

E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación

Luis José Llamas Pérez

Asier Ruperto Marzo Pérez  
Josu Irisarri Erviti

Pamplona, 20/01/2023



Mini Swarm Robots

Grado en Ingeniería Informática

Trabajo Fin de Grado

# Agradecimientos

A mis tutores, Asier Marzo y Josu Irisarri.

A mis compañeros de la carrera.

A Biko2 y 540.

A la Universidad Pública de Navarra

# Resumen

Este proyecto recoge el desarrollo de una inteligencia colectiva que pueda ser ampliable, de fácil implementación y bajo coste. El desarrollo de dicha inteligencia colectiva se divide en diferentes fases que abarcan tanto el diseño del robot y la inteligencia colectiva a nivel de hardware, como la implementación software de los mismos que permita los objetivos expuestos anteriormente.

A nivel de hardware se ha realizado el diseño e implementación de un chasis impreso en 3D, así como los circuitos electrónicos utilizados en los robots. Cabe destacar el uso de componentes fáciles de obtener tanto en el robot como en el sistema centralizado.

A nivel de software, esté proyecto se ha realizado desde cero con la mentalidad de que pueda ser fácilmente ampliable y configurable. Esto se ha conseguido utilizando técnicas de diseño de software como la inyección de dependencias, Mocks y TDD (Desarrollo guiado por pruebas). Para asegurar compatibilidad con múltiples sistemas, se han utilizado librerías que ya implementasen los requisitos funcionales que necesitásemos, como por ejemplo OpenCV.

# Palabras clave

Inteligencia colectiva, swarm robotics, robot, ArUco, TDD, Inyección de Dependencias, Mock.

# Índice

[Agradecimientos 2](#_Toc124069426)

[Resumen 2](#_Toc124069427)

[Palabras clave 2](#_Toc124069428)

[Índice 3](#_Toc124069429)

[Introducción 4](#_Toc124069430)

[Justificación y Objetivos 4](#_Toc124069431)

[Contexto tecnológico 5](#_Toc124069432)

[Metodología empleada 5](#_Toc124069433)

[Mocks y TDD 5](#_Toc124069434)

[Mini Swarm Robots 7](#_Toc124069435)

[Hardware 7](#_Toc124069436)

[Robots 8](#_Toc124069437)

[Sistema centralizado 8](#_Toc124069438)

[Software 8](#_Toc124069439)

[Robots 8](#_Toc124069440)

[Sistema Centralizado 8](#_Toc124069441)

[Líneas futuras 8](#_Toc124069442)

[Bibliografía y referencias 9](#_Toc124069443)

[Anexos 9](#_Toc124069444)

# Introducción

A pesar de que la inteligencia colectiva no es algo nuevo, se ha trabajado poco en investigaciones para conseguir que sea de bajo coste y ampliable para cubrir diferentes necesidades.

Este proyecto surge para dar un primer paso que solucione esos inconvenientes con el fin de atraer la atención hacía el uso de la inteligencia colectiva como forma de resolver problemas.

# Justificación y Objetivos

El proyecto parte de la premisa de que la inteligencia colectiva no tiene que ser ni cara, ni complicada, fácilmente replicable y debe ser capaz de analizar y comunicarse con el entorno.

A raíz de la premisa anterior, surgen los siguientes objetivos:

* Los robots deben poder ser construidos con diferentes componentes, es decir, no tiene por que haber dos robots iguales.
* Añadir un robot que use otro tipo de componentes no debe complicar el desarrollo ni alterar el funcionamiento de los que ya existen.
* El número de robots que puedan participar en la inteligencia colectiva ha de ser alto.
* Escalabilidad
* Los robots cumplirán una de las siguientes opciones:
  + Tener inteligencia propia y ser ellos mismo los responsables de la toma de decisiones
  + Tener un sistema centralizado que se encargue de dichas decisiones.
  + Ambas.
* El sistema de comunicación entre robots y/o el sistema centralizado tendrá que ser flexible, es decir, que se podrán utilizar diferentes tipos de sistemas de comunicación
* Los robots podrán tener comportamientos/patrones y actuar en base al entorno

Debido al amplio campo que es la inteligencia colectiva, no se van a poder abordar todos los objetivos iniciales en su totalidad, sin embargo, sí que se ha creado el camino para poder aumentar en funcionalidad y diseño de manera que en un futuro otra persona pueda continuar con el proyecto sin gran dificultad.

Para ello se han tomado una serie de decisiones que simplifican y a su vez acotan el alcance del proyecto. Estas decisiones se listan a continuación:

* Uso de material fácilmente adquirible por cualquier persona.
* Dicho material ha de ser de bajo coste.
* Software que sea fácilmente ampliable y modificable.
* Los robots serán controlados por un sistema centralizado.
* Solamente se usará un sistema comunicación entre los robots y el sistema centralizado.
* Se crearán dos robots con distintos componentes para la flexibilidad del proyecto.
* El control que ejerce el sistema centralizado será únicamente el movimiento de los robots de un punto a otro.

En los siguientes puntos se hablará sobre el hardware y software concreto que se ha decidido usar en el proyecto.

# Contexto tecnológico

La inteligencia colectiva es definida como el estudio del comportamiento colectivo cuando un sistema este compuesto de múltiples individuos que actúan en busca de un objetivo común. Esto permite abordar problemas de gran complejidad con individuos cuyas capacidades físicas/computacionales por separado sean inferiores a lo requerido.

Esto se consigue mediante el uso de comandos e interacciones simples que son elaboradas por un sistema central o individuos de mayor rango jerárquico.

# Metodología empleada

## Mocks y TDD

La metodología empleada para el desarrollo de este proyecto ha sido una modificación de TDD (Desarrollo guiado por pruebas) junto al uso de Mocks (Objetos simulados). Esto nos ha permitido comenzar a trabajar en el software sin conocer aún como va a ser el hardware, de esta forma, hemos conseguido desacoplar la dependencia del software y hardware.

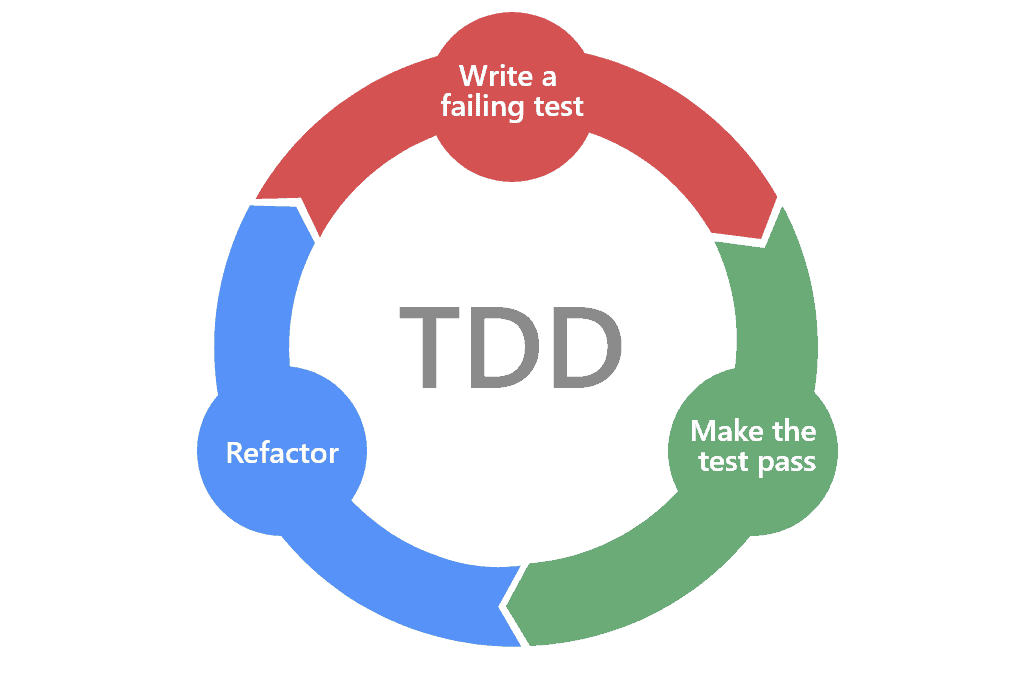


Figura : Diagrama de los estados en TDD (Rojo, Verde, Azul)

La manera en la que lo hemos logrado ha sido mediante la creación de Mocks de objetos que más adelante serán implementados de forma “real”.

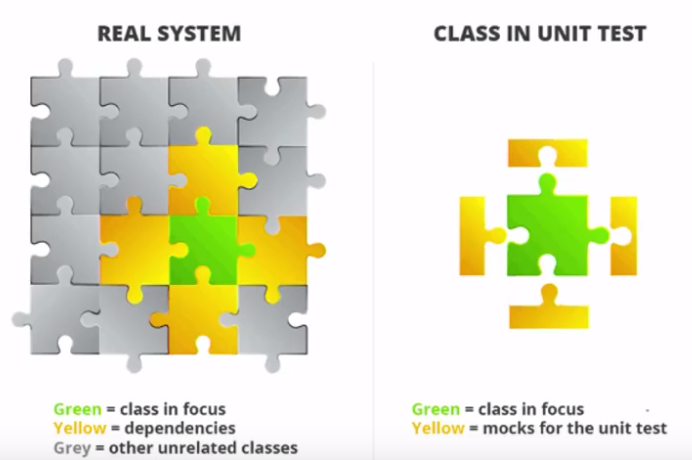


Figura : Ejemplo visual de Mocks

Usando la imagen de la [Figura 2](#Figura_2) como ejemplo, la parte en verde sería el sistema centralizado, mientras que las partes amarillas serían las dependencias del mismo, como, por ejemplo:

* Webcam (Para la obtención de posicionamiento de los robots).
* Sistema de comunicaciones (Para dar las órdenes a los robots).
* Robot (Para ejecutar las ordenes).

Si no hubiésemos usado Mocks, tendríamos que haber esperado a la implementación real de estos componentes antes de ponernos a trabajar en el sistema centralizado, ya que, no tendríamos feedback de si lo que estamos desarrollando es funcional o no.

Para solventar esto, se ha realizado un análisis previo de los componentes/actores necesarios por el sistema. Estos actores son candidatos a poder ser Mockeados, es decir, a crear una implementación “falsa” con las características y métodos que serán utilizados por la implementación real.

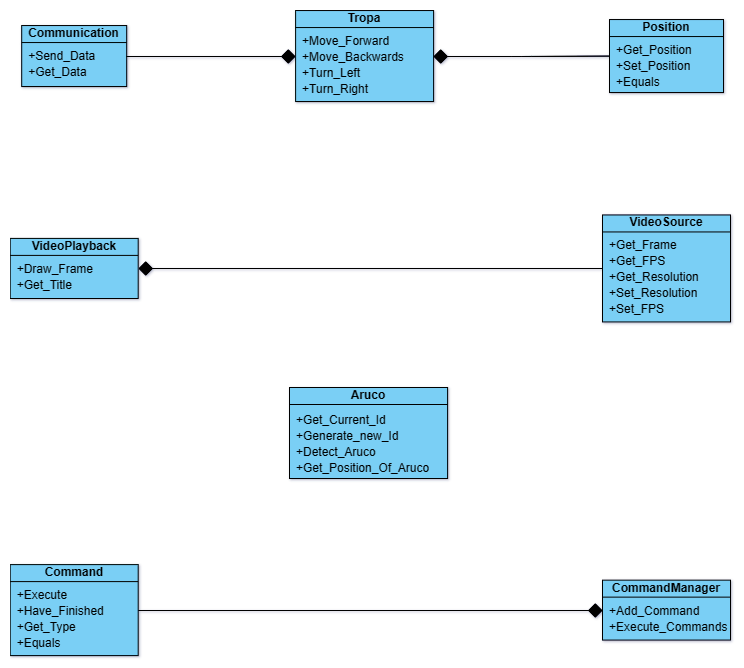


Figura : Actores del sistema centralizado

Los robots (Tropas) son un perfecto candidato a ser Mockeados, ya que nos permite desarrollar sin tener dicho robot. Esto genera un efecto mariposa, ya que, al no tener el robot, tampoco podemos obtener su posición usando una Webcam (VideoSource), por lo que el VideoSource también será mockeado. Lo mismo ocurre también con el sistema de comunicaciones (Communication).

El resto de componentes no tienen necesidad de ser Mockeados, al menos a primera vista, por lo que se puede proceder a la creación del sistema centralizado.

## Inyección de Dependencias

Dado que hemos realizado el análisis de los componentes y sus acciones/métodos, podemos enfocar el desarrollo de estos usando la inyección de dependencias, lo cual nos permitirá el desacople de las implementaciones finales consiguiendo así, un software fácilmente mantenible y actualizable.

El primer paso es crear interfaces en donde antes eran actores simples. Estas interfaces declararan los atributos y métodos/acciones del actor. A raíz de esta interfaz, se crearán las implementaciones finales, las cuales deben cumplir con el “contrato” (interfaz) que hereden.

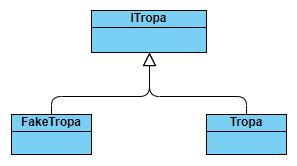


Figura : Primer paso para la Inyección de Dependencias

En el ejemplo de la [Figura 4](#Figura_4) se ha creado la interfaz ITropa la cual tendrá una serie de métodos/acciones que tanto FakeTropa como Tropa deberán de implementar, cada uno a su manera.

Los Robots (Tropas) como hemos visto en la [Figura 3](#Figura_3), dependen del sistema de comunicaciones (Communication), por lo que estamos ante de una dependencia que podemos romper y hacerlo más flexible aplicando la simplificación de la [Figura 4](#Figura_4) al actor Communication.

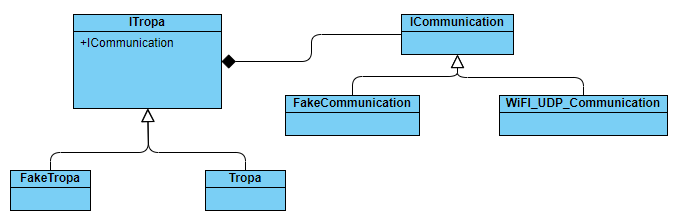


Figura : Inyección del sistema de comunicaciones como dependencia

Como podemos observar en la [Figura 5](#Figura_5), las Tropas ya no dependen de la implementación concreta del sistema de comunicaciones, si no que podemos mezclar, por ejemplo, el uso de una Tropa falsa y un sistema de comunicaciones basado en UDP por WiFi. Esto es lo que nos va a permitir añadir en un futuro de forma sencilla implementaciones concretas (ya sean reales o Mocks) sin tener que modificar el código que ya existía previamente.

Tras realizar este ejercicio sobre el resto de los actores del Sistema Centralizado, obtenemos el diagrama de clases de la [Figura 6](#Figura_6).

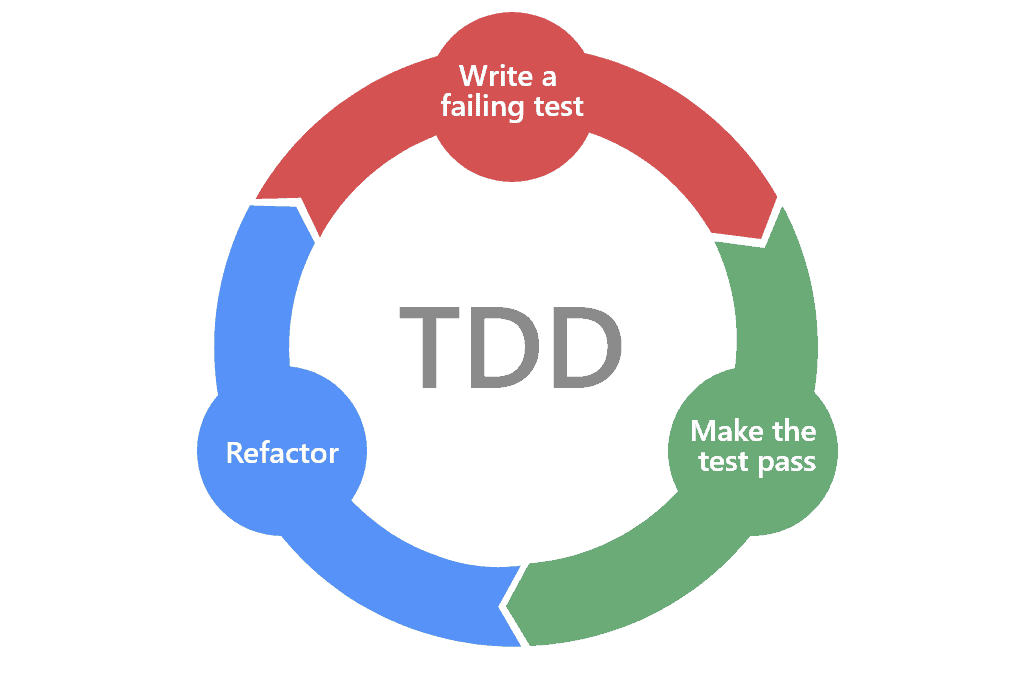


Figura : Diagrama de clases del Sistema Centralizado

Tal y como hemos visto en el sistema centralizado, también se pueden aplicar estas técnicas a la hora de la programación de los robots, donde la abstracción será en los componentes del Robot (Motor, Led, Sistema de Comunicación).

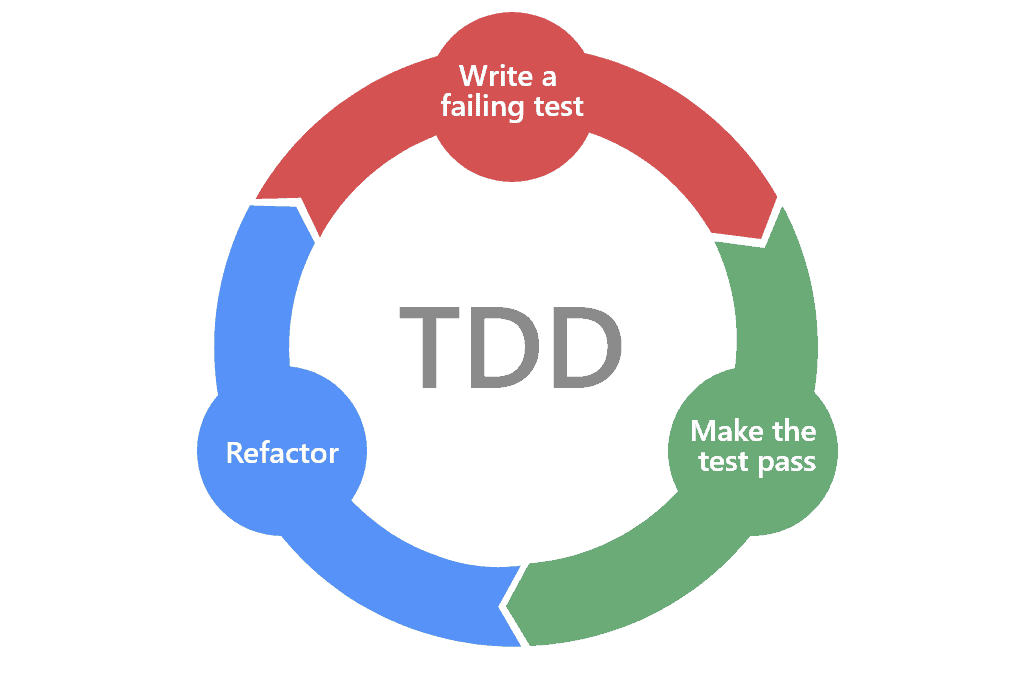


Figura : Diagrama de clases del Robot

Una vez se ha creado el software, pasar al hardware es trivial, ya que, se utilizan componentes fácilmente obtenibles, donde la única preocupación es la “conexión” entre ellos, motores y microcontrolador, por ejemplo.

# Mini Swarm Robots

## Hardware

El hardware como se ha explicado anteriormente, es de bajo coste y accesible. En líneas generales hay dos grandes separaciones, el hardware de los robots y el del sistema centralizado.

### Robots

En el caso de los robots, lo más importante para cumplir los objetivos, es el movimiento. En este caso se ha decidido usar un sistema de dos ruedas motorizadas encargadas de la tracción y una rueda “loca”.

Las ruedas motorizadas como su nombre indica, consisten en dos motores que al girar a distintas velocidades consiguen el movimiento y giro del robot. Para controlar dichos motores, se utiliza un microcontrolador que será el cerebro del robot y para alimentar el robot, un Powerbank de 5V y 1000mAh. Todo esto va integrado en un cuerpo impreso en 3D que permita que todos los componentes tengan un balance de peso optimo.

#### Microcontrolador

El microcontrolador que se ha utilizado es un ESP32 WROOM-32, el cual viene equipado con WiFi y Bluetooth.



Figura : ESP32 WROOM-32

Este microcontrolador es más que suficiente para el proyecto y su elección radica en el hecho de que tiene un precio asequible, es fácil de obtener y tiene una gran cantidad de puertos de entrada/salida, así como comunicaciones WiFi y Bluetooth incluidas.



Figura 9: Pinout del ESP32 WROOM-32

Como características importantes, el microcontrolador cuenta con un regulador de voltaje que permite ser alimentado desde 12V hasta 3.3V. Cabe destacar que el voltaje al que opera este dispositivo es de 3.3V, por lo que puede dar problemas con circuitos integrados cuyo umbral de activación sea superior a dicho voltaje. Esto puede remediarse con un transistor que actúe de Step-Up usando un pin de entrada/salida (GPIO) del microcontrolador.

Otra característica de este microcontrolador es la posibilidad de utilizar un sistema de comunicaciones propio basado en WiFi llamado ESP-MESH, el cual puede ser interesante de cara a aumentar tanto el número de robots, como la distancia entre estos y el sistema centralizado.

#### Motor

Los motores utilizados en los robots que se han prototipado son motores DC simples con una caja reductora y motores Servo modificados para permitir el giro completo.

Para el caso de los motores DC simples se ha utilizado el circuito integrado L293D para controlar hasta dos motores por circuito.

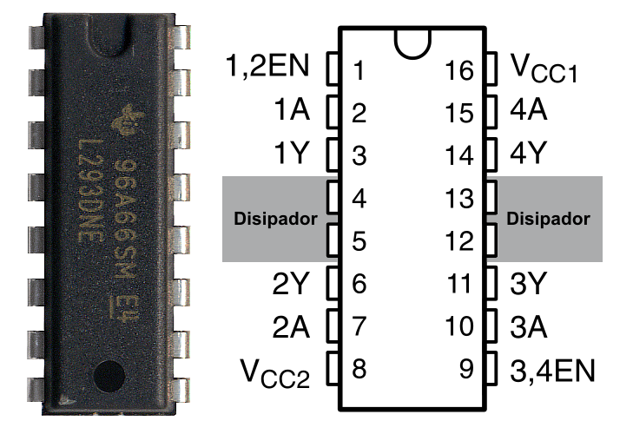


Figura : Pinout del IC L293D

En el caso de los motores Servo, no se ha necesitado el uso de ICs ya que, estos incluyen su propio controlador, así como un sistema de engranajes que actúa de reductora. La desventaja es que por cada Servo hay que conectar tres pines al microcontrolador, mientras que en el caso del L293D, hay que conectar cuatro pines por cada dos motores.



Figura : Motor Servo

#### Chasis

El chasis empleado se ha diseñado para los componentes específicos que se han usado en los robots. Es un prototipo simple que se puede imprimir rápidamente y que tiene en cuenta el equilibrio de los pesos de los componentes para tratar de conseguir un movimiento menos errático.

El chasis se ha diseñado utilizando la herramienta Autodesk Fusion 360 y se ha impreso en una impresora Prusa MK3 utilizando PrusaSlicer para convertir del formato STL a GCODE.

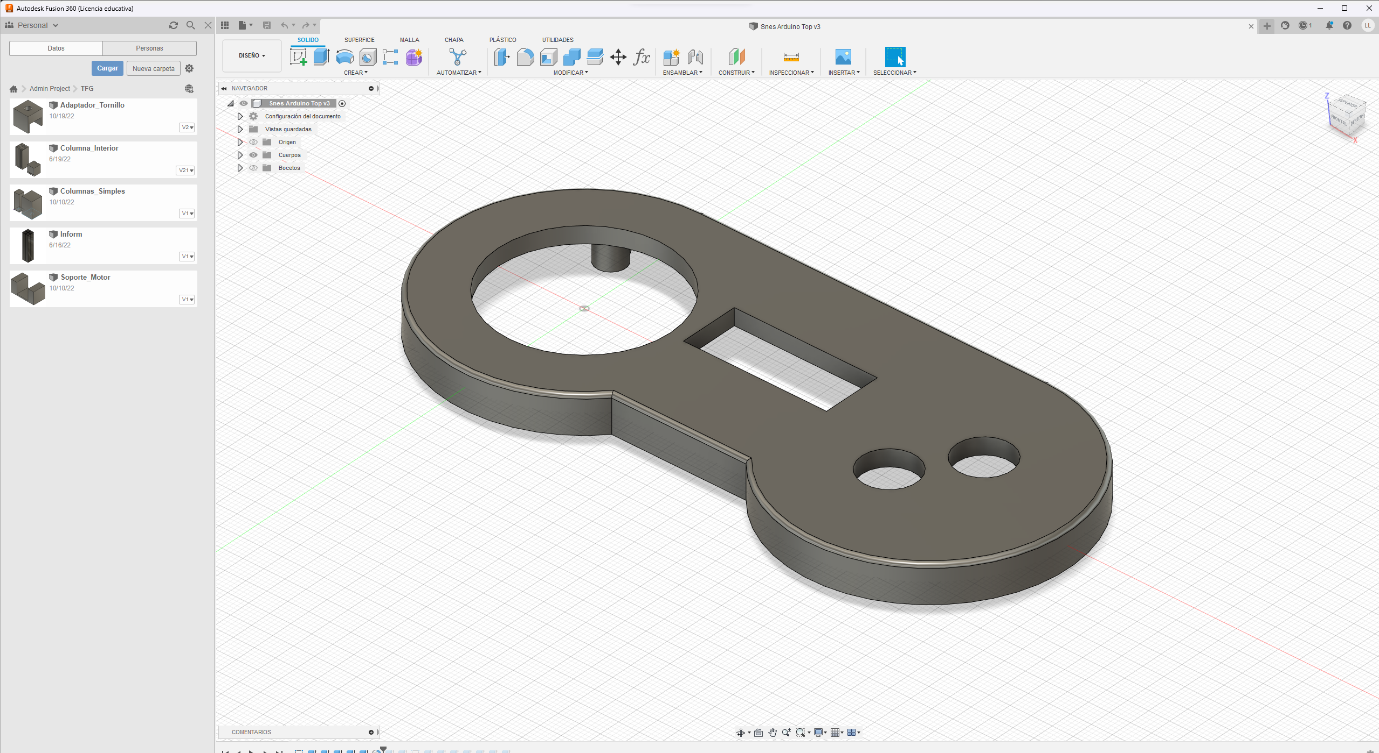


Figura : Autodesk Fusion 360

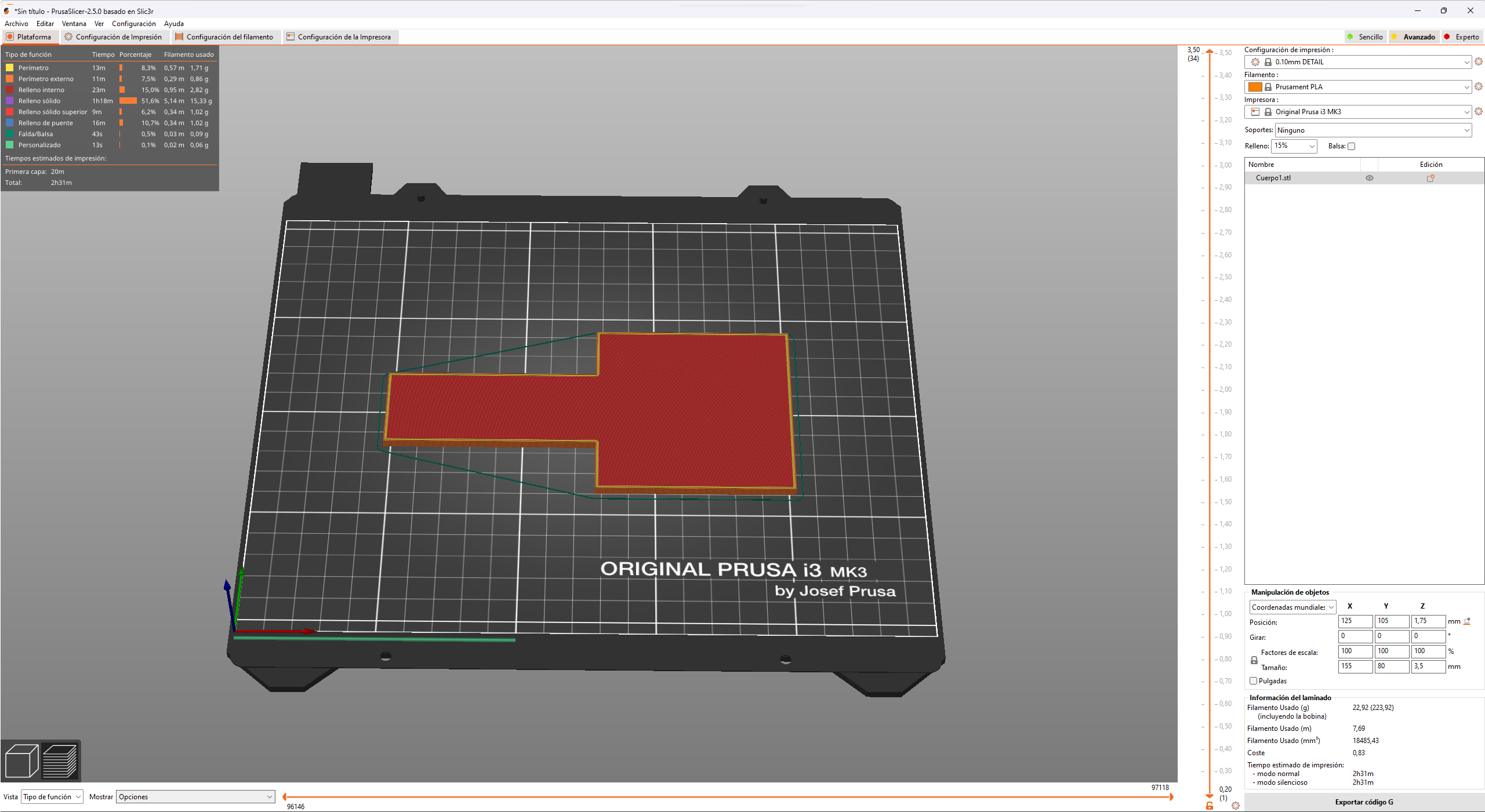


Figura : PrusaSlicer

### Circuitos

La Figura 14 muestra como se ha realizado el conexionado del robot en el caso de los motores DC.

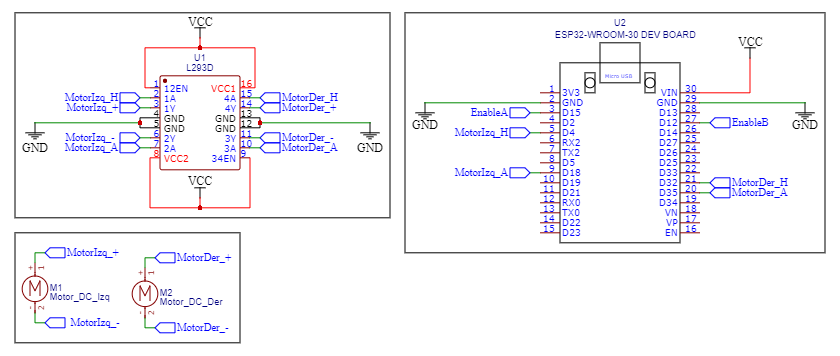


Figura : Circuito del Robot con motores DC

### Sistema centralizado

El hardware del sistema centralizado se compone de las siguientes partes:

* PC donde ejecutar el código
* Webcam que irá conectada al PC
* Red WiFi (para el caso concreto que hemos implementado)

Los requisitos mínimos del PC son el poder ejecutar Python 3.9, al menos tres núcleos/hilos y 4 GB de RAM.

No se especifica arquitectura ya que, es valido tanto en un sistema convencional x86/x64 como en ARM64 (aarch64).

El requisito de los tres núcleos/hilos viene dado de la utilización de multihilo para mejorar el rendimiento del código. Esto viene explicado con detalle en el apartado de Software.

## Presupuesto

Ya que se ha hecho hincapié en que el proyecto sea accesible y barato, a continuación, se recoge una lista de los materiales necesarios para la recreación del proyecto (\*) junto a su precio.

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Cantidad** | **Precio unitario** | **Total** |
| [ESP32-WROOM](https://es.aliexpress.com/item/1005002410521023.html?spm=a2g0o.productlist.main.5.462a74dfUrTuz6&algo_pvid=88e954ce-d38c-4ab3-9080-c7069d48b003&algo_exp_id=88e954ce-d38c-4ab3-9080-c7069d48b003-2&pdp_ext_f=%7B%22sku_id%22%3A%2212000028255222863%22%7D&pdp_npi=2%40dis%21EUR%214.46%213.57%21%21%21%21%21%402145274c16731841645028946d06ca%2112000028255222863%21sea&curPageLogUid=b8SKd9ntGUEf) | 1 | 3.57€ | 3.57€ |
| [IC L293D](https://es.aliexpress.com/item/32844099228.html?spm=a2g0o.productlist.main.1.5c6f4825TuRoZA&algo_pvid=a21a1464-d6ce-4626-b2cb-4fd601a1f36c&algo_exp_id=a21a1464-d6ce-4626-b2cb-4fd601a1f36c-0&pdp_ext_f=%7B%22sku_id%22%3A%2265215204221%22%7D&pdp_npi=2%40dis%21EUR%212.47%212.18%21%21%21%21%21%402100baf316731841869147135d06d7%2165215204221%21sea&curPageLogUid=yA0Ik98ThbJH) | 1 | 0.22€ | 0.22€ |
| [Motor DC](https://es.aliexpress.com/item/1005002264331819.html?spm=a2g0o.productlist.main.43.335920cav0mdNh&algo_pvid=8e608a86-36ae-431a-a51a-eb8e9bd53c4d&algo_exp_id=8e608a86-36ae-431a-a51a-eb8e9bd53c4d-21&pdp_ext_f=%7B%22sku_id%22%3A%2212000019811879381%22%7D&pdp_npi=2%40dis%21EUR%211.79%211.45%21%21%21%21%21%402100b76616731842060561560d06ba%2112000019811879381%21sea&curPageLogUid=qrvC43PMpnFg) | 2 | 0.73€ | 1.46€ |
| [PowerBank](https://es.aliexpress.com/item/1005004227379153.html?spm=a2g0o.productlist.main.43.38c235bbJJd8vR&algo_pvid=8a205a1f-a97f-41df-966f-bc9510d50e9e&algo_exp_id=8a205a1f-a97f-41df-966f-bc9510d50e9e-21&pdp_ext_f=%7B%22sku_id%22%3A%2212000030491195755%22%7D&pdp_npi=2%40dis%21EUR%213.99%213.99%21%21%21%21%21%402100b76616731842760402127d06ba%2112000030491195755%21sea&curPageLogUid=xNM8dMRF95fI) | 1 | 1.29€ | 1.29€ |
|  |  |  |  |
| **Total** | **6.54€** | | |

Tabla : Presupuesto del Robot

\*: En este presupuesto no se han tenido en cuenta el gasto de material 3D ya que es despreciable. A su vez, estos precios se han obtenido de Aliexpress en el momento de su redacción, por lo que pueden estar sujetos a cambios.

## Resultados

Los resultados de esta implementación Hardware pueden ser vistos en la Figura X.

## Software

Asas

### Robots

Asas

### Sistema Centralizado

asasas

## Líneas futuras

Aas

# Bibliografía y referencias

<https://es.wikipedia.org/wiki/Inteligencia_colectiva>

<https://es.wikipedia.org/wiki/Inteligencia_de_enjambre>

<https://www.hiberus.com/crecemos-contigo/todo-lo-que-necesitas-saber-de-tdd-en-3-minutos/>

<https://tienda.bricogeek.com/arduino-compatibles/1274-esp32-wroom-wifi-bluetooth.html>

<https://naylampmechatronics.com/>

<https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>

<http://robots-argentina.com.ar/didactica/wp-content/uploads/pinout-L293D.png>

<https://educarobots.com/producto/mini-servo-motor-sg90/>

# Anexos