



*Universidad Nacional de Lomas de Zamora*  
*Facultad de Ingeniería*



# *Proyecto en Ingeniería* *Mecatrónica*

## *Vehículo de Vigilancia* *Remoto*

***Docentes:*** *Ezequiel Blanca*  
*Juan Ignacio Szombach*  
*Cristian Leandro Lukaszewicz*

***Alumno:*** *Claudio Adrian Gimenez*





## INDICE

|   |    |
|---|----|
| <b>PROPUESTA FINAL</b>                                    | 4  |
| Motivación de la propuesta:                               | 4  |
| Descripción de la propuesta:                              | 4  |
| Alcance:  | 4  |
| Elección de docentes:                                     | 5  |
| Especificación de requerimientos:                         | 5  |
| Diseño funcional macro                                    | 6  |
| Estimación de costos:                                     | 6  |
| Listado de componentes                                    | 7  |
| Cuadro comparativo presupuestos solicitados               | 8  |
| <b>DESARROLLO DEL PROTOTIPO</b>                           | 10 |
| ETAPA DE DISEÑO   | 10 |
| Placa base  | 10 |
| <b>Diseño de la carcasa de protección</b>                 | 11 |
| <b>Diseño del conjunto elevador de sensor ultrasónico</b> | 11 |
| <b>Diseño del soporte/espalda de sensores infrarrojos</b> | 12 |
| <b>COMPONENTES</b>  | 13 |
| Placa de desarrollo Arduino mega 2560                     | 13 |
| Placa de desarrollo Esp32cam                              | 13 |
| Regulador Step down                                       | 13 |
| Servomotor SG90   | 14 |
| Sensor de lluvia  | 14 |
| Sensor ultrasonido  | 14 |
| Sensor infrarrojo   | 14 |
| Driver Puente H L298N                                     | 15 |
| Motor reductor de corriente continua con rueda Arduino    | 15 |
| Pilas 3,7 v – 18650                                       | 15 |
| Regulador cargador de baterías de litio                   | 16 |
| Cargador externo  | 16 |
| Consumo de los componentes                                | 16 |
| <b>CONEXIÓN DE LOS COMPONENTES</b>                        | 17 |
| Placa base con componentes                                | 19 |



|  |   |    |
|--|---|----|
| Vehículo completo  | Vista frontal donde observamos el sensor ultrasónico..... | 19 |
| DESCRIPCIÓN FUNCIONAL MACRO.....   |   | 20 |
| DESCRIPCIÓN DEL FUNCIONAMIENTO DE LOS COMPONENTES .....                                |   | 20 |
| Función de los sensores infrarrojos .....  |   | 20 |
| Observando de frente al vehículo, esta es la secuencia de errores .....                |   | 21 |
| Conjunto sensor ultrasónico (US), servomotor, porta US y plataforma de elevación ..... |   | 22 |
| Sensor de lluvia.....  |   | 23 |
| Driver puente H L298n .....  |   | 23 |
| Función del Arduino Mega 2560 .....  |   | 23 |
| ETAPA DE PROGRAMACIÓN .....  |   | 24 |
| Fragmento de código para Arduino mega 2560.....  |   | 25 |
| Fragmento de código del ESP32cam .....   |   | 30 |
| ETAPA DE ARMADO Y CABLEADO DEL PROTOTIPO .....   |   | 33 |
| Tiempo estimado Vs Tiempo real .....   |   | 39 |
| Costo estimado Vs Costo real .....   |   | 40 |
| MEJORAS PARA EL PROTOTIPO .....  |   | 41 |
| CONCLUSIÓN.....  |   | 42 |
| AGRADECIMIENTOS: .....   |   | 43 |
| ANEXOS .....   |   | 43 |
| SOFTWARE UTILIZADO .....   |   | 43 |
| BIBLIOGRAFÍA, PÁGINAS DE INTERNET CONSULTADAS .....                                    |   | 44 |

### **PROPUESTA FINAL**

#### **Motivación de la propuesta:**

Da origen al presente proyecto los problemas de inseguridad que se da tanto en predios industriales, en los que los materiales y/o elementos representen un valor material, y otras instalaciones como ser patios de establecimientos educativos, patios de hogares, barrios cerrados. En cuyos casos es óptima la utilización de un sistema de seguridad que no sea muy sofisticado en cuanto tecnología de punta en sus cámaras, y además que su instalación y mantenimiento no sean costosos.

En la actualidad se encuentran numerosos proyectos similares al expuesto. Con este proyecto lo que se busca es aplicar los conocimientos adquiridos a lo largo de carrera y enfocarlos en esta disciplina que tanto hace nuestras vidas más eficientes, la cual es la automatización. Puesto en práctica la solución a la problemática planteada, la misma pretende aminorar el riesgo del personal de seguridad.

#### **Descripción de la propuesta:**

Se optó este proyecto dado un listado ofrecido por la cátedra.

Un vehículo de pequeñas dimensiones, liviano, autónomo, que sea capaz de recorrer un circuito dado, a una velocidad acorde, hacer inspecciones en lugares determinados, esquivar obstáculos que se encuentren en su camino. Que en presencia de lluvia active una protección para los sensores.

Como complemento de un sistema de seguridad, le darán utilidad las personas encargadas de la seguridad de los sectores, ya que el mismo, mediante una cámara, enviara imágenes del recorrido que realiza en tiempo real.

#### **Alcance:**

En principio se estima que el prototipo será presentado en su totalidad al final de la cursada de la cátedra, teniendo en cuenta que es a fines demostrativos.

El vehículo será a escala real ya que el mismo no presentará grandes dimensiones (30x20cm aprox.).



*Ilustración 1-imagen de prototipo*



### Elección de docentes:

En cuanto a la elección de docentes a los cuales se realizarán consultas, se tiene en cuenta que el prototipo incluye parte mecánica, eléctrica, circuito electrónico, software, diseño de componentes a imprimir en 3D, ellos son los que dictan cátedras tales como Robótica, Electrónica Analógica y Digital, Medios de representación Gráfica.

Docentes:

- Mauro Maceda
- Diego Lopez
- Matrin Torres

### Especificación de requerimientos:

- Vehículo de 4 ruedas.
- Dimensiones:
  - o Ancho:  $\approx 25$  cm
  - o Largo:  $\approx 35$  cm
  - o Alto:  $\approx 20$  cm
- Con baterías recargables.
- Autónomo: mediante sensores, seguirá un circuito establecido.
- Autonomía:  $\approx 4$ h.
- Distancia de desplazamiento:  $\approx 200$  m.
- De estructura plástica impresa en 3D: una base donde estarán dispuestos los componentes electrónicos; carcasa para proteger los componentes. Base y carcasa estarán unidas mediante tornillos.
- Con detección ante la interposición de obstáculos. Ante este suceso rodeara el obstáculo y volverá al circuito.
- Con detección ante la presencia de lluvia: ante este suceso, se activara una protección del sensor de detección de obstáculos.
- Detención ante la presencia de posible inundación: ante este suceso, activara un aviso al usuario en la interfaz del teléfono móvil.
- Interfaz con el usuario mediante teléfono móvil o monitor al cual enviara imagen en tiempo real, y en caso de lluvia, poder manejar el vehículo desde el teléfono móvil.
- Se mediará la carga de la batería mediante la utilización de diodos LED de colores para identificar en forma visual la cantidad de carga.

### Diseño funcional macro

Mediante diagrama de bloque se expone el diseño funcional

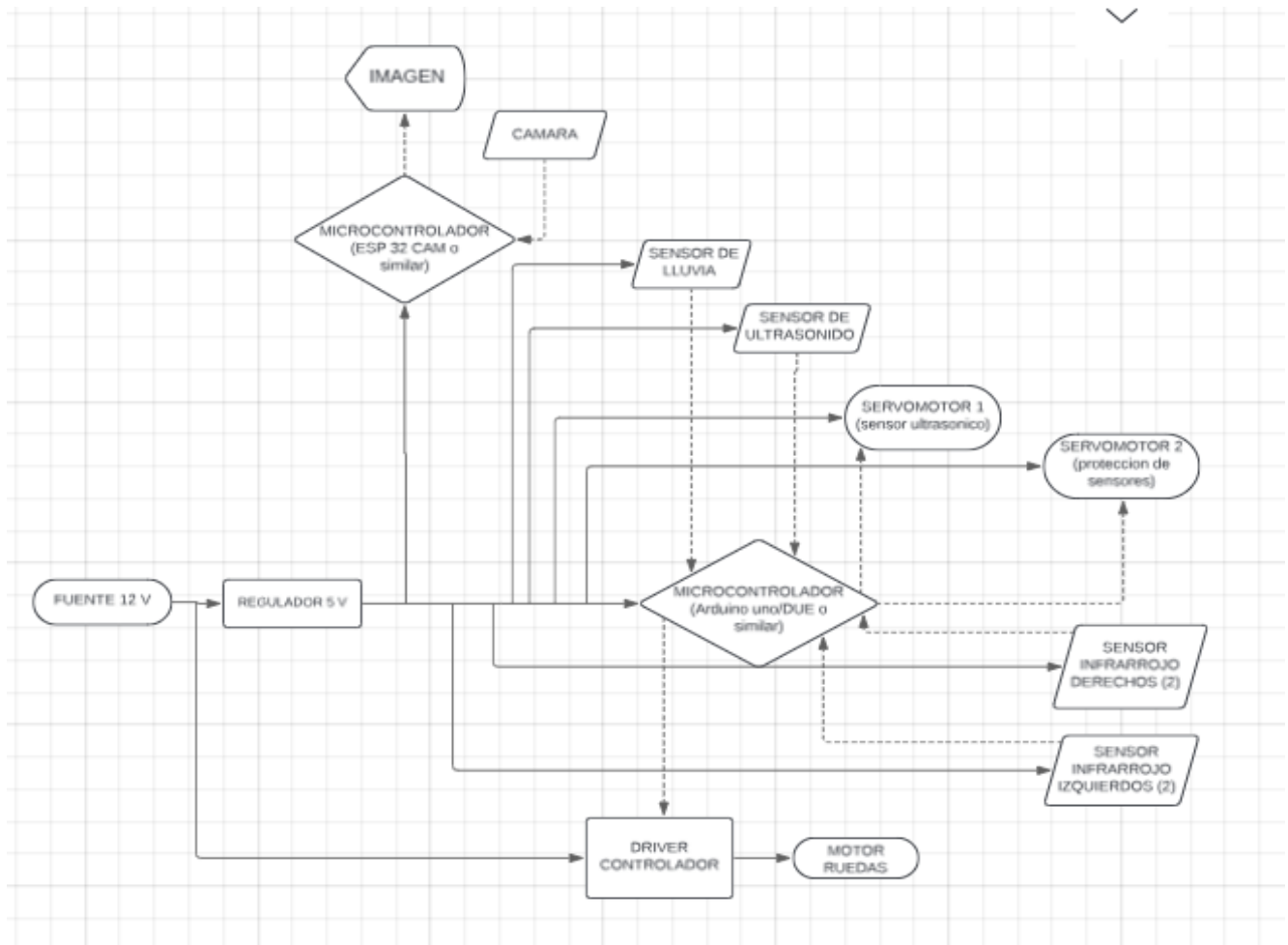


Ilustración 2 - Diseño funcional macro

El prototipo constara un driver controlador, dos servomotores y 4 sensores, los cuales utilizaran 12 pines. La placa de desarrollo **Arduino Uno** posee la cantidad de 14 pines digitales y 6 pines analógicos, suficientes para lograr el funcionamiento.

### Estimación de costos

Listado de posibles proveedores:

- Microelectrónica S.H.
- Starware
- SYC semiconductores y componentes

En la estimación de costos se tiene en cuenta los componentes constructivos y los materiales necesarios para las conexiones de los mismos.



### Listado de componentes

| Componente   |
|--|
| Rueda robot para Arduino – Llanta plástica - con motor |
| Mini servomotor estándar                               |
| Módulo ESP32 CAM                                       |
| Módulo micro controlador Arduino mega 2560 (con cable) |
| Modulo Driver L298N                                    |
| Cables conectores macho-macho 20cm X 40 unidades       |
| Cables conectores macho-hembra 20cm X 40 unidades      |
| Cables conectores hembra - hembra 20cm X 40 unidades   |
| Módulo Arduino Sensor ultrasonido                      |
| Modulo sensor infrarrojo                               |
| Modulo Sensor de lluvia/nivel de agua                  |
| Protoboard mediano                                     |
| Estaño para soldadura electrónica rollo 1mm X 250g     |
| Soldador de estaño tipo lápiz con punta cerámica       |
| Batería Recargable 18650-3.7v 2600mah                  |
| Cargador de batería litio                              |
| Regulador LM7805                                       |
| Disipador de calor 1 cm x 8.5 mm x 8.5 mm              |
| Capacitor 470 uf                                       |
| Capacitor 1uf cerámico x 10 unidades                   |
| Resistencia 1 kΩ                                       |
| Diodo led color rojo                                   |
| Diodo led color verde                                  |
| Diodo led color amarillo                               |
| Diodo 1N4007 – pack por 50 unidades                    |
| Filamento PLA para impresión 3D 1.75mmX1kg             |

Tabla 1 - Lista de componentes





### Cuadro comparativo presupuestos solicitados

| Componente   | SYC |            | STARWARE     |      | MERCADO LIBRE                       |      |
|--|-----|------------|--------------|------|-------------------------------------|------|
|  | \$  | U\$S       | \$           | U\$S | \$                                  | U\$S |
| Rueda robot para Arduino – Llanta plástica - con motor |     | 2,7        | \$ 5.700,00  |      | \$ 4.080,00                         |      |
| Mini servomotor estándar                               |     | 0,7        | \$ 3.129,00  |      | \$ 4.720,00                         |      |
| Módulo ESP32 CAM                                       |     |            | \$ 14.889,00 |      | \$ 10.990,00                        |      |
| Módulo micro controlador Arduino mega 2560 (con cable) |     | 25         | \$ 29.999,00 |      | \$ 27.178,00                        |      |
| Modulo Driver L298N                                    |     | 3,25       | \$ 2.679,00  |      | \$ 4.550,00                         |      |
| Cables conectores macho-macho 20cm X 40 unidades       |     | 2,5        | \$ 2.679,00  |      | \$ 12.522,00                        |      |
| Cables conectores macho-hembra 20cm X 40 unidades      |     | 2,5        | \$ 2.389,00  |      |                                     |      |
| Cables conectores hembra - hembra 20cm X 40 unidades   |     | 2,5        |              |      |                                     |      |
| <a href="#">Módulo Arduino Sensor ultrasonido</a>      |     | 1,7        | \$ 1.969,00  |      | \$ 3.219,00                         |      |
| Modulo sensor infrarrojo                               |     | 1,6        | \$ 869,00    |      | \$ 1.964,00                         |      |
| Modulo Sensor de lluvia/nivel de agua                  |     | 2          | \$ 1.659,00  |      | \$ 2.304,00                         |      |
| Protoboard mediano                                     |     | 2,6        | \$ 1.179,00  |      | \$ 2.670,00                         |      |
| Estaño para soldadura electrónica rollo 1mm X 250g     |     | 21         |              |      | \$ 23.790,00                        |      |
| Soldador de estaño tipo lápiz con punta cerámica       |     | 6,8        |              |      | \$ 7.501,00                         |      |
| <a href="#">Batería Recargable 18650-3.7v 2600mah</a>  |     |            |              |      | \$ 3.750,00                         |      |
| Cargador de batería litio                              |     | 4          | \$ 11.769,00 |      | \$ 10.169,00                        |      |
| Regulador LM7805                                       |     | 0,5        |              |      | \$ 2.016,00                         |      |
| Disipador de calor 1 cm x 8.5 mm x 8.5 mm              |     | 1,05       |              |      |                                     |      |
| Capacitor 470 uf 12V                                   |     | 0,1        |              |      |                                     |      |
| Capacitor 1uf cerámico x 10 unidades                   |     | 0,8        |              |      |                                     |      |
| Resistencia 1 kΩ (1/\$ W x 50 u)                       |     | 2,5        |              |      | \$ 2.019,00                         |      |
| Diodo LED color rojo                                   |     | 0,08       |              |      | \$ 1.999,00<br>(redondo 5 mm x10 u) |      |
| Diodo LED color verde                                  |     | 0,08       |              |      | \$ 1.999,00<br>(redondo 5 mm x10 u) |      |
| Diodo LED color amarillo                               |     | 0,08       |              |      | \$ 1.999,00<br>(redondo 5 mm x10 u) |      |
| Diodo 1N4007 (x 50 u)                                  |     | 1,25       |              |      | \$ 2.567,00                         |      |
| Filamento PLA para impresión 3D 1.75mmX1kg             |     |            |              |      | \$ 27.900,00                        |      |
| Totales  | 0   | U\$S 85,29 | \$ 78.909,00 | 0    | \$ 159.906,00                       | 0    |

Tabla 2 - Cuadro comparativo de precios



A modo de comparación, se tomó además los precios de los componentes y demás elementos extraídos del sitio web: [www.mercadolibre.com.ar](http://www.mercadolibre.com.ar), al día 05/04/2025

El valor del dólar según el sitio web del Banco Nación Argentina al día 04/04/2025:

Dólar U.S.A:    Compra: \$ 1055,50    Venta: 1095,50

Los proveedores disponen de los componentes en inventarios. La entrega dependerá de los horarios de atención al público del local comercial; además ofrecen la posibilidad del realizar envíos con un costo adicional, con entrega un (1) día después de realizada la solicitud de compra.

## DESARROLLO DEL PROTOTIPO

En principio, la idea fue desarrollar un vehículo que pueda ser manejado mediante un teléfono móvil o notebook. Sin embargo, se estableció que debía realizar el recorrido de un circuito fijo, por ello se optó por realizar un sistema de seguidor de línea a fin de cumplir con lo fijado.

Para ello, con software de diseño, se logró realizar la estructura base del vehículo, el posicionamiento de los componentes, sujetores de los mismos, y la carcasa que protege a los componentes.

Las tareas a realizar fueron divididas en etapas definidas.

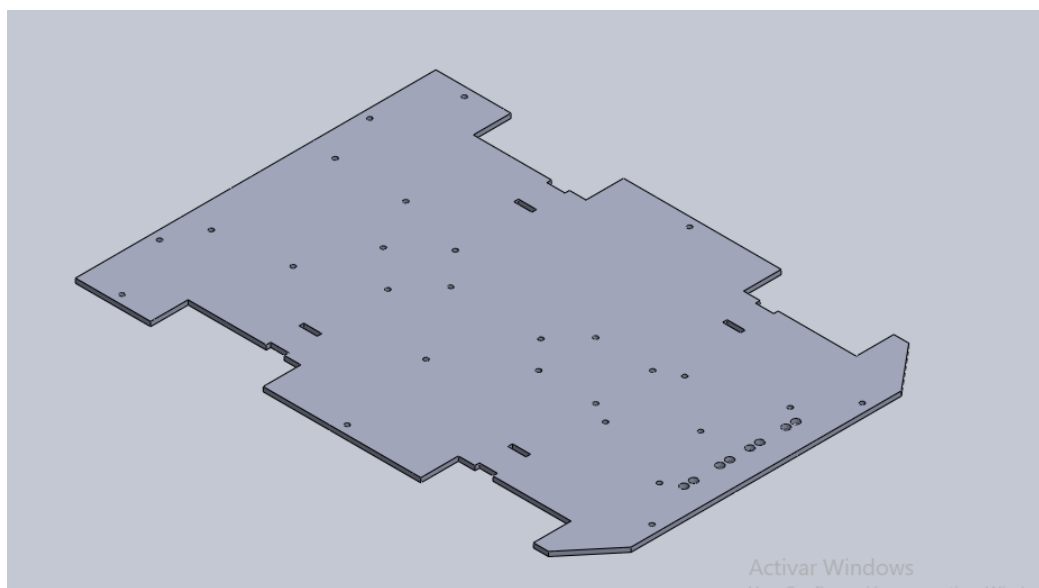
### ETAPA DE DISEÑO

Diseño de la base para los componentes

Para cumplir con la condición de parte mecánica del proyecto, se diseña placa base, carcasa y el conjunto elevador de sensor ultrasónico, en software de diseño SolidWorks.

Para la placa base se consideran las dimensiones de los componentes que serán utilizados, según la información suministrada por la hoja de datos de cada uno de ellos, en cuanto a las medidas, el posicionamiento de los mismos, se realizan los orificios de 3mm para los tornillos de sujeción para la carcasa, Arduino mega, puente H I298n, el porta pilas, el sensor de lluvia y reguladores Step down, las ranuras de las ruedas, el conjunto elevador de sensor ultrasónico, y los orificios de 5mm donde estarán los diodos de los sensores infrarrojos. Se disponen en línea, longitudinal al centro, y en el centro de placa, se colocan las pilas/batería de litio recargable, ya que este es el conjunto con más peso.

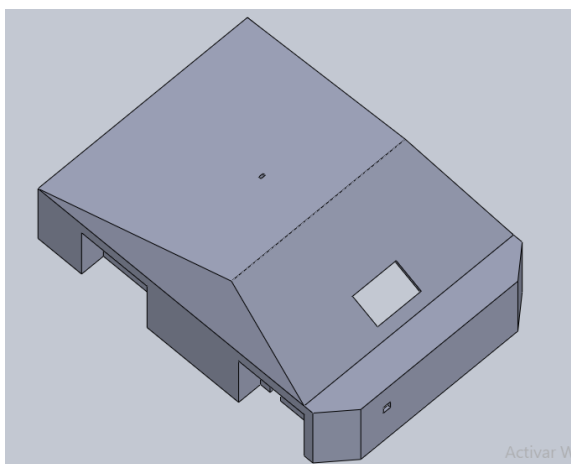
Placa base



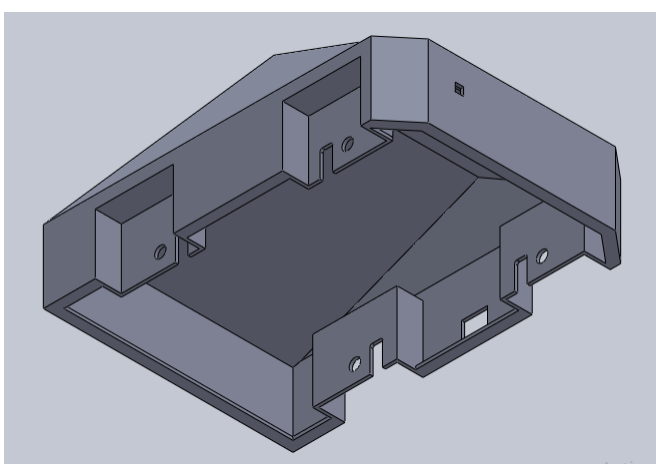
*Ilustración 3 - Diseño de la placa base de componentes*

**Diseño de la carcasa de protección**

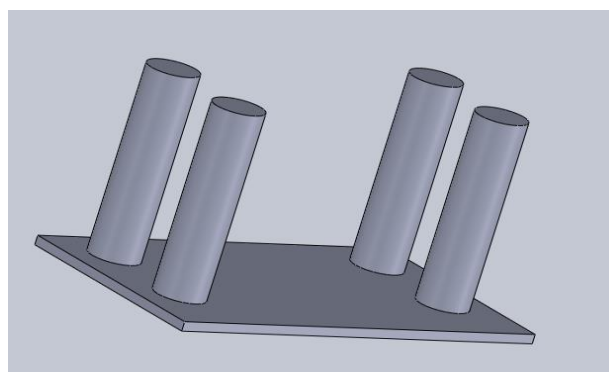
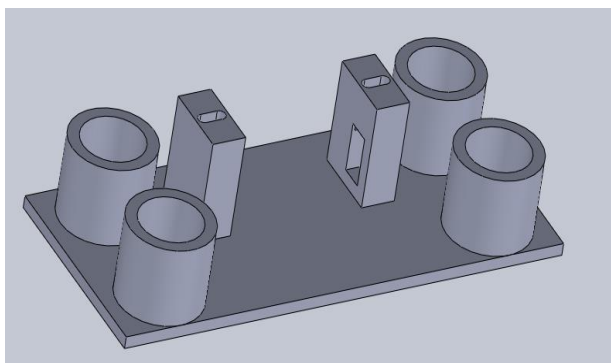
Para el diseño de la carcasa de protección se tuvo en cuenta la sujeción mediante tornillos a la placa base, para ello es el reborde inferior interior, los orificios laterales para las ruedas y su sujeción, switch de encendido y el enchufe del cargador de batería, frontal para la cámara y el flash del ESP32cam, superior trasero para el cable del sensor de lluvia exterior, y superior delantero donde estará el conjunto sensor ultrasónico.

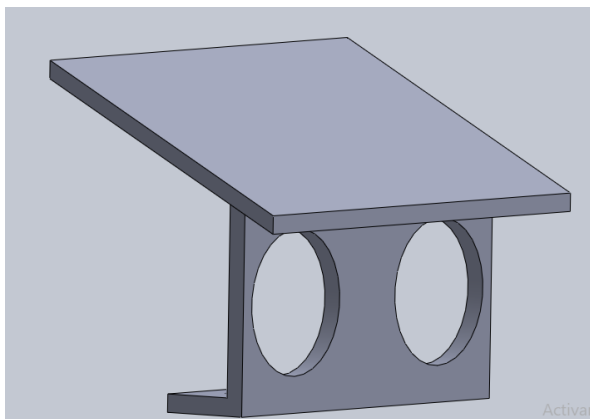
**Carcasa**

*Ilustración 4 - Carcasa-Vista en perspectiva superior*



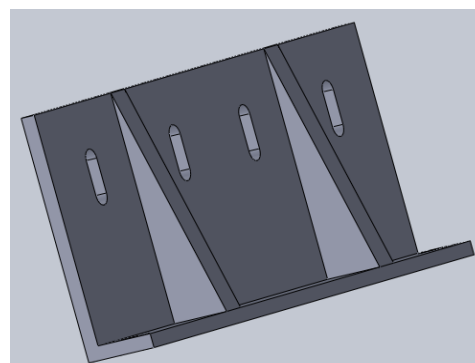
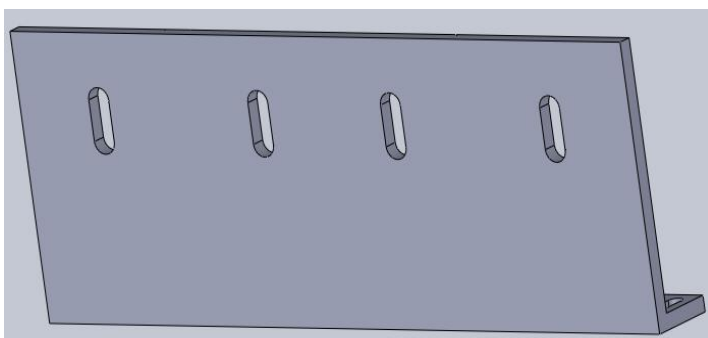
*Ilustración 5 - Carcasa-Vista en perspectiva inferior*

**Diseño del conjunto elevador de sensor ultrasónico**



*Ilustración 6 - a)-b)-c) conjunto elevador de sensor ultrasónico*

### **Diseño del soporte/espalda de sensores infrarrojos**



*Ilustración 7 a)-b) espaldas de sensores infrarrojos*

En un principio, el prototipo fue pensado para ser impreso al completo en 3d. A medida que se avanzaba el diseño de los componentes plásticos, surgieron los inconvenientes:

Primero, independientemente del espesor de la placa base, no fue conveniente imprimirlo en 3d, ya que es propenso, por su forma y disposición de las fibras de impresión, a sufrir de quebraduras y rajaduras, y además del costo y del tiempo de impresión. Por ello se propuso otro material, acrílico, el cual es más resistente y de fácil corte en maquina laser. Pero el problema fue el costo del mismo. Entonces surgió otro material, más económico pero resistente, y que también se realiza el corte mediante maquina laser, el material seleccionado fue MDF.

En cuanto a la carcasa, a diferencia de la placa base, por su diseño, la idea fue imprimirlo en 3d en un solo bloque. Surgió entonces el segundo problema. El tamaño del prototipo vs. las medidas estándar de las mesas de impresión. Por este motivo, la solución fue dividir la carcasa en varias secciones de impresión, para luego armar el vehículo, unirlo mediante pegamento adecuado para plástico de impresión.

Otra condición obligatoria del proyecto es lo relacionado al circuito electrónico y eléctrico. Por ello se detalla a continuación los componentes utilizados, y en software Proteus se detallan las conexiones entre ellos.

### COMPONENTES

#### Placa de desarrollo Arduino mega 2560

La placa de desarrollo Arduino Mega 2560 es el núcleo de control. Esta elección obedece su versatilidad para la integración de los sensores y componentes, que permite la lectura de las señales en tiempo real, es una solución integral y adaptable a diferentes escenarios.



Ilustración 8 - Arduino mega 2560

#### Placa de desarrollo Esp32cam

Componente seleccionado por su ventaja de poseer una cámara integrada, y de esa manera realizar el monitoreo en tiempo real el recorrido que realiza el vehículo.



Ilustración 9 - ESP32cam

#### Regulador Step down

Se encarga de regular y suministrar la tensión de alimentación a los componentes. Este proyecto cuenta con 2 reguladores, uno para los componentes electrónicos que se manejan con 5 voltios, y otro independiente para la alimentación del Driver Puente H L298n para el manejo de los motores de las ruedas. En el primer prototipo se colocó un regulador integrado 7805 el cual se perdía potencia por disipación de calor. Por ello se optó por el regulador Step down.

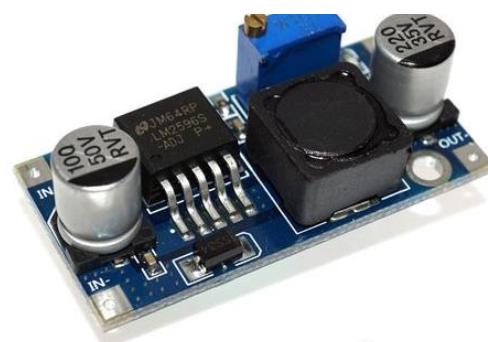


Ilustración 10 - Regulador Step Down

### Servomotor SG90

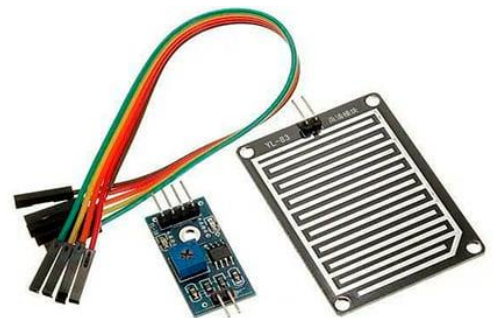
Se encarga del movimiento del porta sensor de ultrasónico, y de elevar y bajar el sistema que sostiene es sensor de distancia.



*Ilustración 11 - Servomotor*

### Sensor de lluvia

En caso de la presencia de lluvia, se encarga de advertir y enviar la señal este suceso al Arduino.



*Ilustración 12 - Sensor de lluvia*

### Sensor ultrasonido

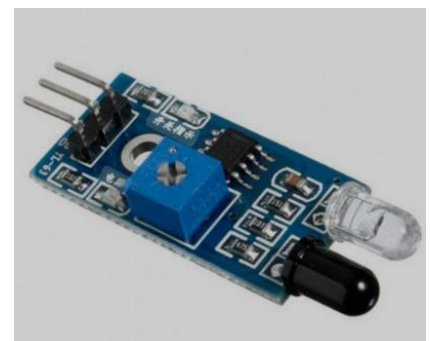
Encargado de verificar que no hay objetos que interrumpan el paso del vehículo.



*Ilustración 13 - Sensor ultrasónico*

### Sensor infrarrojo

Encargado de la lectura del circuito a seguir.



*Ilustración 14 - Sensor infrarrojo*



### Driver Puente H L298N

Encargado de manejar la velocidad de los motores según la información que recibe del Arduino mega 2560.

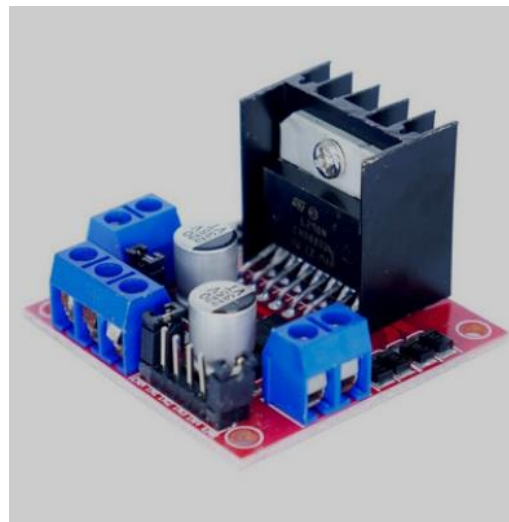


Ilustración 15 - Driver puente H

### Motor reductor de corriente continua con rueda Arduino

Para realizar el recorrido, es indispensable. Cumple, a través del driver puente H, las ordenes enviadas por el Arduino. Su tensión de trabajo es de 3 a 12v, la cual no es conveniente superar. Su tensión óptima es de 6 a 8v.



Ilustración 16 - Motor

### Pilas 3,7 v – 18650

Pilas-baterías de litio recargables, utilizamos 4 en serie como fuente de alimentación para abastecer el circuito. En las pruebas realizadas en banco de trabajo, el prototipo funciono durante 1:25 hs. sin interrupción.



Ilustración 17 - Pila-batería de litio recargable



### Regulador cargador de baterías de litio

Para la carga de las baterías mediante cargador externo.

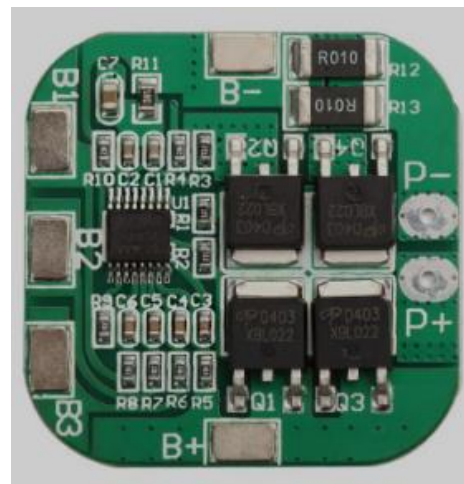


Ilustración 18 - modulo cargador de batería

### Cargador externo

Cargador de notebook externo, utilizado para alimentar la placa regulador cargador de litio



Ilustración 19 - Cargador estándar

### Consumo de los componentes

| Componente                                     | Cantidad       | Consumo total (mA) |
|--|----------------|--------------------|
| ESP32cam                                       | 1              | 310                |
| Arduino Mega 2560                              | 1              | 200                |
| Sensor ultrasónico                             | 1              | 15                 |
| Sensor infrarrojo                              | 4 (25 mA c/u)  | 100                |
| Sensor de lluvia                               | 1              | 15                 |
| Servomotor                                     | 2 (250 mA c/u) | 500                |
| Puente H                                       | 1              | 40                 |
| Regulador Step Down                            | 2              | --                 |
| Consumo total del conjunto de componentes (mA) |                | 970 mA (aprox.)    |

Tabla 3 - Consumo de los componentes

## CONEXIÓN DE LOS COMPONENTES

El software **Proteus** proporciona un entorno visual práctico para ilustrar las conexiones de los componentes.

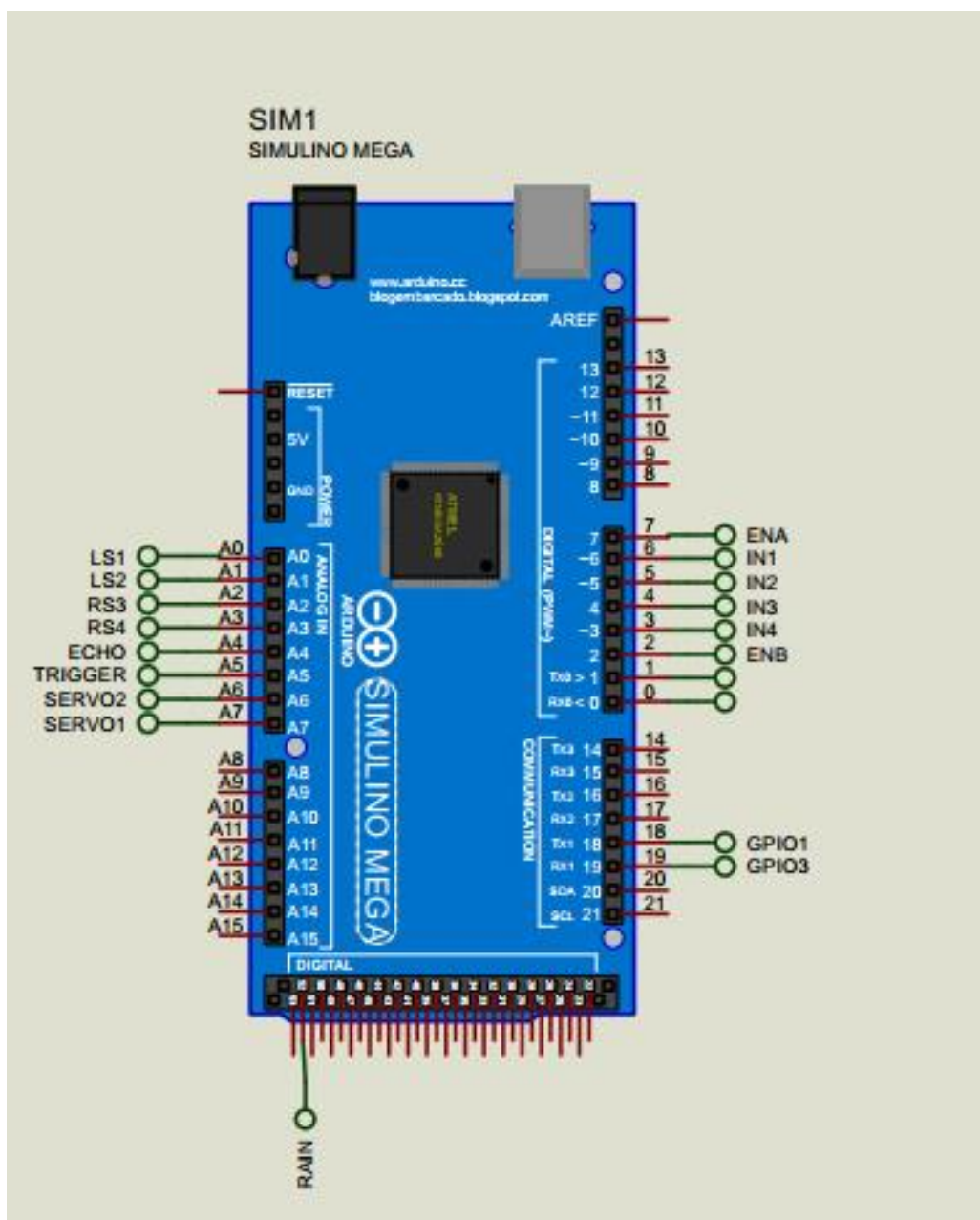


Ilustración 20 - Pines de Arduino a utilizar

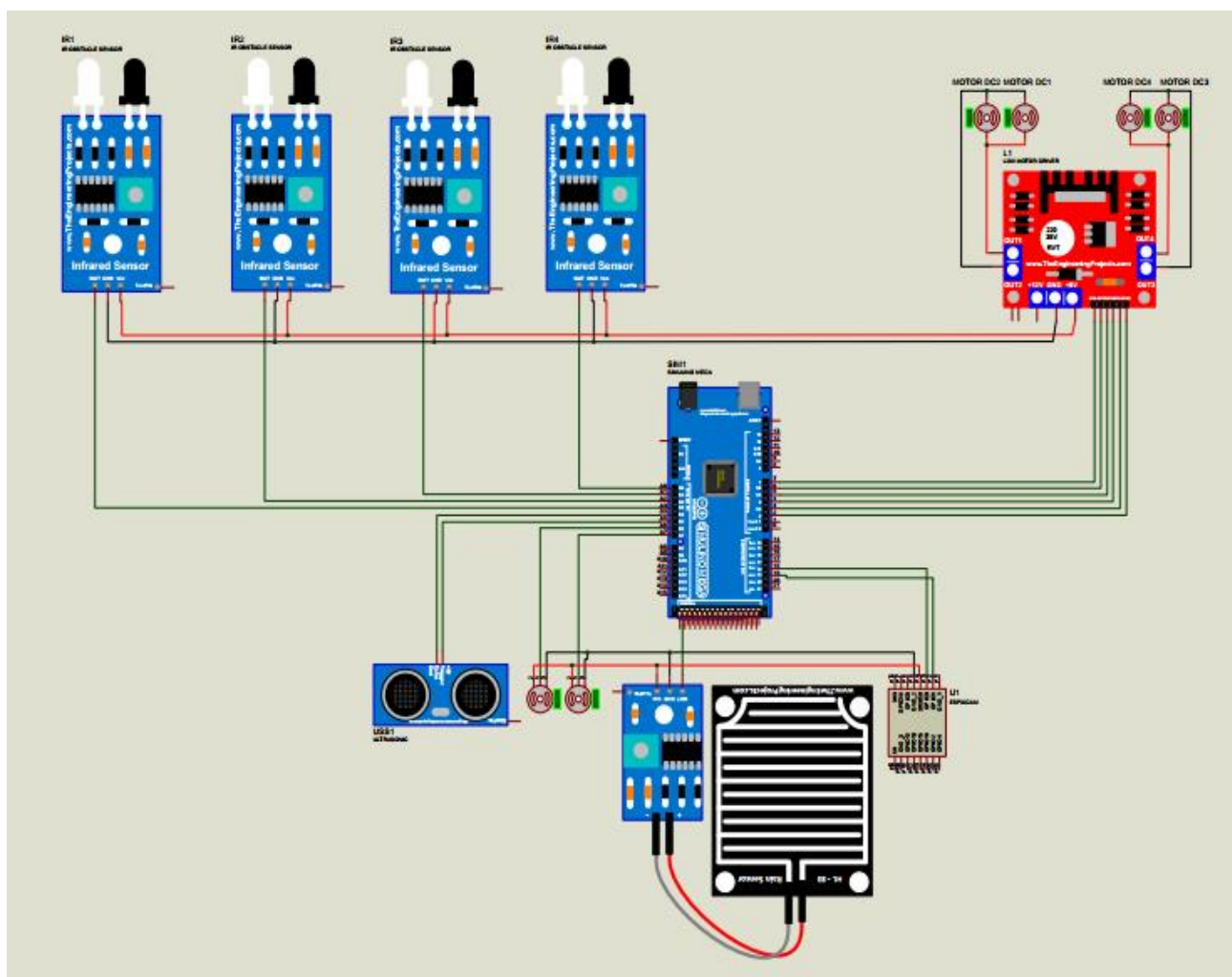


Ilustración 21 - Conexión de los componentes

### Placa base con componentes

En la imagen se observa lo ideal, sin el conexionado con los cables

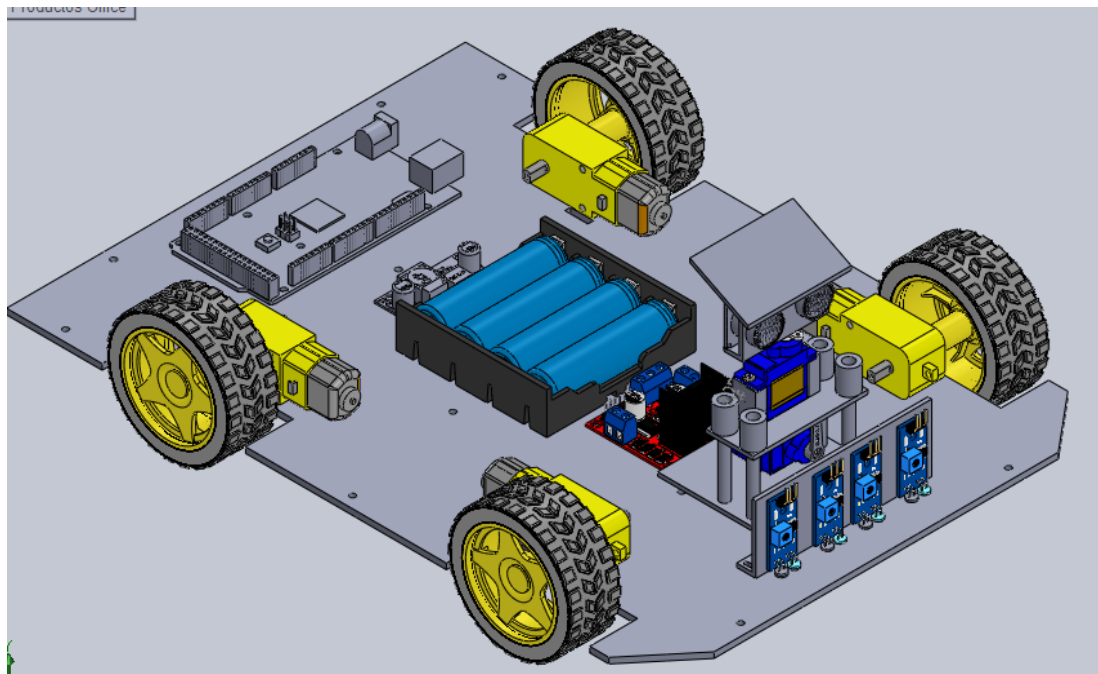


Ilustración 22 - Placa base y componentes

### Vehículo completo

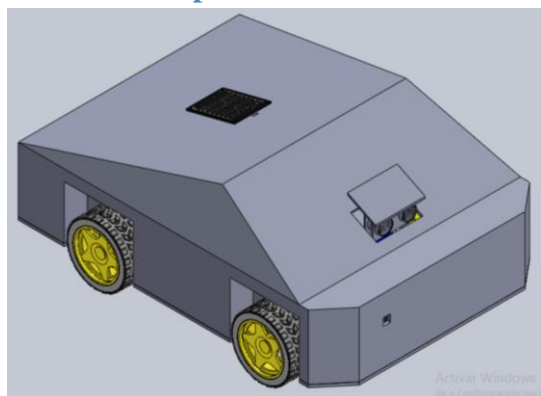


Ilustración 24 - Vehículo completo

### Vista frontal donde observamos el sensor ultrasónico

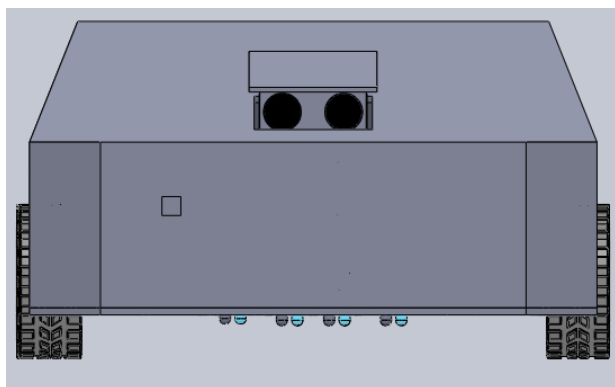


Ilustración 23 - Vehículo-vista frontal

## DESCRIPCIÓN FUNCIONAL MACRO

El vehículo debe seguir un circuito fijo. Para lograrlo, en la primera pruebas se colocaron 2 sensores infrarrojos, logrando así un seguimiento de línea poco eficaz, el cual perdía la línea a medida que se aumentaba la velocidad, a pesar del grosor de la línea.

Para conseguir un mejor seguimiento y velocidad, se agregan más sensores infrarrojos, llegando a un total de 4, los cuales están dispuestos de tal forma que el vehículo siga la línea establecida. Estos sensores, envían señales al Arduino mega. El Arduino procesa la información y envía señales de velocidad al drives l298n. El driver recibe la señal y aumenta o disminuye la tensión de salida a los motores, y de esta forma el vehículo avanza, retrocede o dobla hacia la izquierda o derecha.

El sensor de ultrasonido verifica que no haya objetos que interrumpan el paso del vehículo en el circuito, en caso de detectar alguno a una distancia fijada, envía la señal al Arduino, y procede a “esquivar” el obstáculo.

Es sensor de lluvia tiene como objeto verificar la presencia de agua, envía la señal al Arduino, y con ello se realiza la protección el sensor de ultrasónico y se establece que el vehículo debe volver al lugar de salida.

Para volver al lugar de salida en caso de lluvia, el vehículo será manejado por un operario, quien en todo momento está al tanto del recorrido. De esto se ocupa el micro controlador ESP32cam. Este micro controlador posee una interfaz de cámara y mandos mediante un dispositivo móvil o notebook, que ante la detección de lluvia, manejara en forma remota al vehículo hasta su lugar de salida.

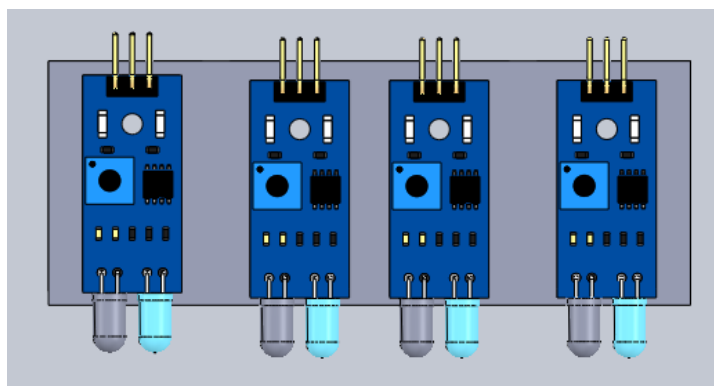
## DESCRIPCIÓN DEL FUNCIONAMIENTO DE LOS COMPONENTES

### Función se los sensores infrarrojos

Los encargados de seguir la línea son sensores infrarrojos, están compuestos por un emisor y un receptor de luz infrarroja, inciden la superficie, y según sea, refleja en mayor o menor media la luz infrarroja.

Son 4 sensores dispuestos en línea, 2 en el centro están separados a una distancia de 10 mm, mientras que otros 2 separados a 15 mm de los del centro (figura), todos a una altura de 15 mm del piso.

La línea es de color oscuro, el funcionamiento de estos sensores, según estén posicionados por sobre la línea, entregan un código de error, y cada código establece una



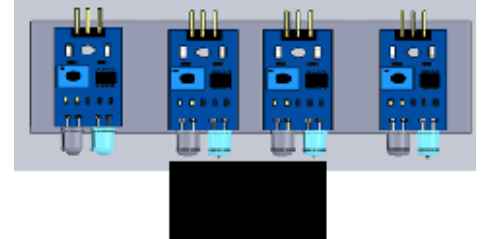
*Ilustración 25 - Línea de sensores infrarrojos*



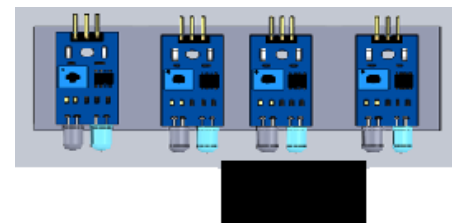
acción. El sensor envía 1 (uno) en caso de estar sobre superficie clara reflectante, y 0 (cero) cuando esta sobre superficie oscura no reflectante.

Observando de frente al vehículo, esta es la secuencia de errores

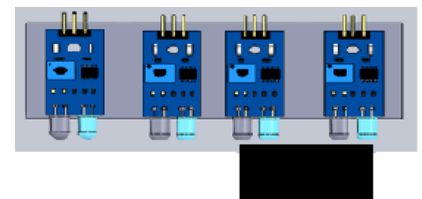
Error 0 (1-0-0-1): 2 sensores del centro están posicionados sobre la línea negra.



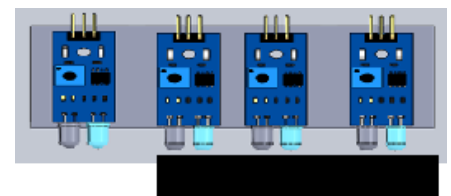
Error 1 (1-1-0-1): 1 sensor del centro está posicionados sobre la línea negra, el vehículo esta levemente hacia la izquierda.



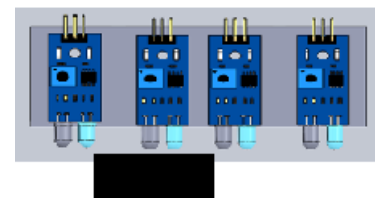
Error 2 (1-1-0-0): 2 sensores del lado derecho están posicionados sobre la línea negra, el vehículo esta mayormente sobre la izquierda.



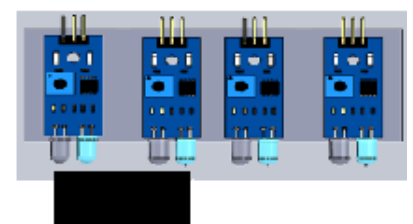
Error 3 (1-0-0-0): 3 sensores del lado derecho está posicionado sobre la línea negra, el vehículo está en una curva pronunciada o de 90°.



Error -1 (1-0-1-1): 1 sensor del centro está posicionados sobre la línea negra, el vehículo esta levemente hacia la derecha.



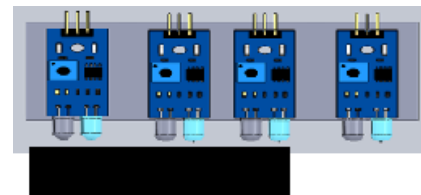
Error -2 (0-0-1-1): 2 sensores del lado izquierdo están posicionados sobre la línea negra, el vehículo esta mayormente sobre la derecha.



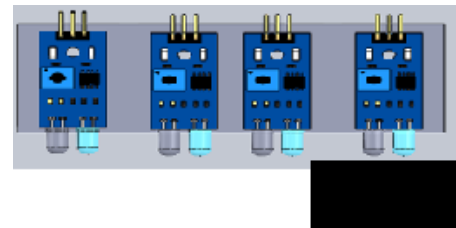




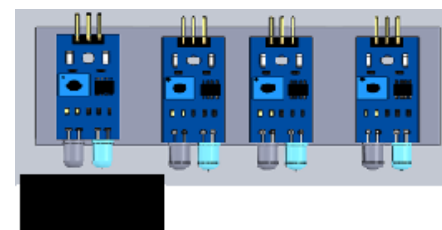
Error -3 (0-0-0-1): 3 sensores del lado izquierdo está posicionado sobre la línea negra, el vehículo está en una curva pronunciada o de 90°.



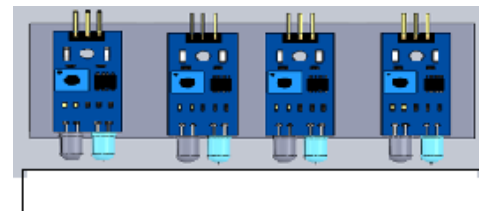
Error 100 (1-1-1-0): 1 sensor del lado derecho está posicionado sobre la línea negra, el vehículo está totalmente sobre la izquierda.



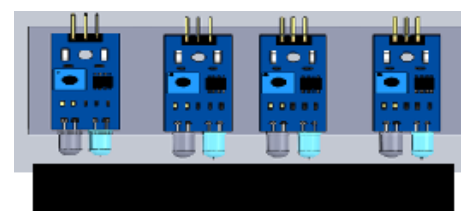
Error 101 (0-1-1-1): 1 sensor del lado izquierdo está posicionado sobre la línea negra, el vehículo está totalmente sobre la derecha.



Error 102 (1-1-1-1): los 4 sensores están posicionados fuera de la línea negra, el vehículo está totalmente fuera del circuito.



Error 103 (0-0-0-0): 4 sensores del lado izquierdo está posicionado sobre la línea negra, el vehículo finalizo su recorrido.



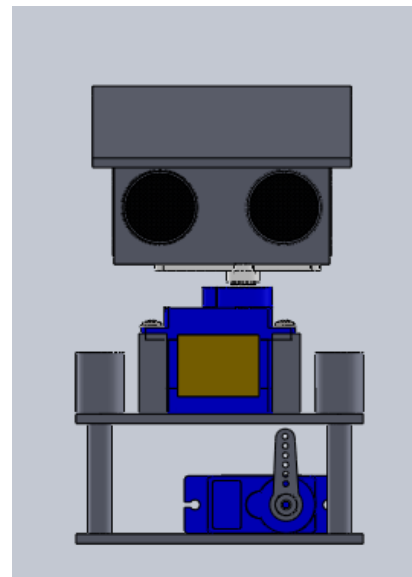
### Conjunto sensor ultrasónico (US), servomotor, porta US y plataforma de elevación

Este conjunto fue diseñado para elevar el sensor ultrasónico por encima de la carcasa, en posición frontal apuntando hacia adelante, y de esa manera detectar si algún objeto obstaculiza el normal recorrido del vehículo. El soporte inferior esta fijo a la placa base, en él se posiciona un servomotor encargado de elevar y bajar, mediante 4 guías, el soporte medio. El soporte medio dispone de un alojamiento para otro servomotor, el cual, en caso de detectar un

obstáculo, tiene la función de girar el sensor ultrasónico de 0 a 180°, con el fin de detectar los obstáculos de los laterales y así seleccionar la dirección hacia la cual será el desvío del vehículo. En la parte superior se encuentra el sensor ultrasonido y su porta sensor, el cual está diseñado con un “techo” sobredimensionado, de forma tal que evita que ingrese algún objeto al interior, en donde se encuentran los demás componentes electrónicos.

### Sensor de lluvia

Este sensor está dispuesto en el exterior, por sobre la carcasa en la parte trasera. Se encarga de verificar el estado de las precipitaciones. En caso de detectar la presencia de lluvia, envía la señal de alerta al Arduino, el cual actúa en consecuencia.



*Ilustración 26 - Conjunto elevador*

### Driver puente H l298n

Componente encargado de manejar las diferentes velocidades del vehículo, según las sean las señales que le envía el Arduino. Esta señales son mediante Modulación por Ancho de Pulso (PWM) es una técnica de control que simula una salida de voltaje analógica utilizando una señal digital. Aunque las señales digitales solo pueden estar en dos estados (encendido o apagado, HIGH o LOW, 1 o 0), PWM logra variar la "cantidad" de energía entregada a una carga al controlar el tiempo que la señal permanece en estado "ENCENDIDO" dentro de un período fijo. Puede asemejarse al funcionamiento de un interruptor que se enciende y apaga muy rápidamente. Si lo dejas encendido por más tiempo dentro de un ciclo y apagado por menos tiempo, el dispositivo conectado recibirá, en promedio, más energía. Si lo dejas encendido por menos tiempo y apagado por más, recibirá menos.

### Función del Arduino Mega 2560

La placa de desarrollo Arduino mega 2560 es la encargada de recibir los datos de los sensores y enviar las ordenes a los actuadores.

En principio, recibe del conjunto de sensores infrarrojos los códigos de error, con ello envía la señal al puente H l298n, el cual según sea valor del error, acciona los motores en cuanto a dirección y velocidad.

Al mismo tiempo, esta recibe la señal del sensor ultrasónico, de esta manera realizar la acción correspondiente al “esquive” del objeto, y según la lectura de distancias, será a la derecha o izquierda.

También, y en forma simultánea, está recibiendo los datos enviados por el sensor de lluvia. El mismo, al detectar la lluvia, envía la señal, y aquí se procede a algo importante. Ante la presencia de la lluvia, se envía la señal al conjunto “sensor ultrasónico (US), servomotor, porta





US y plataforma de elevación”, para que active el servomotor inferior y de esa forma proteger la parte electrónica del conjunto.

En ese instante, entran en juego el ESP32cam. En todo momento del recorrido del circuito, tenemos visión mediante la cámara incorporada en la placa de desarrollo, y además tiene en la interfaz una “botonera” para, de esa manera, un usuario tenga la opción de manejar el vehículo en forma remota y poder traerlo de la manera más rápida y lugar del final de recorrido.

La comunicación entre Arduino mega 2560 y ESP32cam es serial, en cuanto que las órdenes las emite el esp32cam, y las recibe e interpreta el Arduino.

## ETAPA DE PROGRAMACIÓN

Otra condición mínima que debe cumplir el proyecto, es la parte de programación, que puede ser en micro controladores, en este proyecto utilizamos 2 (dos).

La programación de las placas de desarrollo es mediante el IDE de Arduino en lenguaje C++.

En cuanto a los códigos utilizados, no son de creación propia del alumno. Son extraídos de modelos de utilización libre en la web, así como de ejemplos que brinda IDE de Arduino en su lista de ejemplos. Se adaptan los códigos a las necesidades del proyecto.

Al tener 2 placas de desarrollo independientes, cada una tiene cargado su código para el manejo del prototipo

A continuación se procede a la explicación de las partes esenciales del código en principal (.ino). es código fuente no utiliza librerías específicas, pero tiene varios condicionales “if”, “else if”, “do”, “while” , “flag” (bandera) para asegurar la correcta de condicionales, y funciones de aplicación específica “void...()”.

En los códigos de Arduino, void setup() y void loop() son las dos funciones principales y obligatorias en cada programa. El “sketch” se divide en dos fases fundamentales:

Void setup(): Esta función se ejecuta una sola vez cuando el Arduino se enciende o se reinicia. Configura lo que se necesita antes de iniciar a funcionar de forma continua, la preparación del escenario antes de que empezar. Todo lo que se define aquí solo ocurre una vez. Aquí se hace lo siguiente:

- Inicializar pines: Configuras si un pin digital va a funcionar como entrada (INPUT) o como salida (OUTPUT) usando pinMode().
- Iniciar comunicación serial: para comunicarse con la computadora (enviar mensajes o recibir comandos), inicia comunicación serial con Serial.begin() y establecer la velocidad de comunicación.



- Inicializar librerías: para sensores, pantallas o módulos que requieren librerías específicas, aquí se inicializa.
- Establecer estados iniciales: encender o apagar LEDs, mover un servo a una posición inicial, etc.

void loop(): Después de que setup() termina de ejecutarse, el programa pasa a void loop() , lo se ejecuta continuamente en un ciclo infinito. Esta es la parte donde realiza la tarea/ acción principal para la que fue programado suceda una y otra vez sin parar, reaccionando y actuando de forma dinámica. Aquí se coloca el código que se repite una y otra vez, por ejemplo:

- Leer sensores: Obtener datos de temperatura, luz, distancia, etc.
- Controlar actuadores: Encender/apagar LEDs, mover motores, activar relés.
- Responder a eventos: Tomar decisiones basadas en las entradas (por ejemplo, si un botón es presionado).
- Enviar datos: Enviar lecturas de sensores o estados a través de la comunicación serial.
- 

### Fragmento de código para Arduino mega 2560

Fragmento de código lee los sensores infrarrojos, establece los errores a fin de ejecutar las correcciones de velocidad necesarias para continuar con el circuito.

```
void read_sensor_values()
```

```
{
```

```
  sensor[0] = !digitalRead(L_S1); //sensor 1 mirado de frente al vehiculo de izq a der
```

```
  sensor[1] = !digitalRead(L_S2); //sensor 2 mirado de frente al vehiculo de izq a der
```

```
  sensor[2] = !digitalRead(R_S3); //sensor 3 mirado de frente al vehiculo de izq a der
```

```
  sensor[3] = !digitalRead(R_S4); //sensor 4 mirado de frente al vehiculo de izq a der
```

```
// errores segun la posocion de los sensores sobre la linea
```

```
if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0))
```

```
  error = 3; //llego a una esquina de 90° - el vehiculo debe girar a la derecha mirando de frente
```

```
  else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 0) && (sensor[3] == 0))//debe girar a la derecha con mas fuerza
```

```
  error = 2;
```



```
else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 0) && (sensor[3] == 1))//debe girar a la derecha levemente
```

```
error = 1;
```

```
else if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 1))//avanza
```

```
error = 0;
```

```
else if ((sensor[0] == 1) && (sensor[1] == 0) && (sensor[2] == 1) && (sensor[3] == 1))//debe girar a la izquierda levemente
```

```
error = -1;
```

```
else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 1) && (sensor[3] == 1))//debe girar a la izquierda con mas fuerza
```

```
error = -2;
```

```
else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 1))
```

```
error = -3; // llego a una esquina de 90° - el vehiculo debe girar a la izquierda mirando de frente
```

```
else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 0)) // gira a la izquierda
```

```
error = 100;
```

```
else if ((sensor[0] == 0) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 1)) // gira a la derecha
```

```
error = 101;
```

```
else if ((sensor[0] == 1) && (sensor[1] == 1) && (sensor[2] == 1) && (sensor[3] == 1)) // esta fuera de la linea al completo
```

```
error = 102;
```

```
else if ((sensor[0] == 0) && (sensor[1] == 0) && (sensor[2] == 0) && (sensor[3] == 0)) // estan todos los sensores sobre la linea
```

```
error = 103;
```

```
}
```

Fragmento de código, función para el manejo de los servomotores

```
void servoPulse (int pin, int angle){
```

```
int pwm = (angle*11) + 500; //conversion del angulo en microsegundos
```



```
digitalWrite(pin, HIGH);  
delayMicroseconds(pwm);  
digitalWrite(pin, LOW);  
delay(50);  
}
```

Fragmento de código, función que compara las distancias. El sensor ultrasónico toma los datos de las distancias derecha e izquierda y los guarda en las variables para realizar la comparación

```
void Check_side(){  
    Stop();  
    delay(100);  
    for (int angle = 100; angle <= 150; angle += 10) {  
        servoPulse(servo2, angle); }  
    delay(800);  
    dist_der = Ultrasonic_read();  
    //Serial.print("D R=");Serial.println(dist_der);  
    delay(400);  
    for (int angle = 150; angle >= 50; angle -= 10) {  
        servoPulse(servo2, angle); }  
    delay(800);  
    dist_izq = Ultrasonic_read();  
    //Serial.print("D L=");Serial.println(dist_izq);  
    delay(400);  
    for (int angle = 50; angle <= 100; angle += 10) {  
        servoPulse(servo2, angle); }  
    delay(600);  
    compareDistance();  
}
```

Fragmento de código, funciones que realiza el cálculo para manejar las velocidades para cambio de dirección de las ruedas. Se adopta un sistema de control PD. Si el vehículo se mueve



hacia la izquierda o derecha, tendremos un error positivo o negativo, o 0 si el vehículo está en el centro. Debido a la dificultad para realizar el modelado del vehículo, por tener muchas variables tales como el torque de los motores, el agarre de las ruedas a la superficie, los pesos y su distribución, la longitud, tensión de las baterías, los valores para las constantes  $K_p$  y  $K_d$  no son calculadas, sino que se establecen según las pruebas que se realizan, hasta que la velocidad, dirección, giro y oscilaciones son tales se pueda asegurar el recorrido del vehículo.

```
void calculate_pid()
```

```
{  
    P = error;  
    //Serial.print("P= ");  
    //Serial.print(P); //4° valor en la linea del monitor serial  
    //Serial.print("\t");  
    I = I + previous_I;  
    D = error - previous_error;  
    //Serial.print("D= ");  
    //Serial.print(D); //5° valor en la linea del monitor serial  
    //Serial.print("\t");  
    PID_value = (Kp * P) + (Ki * I) + (Kd * D);  
    previous_I = I;  
    previous_error = error;  
}
```

```
void motor_control()
```

```
{  
    // Cálculo de la velocidad efectiva del motor  
  
    int left_motor_speed = vel_inic-PID_value; // motores manejados por enA - derecho mirados  
    de frente al vehiculo  
  
    int right_motor_speed = vel_inic+PID_value; // motores manejados por enB - izq mirados de  
    frente al vehiculo  
  
    // Limita la velocidad del motor, no debe exceder el valor máximo de PWM.
```



```
left_motor_speed = constrain(left_motor_speed, 0, 255);
right_motor_speed = constrain(right_motor_speed, 0, 255);

//Serial.print(PID_value); //6° valor en la linea del monitor serial
//Serial.print("\t");
//Serial.print(left_motor_speed); //7° valor en la linea del monitor serial
//Serial.print("\t");
//Serial.println(right_motor_speed); // 8° valor en la linea del monitor serial y salta de linea
analogWrite(enA, left_motor_speed); //son los motores derechos mirando el auto de frente
analogWrite(enB, right_motor_speed); //son los motores izquierdos mirando el auto de frente
forword();
}
```

Fragmento del código, función para la placa arduino que interpreta las ordenes que envía la placa ESP32cam a través de comunicación serial. Para lograr esto, Arduino debe “leer” la palabra exacta que envía el ESP32cam.

```
void mando_esp32cam() {
  if (Serial1.available()) {
    comando = Serial1.readStringUntil('\n'); // Espera hasta recibir \n
    comando.trim(); // Elimina \r, espacios, etc.

    if (comando == "Forward") {
      Serial.println("Comando: Forward");
      analogWrite(enA, 140);
      analogWrite(enB, 140);
      forword();

      // Acción para avanzar
    }

    else if (comando == "Left") {
```



```
Serial.println("Comando: Left");
sharpLeftTurn();
}
else if (comando == "Right") {
    Serial.println("Comando: Right");
    sharpRightTurn();
}
else if (comando == "Back") {
    Serial.println("Comando: Back");
    backward();
}
else if (comando == "Stop") {
    Serial.println("Comando: Stop");
    Stop();
}
else {
    Serial.print("Comando desconocido: ");
    Serial.println(comando);
}
}
}
```

### Fragmento de código del ESP32cam

En cuanto al código del ESP32cam, posee el archivo principal (\*.ino) donde establecemos la dirección IP fija para poder manejar el vehículo en forma remota; un archivo de cabecera (\*.h) donde está la matriz de la cámara; y un archivos de código fuente (\*.cpp) donde se definen las funciones de la cámara, esta parte del código es un ejemplo de que ofrece la placa en el IDE de Arduino>>Archivo>>Ejemplos>>ESP32>>Camera>>CameraWebServer. Posee una licencia



de “software colaborativo y de código abierto”, es por esto q se permite editar para agregar funciones de la página en lenguaje HTML – CCS para la interfaz gráfica con el usuario.

Se logró operar el vehículo mediante el teléfono móvil y la notebook, el problema surge que al conectamos a un router inalámbrico, después de un tiempo se generaban IP aleatorias, lo que nos obligaba a abrir el monitor serie a fin de obtener la nueva IP, lo que genera un inconveniente importante debido a que la placa está posicionada dentro del vehículo. Encontramos la solución en la misma placa de desarrollo ESP32cam, la cual mediante código, ofrece la posibilidad de configurarla para funcionar como estación Wi-Fi, como punto de acceso (AP) o ambos. La limitación es la distancia de manejo, tiene un radio de 5 metros. Para más distancia, la opción es configurar el router con una puerta de enlace y realizar la conexión, este es un servicio de IP pública que ofrecen los distribuidores de internet el cual tiene un costo. Por ejemplo conexiones domésticas de la empresa “Telecentro” se puede acceder desde cualquier lugar con una cuenta DynDNS. Existen servicios gratuitos como No-IP o DuckDNS y ejecutar el cliente actualización en una PC o Raspberry Pi.

Fragmento de código para configurar la IP fija con conexión directa al router. Esta es la forma seleccionada para el proyecto.

```
// --- CONFIGURACIÓN DE IP FIJA ---
```

```
// IMPORTANTE: Debes ajustar estos valores a la configuración de tu red.
```

```
// La IP fija debe estar dentro del rango de tu red, pero fuera del rango DHCP
```

```
// para evitar conflictos con otras IPs asignadas automáticamente.
```

```
IPAddress local_IP(192, 168, 137, 100); // IP que quieres asignar a tu ESP32-CAM///ip fija...  
cargo en el buscador
```

```
IPAddress gateway(192, 168, 137, 1); // Puerta de enlace de tu red (normalmente la IP de tu  
router)// de mi compu en este caso
```

```
IPAddress subnet(255, 255, 255, 0); // Máscara de subred (la más común es 255.255.255.0)
```

```
// --- FIN CONFIGURACIÓN DE IP FIJA ---
```

```
--
```

```
--
```

```
// --- APLICAR CONFIGURACIÓN DE IP FIJA ---
```

```
WiFi.config(local_IP, gateway, subnet);
```

```
// --- FIN APLICAR CONFIGURACIÓN DE IP FIJA ---
```

```
WiFi.begin(ssid, password);
```





```
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("");  
Serial.println("WiFi connected");  
startCameraServer();  
Serial.print("Camera Ready! Use 'http://'");  
Serial.print(WiFi.localIP());  
WiFiAddr = WiFi.localIP().toString();  
Serial.println(" to connect");
```

Fragmento de código incorporado al archivos de código fuente (\*.cpp), esta líneas están redactadas en de lenguaje HTML-CCS, que da formato a la interfaz al usuario.

```
static esp_err_t index_handler(httpd_req_t *req){  
    httpd_resp_set_type(req, "text/html");  
    String page = "";  
    page += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0\">\n";  
    page += "<script>var xhttp = new XMLHttpRequest();</script>";  
    page += "<script>function getsend(arg) { xhttp.open('GET', arg +'?' + new Date().getTime(), true); xhttp.send() } </script>";  
    page += "<p align=center><IMG SRC='http://\" + WiFiAddr + \":81/stream' style='width:280px;'></p><br/><br/>";  
    // page += "<p align=center><IMG SRC='http://\" + WiFiAddr + \":81/stream' style='width:300px; transform:rotate(180deg);'></p><br/><br/>";  
    page += "<p align=center style= font-size: 1.2em; color: #555; margin-bottom: 20px; font-family: 'comic Sans', sans-serif;><b>PROYECTO MECATRONICO</b></p>";  
    page += "<p align=center><button style=background-color:lightgrey;width:90px;height:90px;border-radius:25%;padding:10px;font-size:1em onmousedown=getsend('go') onmouseup=getsend('stop') ontouchstart=getsend('go') ontouchend=getsend('stop') ><b>Forward</b></button> </p>";
```



```
page += "<p align=center>";

page += "<button style=background-color:lightgrey;width:90px;height:90px;border-
radius:25%;padding:10px;font-size:1em;                                onmousedown=getsend('left')
onmouseup=getsend('stop')                                           ontouchstart=getsend('left')
ontouchend=getsend('stop')><b>Left</b></button>&nbsp;";

page += "<button style=background-color:indianred;width:90px;height:80px
onmousedown=getsend('stop') onmouseup=getsend('stop')><b>Stop</b></button>&nbsp;";

page += "<button style=background-color:lightgrey;width:90px;height:90px;border-
radius:25%;padding:10px;font-size:1em                                onmousedown=getsend('right')
onmouseup=getsend('stop')                                           ontouchstart=getsend('right')
ontouchend=getsend('stop')><b>Right</b></button>";

page += "</p>";

page += "<p align=center><button style=background-
color:lightgrey;width:90px;height:90px;border-radius:25%;padding:10px;font-size:1em
onmousedown=getsend('back') onmouseup=getsend('stop') ontouchstart=getsend('back')
ontouchend=getsend('stop') ><b>Backward</b></button> </p>";

page += "<p align=center>";

page += "<button style=background-color:yellow;width:140px;height:40px ;border-
radius:10px; onmousedown=getsend('ledon')><b>Light ON</b></button>";

page += "<button style=background-color:yellow;width:140px;height:40px ;border-
radius:10px; onmousedown=getsend('ledoff')><b>Light OFF</b></button>";

page += "</p>";

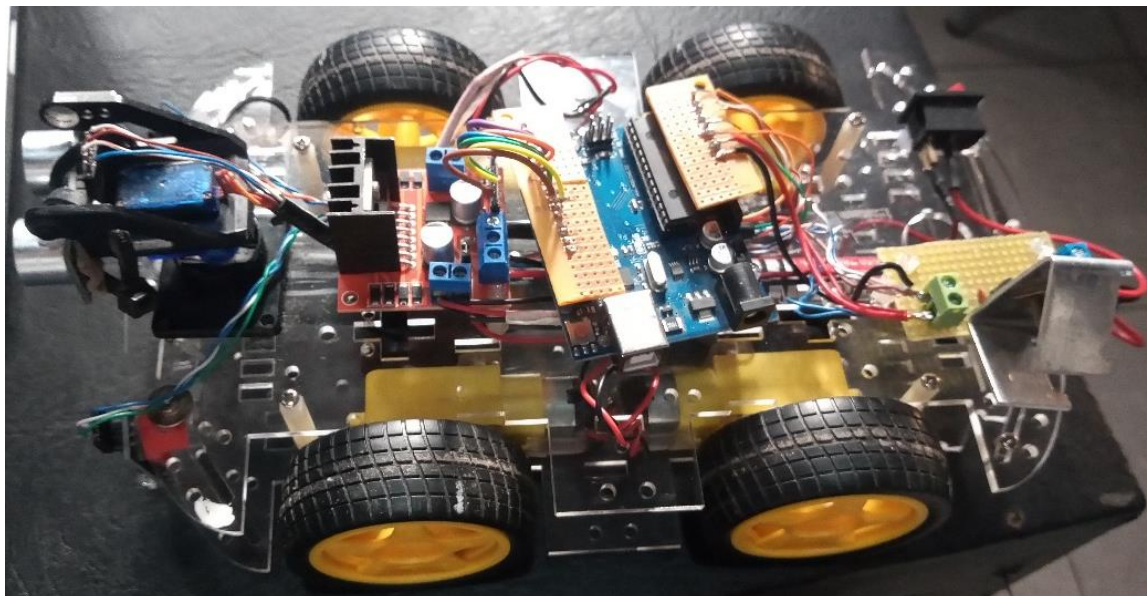
return httpd_resp_send(req, &page[0], strlen(&page[0]));

}
```

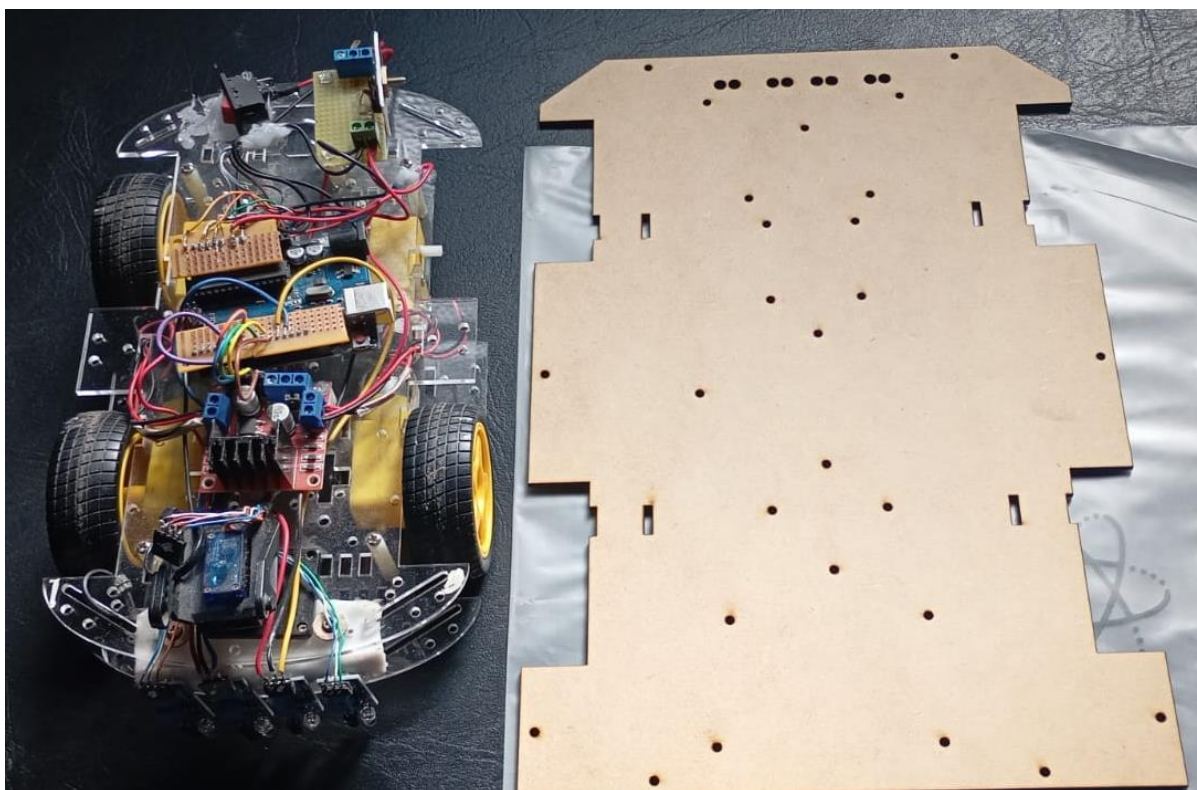
## ETAPA DE ARMADO Y CABLEADO DEL PROTOTIPO

Continuación de la condición de circuito eléctrico y electrónico, en esta etapa, se realiza el montaje de los componentes electrónicos y de las piezas que se diseñaron para los mismos. Las imágenes muestran el proceso de traspaso de componentes del primer prototipo de pruebas, a la placa base diseñada.

Primer prototipo de pruebas

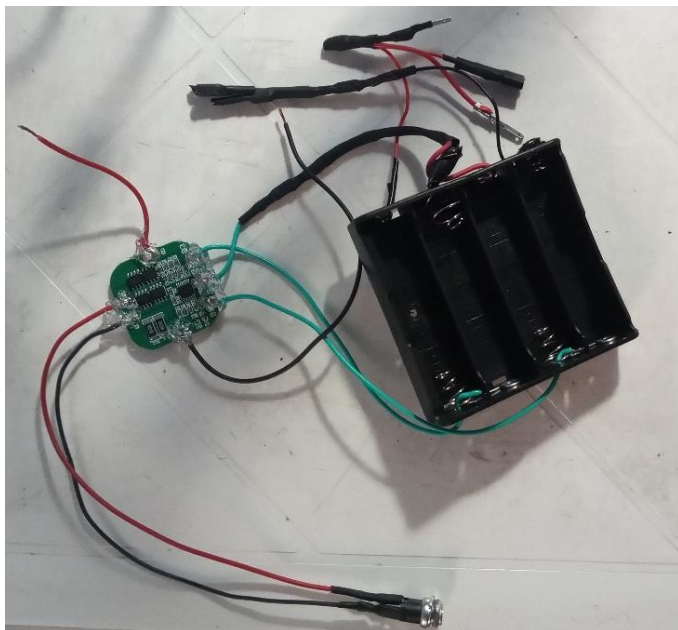


Placa base diseñada, material MDF, a la cual se colocan los componentes.

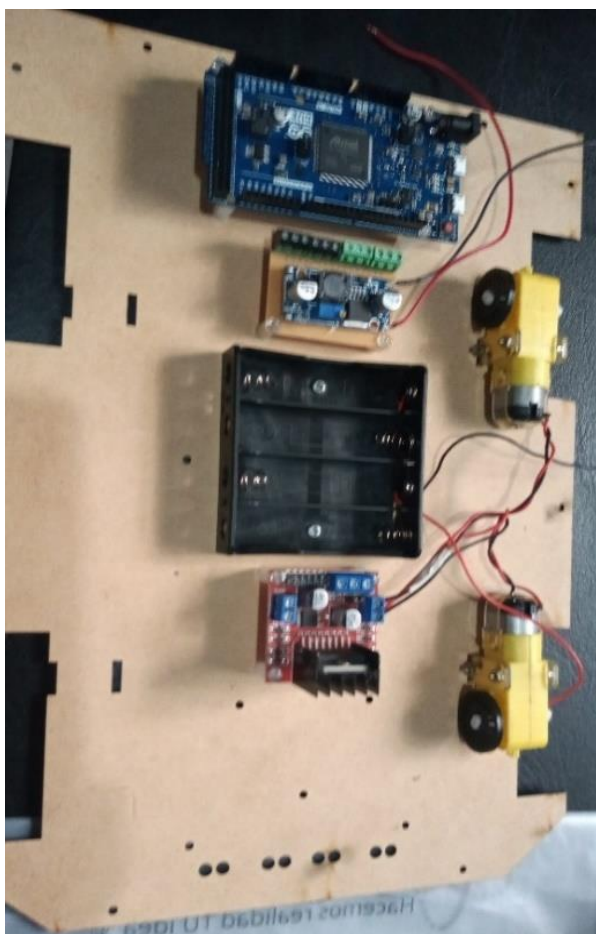




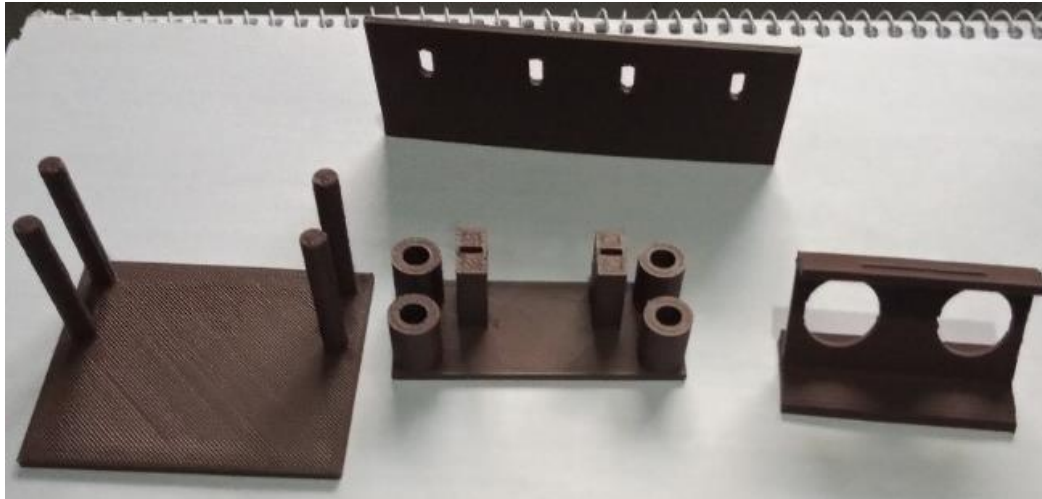
Porta pilas con la conexión de la placa cargador y protección de batería de litio



Disposición de los componentes en la placa base con los separadores

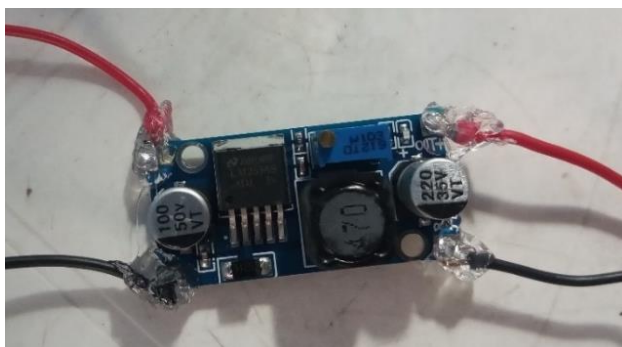


Piezas diseñadas e impresas en 3d del conjunto elevador del sensor ultrasónico y espalda de sensores infrarrojos.

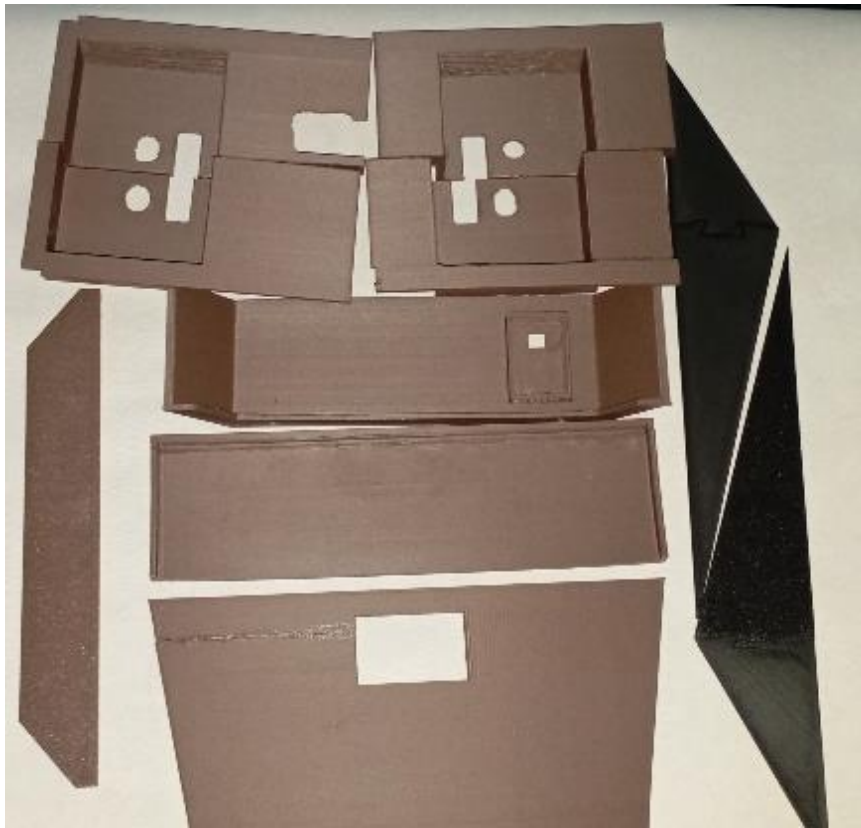


Reguladores Step down

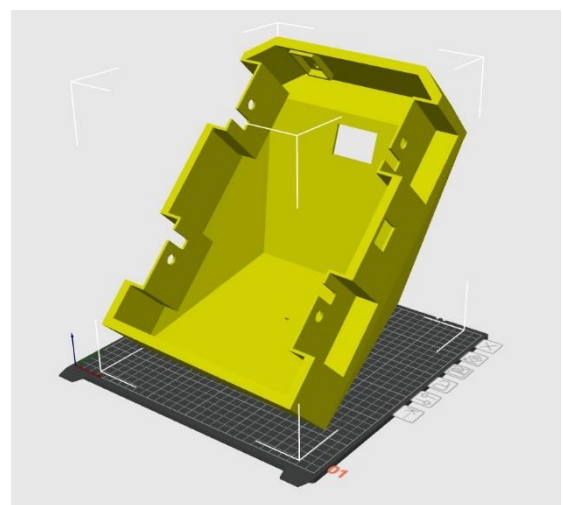
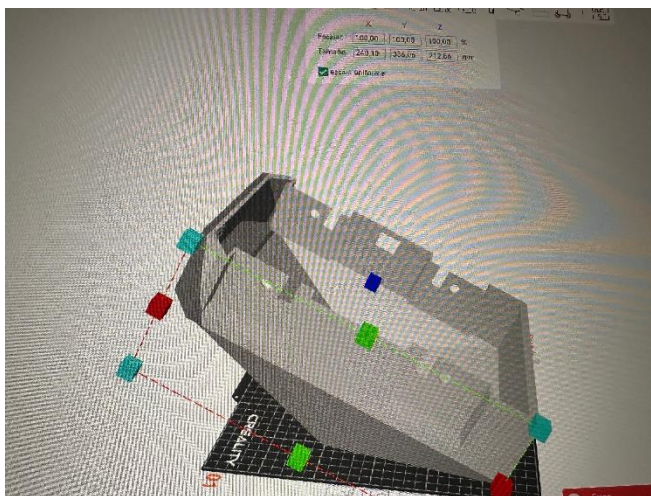
Regulador para driver puente H y regulador para los demás componentes al cual se le agregan terminales de conexión



Carcasa diseñada impresa en 3d en varias piezas debido al límite de mesa de impresión

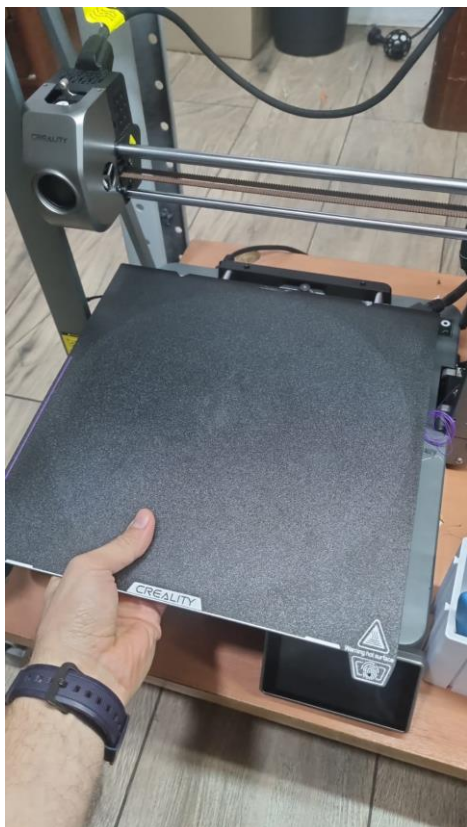


Las siguientes imágenes muestran los límites de las “camas” de las impresoras 3d, por ello se justifica el fraccionamiento de la carcasa y la correspondiente impresión en partes, ya que el costo de hacerlo en un solo bloque aumentaría el costo total. Ello está reflejado en el anexo IV-Mail de solicitud, respuesta de presupuestos, cotización de carcasa.



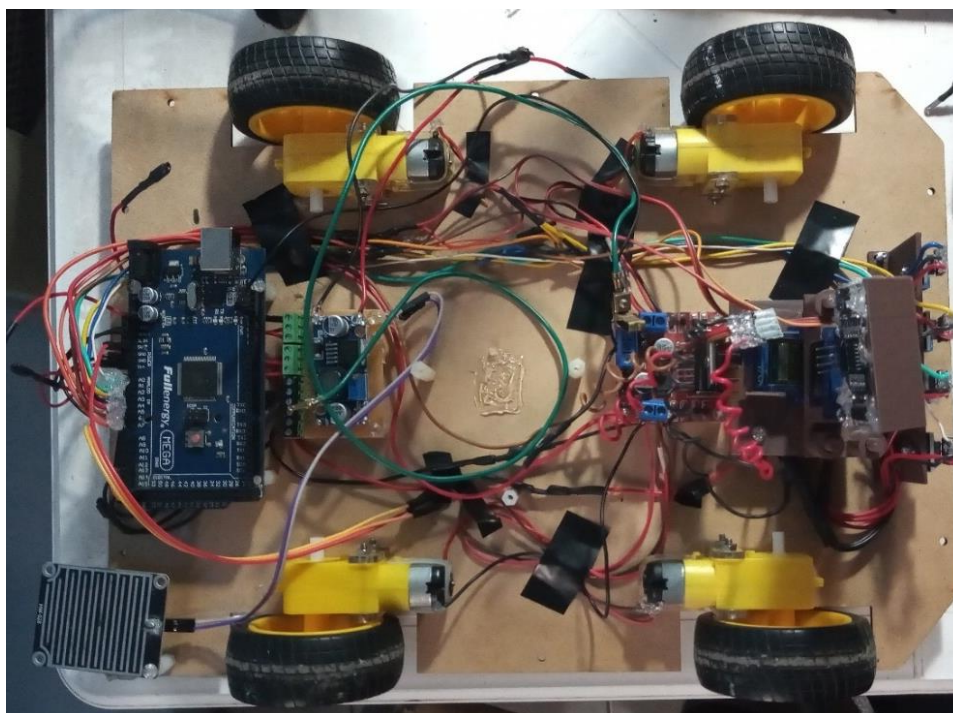
*Ilustración 27 - Prototipo proyectado en mesa de impresión*

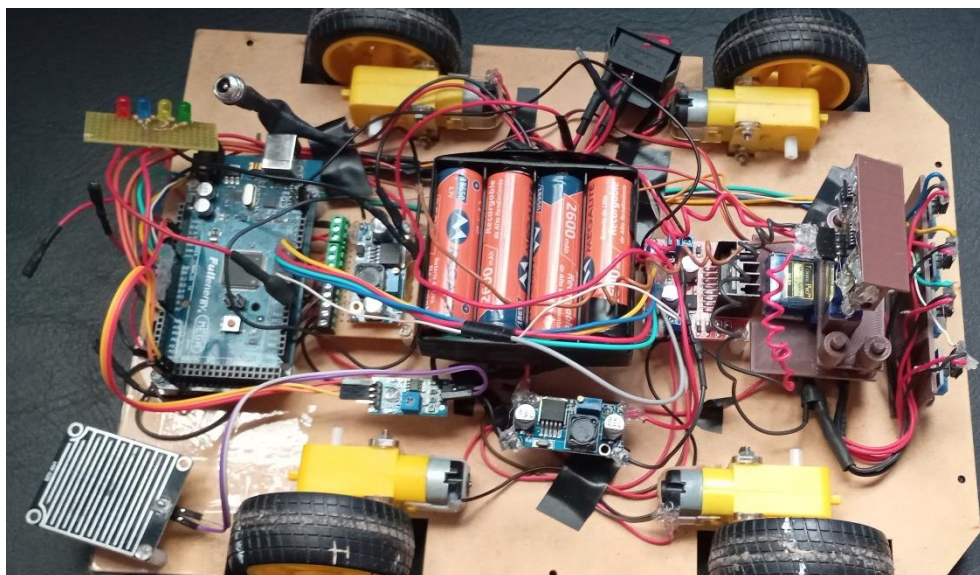




*Ilustración 28 - Fotografía del volumen total de una mesa de impresión 3d*

#### Placa con componentes y cableado





### Tiempo estimado Vs Tiempo real

Al iniciar el proyecto se realizó un cronograma detallando las diferentes actividades y el tiempo estimado de cada una, además de un cuadro donde se debía constar las horas que nos ocupábamos de cada actividad y tiempo total de finalización.

En el transcurso del proyecto, se verificó que algunas actividades fueron sobrestimadas, y otras subestimadas, tanto en tiempo como en su fecha de inicio y final. A pesar de ello, las horas totales estimadas son similares a las horas totales reales que llevo cada actividad. Es siguiente cuadro refleja las variaciones.

| TAREA  | HORAS REALES | HORAS ESTIMADAS | DIFERENCIA | COCIDERACION |
|--|--------------|-----------------|------------|--------------|
| Selección del proyecto                       | 3            | 3               | 0          |              |
| Determinación de viabilidad y alcance        | 4            | 4               | 0          |              |
| <b>Búsqueda de información</b>               |              |                 | 0          |              |
| Software a utilizar                          | 4            | 8               | 4          | SOBRESTIMADA |
| Componentes para el desarrollo del prototipo | 10           | 8               | -2         | SUBESTIMADA  |
| <b>Diseño general del prototipo</b>          |              |                 |            |              |
| Diseño de la base para los componentes       | 28           | 16              | -12        | SUBESTIMADA  |
| Diseño de la carcasa superior                | 32           | 16              | -16        | SUBESTIMADA  |
| <b>Confección de prototipo</b>               |              |                 |            |              |
| Solicitud de presupuesto de componentes      | 5            | 3               | -2         | SUBESTIMADA  |
| Compra de componentes                        | 19           | 8               | -11        | SUBESTIMADA  |





| Programación                        |     |     |     |              |
|-------------------------------------|-----|-----|-----|--------------|
| Arduino y componentes               | 32  | 45  | 13  | SOBRESTIMADA |
| Comunicación serial arduino-esp32   | 20  | 45  | 25  | SOBRESTIMADA |
| Electrónica                         |     |     |     |              |
| Armado de componentes en la base    | 32  | 15  | -17 | SUBESTIMADA  |
| Conexión y soldadura de componentes | 20  | 15  | -5  | SUBESTIMADA  |
| Consulta a profesores               |     |     |     |              |
| Consulta de electrónica             | 8   | 16  | 8   | SOBRESTIMADA |
| Consulta de programación            | 4   | 16  | 12  | SOBRESTIMADA |
| Consulta de diseño (SolidWorks)     | 13  | 16  | 3   | SOBRESTIMADA |
| Testeo del prototipo                | 55  | 32  | -23 | SUBESTIMADA  |
| PRIMER ENTREGA                      | 1   | 1   | 0   |              |
| Revisión general del prototipo      | 14  | 15  | 1   |              |
| Corrección de errores               | 19  | 20  | 1   |              |
| SEGUNDA ENTREGA                     |     |     |     |              |
| Determinación de mejoras            | 22  | 45  | 23  | SOBRESTIMADA |
| ENTREGA FINAL                       |     |     | 0   |              |
| Confección de informe               | 51  | 45  | -6  | SUBESTIMADA  |
| Total de horas                      | 394 | 392 | 2   |              |

Tabla 4 - Horas estimadas vs Horas reales

### Costo estimado Vs Costo real

Cada vez que se realizó una compra, se solicitó de ser la emisión de un comprobante o factura. La suma de todas sin contar con las compras abonadas con otros medios de pago es de \$ 140.736,70. A esto deberíamos agregar el monto de los componentes que, a través de las distintas cursadas de la carrera, adquiridos de antemano, ellos son: Arduino mega 2560, ESP32cam, sensor ultrasónico, driver puente H y motores con ruedas, tomando como referencia el precio más económico de las ofertas, el costo real del prototipo asciende a \$ **203.001,70**.

Teniendo en cuenta el monto de las ofertas de la totalidad de los componentes de \$ 159.906,00; debe sumarse los servicios de impresión en 3d en un solo bloque de la carcasa y el corte en acrílico de la base, por ello el costo estimado es de **\$275.556**.

Se realizó consultas, a alumnos que desarrollan actividad en la industria, respecto la posibilidad de que la carcasa para el prototipo sea fabricada en plástico mediante un proceso de inyección o extrusión. Esto implicaría la fabricación de una matriz con las dimensiones de la carcasa y la producción de ella en cantidades. En este sentido, no es viable esa clase de fabricación para el prototipo.



## MEJORAS PARA EL PROTOTIPO

En cuanto al diseño, no fue conveniente hacerlo en forma prismática, por la limitación en cuanto al movimiento circular del conjunto sensor ultrasónico y de la posibilidad de que por allí ingrese agua u otros objetos que hagan problemático el funcionamiento de los componentes electrónicos dentro del vehículo. También, la placa ESP32cam podría ir en un pequeño gabinete por encima del vehículo y con un servomotor ser direccionado para observar los alrededores.

Además, en cuanto al tamaño del prototipo, puede ser más compacto a fin de que la carcasa pueda ser impresa en un solo bloque.

En cuanto a la carga de las baterías. Tener una estación de carga fija, una vez que el vehículo finaliza su recorrido, detecte su estación y se dirija a ella. La carga se realiza mediante contactos metálicos dispuestos en la base del vehículo, y la navegación hacia la estación de carga mediante sensores infrarrojos o de radiofrecuencia.

En cuanto a la limitación de manejo del vehículo a más distancia, realizar la configuración de la placa de desarrollo ESP32cam, a fin de realizar esta acción desde cualquier lugar.

También es posible aplicar la tecnología de las máquinas de aspirar autónomas, y así realizar un “mapeo” del recorrido deseado y de esa manera evitar el uso de línea.



## CONCLUSIÓN

El proyecto fue seleccionado de un listado ofrecido por la cátedra. A primera lectura, parecía ser el apropiado para poder realizarlo en el tiempo de cursada de la cátedra, con abundante información, ejemplos, modelos, páginas de internet, componentes de bajo costo y de acceso en el mercado. Nada más lejano de la realidad. Las exigencias del proyecto, es decir la parte mecánica, electrónica, de código y programación, hicieron que todo aquello que parecía fácil, se volviera en contra de esa idea.

En cuanto a la parte mecánica, fue iniciar casi de cero el software de diseño, en este proyecto se utilizaron SolidWorks para la placa base, carcasa y la visualización de los componentes en la posición que tendrían en el espacio físico el vehículo.

Otro reto fue realizar la impresión en 3d del diseño logado, ya que al diseñar el vehículo, no se tuvo en cuenta las dimensiones estándar de las impresoras, por ello fue que luego de todo, se tuvo que dividir en varias partes la carcasa, y después unirlos mediante pegamento. En cuanto a la base, se optó por corte en material MDF, económico y resistente.

En cuanto a la parte de software, otro problema. Se utiliza en el proyecto 2 placas de desarrollo, Arduino y ESP32CAM. Por ello se buscaron los modelos y se optó por adaptarlo al requerimiento. Increíblemente, nada salió como lo planeado. El tema de adaptar código, fue difícil. Por ejemplo, la comunicación serial entre placas. Todo un desafío, no fue solo “conectar un cable entre ellas”, varios días de lectura y “tutoriales” que no funcionaban. Y un sábado, con la ayuda del docente, a prueba y error, conectar y desconectar, y “SHAZAM” hubo comunicación y se produjo el reconocimiento de las sentencias y lo más importante, respuesta del vehículo a los comandos enviados desde el teléfono móvil.

Luego está la parte electrónica, donde entra otro elemento importante e indispensable que al momento de diseño no se consideró, “el cable”, ocupa mucho espacio y dificulta todo lo que se diseñó. Los terminales de conexión, la soldadura, la flexibilidad, la cantidad, movimiento de algunos componentes, no fueron tenidos en cuenta al momento de seleccionar la disposición de los ellos en la placa base, y una vez montados los mismos, “ups”, que desgracia, cortar y empalmar, colocar cinta o termo contraíble.

En todo momento se contó con la ayuda de los profesores de la cátedra y de otras materias tales como medios de representación gráfica, robótica, sistemas de control, también ingenieros ya egresados de esta casa, fundamentales para evacuar dudas, corregir, brindar ideas, y los alumnos de la misma cátedra quienes también estaban a disposición. Así, considero que se formó un gran grupo de trabajo, que a fin de todo es lo que se debe encontrar, ya que hacer todo uno solo resulta poco práctico, y la solución a todo es “preguntar y hacer”. La comunicación efectiva y clara es un aspecto importante los grupos de trabajo, y en todos los ámbitos se adquieren experiencias nuevas, que son puesta en práctica en todos desafíos que en se enfrentan en la vida profesional.



---

### AGRADECIMIENTOS:

- A mi familia
- Docentes de la catedra:
  - Ing. Ezequiel Blanca
  - Ing. Cristian Leandro Lukaszewicz
  - Ing. Juan Ignacio Szombach
- Ing. Luis Solís
- Ing. Joaquín Gómez
- Lic. Psicología Ricardo Troullett
- Tec. sup. Tel. Inf. Juan Ottino

### ANEXOS

- I- Constancias de consultas realizadas.
- II- Planos de vistas y cotas de placa base, carcasa y conjunto elevador.
- III- Datasheet de los componentes.
- IV- Mail de solicitud, respuesta de presupuestos, cotización de carcasa.

### SOFTWARE UTILIZADO

- SolidWorks
- Arduino IDE
- Proteus



---

BIBLIOGRAFÍA, PÁGINAS WEB

<https://www.apache.org/licenses/LICENSE-2.0>

<http://www.apache.org/licenses/>

<https://grabcad.com/>

<https://www.alldatasheet.com/>

<https://ria.utn.edu.ar/server/api/core/bitstreams/0cdaa4d1-fe3c-4fad-87ad-7e852d82ebb5/content>

<https://www.youtube.com/watch?v=-ACctHjLzts>

<https://www.prometec.net/timers/>

<https://pasionelectronica.com/pines-y-conectores-de-arduino-uno/>

<https://www.sunrom.com/>

<https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/>

<https://www.youtube.com/watch?v=hsz6MLl83f0>

<https://www.youtube.com/watch?v=9gfZnNPBlVg>

[https://www.youtube.com/watch?si=tFBA\\_p6hnMlsXIBf&v=QEZMRUZpc14&feature=youtu.be](https://www.youtube.com/watch?si=tFBA_p6hnMlsXIBf&v=QEZMRUZpc14&feature=youtu.be)

[https://www.youtube.com/watch?v=\\_35XK2IpYq4&t=5s](https://www.youtube.com/watch?v=_35XK2IpYq4&t=5s)

<https://acortes.co/proyecto-26-sensor-de-lluvia/>

<https://docs.espressif.com/projects/esptool/en/latest/esp32/esptool/serial-connection.html#serial-port-settings>

[https://github.com/un0038998/4\\_In\\_1\\_Arduino\\_Robot/blob/main/Four\\_in\\_One\\_Car/Four\\_in\\_One\\_Car.ino](https://github.com/un0038998/4_In_1_Arduino_Robot/blob/main/Four_in_One_Car/Four_in_One_Car.ino)

<https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>

<https://descubrearduino.com/sensor-de-lluvia/>

<https://programarfacil.com/blog/arduino-blog/pines-digitales-de-arduino/>

<https://lastminuteengineers.com/esp32-mdns-tutorial/>

<https://randomnerdtutorials.com/esp32-static-fixed-ip-address-arduino-ide/>

[https://crm-uam.github.io/actividades/2012\\_taller\\_arduino/Control\\_PD\\_Siguelineas.pdf](https://crm-uam.github.io/actividades/2012_taller_arduino/Control_PD_Siguelineas.pdf)

<https://programarfacil.com/categoria/esp32/>