Logotipo

Descripción generada automáticamente

**Plataforma dibujante con reconocimiento de imágenes**

**Informe final**

Grupo 2

Caciani Toniolo, Melina 02866/1

Chanquía, Joaquín 02887/7

Ollier, Gabriel 02958/4

UNLP

Facultad de Ingeniería

Departamento de Electrotecnia

Taller de Proyecto I (E0306)

13 de febrero de 2025

# ÍNDICE

[INTRODUCCIÓN 4](#_Toc190428730)

[OBJETIVOS 5](#_Toc190428731)

[PRIMARIOS 5](#_Toc190428732)

[SECUNDARIOS 5](#_Toc190428733)

[REQUERIMIENTOS 5](#_Toc190428734)

[FUNCIONALES 5](#_Toc190428735)

[HARDWARE 5](#_Toc190428736)

[SOFTWARE 6](#_Toc190428737)

[NO FUNCIONALES 6](#_Toc190428738)

[DISEÑO DE HARDWARE 7](#_Toc190428739)

[CIRCUITO ESQUEMÁTICO 8](#_Toc190428740)

[DISEÑO DE LA PCB 10](#_Toc190428741)

[ENSAMBLAJE DEL ROBOT 11](#_Toc190428742)

[MATERIALES UTILIZADOS 13](#_Toc190428743)

[DISEÑO DE SOFTWARE 13](#_Toc190428744)

[INTERFAZ WEB 14](#_Toc190428745)

[COMUNICACIÓN WEBSOCKET 15](#_Toc190428746)

[RECONOCIMIENTO DE IMÁGENES 16](#_Toc190428747)

[MOVIMIENTO DE MOTORES 17](#_Toc190428748)

[CONEXIÓN Y COMUNICACIÓN UART 19](#_Toc190428749)

[CÓDIGO 19](#_Toc190428750)

[VALIDACIÓN 20](#_Toc190428751)

[CONCLUSIÓN 20](#_Toc190428752)

[CUMPLIMIENTO DE OBJETIVOS 20](#_Toc190428753)

[EVALUACIÓN DE REQUERIMIENTOS 21](#_Toc190428754)

[ESCALABILIDAD 24](#_Toc190428759)

[BIBLIOGRAFÍA 25](#_Toc190428760)

[ANEXO 26](#_Toc190428761)

[DIVISIÓN DE TAREAS 26](#_Toc190428762)

[CALENDARIO DE ACTIVIDADES 27](#_Toc190428763)

# INTRODUCCIÓN

Este proyecto nació con la idea original de desarrollar una plataforma mecánica capaz de jugar al ta-te-ti de manera autónoma. Sin embargo, a medida que se avanzó en el diseño y análisis del sistema, se identificó que un enfoque más versátil y escalable sería desarrollar un robot dibujante. Este cambio no solo simplificó el desarrollo inicial, sino que también abrió la posibilidad de expandir sus capacidades en el futuro.

El robot, en su versión actual, es capaz de realizar dibujos básicos con una serie de figuras predefinidas. Sin embargo, su potencial va mucho más allá: con la implementación de un sistema basado en instrucciones tipo G-code (un lenguaje utilizado para controlar máquinas de dibujo y corte), podría evolucionar hasta ser capaz de dibujar cualquier diseño con precisión. Además, al integrar una ESP32-CAM, el sistema puede reconocer figuras como círculos y cruces, lo que ya permite una funcionalidad clave para jugar al ta-te-ti. En el futuro, si se mejora la detección de la posición exacta de cada figura, el robot podría jugar al ta-te-ti de manera autónoma y con mayor precisión.

Más allá de este juego, el proyecto puede escalar a aplicaciones aún más interesantes, como escribir textos, reproducir bocetos complejos o incluso la interacción con otros sistemas inteligentes. De este modo, el proyecto no solo cumple con su objetivo inicial, sino que también se convierte en una herramienta escalable con múltiples aplicaciones potenciales.

Este informe detalla el diseño, implementación y funcionamiento del robot dibujante, así como las bases para futuras mejoras que permitirán ampliar sus capacidades.

# OBJETIVOS

## PRIMARIOS

* Diseñar y ensamblar una estructura capaz de soportar el peso de los componentes electrónicos y mecánicos.
* Implementar un sistema que permita al robot dibujar figuras predeterminadas de manera clara.
* Desarrollar un sistema de reconocimiento de imágenes que pueda identificar determinadas figuras realizadas por el usuario.
* Implementar un sistema que permita al usuario seleccionar los distintos modos de uso.
* Implementar un modo de uso que permita al usuario trazar una figura, que el robot la identifique y, si corresponde a una de las figuras predeterminadas, que la dibuje.

## SECUNDARIOS

* Programar un algoritmo eficiente que permita al robot jugar al TA-TE-TI, tomando decisiones estratégicas según las jugadas detectadas.
* Incorporar diferentes niveles de dificultad para el juego del TA-TE-TI.
* Notificar al jugador el resultado final de la partida, ya sea victoria, derrota o empate.

# REQUERIMIENTOS

## FUNCIONALES

### HARDWARE

* El sistema debe incorporar una cámara con resolución suficiente para detectar de manera precisa las figuras realizadas por el usuario.
* Los motores paso a paso deben ser capaces de posicionar el marcador con precisión en los diferentes sectores de la hoja.
* El sistema debe contar con rieles que permitan el movimiento lineal de la estructura.
* La máquina debe tener un tamaño y peso adecuado para facilitar su transporte.
* Se debe integrar una pantalla LCD para comunicar los modos de uso.
* Deben incluirse botones físicos que permitan una configuración sencilla de la máquina.

### SOFTWARE

* El software de reconocimiento de imágenes debe analizar las ilustraciones capturadas por la cámara y detectar las figuras (preestablecidas) hechas por el usuario.
* El software debe enviar comandos precisos a los motores paso a paso para mover el marcador en la hoja, siguiendo las decisiones del algoritmo de dibujo.
* El sistema será probado y validado para garantizar su funcionamiento óptimo y el cumplimiento de los requerimientos establecidos.

## NO FUNCIONALES

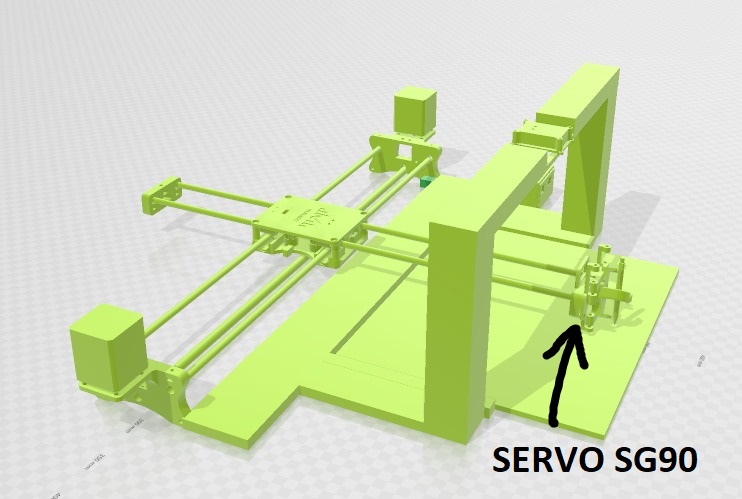
* El proyecto debe implementarse utilizando la placa EDU-CIAA.
* El proyecto debe estar finalizado, junto con su documentación correspondiente, antes del final de la cursada de "Taller de Proyecto 1" del segundo semestre de 2024.
* El desarrollo del proyecto debe ajustarse a un presupuesto preestablecido.
* La máquina debe diseñarse de manera que garantice la seguridad del usuario durante su uso.
* La interfaz del sistema debe ser fácil de usar, con instrucciones claras, y adecuada para personas de diferentes edades y niveles de habilidad.

# DISEÑO DE HARDWARE

Luego de varios análisis y pruebas, se definió la siguiente arquitectura del sistema para alcanzar los objetivos propuestos:

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.También se ideó una estructura del robot similar a la que se puede observar a continuación:

## CIRCUITO ESQUEMÁTICO



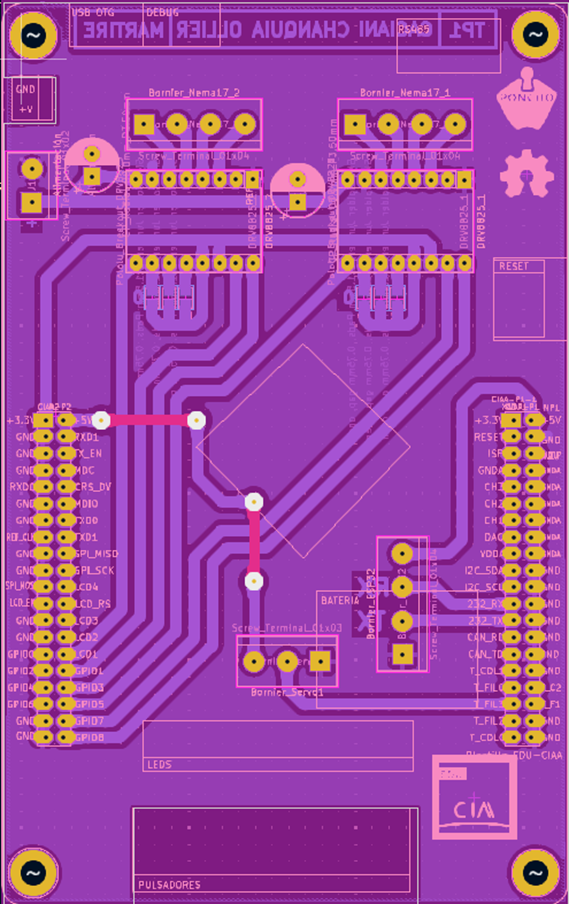
La única modificación que sufrió este esquema se debió a que, durante los ensayos, se observó que ni la ESP32-CAM ni el servo SG90 recibían correctamente los 5V provenientes de la EDU-CIAA. Tras realizar mediciones, se determinó que la EDU-CIAA sí recibía 5V desde la PC, pero al aumentar la demanda de corriente, la tensión comenzaba a caer. Esto ocurría principalmente porque el servo SG90 tiene un consumo elevado de corriente, lo que generaba caídas de tensión en la alimentación de la EDU-CIAA.

Cuando se alimenta desde un puerto USB, el límite típico de corriente es de 500 mA, de los cuales la EDU-CIAA ya consume aproximadamente 200 mA en funcionamiento. Esto dejaba un margen muy reducido para alimentar otros dispositivos. Al activar el servo, se producían picos de corriente que superaban la capacidad del puerto USB, lo que resultaba en una disminución de la tensión en la entrada de la EDU-CIAA. Inicialmente, esto afectaba al servo, ya que no recibía la tensión ni la corriente adecuadas, pero en algunos momentos la caída de tensión llegaba a afectar al propio microcontrolador de la EDU-CIAA, reduciendo su alimentación por debajo de los 3.3V. Esto causaba fallos en la ejecución de los programas e incluso impedía la carga de nuevos códigos en la placa.

Para solucionar este problema, se decidió alimentar la ESP32-CAM y el servo SG90 directamente desde la fuente de alimentación de PC, la cual inicialmente solo suministraba 12V a los motores paso a paso, pero ahora también proporciona 5V estables para la ESP32 y el servo.

## DISEÑO DE LA PCB

A continuación, se presenta el diseño final de la PCB, que difiere en algunos detalles de la última versión descripta en el informe anterior. Estos cambios se hicieron con el objetivo de agregar un capacitor más para la entrada de la alimentación del motor en cada controlador, ya que en el diseño anterior se utilizaba uno solo para ambos controladores.



## ENSAMBLAJE DEL ROBOT

Con todos los componentes necesarios en funcionamiento y previamente testeados, incluyendo las piezas impresas en 3D, la PCB fabricada y otros elementos como varillas, correas y rodamientos, se pudo dar inicio al armado del robot. El proceso comenzó con la soldadura de los componentes en la PCB.

Imagen que contiene tabla, circuito

El contenido generado por IA puede ser incorrecto.Imagen que contiene electrónica, circuito

El contenido generado por IA puede ser incorrecto.

Seguido por el ensamblaje de la estructura mecánica y la instalación de un soporte para la ESP32-CAM.

Imagen que contiene persona, mujer, tabla, hombre

El contenido generado por IA puede ser incorrecto.Imagen que contiene interior, tabla, equipo, colgando

El contenido generado por IA puede ser incorrecto.

## MATERIALES UTILIZADOS

* 1 EDU-CIAA-NXP
* 1 ESP-32 CAM
* 2 Motor NEMA17
* 2 DRV8825
* 1 Servo SG90
* Fuente de alimentación para PC
* Cable UTP
* Borneras
* 2 capacitores 100 uF x 16V
* Placa perforada
* 2 varilla de rodamiento lineal M8 x 450 mm (eje x)
* 2 varilla de rodamiento lineal M8 x 350 mm (eje y)
* 2 varilla lineal M3 x 75mm (eje z)
* 8 rodamientos lineales LM8UU
* 2 polea GT2 16 dientes
* 1 correa GT2 de 2000 mm.
* 5 rodamientos 624zz
* 4x [idler\_18x13x6\_01.stl](https://cults3d.com/en/3d-model/tool/idler-pulley-for-624-bearing-and-gt2-belt)
* Tornillos, tuercas y arandelas.
* [Piezas impresas en 3D.](https://drive.google.com/drive/folders/1rTYZXP_jB97mm8S2osZCV3PcsvxQlc-Y?usp=sharing)
* Estructura de metal para sostener la ESP32-CAM

# DISEÑO DE SOFTWARE

El software del sistema está dividido en tres componentes principales:

* **Interfaz Web:** Permite la selección del modo de uso, la elección de figuras y la visualización del estado del robot.
* **ESP32-CAM:** Actúa como intermediario entre la interfaz web y la EDU-CIAA, estableciendo una comunicación bidireccional mediante WebSocket y UART. Además, realiza el reconocimiento de imágenes para el modo de replicación de dibujos.
* **EDU-CIAA NXP:** Ejecuta las instrucciones para controlar los motores paso a paso y el servomotor, permitiendo el dibujo de las figuras.

## INTERFAZ WEB

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

La interfaz web está diseñada utilizando tecnologías web estándar (HTML5, CSS3 y JavaScript vanilla), sin depender de frameworks externos para mantener la aplicación ligera y de rápida carga. El diseño implementa un sistema responsive que se adapta a diferentes dispositivos, utilizando Flexbox para layouts flexibles y Material Icons para la iconografía. La manipulación del DOM (Document Object Model) permite actualizaciones dinámicas de la interfaz en tiempo real, proporcionando feedback visual inmediato y una experiencia de usuario intuitiva durante la interacción con el robot.

En ella se permiten tres modos de uso:

* **Selección de figuras predefinidas:** El usuario puede elegir entre círculo, cuadrado, triángulo, cruz, corazón o estrella.
* **Modo replicación:** Utiliza la cámara de la ESP32-CAM para capturar la figura y enviarlo al robot para su reproducción.
* **Modo TaTeTi:** A pesar de que no se podrá jugar físicamente contra el robot como se había planeado, se incluyó el juego en la interfaz web de manera simbólica o como ampliación futura.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

## COMUNICACIÓN WEBSOCKET

El sistema utiliza WebSocket para establecer una conexión entre la página web y la ESP32-CAM. Cuando la ESP32-CAM inicia el servidor WebSocket, se conecta a una red WiFi existente utilizando datos de acceso predefinidos (SSID y contraseña) y comienza a escuchar en un puerto específico. La página web se conecta a este servidor y establece una comunicación bidireccional.

El servidor WebSocket en la ESP32-CAM maneja eventos de conexión, recepción de mensajes y desconexión. Cuando recibe un comando desde la interfaz web, lo transmite a la EDU-CIAA a través de UART.

El flujo de comunicación es el siguiente:

1. La interfaz web se conecta a la ESP32-CAM mediante WebSocket al cargar la página.
2. Cuando el usuario selecciona una figura o inicia el modo de replicación, la página envía un mensaje a la ESP32-CAM.
3. La ESP32-CAM reenvía la instrucción a la EDU-CIAA a través de UART.
4. La EDU-CIAA procesa el comando y acciona los motores para realizar el dibujo.
5. Una vez completado el dibujo, la EDU-CIAA notifica a la ESP32-CAM mediante UART
6. La ESP32-CAM notifica a la interfaz web mediante WebSocket para que habilite nuevamente la selección de figuras.

Además, la ESP32-CAM realiza el reconocimiento de imágenes para el modo de replicación de dibujos, procesando la imagen capturada por su cámara y enviando la información correspondiente a la EDU-CIAA.

## RECONOCIMIENTO DE IMÁGENES

El modelo de reconocimiento fue desarrollado utilizando la plataforma Edge Impulse, una herramienta que permite crear y entrenar modelos de machine learning optimizados para dispositivos embebidos como la ESP32-CAM.

Estos modelos se basan en una red neuronal diseñada para la detección de objetos, la cual fue entrenada con más de mil imágenes capturadas con el módulo ESP32-CAM en un área con buena iluminación generando un contraste de la figura en el papel. Estas imágenes se dividieron y clasificaron en partes iguales entre las cuatro figuras geométricas, con variaciones en tamaño y posición para mejorar la robustez del modelo. El conjunto de datos se dividió en un 80% para entrenamiento y un 20% para validación, una práctica común en aprendizaje automático para evaluar el desempeño del modelo en imágenes nuevas.

Para la detección de objetos se utilizó la técnica FOMO (Faster Objects, More Objects) basada en MobileNetV2 (0.35), una arquitectura diseñada para dispositivos con recursos limitados. Siguiendo parámetros recomendados para este tipo de problemas los adaptamos a nuestro modelo como resolución de imagen en escala de grises, número de ciclos de entrenamiento: 40, tasa de aprendizaje: 0.005 (controla qué tan rápido o lento el modelo ajusta sus parámetros durante el entrenamiento para minimizar los errores).

Además, se activó la opción de Data Augmentation, lo que permite aumentar la variabilidad de los datos mediante modificaciones aleatorias en las imágenes de entrenamiento. Esto ayuda a mejorar la capacidad del modelo para generalizar y reconocer figuras en distintas posiciones o condiciones. El modelo alcanzó un F1 score (métrica que mide la exactitud de un modelo de clasificación) de 99.4%, lo que indica un alto nivel de precisión en la detección de las figuras geométricas.

Durante el desarrollo del modelo, se identificó restricción en la cantidad de imágenes de entrenamiento impuesta por la versión gratuita de la plataforma Edge Impulse, la imposibilidad de cargar grandes volúmenes de datos afectó la capacidad del modelo para generalizar correctamente ante variaciones en las formas y tamaños de las figuras detectadas, incluso se redujo la cantidad de figuras que se habían marcado como reconocibles para nuestro sistema ante esta dificultad, permitiendo que mejore la precisión del algoritmo al tener menos clases distinguibles, el tiempo de entrenamiento también es limitado, lo que puede restringir la capacidad de ajustar y optimizar el modelo con más ciclos de entrenamiento o configuraciones más complejas. A pesar de las limitaciones de la versión gratuita, se logró desarrollar un modelo funcional con alta precisión, sin embargo, podría serlo aún más con las funcionalidades avanzadas de la versión paga de Edge Impulse.

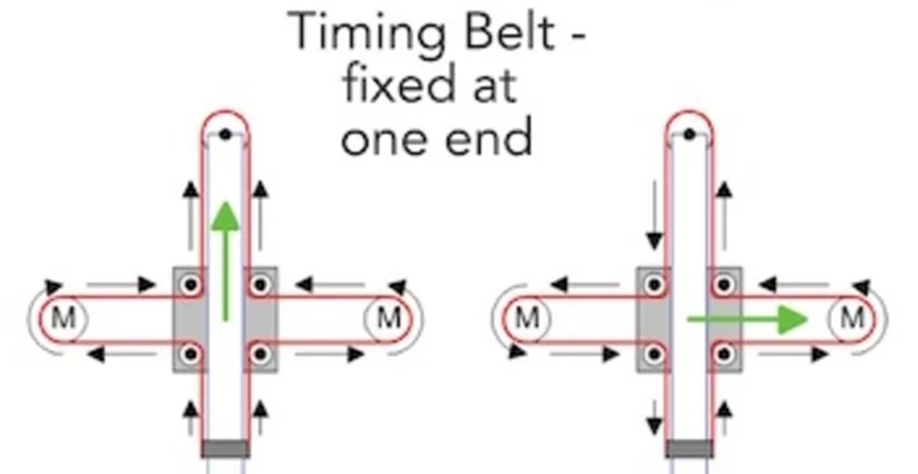
Una vez entrenado, el modelo fue convertido en una librería de Arduino, para integrarlo en el módulo ESP32-CAM. Como resultado de cada procesamiento de imagen se obtiene en formato de cuadros delimitadores, que indican la posición y la precisión de las figuras detectadas en la imagen, cuando se llama a la función reconocerFigura() retorna la figura reconocida sí reconoció alguna en la imagen con el fin de modularizar el código.

Para reducir el margen de error, por un lado se evaluó tomar únicamente las figuras que fueron reconocidas con una precisión mayor a 0.6 y arrojo un mejor resultado ya que por debajo de ese nivel de precisión generalmente no son ciertas, por otro lado, se consideró que la detección de figuras se base en un sistema de filtrado por iteraciones, en el cual se realizan una cantidad de inferencias consecutivas, donde se obtiene la figura más reconocida y su precisión promedio, ambas técnicas de mitigación de error disminuyen la posibilidad de falsos positivos y se notó una mejora en la fiabilidad de la detección. Finalizado el proceso de detección de figuras los resultados se transmiten a la EDU-CIAA que se encarga de procesarlos.

Otra dificultad que debió ser consideradas a la hora de desarrollar nuestro sistema de detección de figuras fue el impacto de la iluminación sobre la precisión del algoritmo, dado que variaciones en la intensidad y dirección de la luz teniendo en cuenta la estructura necesaria para sostener la ESP32-CAM generan sombras y contrastes los cuales pueden alterar la identificación correcta de la imagen, para mitigar este error se consideró la posibilidad de incluir algún sistema de iluminación o el propio led que tiene integrado el módulo ESP32-CAM.

## MOVIMIENTO DE MOTORES

El código del movimiento de los motores se desarrolló siguiendo las pautas de movimiento del sistema de motores elegido para este proyecto. Para mover la punta de la impresora en el eje vertical se deben mover ambos motores en sentidos opuestos. Y para moverla en el sentido horizontal se deben mover ambos motores en el mismo sentido.



Para esto se desarrollaron varias funciones con el objetivo de simplificar estos movimientos, ya sea para mover un solo paso en una de las direcciones o varios en simultaneo. A su vez para mover la punta en distintos ángulos se utilizaron funciones trigonométricas para definir cuantos pasos debían realizarse en cada eje para distintos pasos para lograr un movimiento en el ángulo deseado. Para un ángulo de 45 grados se creó otra función que mueve uno solo de los motores. Esto logra que, al dejar fijo uno de los extremos de la correa, la punta se mueva en diagonal.

Luego de ya tener las funciones básicas de movimiento se prosiguió con la creación de distintas funciones para el dibujo de figuras básicas: Un cuadrado, rectángulo, un triángulo, una cruz y una estrella. Al finalizar con estas figuras se siguió con la creación de una función capaz de crear un círculo. Este fue un desafío al tener que trabajar con varios ángulos diferentes para las distintas partes del círculo. Otro error con el que nos encontrábamos era que al hacer tantos pasos se lograba un círculo mucho más grande que el resto de las figuras, y al intentar escalarlo se perdía la definición de este y pasaba a dibujarse un octógono en lugar de un círculo. Luego de solucionar estos problemas se logró hacer un círculo de un tamaño acorde al resto de figuras con una definición adecuada.

Para probar las distintas figuras creadas se creó una función con el objetivo de dibujar una figura al presionar uno de los botones ubicados en la EDU-CIAA.

Para lograr mover la punta del lápiz por la hoja se creó un sistema de coordenadas relativas al punto inicial del lápiz, y los movimientos que requieran ubicarse en cierta posición del tablero utilizan ese punto como referencia. Utilizando estas coordenadas y la función creada para desplazarse entre ellas se logró dibujar el tablero para el juego de TA-TE-TI y luego de varias pruebas se logró que el lápiz se ubique en posición para poder dibujar las figuras que sean necesarias para el correcto flujo del juego.

## CONEXIÓN Y COMUNICACIÓN UART

Para la conexión de la CIAA con la ESP se creó una función capaz de recibir datos mediante el protocolo UART de comunicación. En un principio se había planeado enviar un único carácter para la decodificación de la información, pero en una de las pruebas realizadas se conectó primero la CIAA en lugar de la ESP y esto produjo que por el canal de UART se recibieran datos sin sentido, entre ellos se descifro un carácter valido y se comenzó a dibujar sin haberse enviado ningún carácter desde la aplicación. Por este motivo, luego de consultar con el ayudante, se decidió implementar un formato para la información en la comunicación:

Antes de la información se deben recibir los caracteres ‘.’ y ’\*’, en ese orden. Luego se proceden a leer todos los caracteres enviados hasta encontrar los caracteres ‘\*’ y ‘.’, en ese orden. Con este sistema se decodifican los caracteres recibidos para dibujar la figura correspondiente. El lápiz se mueve hacia adelante y luego dibuja la figura indicada por el siguiente parámetro:

1. C: Cuadrado
2. O: Circulo
3. X: Cruz
4. T: Triangulo
5. S: Estrella
6. H: Corazón
7. Si se recibe un carácter no definido en esta lista se procede a parpadear las luces rojas ubicadas en la EDU\_CIAA como modo de advertencia.

## CÓDIGO

Todo el código está disponible en <https://github.com/joacochanquia/TdP-Grupo2-2024>, junto con sus librerías necesarias. Tanto de la interfaz web, como de la ESP32 y de la EDU-CIAA. Además, contiene fotos y videos del proceso del proyecto.

# VALIDACIÓN

Con los siguientes enlaces se puede acceder a los videos que demuestran la correcta funcionalidad del sistema, en ambos modos.

* [Demostración del modo de uso en el que en la aplicación se elige la figura que se desea que el robot dibuje.](https://drive.google.com/drive/folders/1U7fdQDwiTWnAouzxbulKEgNdV_WfH2Z-?usp=sharing)
* [Demostración del modo en el que la cámara reconoce la figura dibujada y traza una similar.](https://drive.google.com/drive/folders/1QG9Qkif7tV6zo3cGPZ-MGpje1YTTLAfl?usp=sharing)

# CONCLUSIÓN

## CUMPLIMIENTO DE OBJETIVOS

Se considera que los **objetivos primarios** han sido cumplidos satisfactoriamente, ya que se logró diseñar y ensamblar una estructura funcional, implementar un sistema de dibujo preciso con figuras predeterminadas y desarrollar un sistema de reconocimiento de imágenes capaz de diferenciar entre distintas figuras. Además, se implementó una interfaz que permite al usuario seleccionar los diferentes modos de uso, incluyendo la replicación de dibujos previamente reconocidos.

Por otro lado, los **objetivos secundarios** no pudieron ser alcanzados dentro del tiempo disponible. Sin embargo, el progreso realizado sienta las bases para su futura implementación. Actualmente, el sistema ya es capaz de distinguir entre un círculo y una cruz utilizando la ESP32-CAM, lo que representa un avance significativo hacia la programación del juego de TA-TE-TI. La principal tarea pendiente es la localización exacta de las figuras dentro del tablero, lo que permitiría que el robot detecte las jugadas y tome decisiones estratégicas.

## EVALUACIÓN DE REQUERIMIENTOS

## FUNCIONALES

### HARDWARE

* *El sistema debe incorporar una cámara con resolución suficiente para detectar de manera precisa las figuras realizadas por el usuario.*

Para cumplir con este requerimiento, se utilizó la ESP32-CAM, un módulo con una cámara integrada capaz de capturar imágenes con una resolución adecuada para el procesamiento de visión por computadora, al cual se le integró un algoritmo de red neuronal entrenado para detectar las figuras de la imagen.

* *Los motores paso a paso deben ser capaces de posicionar el marcador con precisión en los diferentes sectores de la hoja.*

Para cumplir con este requerimiento, se utilizaron dos motores paso a paso NEMA 17 controlados por DRV8825, los cuales permiten un desplazamiento preciso y estable del marcador. En conjunto con el sistema de rieles y correas, se logra un movimiento controlado y repetible en toda la superficie de dibujo. Además, la EDU-CIAA NXP gestiona el posicionamiento de los motores, asegurando que el marcador alcance con precisión cada sector de la hoja según las instrucciones enviadas.

* *El sistema debe contar con rieles que permitan el movimiento lineal de la estructura.*

Tomando como referencia un modelo desde internet, logramos crear una estructura capaz de soportar su propio peso y de moverse libremente en todas direcciones en un sistema de rieles y rodamientos lineales, logrando desplazamiento sin vibraciones del mecanismo.

* *La máquina debe tener un tamaño y peso adecuado para facilitar su transporte.* El robot puede ser transportado ligeramente en una caja, la estructura impresa en 3D y los rieles otorgan firmeza en el equipo para su traslado, al igual que la estructura metálica para la ESP32-CAM.
* *Se debe integrar una pantalla LCD para comunicar los modos de uso.*
* *Deben incluirse botones físicos que permitan una configuración sencilla de la máquina.*

Inicialmente, se había previsto la inclusión de una pantalla LCD para mostrar los modos de uso y botones físicos en la EDU-CIAA para la selección de los modos de uso y funciones. Sin embargo, a lo largo del desarrollo, se identificó la posibilidad de implementar una interfaz web como medio principal de interacción con el usuario, lo que resultó en una solución más moderna, practica, escalable y accesible que reemplazó por completo estos elementos y simplificó el hardware.

### SOFTWARE

* *El software de reconocimiento de imágenes debe analizar las ilustraciones capturadas por la cámara y detectar las figuras (preestablecidas) hechas por el usuario.*

Se desarrolló un modelo de aprendizaje automático para detección de figuras el cual captura una imagen y procesa la misma para detectar alguna figura, indicando que figura reconoció, precisión del reconocimiento y coordenadas de su posición.

* *El software debe enviar comandos precisos a los motores paso a paso para mover el marcador en la hoja, siguiendo las decisiones del algoritmo de dibujo*. Para cumplir con este requerimiento, se desarrolló un sistema de control en la EDU-CIAA NXP, el cual recibe instrucciones de movimiento desde la ESP32-CAM mediante comunicación UART. El software interpreta estas instrucciones y genera los pulsos necesarios para controlar los motores paso a paso NEMA 17 a través de los controladores DRV8825, asegurando desplazamientos precisos y coordinados. Además, el algoritmo implementado garantiza que los movimientos del marcador sigan fielmente el diseño deseado, permitiendo la correcta ejecución de los dibujos.
* *El sistema será probado y validado para garantizar su funcionamiento óptimo y el cumplimiento de los requerimientos establecidos.*

Se realizaron pruebas y ensayos con el sistema completo en acción y no se observaron alteraciones en el funcionamiento de este, se mantiene el control en todo momento del sistema y permite una experiencia de usuario acorde.

## NO FUNCIONALES

* *El proyecto debe implementarse utilizando la placa EDU-CIAA.*

Para cumplir con este requerimiento, se utilizó la EDU-CIAA NXP como unidad principal de control, encargada de gestionar el movimiento de los motores paso a paso y el servomotor. La placa permitió implementar el control de movimiento con precisión y estableció una comunicación eficiente con la ESP32-CAM mediante UART. Su capacidad de procesamiento y sus puertos de entrada/salida fueron adecuados para las necesidades del sistema, asegurando un desempeño estable y confiable.

* *El proyecto debe estar finalizado, junto con su documentación correspondiente, antes del final de la cursada de "Taller de Proyecto 1" del 2do semestre de 2024.*

El desarrollo del sistema, junto con los informes y documentación, se completó dentro del plazo establecido para la materia, permitiendo su validación y presentación sin inconvenientes.

* *El desarrollo del proyecto debe ajustarse a un presupuesto preestablecido*.

El proyecto se mantuvo dentro del presupuesto estimado, ya que se optó por el uso de componentes accesibles sin comprometer el rendimiento del sistema, además gran parte de los materiales y componentes electrónicos fueron proporcionados por la cátedra, lo que permitió reducir aún más los costos

* *La máquina debe diseñarse de manera que garantice la seguridad del usuario durante su uso*.

Para cumplir con este requerimiento, se diseñó un sistema que minimiza la necesidad de intervención directa del usuario durante su funcionamiento. La interfaz web permite el control remoto del robot, evitando el contacto con partes móviles mientras está en operación. Además, el sistema funciona con tensiones de 12V y 5V, valores seguros para la electrónica utilizada y sin riesgo significativo para el usuario. La estructura impresa en 3D y los rieles lineales garantizan estabilidad, evitando movimientos bruscos o descontrolados que puedan representar un peligro.

* *La interfaz del sistema debe ser fácil de usar, con instrucciones claras, y adecuada para personas de diferentes edades y niveles de habilidad.*

La implementación de una interfaz web permitió una experiencia de usuario más intuitiva y accesible. La interfaz es clara, con opciones simples para seleccionar figuras y controlar el sistema, lo que facilita su uso sin necesidad de conocimientos técnicos avanzados.

## ESCALABILIDAD

**- Proyecto adaptable a G-CODE**(lenguaje de programación que sigue una estructura específica para comunicar instrucciones a máquinas CNC): se podría implementar un intérprete de G-code de algunas instrucciones, el cual recibirá los comandos, los analizará y ejecutará las acciones correspondientes o también se podría conectar el dispositivo ESP32 a una computadora mediante UART, utilizar un programa que permita trabajar con G-code como Universal G-code Sender, seleccionar el puerto serial correspondiente, configurar la velocidad de comunicación y finalmente enviar comandos G-code al dispositivo.

- La **escalabilidad del reconocimiento de imágenes** en el proyecto ofrece múltiples posibilidades de expansión. Actualmente, el sistema está diseñado para identificar un conjunto limitado de figuras geométricas, como círculos, triángulos y cuadrados, pero con un entrenamiento más amplio del modelo, podría extenderse a reconocer letras, números o incluso símbolos más complejos. Esto permitiría no solo mejorar la funcionalidad actual, sino también habilitar nuevas aplicaciones, como la escritura automatizada o el reconocimiento de caracteres en distintos contextos. Además, el mismo modelo ya es funcional para el desarrollo del juego del Ta-Te-Ti, procesando las cruces y círculos en el tablero para interpretar jugadas y permitir una interacción más dinámica con el usuario.

**- Implementación de microstepping para mayor precisión:** Actualmente, el sistema no utiliza microstepping en los motores paso a paso, lo que limita la resolución del movimiento. La incorporación de microstepping permitiría trazos más suaves y precisos, lo que mejoraría la calidad de los dibujos, permitiendo realizar detalles más finos y líneas más delicadas.

**- Ampliación del área de dibujo:** Modificando la estructura mecánica y el alcance de los rieles, el sistema podría escalarse para trabajar en superficies más grandes, lo que lo haría útil para aplicaciones como la cartelería automatizada o el trazado de circuitos en placas de gran tamaño.

**- Uso de diferentes herramientas de dibujo y corte:** En lugar de un solo marcador, se podría desarrollar un sistema intercambiable de herramientas, como distintos tipos de lápices, pinceles o incluso un módulo de grabado láser. Además, el sistema podría adaptarse para portar una cuchilla tipo cúter, permitiendo cortar materiales como papel, cartón o vinilo, lo que ampliaría sus aplicaciones en diseño gráfico, fabricación de plantillas y modelado.

# BIBLIOGRAFÍA

* [CIAA](https://www.proyecto-ciaa.com.ar/index.html)
* [Estructura robot dibujante](https://www.thingiverse.com/thing:2349232)
* [Enfoque FOMO](edgeimpulse/tutorials)
* [Modelo Edge Impulse](https://studio.edgeimpulse.com/public/609742/latest)

**Especificaciones Técnicas**

* [ESP32](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
* [DRV8825](https://www.ti.com/lit/ds/symlink/drv8825.pdf)
* [Motor Nema 17](https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf)
* [Servo SG90](https://www.friendlywire.com/projects/ne555-servo-safe/SG90-datasheet.pdf)

**Herramientas utilizadas**

* [KiCad](https://www.kicad.org/)
* [3D Builder](https://apps.microsoft.com/detail/9wzdncrfj3t6?hl=en-us&gl=US)
* [Arduino IDE 2.3.4](https://www.arduino.cc/en/software)
* [Edge Impulse](https://edgeimpulse.com/)
* [Cursor](https://www.cursor.com/)

# ANEXO

## DIVISIÓN DE TAREAS

Con respecto a las tareas relacionadas con Hardware, fueron hechas mayoritariamente en conjunto, como la fabricación de la PCB, la adquisición de los componentes, el armado del robot, etc. Exceptuando por el diseño de la PCB, que fue efectuado por Joaquín Chanquía.

En relación con el Software, a pesar de que ningún integrante debió realizar ninguna tarea de forma completamente individual, sino que, siempre contó con la colaboración de los demás integrantes, sugerencias, ayuda ante problemas, etcétera; al tener tres bloques importantes que desarrollar, cada integrante tenia a cargo un bloque específico:

- Joaquín Chanquía se encargó de implementar el control de movimiento del robot, asegurando el funcionamiento adecuado de los motores. Desarrolló las funciones necesarias para dibujar correctamente cada figura específica, considerando el posicionamiento inicial y final del robot.

- Gabriel Ollier desarrolló el software de reconocimiento de figuras utilizando la cámara de la ESP32. A través de Edge Impulse, implementó un modelo capaz de distinguir con alta precisión entre cuatro figuras geométricas.

- Melina Caciani Toniolo fue responsable del diseño y desarrollo de la interfaz web, así como de la implementación de las comunicaciones, incluyendo la conexión entre la interfaz web y la ESP32 a través de WebSocket, y la comunicación entre la ESP32 y la EDU-CIAA mediante UART.

## CALENDARIO DE ACTIVIDADES

