

# Rockobot V1.0

...

Realizado por:

Ángel Hurtado

José Hinojosa

# Índice

- **Objetivos**
- **Diseño 3D:**
  - Estructura
  - Mecánica
- **Electrónica**
  - Componentes
  - PCB
- **Programación**
- **Problemas**

# Objetivos

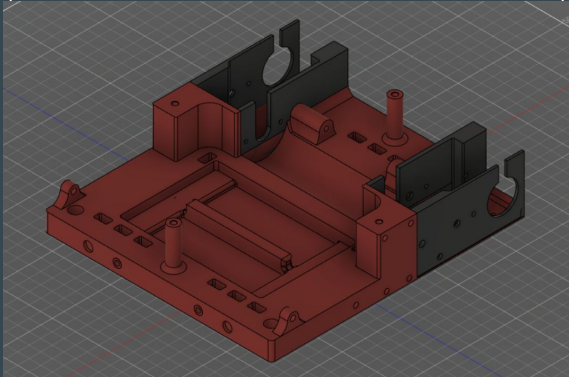
Queríamos hacer un robot que cumpliera lo siguiente:

- Fuerte y grande
- Modular y reparable
- Impreso en 3D en su mayoría
- Usar Orugas (Tanque)
- Circuitería ordenada
- Programación simple

# Diseño 3D: Estructura

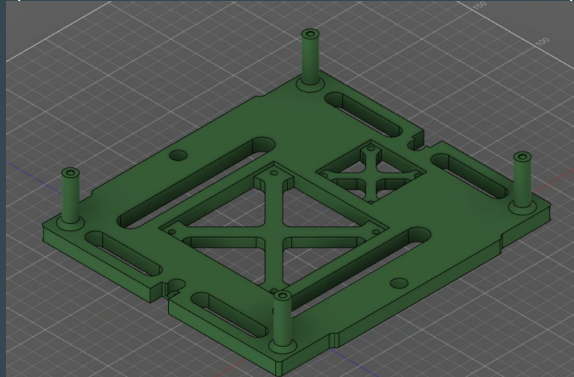
## Piso Motor

Base del robot. Incluye los motores, sensores de infrarrojos, reductora, conexión a la oruga.



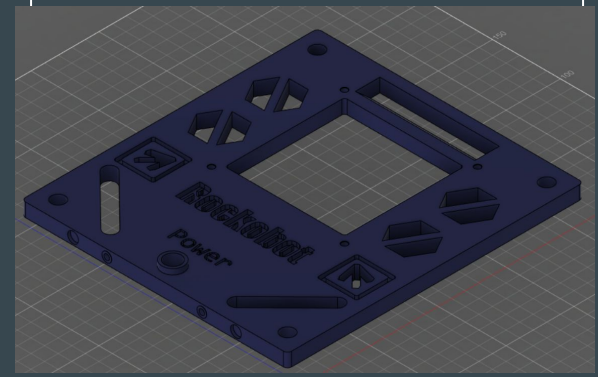
## Piso Electrónica

Segundo piso, incluye la conexión electrónica, driver y el Arduino.

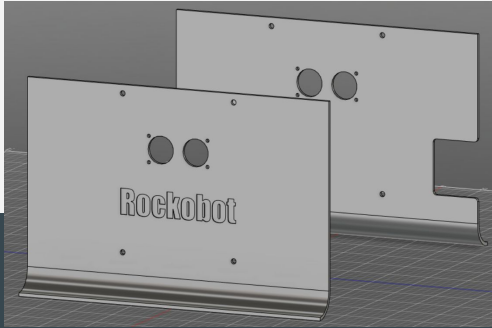


## Piso Superior

Principalmente decorativo pero también incluye el botón de encendido. Brilla en la oscuridad.

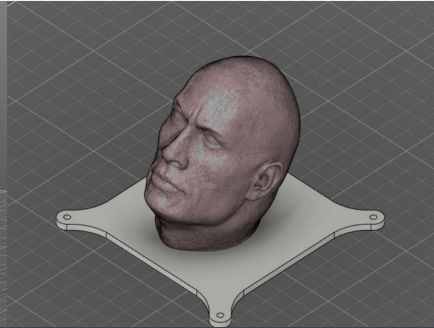


# Diseño 3D: Estructura



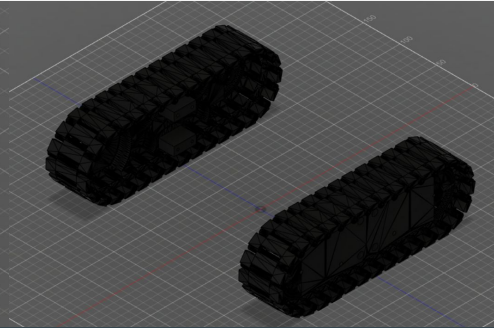
## Shields

Escudos de acero inoxidable. Integran en ellos unos agujeros para el sensor de ultrasonidos



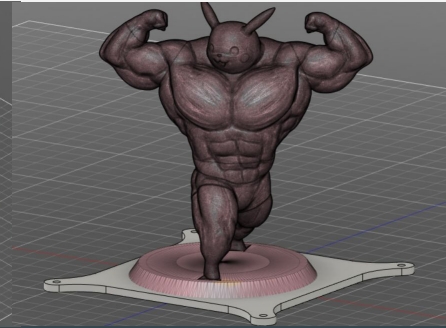
## Rockins

Módulos para el piso superior. Principalmente decorativos, pero pueden ser funcionales (iluminación, por ejemplo)



## Tracks

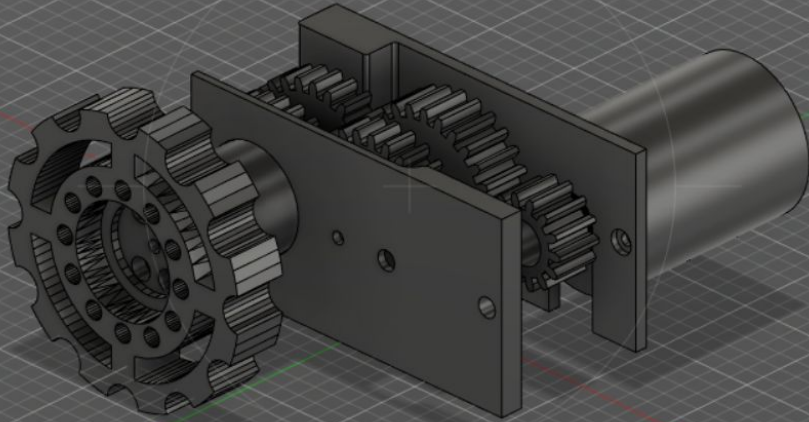
4 piezas: Oruga de plástico flexible, frame para conexión oruga-cuerpo y 2 tipos de engranajes: para movimiento y para tensión



## Otros componentes

Para conexión de las piezas se usaron tornillos, tuercas, threaded inserts, arandelas, silicona para algunos apañes etc...

# Diseño 3D: Mecánica



*Gearbox, motor y engranaje principal, que engancha a la oruga flexible*

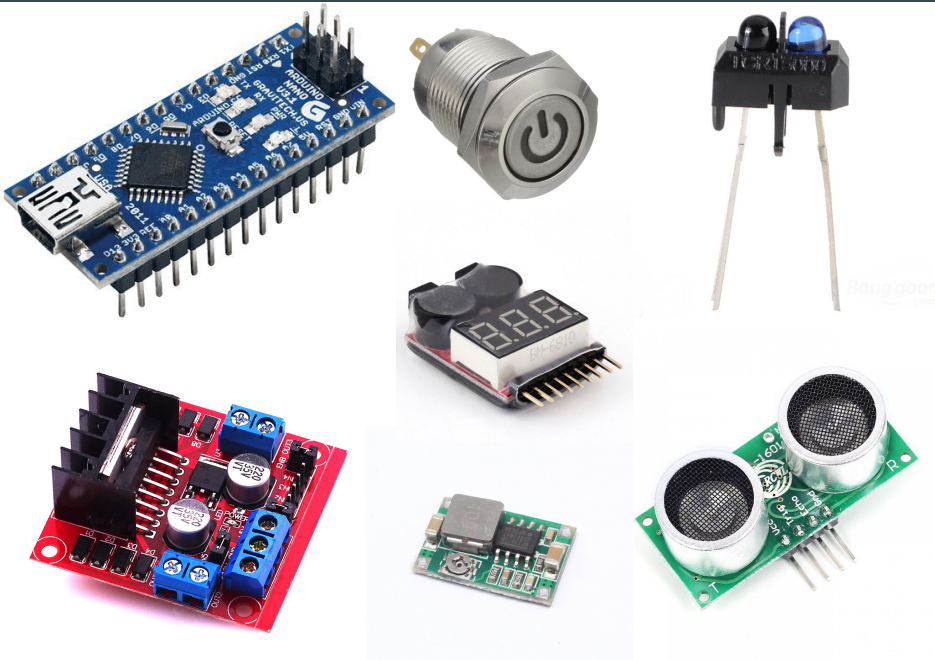
- 3 partes:
  - Caja de engranajes
  - Oruga con filamento flexible (TPU)
  - Frame para la oruga y conexión con engranajes
- Lubricación de engranajes con vaselina (el plástico para impresoras 3D lo prefiere a otros lubricantes)
- Muchos tornillos (unos 120, 64 para la oruga)
- Reducción de velocidad aproximadamente 86:1 (86 veces más lento, pero mayor par motor)

# Construcción del diseño 3D



Para ver el diseño  
online:  
[a360.co/3mfHlEx](https://a360.co/3mfHlEx)

# Electrónica: Componentes

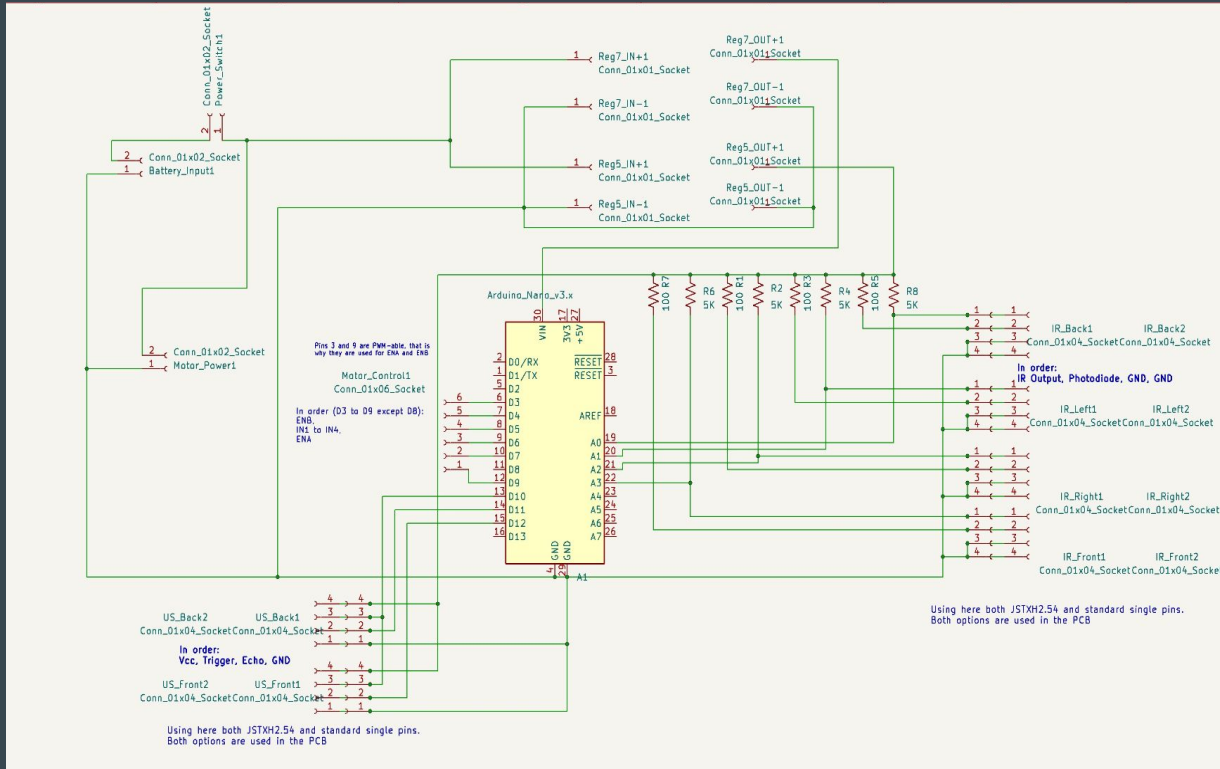


Componentes usados:

- **Arduino NANO v3**
  - **L298N motor driver**
  - **RCWL-1601 US sensor x2**
  - **TCRT5000 IR sensor x4**
  - **Step-Down converter Mini 360 x2**
  - **Batería de 14.8V y 1550 mAh**
  - **Alarma para batería baja**
  - **Motores de un VHS antiguo x2**
  - **Rockobot V1.0 PCB**
  - **Botón guay de Aliexpress**
  - **Conector XT-60 (para batería)**
  - **Conectores y cables JSTXH2.5, para los sensores**
  - **Resistencias, cables, estaño...**
-



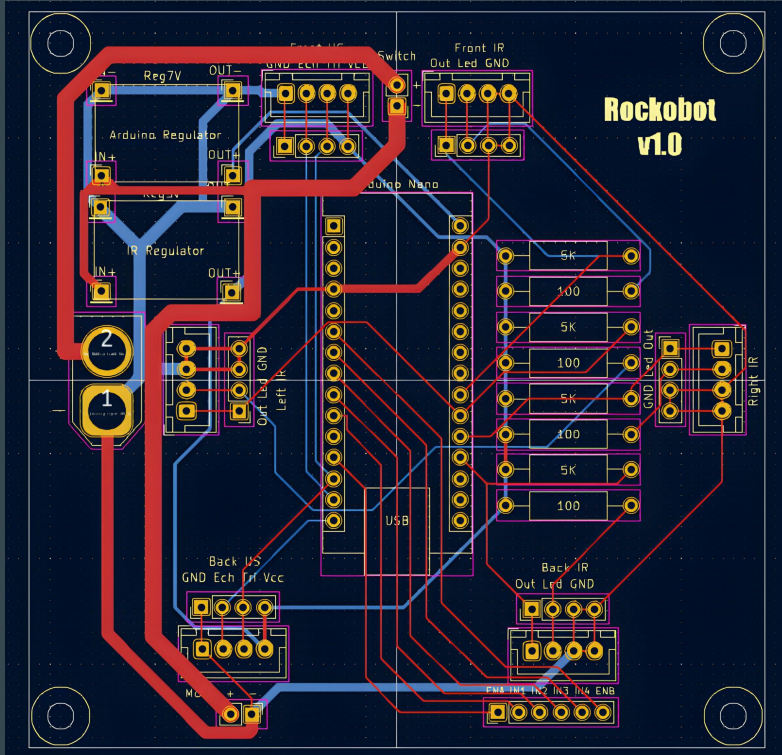
# Electrónica: Componentes



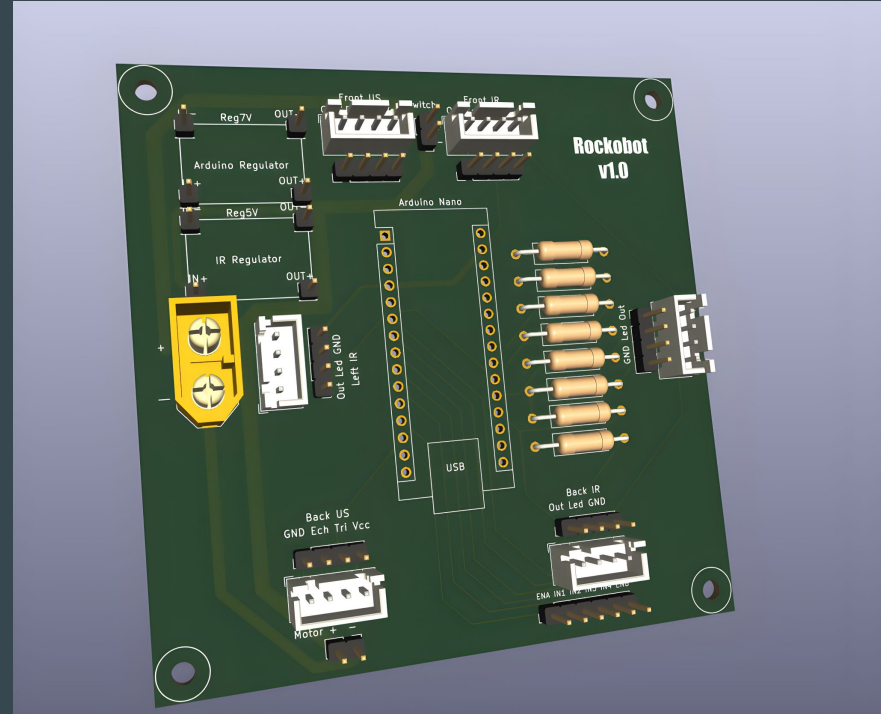
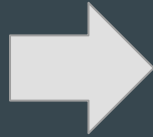
Esquema electrónico del robot. Se divide tal que:

- Entrada de la batería
- Botón de encendido
- Conexión a reguladores de voltaje
- Regulador 1 alimenta sensores de IR y sensores de US a ~5.1V
- Regulador 2 alimenta el Arduino mediante pin Vin a unos ~7.5V (el regulador interno reduce a 5V)
- Conexión de pines del Arduino a sensores y driver
- Alimentación directa de la batería al driver de motor

# Electrónica: PCB

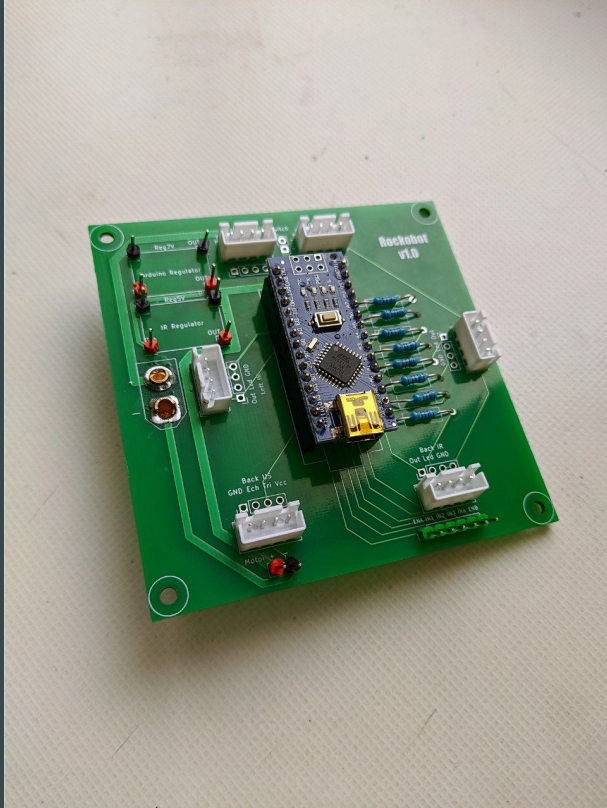


*Diseño en KiCad*

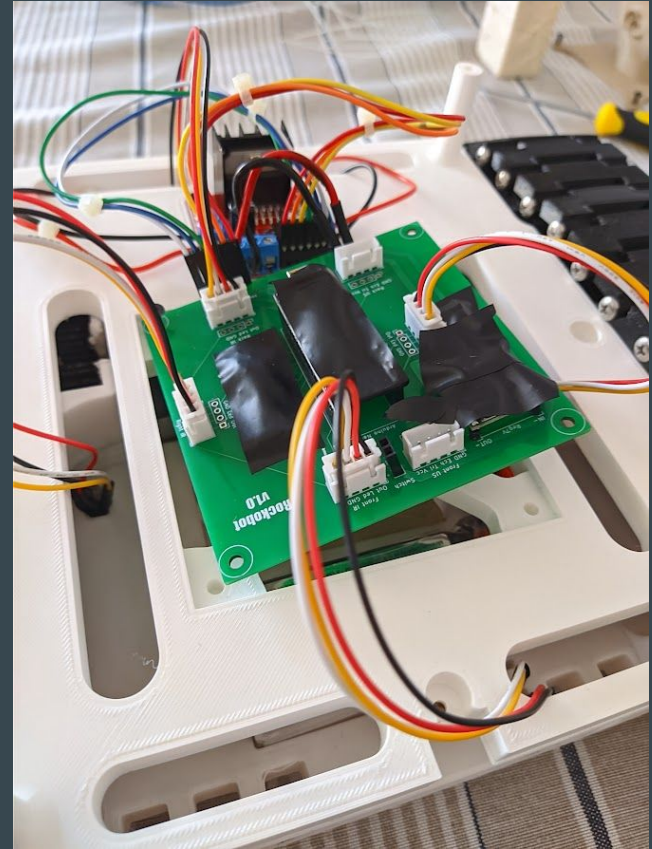


*Diseño 3D en KiCad*

# Electrónica: PCB

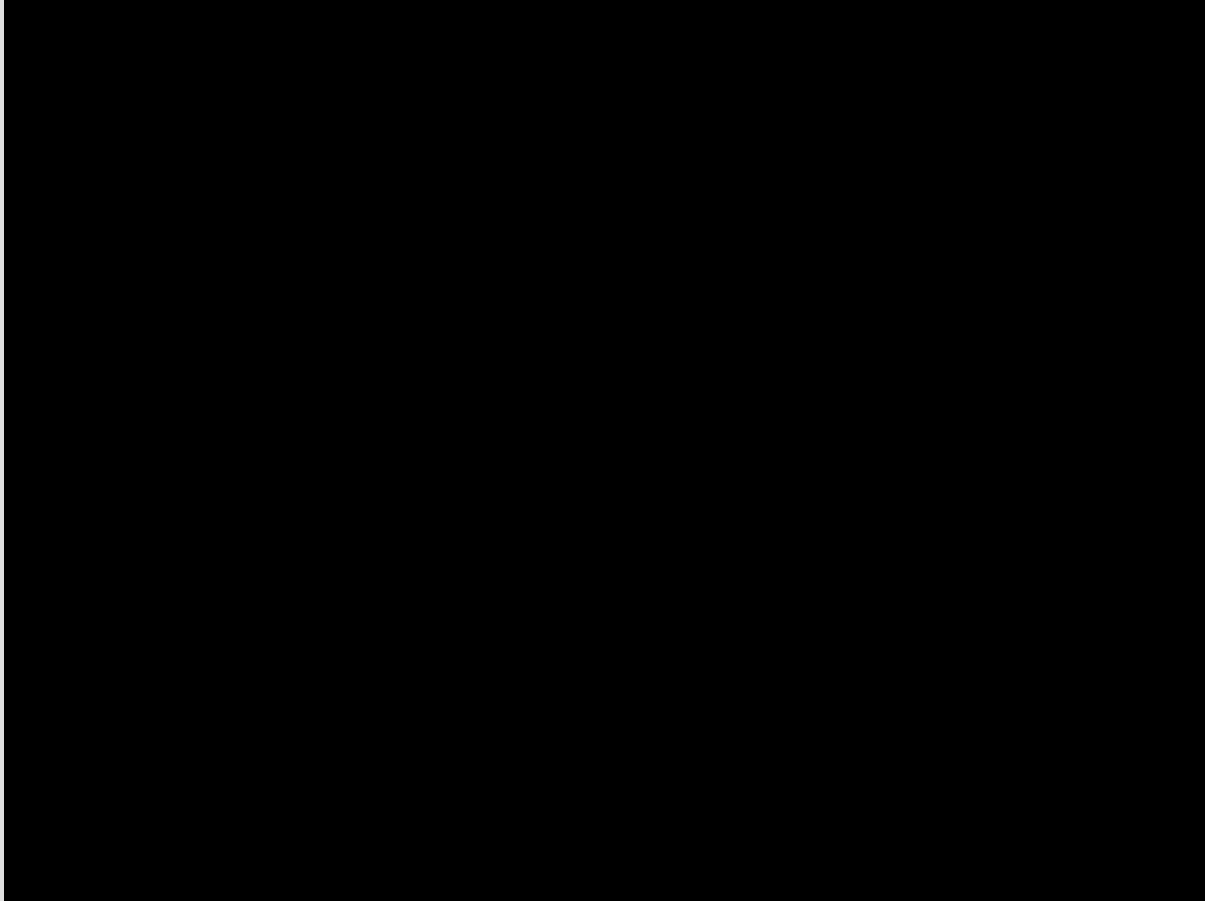


*Montaje casi final*



*PCB montada en el robot*

# Probando la fuerza del motor



# Programación

## Sensores

- **Ultrasonidos:** librería para HC-SR04, funciona con el sensor que usamos perfectamente.
- **IR reflexivo:** Solo hay que usar `analogRead()`

## Arduino

- Recibe señales de los sensores, procesa la información y decide en un conjunto de if, si avanzar, girar, retroceder, cambiar la velocidad... **cada 50 ms.** Controla el driver

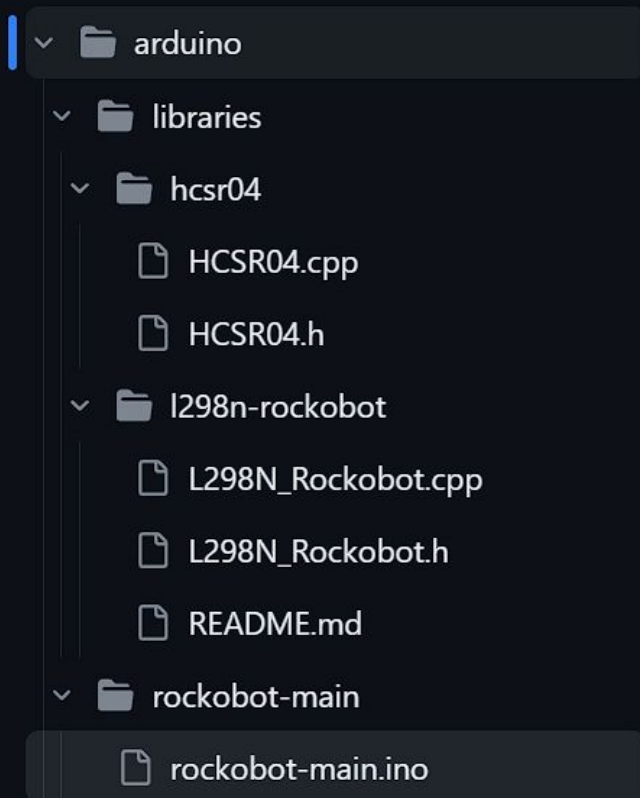
## L298N Driver

- Alimentación directa de la batería, conexión con Arduino mediante 6 pines (4 para puente H, 2 para PWM)
- Controla ambos motores mediante librería propia

## Motores

- Reciben el voltaje del driver, variable según PWM y batería restante. Se mueven a unas **20-80 RPM**

# Programación



*Estructura de ficheros para  
programar el Arduino*

La programación incluye los siguientes ficheros

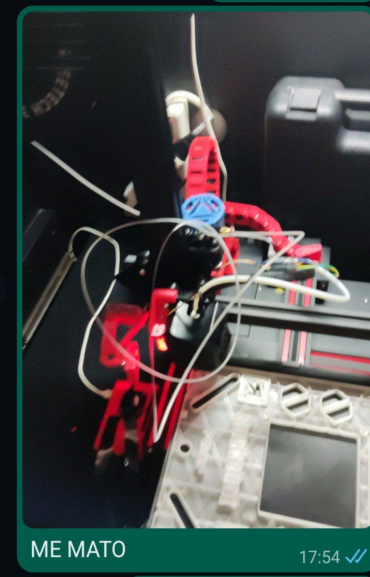
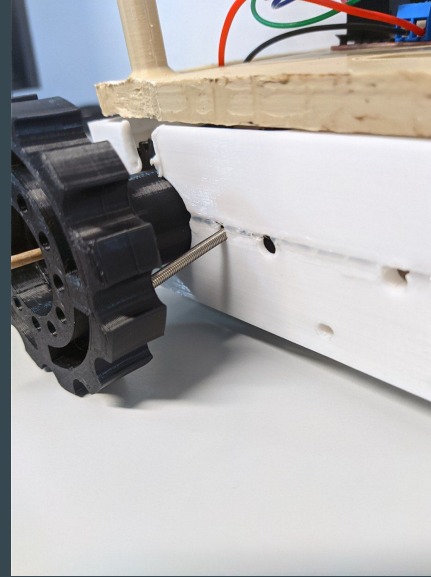
- **HCSR04.cpp**: Librería extraída de GitHub para control fácil del sensor de ultrasonidos. Funciona con nuestro modelo de ultrasonidos
- **L298N\_Rockobot.cpp**: Librería propia para el control de motores. Incluye una pequeña clase y enumerados para indicar fácilmente si se quiere girar, avanzar, retroceder, a qué velocidad, saber velocidad, rotación actual...
- **rockobot-main.ino**: Archivo principal, incluye las constantes, librerías previas, código principal... Se divide en 3 partes:
  - Leer de sensores
  - Decidir
  - Enviar a motores



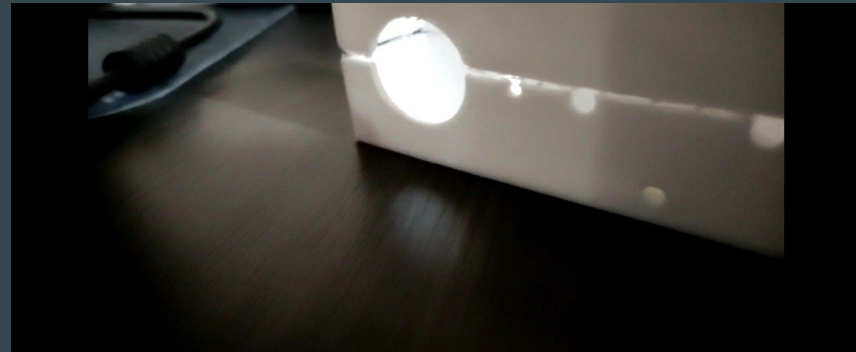
# Problemas

Lista de problemas que sufrimos:

- Cortocircuito de un Arduino
- Problemas a la hora de imprimir, tanto José como Ángel
  - Atasco en la boquilla de la impresora de Ángel
  - Rotura de una pieza en medio de una impresión (Ángel también)
  - La pieza más grande (300 gramos) tuvo un problema de impresión (José)
- El montaje, tanto de los motores, como la oruga, como de un piso a otro...
- Los malditos ejes de la reductora (salían y se doblaban)
- En una ocasión se nos fusionaron 2 engranajes de la reductora con el eje. Esto se debió al roce entre un engranaje y el otro. El PLA técnicamente empieza a derretirse a los 50-60°C, pero creemos que llegó a más.
- El diseño 3D tomó mucho tiempo, es más, el diseño actual tiene cambios que no tienen las piezas impresas del robot, tuvimos que hacer adaptaciones con silicona y (des)soldando plástico. Hay cerca de 700 dimensiones en el robot.



*Varilla doblada (izquierda), impresora se le sale el tubo en medio de impresión (derecha) y pieza de 300 g (24 horas de impresión) se salta capas de impresión (abajo)*



# ¿Github? **Sí**

[github.com/Pelochus/rockobot](https://github.com/Pelochus/rockobot)

Incluimos diseños 3D, diseño electrónico, código, archivos para imprimir, consejos para imprimir piezas resistentes, explicación de la mecánica, datasheets de los componentes electrónicos, listas de componentes, explicación de la librería para el driver, ideas para mejorar el robot, instrucciones para crear el escudo de acero, esta presentación...

# ¿Instrucciones de montaje? **No**

Ni siquiera nosotros sabemos montarlo bien



**Fin**

**Gracias**