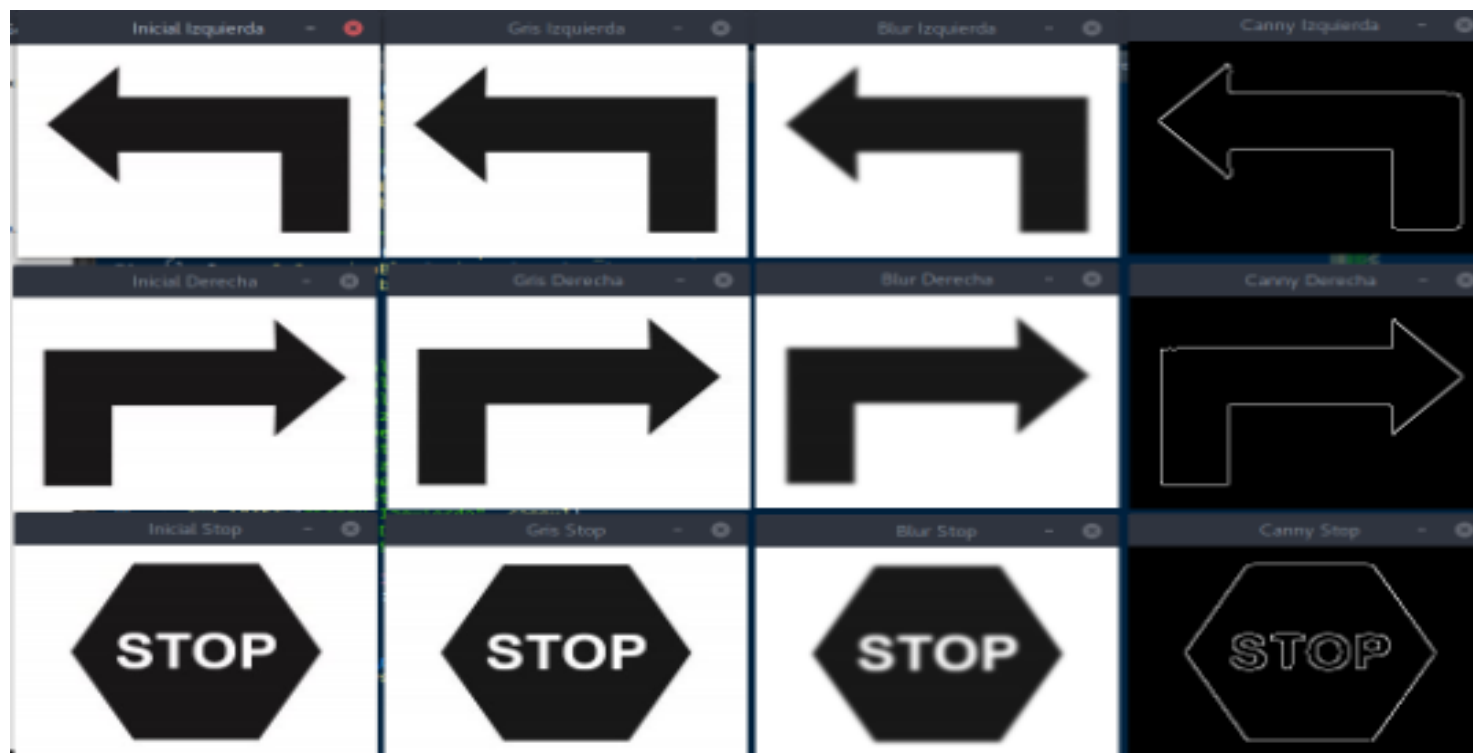
# PROCESAMIENTO DE IMÁGENES: ESQUEMA GENERAL



# PROCESAMIENTO DE IMÁGENES: TRATAMIENTO DE IMAGEN DE LA CÁMARA



# PROCESAMIENTO DE IMÁGENES: TRATAMIENTO DE IMÁGENES DE MUESTRA



# PROCESAMIENTO DE IMÁGENES: COMPARACIÓN DE IMÁGENES

## Error Cuadrático Medio. (MSE)

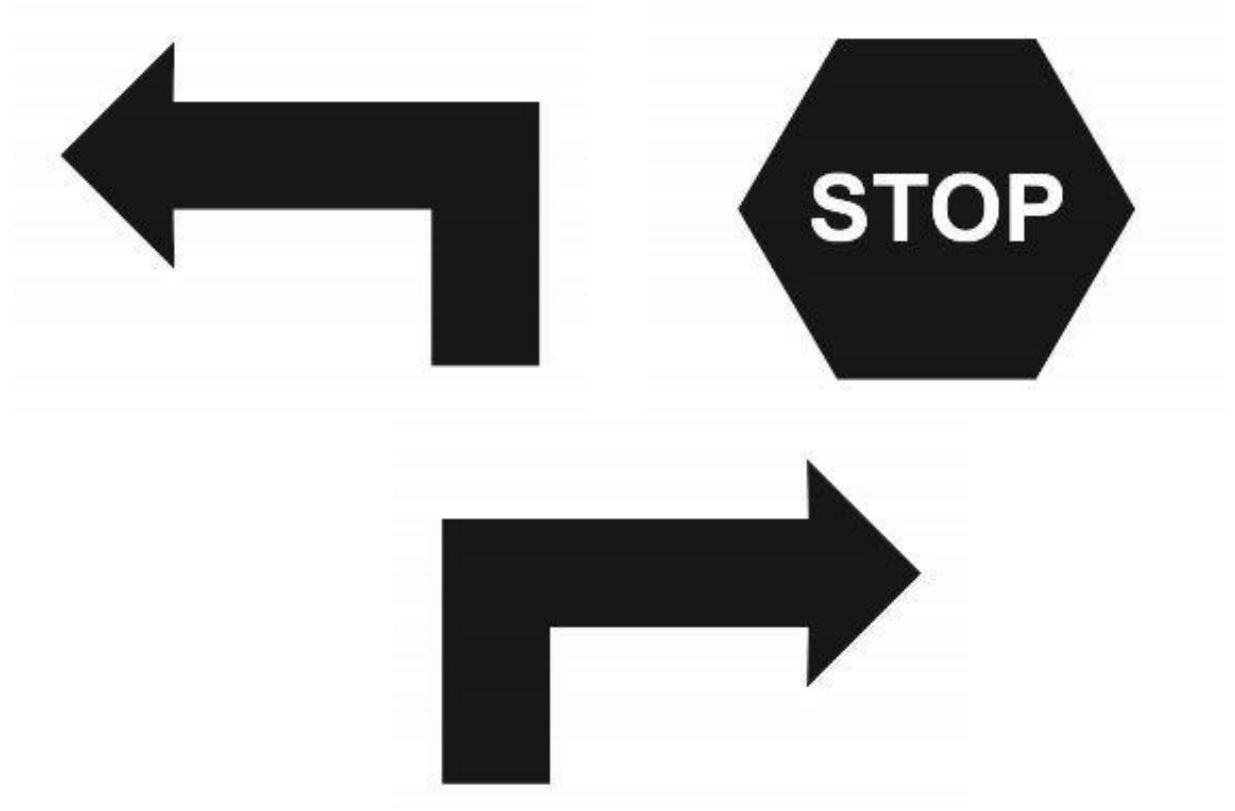
- Es el promedio de los errores al cuadrado.
- Muy usado en estadística.
- A menor error, mayor similitud.
- Se aplica la fórmula.

## Índice de similitud estructural. (SSIM)

- Mide la similitud entre dos imágenes.
- Calcular la calidad de imagen emitida por televisión.
- A menor error, mayor similitud.
- Método de la librería *skimage*.



# PROCESAMIENTO DE IMÁGENES: COMPARACIÓN DE IMÁGENES



# PROCESAMIENTO DE IMÁGENES: COMPARACIÓN DE IMÁGENES

Siendo: M1, M2 y M3

Error Cuadrático Medio al comparar:

1. Original con Giro Izquierda.
2. Original con Giro Derecha.
3. Original con Stop.

Siendo: S1, S2 y S3

Índice de Similitud Estructural:

1. Original con Giro Izquierda.
2. Original con Giro Derecha.
3. Original con Stop.

M1:  
3871.95462201

M2:  
3753.92453401

M3:  
4583.40679345

S1:  
0.616284588556

S2:  
0.645181809892

S3:  
0.538743503856  
Stop

Comparando  
con MSE

M1:  
3871.95462201

M2:  
3753.92453401

M3:  
4583.40679345

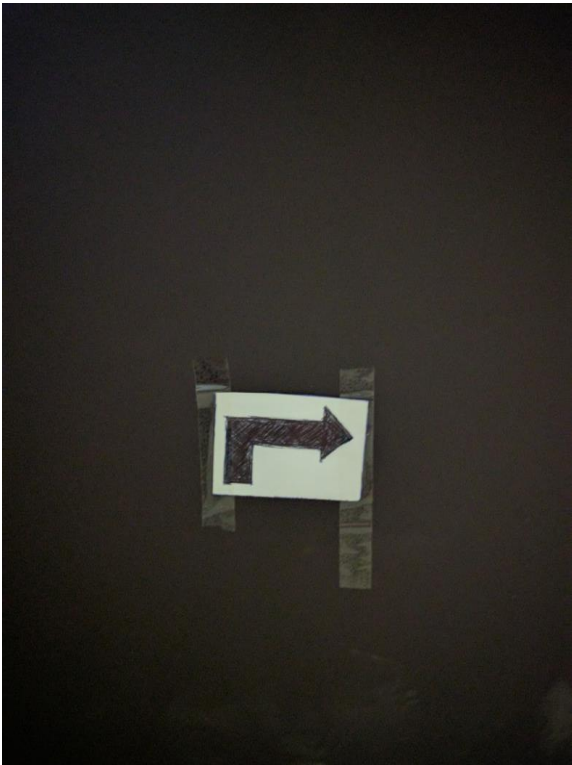
S1:  
0.616284588556

S2:  
0.645181809892

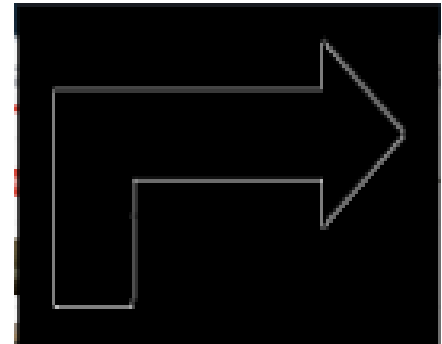
S3:  
0.538743503856  
Giro Derecha

Comparando  
con SSIM

# PROCESAMIENTO DE IMÁGENES: EXPERIMENTO

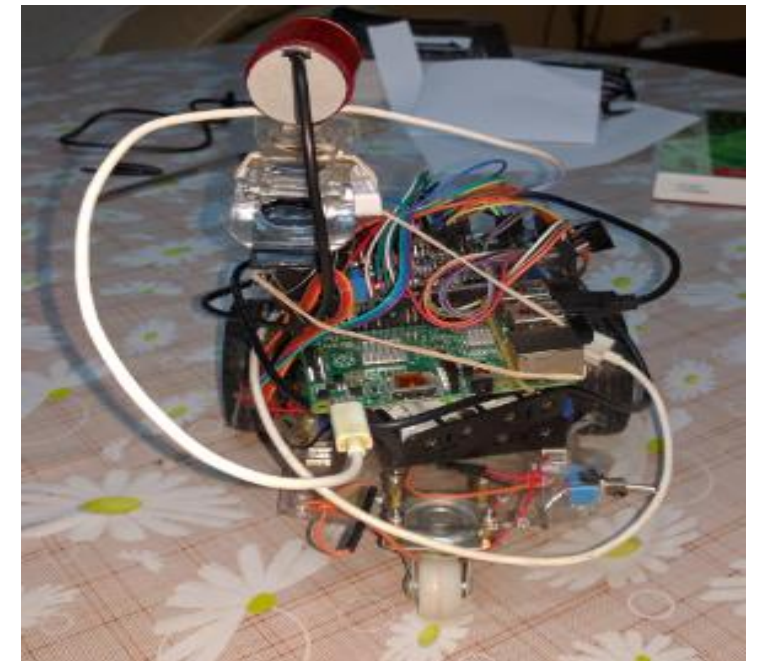
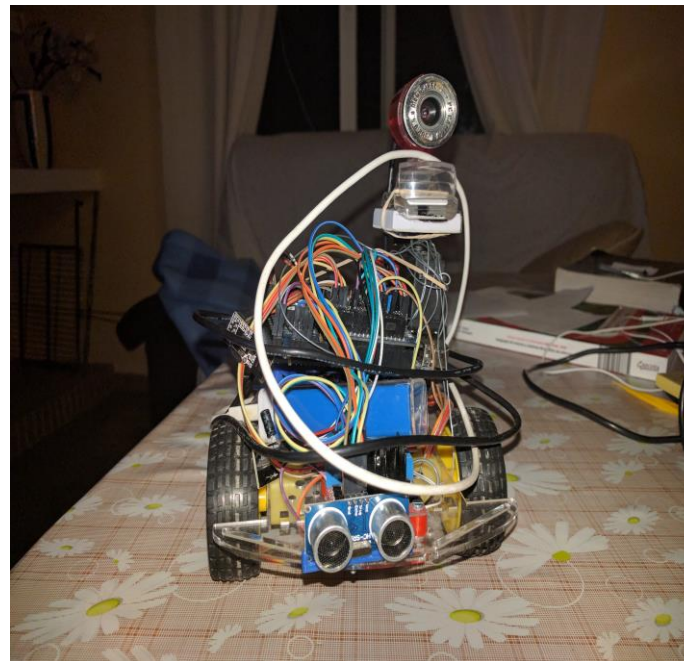
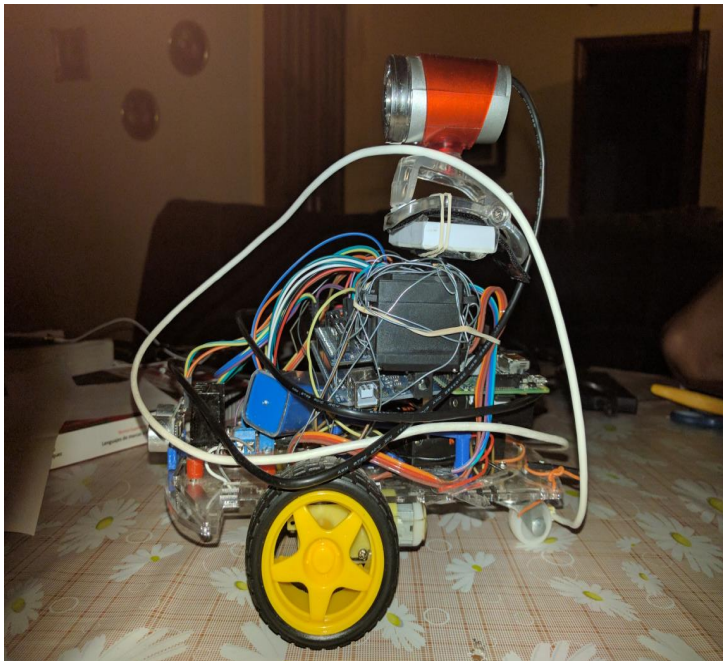


S1:  
 0.616284588556  
  
 S2:  
 0.645181809892  
  
 S3:  
 0.538743503856  
 Giro Derecha

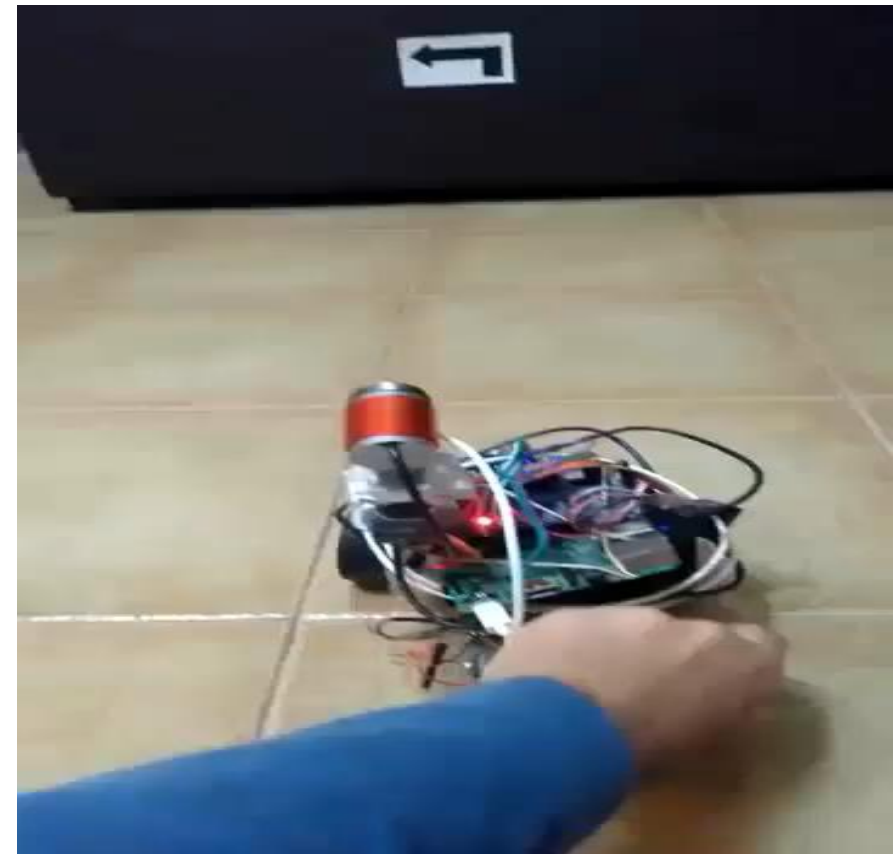
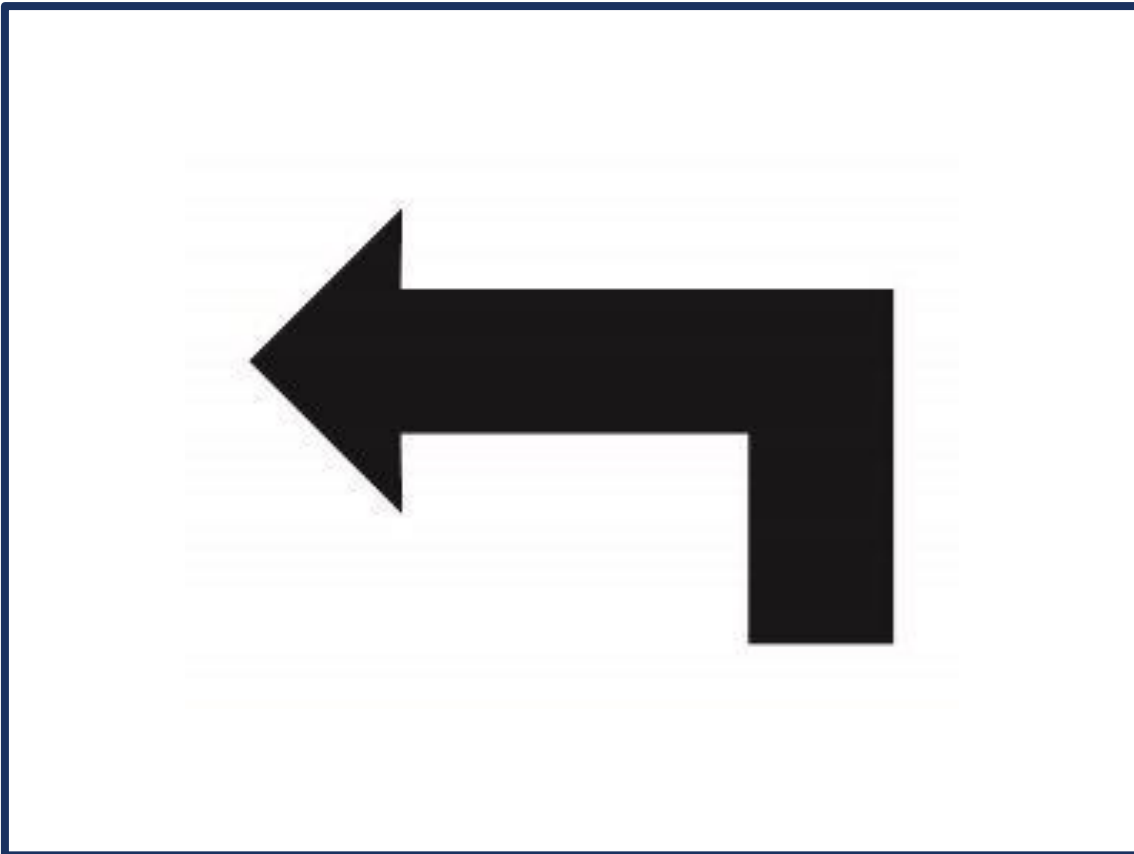


# RESULTADOS EXPERIMENTALES

## Robot móvil

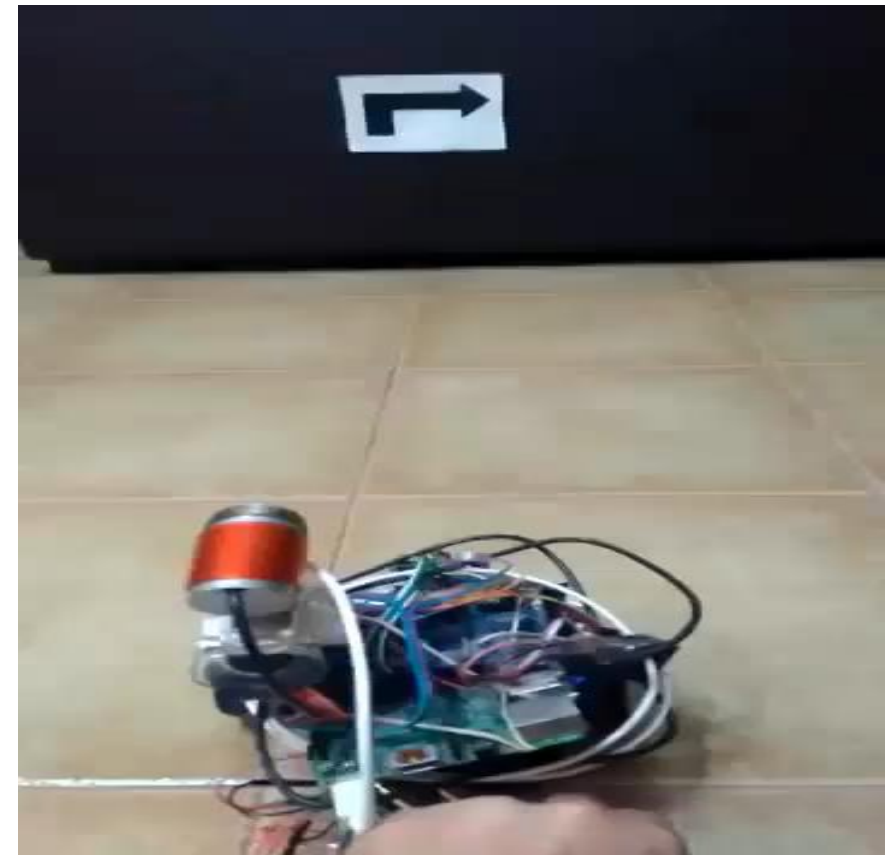
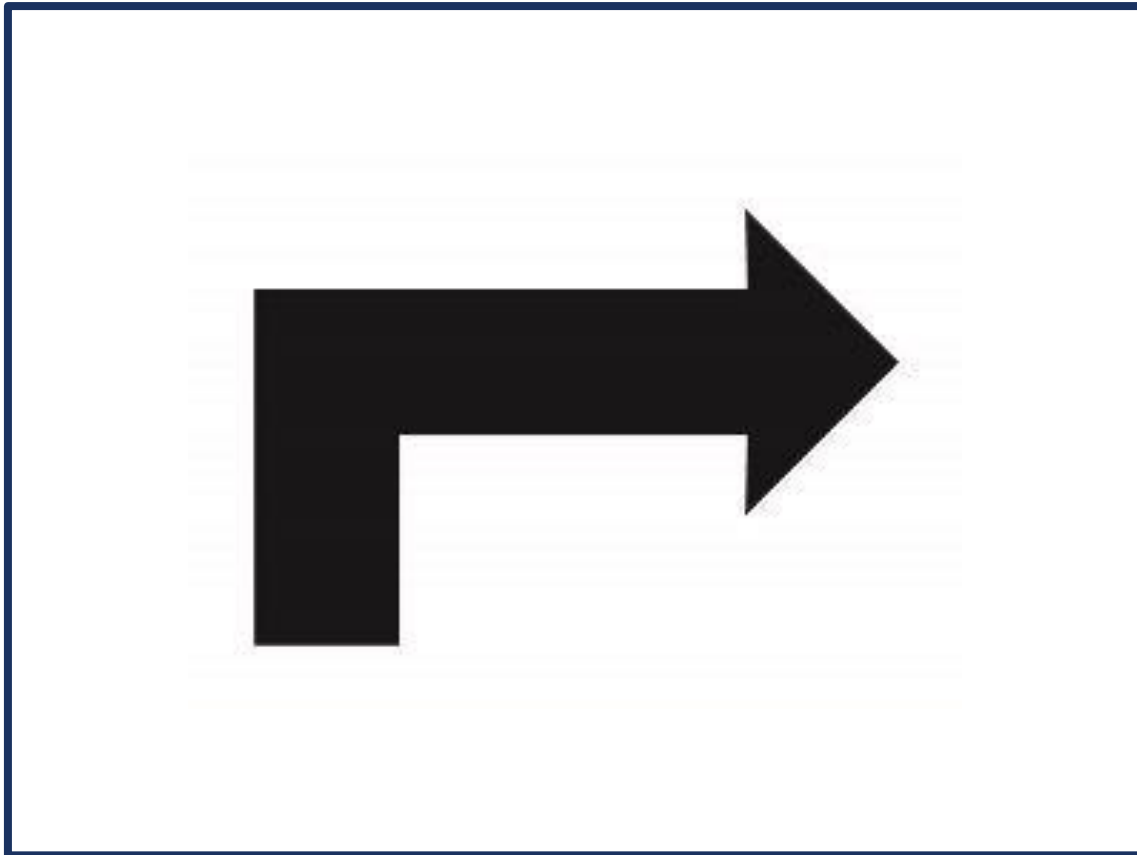


## RESULTADOS EXPERIMENTALES: GIRO IZQUIERDA 90°





## RESULTADOS EXPERIMENTALES: GIRO DERECHA 90°

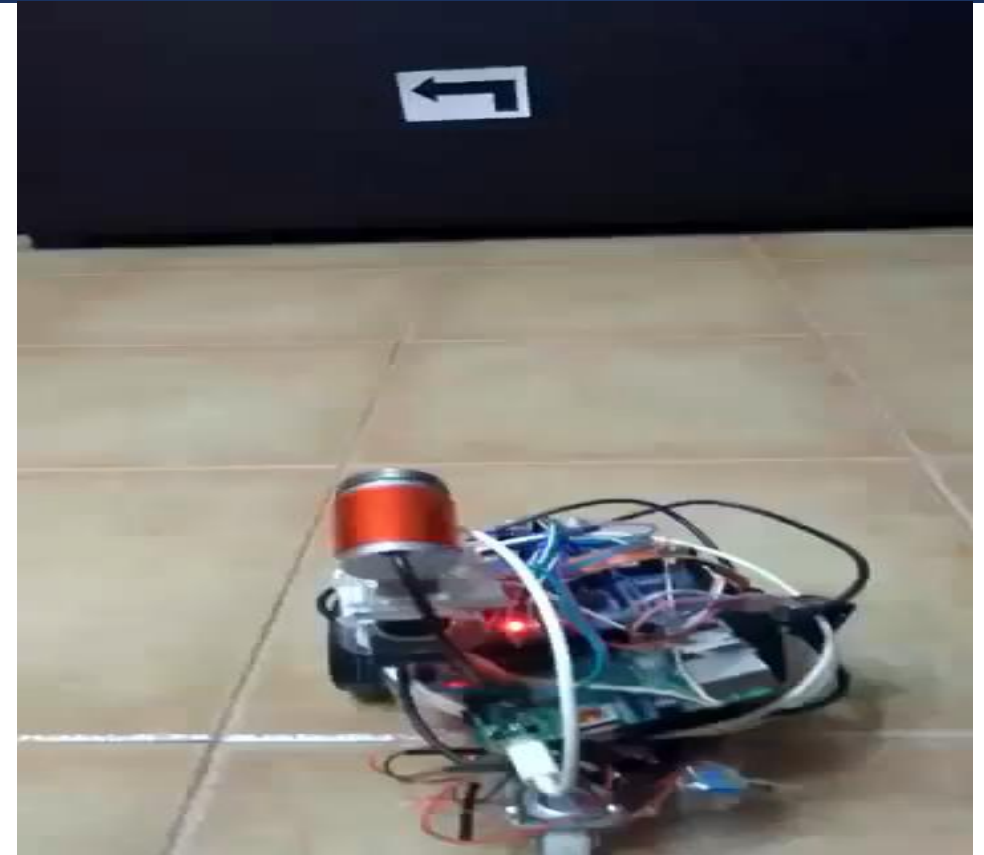




## RESULTADOS EXPERIMENTALES: SEÑAL STOP. FIN



## RESULTADOS EXPERIMENTALES



# CONCLUSIONES

## Robot móvil

### ■ Problemas:

1. Errores de odometría.
2. Error de alimentación.

### ■ Posibles ampliaciones:

1. Poder dirigirlo de manera remota.
2. Recibir órdenes desde un mando de radiofrecuencia.
3. Añadir fácilmente más funciones de movimiento.

## Procesamiento de imágenes

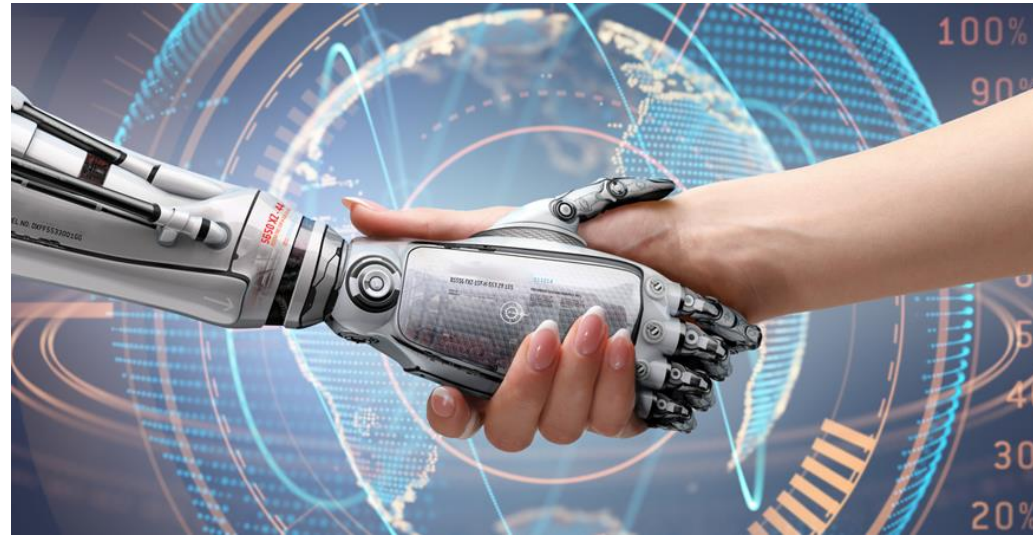
### ■ Problemas:

1. Errores de iluminación.
2. Errores de ángulo de visión.
3. Necesidad de contraste para reconocer señales.

### ■ Posibles ampliaciones:

1. Añadir servomotor para ampliar ángulo de visión.
2. Ampliar número de señales.

# Gracias por su atención



Francisco Márquez Chaves  
[fran29400@gmail.com](mailto:fran29400@gmail.com)