Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Maestría en Ingeniería Electrónica,
opción Instrumentación Electrónica

Tesis para obtener el grado de:
Maestro en Ingeniería Electrónica

SISTEMA DE DEPÓSITO DE MATERIALES PARA CONTACTOS METÁLICOS DE CELDAS FOTOVOLTAICAS

Presenta:
Ing. Guillermo Colorado Jiménez*

Asesores:
Dr. Gerardo Mino Aguilar
Dr. César Augusto Arriaga Arriaga
Dr. Luis Armando Moreno Coria

*Becario CONAHCYT

Puebla, Pue. Enero 2025

Agradecimientos

Quiero darle las gracias a Dios por darme la oportunidad de estudiar un posgrado, a mi abuelita Mari quien se encuentra en el cielo y a todas las personas que se han puesto en mi camino porque me han dejado una enseñanza o me han motivado a dar mi mayor esfuerzo en el cumplimiento de mis sueños; como lo es el desarrollo de esta tesis.

Doy gracias a la Mtra. Ma. Lydia Jiménez Morales quien es mi madre y ha sido mi ejemplo de superación personal desde mi infancia; así mismo a mis hermanas Guadalupe y Gabriela por acompañarme en mis momentos tristes y de alegría. A mis abuelos José, Guillermo y Sofía quienes han estado en todo momento y me han enseñado que la juventud nunca envejece, ya que nunca es tarde para aprender algo nuevo.

Agradezco a mis asesores Dr. Gerardo Mino Aguilar, Dr. Cesar Augusto Arriaga Arriaga y al Dr. Luis Armando Moreno Coria por ayudarme en el desarrollo de este proyecto, darme la motivación y la confianza de trabajar con ellos. Al Dr. Julio Villanueva Cab y Dr. Eulises Regalado Pérez investigadores del Instituto de Física de la Benemérita Universidad Autónoma de Puebla, quienes me apoyaron en realizar mi estancia, brindándome el conocimiento y los equipos científicos para realizar las pruebas de depósito.

A mis compañeros Mauricio, Fernando y Mauro porque nos decidimos a iniciar está maestría; buscando superarnos, tanto profesionalmente como personalmente, trabajando codo a codo, motivándonos cuando no veíamos la salida a las dificultades que se presentaban, siempre nos tendíamos la mano. También darle las gracias a Enrique, Bruno y compañeros de la maestría por la compañía, esto nos unió a la investigación; a Yair Romero, compañero de la generación pasada de la maestría quien siempre me aconsejó e impulsó en cada etapa de este proyecto.

Darle gracias a Julio, Diego, Jesús, Nevai y Antonio tesistas del Dr. Luis Armando; y a su vez a los chicos del laboratorio del prototipado rápido Leo, Carlos y Osvaldo, por todo el apoyo que me brindaron en el diseñando y armando del sistema, al contagiarme con su ánimo y por darme el acompañamiento en los largos días de trabajo para poder alcanzar esta meta.

A mis mejores amigos, Karla, Alexis y Héctor, quienes han sido una parte fundamental en mi vida. Gracias por compartir conmigo tantos momentos, por estar a mi lado en los momentos buenos y malos, y por la increíble amistad que hemos construido a lo largo del tiempo. Su apoyo incondicional en los momentos más críticos ha significado todo para mí, y valoro profundamente el lazo especial que nos une.

También quiero agradecer a la Mtra. Ana María Rodríguez Domínguez Coordinadora de la Maestría en Ingeniería Electrónica, opción Instrumentación Electrónica y a todos sus profesores, quienes conforman a esta maestría, por transmitir su conocimiento sobre el desarrollo científico y tecnológico de mi tesis. A la Benemérita Universidad Autónoma de Puebla por ser mi segundo hogar y mi alma mater en toda mi formación académica, y a la Facultad de Ciencia de la Electrónica por prestarme sus instalaciones para el desarrollo de este trabajo.

Por último, reconozco el apoyo del Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el valioso ingreso económico brindado, el cual ha sido fundamental para la realización de mis estudios de Maestría y la conclusión de esta tesis.

Resumen

En este proyecto se desarrolló un sistema para depositar material conductivo sobre los patrones geométricos de los contactos de la celda fotovoltaica, utilizando procesamiento de imagen para asegurar su colocación en los patrones. Esta tarea es esencial para garantizar el adecuado desempeño en la generación de energía eléctrica de las celdas, ya que un depósito incorrecto puede alterar las propiedades de la celda.

El trabajo se realizó con una celda fotovoltaica monocristalina ya fabricada, que no contaba con contactos superior e inferior. La celda tiene dos contactos para extraer la corriente eléctrica generada. La zona N presenta un patrón de contactos finos que cubre entre el 5 % y el 10 % del área, correspondiente a la terminal negativa, mientras que la zona P está completamente recubierta con una capa metálica, representando la terminal positiva.

El sistema de depósito fue diseñado utilizando diversos materiales, cuyas características y aplicaciones se detallan en este documento. Además, se desarrolló una tarjeta de circuito electrónico para conectar los arneses de potencia y control del sistema, pasando por varios prototipos hasta obtener una placa que cumpliera con todos los requisitos eléctricos y electrónicos. Se creó también una interfaz gráfica que integró el control del sistema de depósito y el procesamiento de imágenes, utilizando las tarjetas de desarrollo Raspberry Pi 5 y Esp32. Esta solución mejoró la operación del sistema para el usuario.

Como resultado, se realizó la caracterización eléctrica de la celda antes y después de aplicar tinta conductiva de grafito y una pasta de carbono disuelta en Xilol. Aunque este no fue el objetivo principal de la tesis, los resultados obtenidos aportan información valiosa y sientan las bases para trabajos futuros centrados en mejorar la eficiencia de las celdas fotovoltaicas.

Abstract

In this project a system was developed to deposit conductive material on the geometric patterns of the contacts of the photovoltaic cell, using image processing to ensure their placement in the patterns. This task is essential to ensure proper performance in the generation of electrical energy from cells, since an incorrect deposit can alter cell properties. The work was carried with a monocrystalline photovoltaic cell, which had no upper and lower contacts. The cell has two contacts to draw the generated electrical current. Zone N has a fine contact pattern covering between 5 % and 10 % of the area corresponding to the negative terminal, while zone P is completely covered with a metallic layer representing the positive terminal.

The deposit system was designed using various materials, whose characteristics and applications are detailed in this document. In addition, an electronic circuit board was developed to connect and control the system. They went through some prototypes to obtain a board that met all the electrical and electronic requirements. A graphical interface was created that integrated the deposit system control and image processing, it is using the Raspberry Pi 5 and Esp32 development cards. This solution improved the system operation for the user.

As a result, the electrical characterization of the cell was performed before and after applying conductive graphite ink and a carbon paste dissolved in Xilol. Although this was not the main objective of the thesis, the results obtained provide valuable information and lay the foundation for future work focused on improving the efficiency of photovoltaic cells.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La demanda de energía eléctrica en el mundo ha aumentado considerablemente debido al crecimiento poblacional y a la necesidad de satisfacer las demandas actuales de la sociedad. Esto ha llevado al agotamiento de los recursos naturales, como los combustibles fósiles, provocando impactos en el cambio climático global. Esto ha motivado a buscar nuevas formas de obtener la energía de una manera más limpia, por lo cual el uso de energías renovables y sostenibles, como lo es la energía solar, es una alternativa para solucionar estas necesidades, ya que es limpia, silenciosa y confiable. [9].

En México el consumo de energía per cápita en 2023 fue de 77.92 GJ, lo que representó una disminución del 3.49 % respecto al año previo, como lo muestra en la Fig. 1.1. Además, la población mexicana creció 0.9 % entre 2022 y 2023, pasando de 129.96 a 131.14 millones de habitantes. [1].

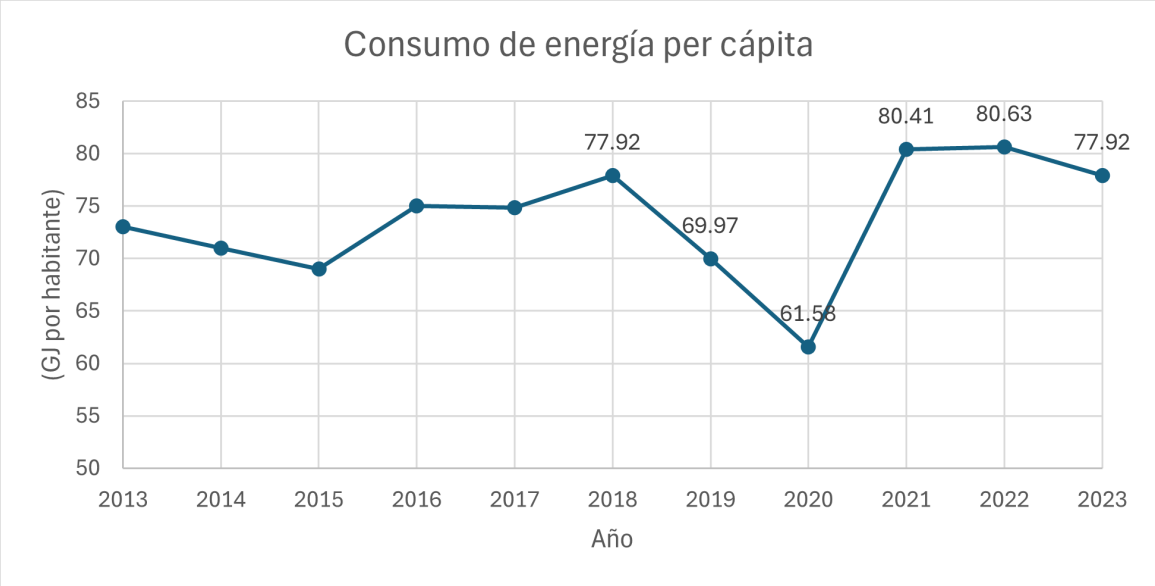


Fig 1.1: Elaboración propia con información de CONAPO e INEGI [1].

México obtiene más del 80 % de su suministro energético total de los combustibles fósiles. En 2023 el petróleo crudo contribuyó un 52,65 %, seguido del gas natural con el 21.48 %, el carbón con el 1,76 %, la nuclear 2.39 % y las energías renovables con el 13.97 %, en donde se engloba la geoenergía, la energía eólica y solar 6.53 %, la hidroenergía 2.59 % y la biomasa 4.63 % [2]. Sin embargo, el suministro energético proveniente de la energía solar aún no es suficiente para generar un cambio considerable, ya que, según el Balance Nacional de Energía 2023 se tuvo una participación de las energías limpias de 24.32 % [1].

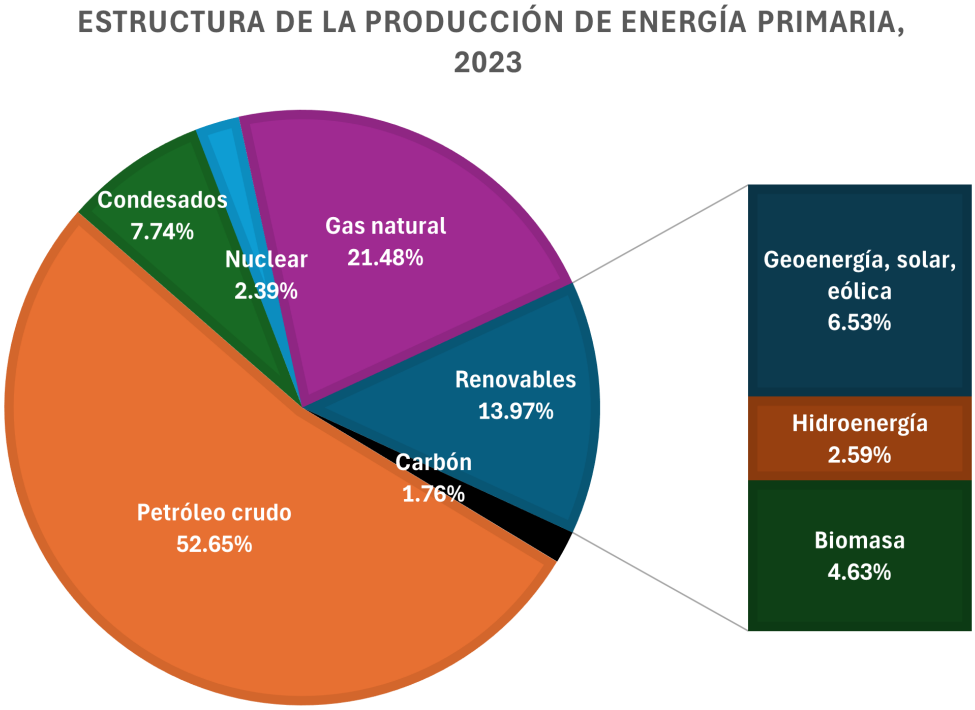
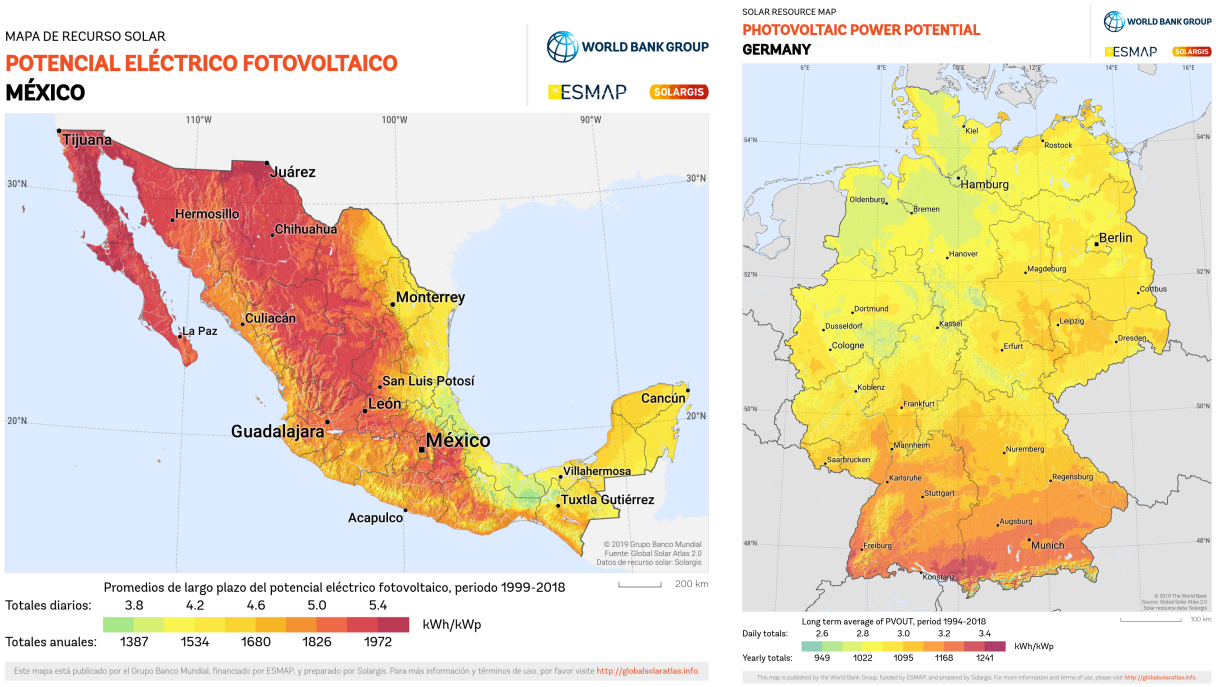


Fig 1.2: Gráfica de los resultados obtenidos en el Sistema de Información Energética (SIE) [2].

El uso de energías renovables es una oportunidad de resolver problemas ambientales y garantizar el abastecimiento energético del país. Hablando específicamente de la energía solar, que se puede obtener energía durante todo el día, pues la cantidad de energía solar promedio que se recibe es de 5 kWh por cada metro cuadrado al día, por lo que se estima que el potencial solar bruto representa alrededor de 50 veces el total de energía eléctrica generada en el país [10]. En comparación con otros países que son potencia en general energía solar, como lo son Alemania y China, que cuentan con toda la infraestructura para poder aprovechar las energías limpias, no poseen una ubicación geográfica tan favorable como la de México para recibir altos niveles de radiación solar.

La Fig. 1.7 extraída del Solargis [11] muestra que el lugar con mayor potencial eléctrico fotovoltaico de Alemania registra aproximadamente 3.4 kWh por día. En contraste, el potencial eléctrico fotovoltaico más bajo en México se registra alrededor de 3.8 kWh por



(a) México (b) Alemania

Fig 1.3: Mapas del potencial eléctrico fotovoltaico.

día. Esto indica que, en cualquier parte del país, México cuenta con condiciones más óptimas que Alemania para generar energía limpia a través del sol.

1.1. Justificación

El Programa Nacional Estratégico de Energía y Cambio Climático busca promover una reflexión amplia y acciones específicas para alcanzar un sistema energético sostenible y equitativo. Para ello, es necesario adoptar un enfoque interdisciplinario e integral de la problemática energética que se enfrenta el país [12], dado que el aumento de población va directamente proporcional al incremento en el consumo de energía. Esto provoca que se ocupen más recursos de nuestro planeta, de modo que se requiere de métodos de generación de energía que sean limpios e inagotables; por lo tanto, la fabricación de celdas solares es una posibilidad para poder dar una solución a esta problemática; sin embargo, la tecnología fotovoltaica representa un verdadero reto para la ciencia.

En la actualidad, existen diversos mecanismos que afectan la eficiencia en la conversión de energía radiante a eléctrica. Uno de los principales retos está relacionado con la reflexión de la luz sobre la superficie. Para resolver este problema, se han desarrollado procesos de texturización que modifican la superficie del silicio, reduciendo la reflexión superficial a valores inferiores al 10 % [13].

Otro reto por resolver es sobre la fotolitografía que es la técnica que se utiliza para colocar contactos, sin embargo, esta técnica se basa en la reflexión de la luz sobre una superficie y puesto que se tiene una superficie texturizada, la luz que se refleja no es suficiente para poder alinear la mascarilla y entonces, colocar contactos se vuelve una tarea compleja, que a su vez requiere de una numerosa cantidad de recursos para poder lograr su objetivo que es unir las celdas solares.

1.2. Antecedentes

El primer paso que se dio en el desarrollo de la celda fotovoltaica fue el descubrimiento del efecto fotovoltaico, que se le atribuye al científico francés Edmond Becquerel en el año 1839. El observó el proceso que ocurre cuando la luz es absorbida por un material, lo que permite generar un voltaje.
Posteriormente, se descubrió la fotoconductividad del selenio, lo que significa que el material es conductor cuando absorbe la luz. Este descubrimiento fue realizado por el ingeniero inglés Willoughby Smith. Al paso de unos años una publicación de William G. Adams y Richard E. Day, consideraron los efectos de la luz solar sobre el selenio sin calor ni partes móviles, lo que demostró que la energía solar podía ser aprovechada de manera accesible. No fue hasta 1883 que Charles Edgar Fritts desarrolló el primer prototipo de una celda solar, utilizando una delgada capa de oro. Sin embargo, este prototipo resultó ineficiente, con una eficiencia de apenas $1 - 2\,\%$ [14].

Aún quedaba mucho por descubrir, al paso del tiempo se desarrolló la teoría del Rectificador de Estado Sólido que dio origen al diodo, llevado a cabo por Mott y Schottky a principios de la década de 1940 y el invento del transistor por Bardeen, Brattain y Shockley en el año 1949, lo que ocasionó un cambio al establecer la primera celda solar de silicio cristalino, desarrollada por los Laboratorios Bell en 1954.

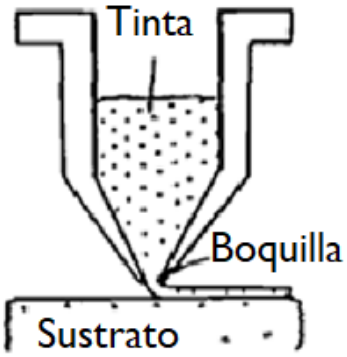


Fig 1.4: Escritura directa en película gruesa [3].

Este avance atrajo la atención sobre el potencial de las celdas solares, en 1958 fue el protagonista del Vanguard I, el primer satélite en utilizar celdas solares para generar su propia energía eléctrica. Desafortunadamente, desarrollar las celdas solares era de alto costo, lo que ocasionaba que solo fueran objeto de estudios en laboratorios o en proyectos espaciales [10].

1.2.1. Trabajos relacionados

En la literatura hablan del avance tecnológico de las celdas fotovoltaicas, desde su aparición por primera vez en los satélites como fuente de energía, hasta los importantes aportes que han realizado con el paso del tiempo.

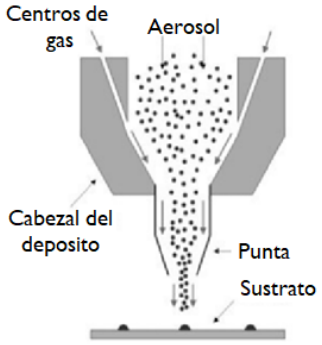


Fig 1.5: Impresora de aerosol de metal [4].

La escritura directa en película gruesa [3], describe un método innovador para colocar el material conductor sobre los contactos. Este sistema implica la deposición de tintas en una película gruesa, mediante la acción capilar de un lápiz de un metal precioso, que también funciona como depósito de tinta, en la Fig. 1.4 muestra un esquema del sistema.

La impresión de jeringa [15] no es una tecnología comúnmente utilizada para el depósito de líneas en la industria fotovoltaica. Esta se usa para depositar las barras colectoras sobre la superficie de silicio texturizado de forma mecánica, lo cual representa un gran desafío, ya que las partículas de pasta deben presionarse a través de la fina punta de la jeringa sin obstruirse. Además, las líneas de contacto deben depositarse sin interrupción, lo que requiere control preciso. Una ventaja de esta técnica es la posibilidad de depositar las líneas de contacto sobre superficies muy rugosas siempre y cuando se controle adecuadamente la distancia entre la punta de la jeringa y el

La impresión por chorro de aerosol metálico [4] es una técnica de escritura directa sin contacto, utilizada para la metalización frontal de celdas solares de silicio de alta eficiencia. Con esta técnica se crea una estructura de contacto de dos capas con baja resistencia y una buena adherencia mecánica a la superficie de silicona, como se muestra en la Fig. 1.5.

La impresión por chorro de tinta [5] es una técnica de impresión adecuada en el caso de sustratos flexibles, se basa en la formación de una pequeña gota de pasta que se deposita sobre el sustrato. Es posible realizar una impresión de dos formas posibles para generar las gotas: mediante el uso de un mecanismo piezoeléctrico que produce una compresión mecánica a través de una boquilla o por calentamiento de la pasta para aumentar el volumen y la presión del chorro de tinta, el ejemplo de sistema se muestra en la Fig. 1.6.

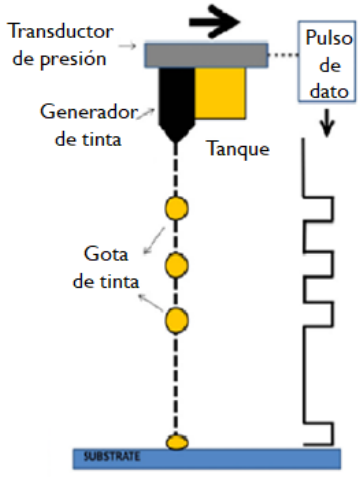


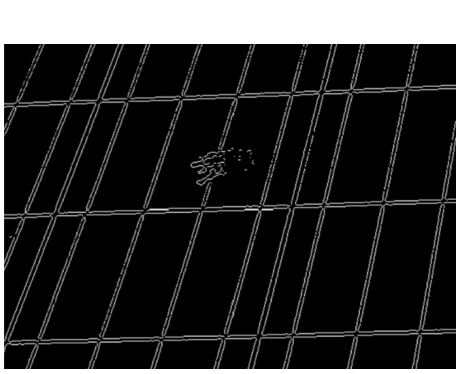
Fig 1.6: Impresión por chorro de tinta [5].

Tabla 1.1: Tabla de técnicas de depósito de contacto.

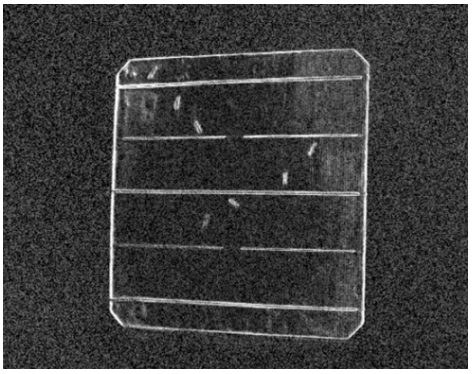
Técnica	**Característica**	**Ventaja**	**Desventaja**
Depósito físico de vapor (PVD) [16].	Se fabrica una plantilla o máscara de cromo o titanio sobre la superficie de silicio mediante un proceso de deposición de vapores.	Ancho de canal reducido.	Costo elevado debido a las condiciones de operación.
Deposito Químico de vapor (CVD) [17], [18].	Depositar una capa delgada de oro mediante deposición química.	Ancho de canal reducido.	Costo elevado debido a las condiciones de operación.
Envolventes [19].	Las celdas tienen contactos positivos y negativos ubicados en la parte posterior del dispositivo. Existen dos tipos de celdas envolventes: la unión envolvente y el aislante envolvente.	Ancho de canal reducido	Costo elevado debido a las condiciones de operación.
Escritura directa en película gruesa [3].	Este es un sistema de escritura que consiste en depositar tintas sobre una película gruesa mediante la acción capilar de un lápiz hecho de metal precioso.	Bajo costo de materiales e implementación.	Tamaño del canal.
Impresora de aerosol de metal [4].	Este es un sistema de escritura que consiste en depositar tintas sobre una película gruesa mediante la acción capilar de un lápiz hecho de metal precioso.	Bajo costo de materiales e implementación.	Tamaño del canal.
Impresión por chorro de tinta [5].	Técnica de impresión adecuada para sustratos flexibles, que se basa en la formación de una pequeña gota de pasta depositada sobre el sustrato.	Bajo costo de materiales e implementación.	Tamaño del canal.
La impresión de jeringa [15].	Técnica que deposita las barras colectoras sobre la superficie de silicio texturizado de forma mecánica, lo cual representa un gran desafío, ya que las partículas.	Bajo costo de materiales e implementación.	Tamaño del canal.

Tabla 1.2: Tabla de trabajos relacionados con el procesamiento de imágenes en celdas fotovoltaicas.

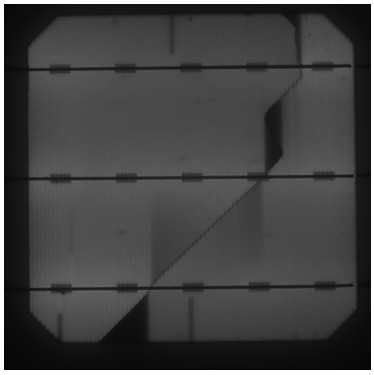
Trabajos	**Característica**
Investigación sobre un método de detección de suciedad en la superficie de las celdas fotovoltaicas basado en tecnología de procesamiento de imágenes [20].	Este trabajo propone un método para detectar el polvo en la superficie de las celdas fotovoltaicas, utilizando tecnología de procesamiento de imágenes para identificar riesgos ocultos y mejorar la eficiencia en la generación de energía.
Detección de objetivos pequeños en imágenes de polarización de defectos de celda fotovoltaicas basadas en YOLOv7 [21].	Este artículo propone un método para la detección de objetivos pequeños en imágenes de polarización de defectos en celdas fotovoltaicas, basado en YOLOv7, con el propósito de abordar el problema de la baja precisión en la detección.
Procesamiento automático y detección de células solares en imágenes de electroluminiscencia fotovoltaica [22].	Este trabajo utiliza imágenes de electroluminiscencia (EL), una técnica potente y ampliamente utilizada para evaluar la calidad de los módulos fotovoltaicos (PV), los cuales están compuestos por múltiples celdas solares conectadas eléctricamente y dispuestas en una red. El análisis de imágenes imperfectas del mundo real requiere métodos confiables de preprocesamiento. Este artículo aborda cómo mejorar dichas imágenes.



(a) Detección de suciedad en la superficie de las celda [20].



(b) Detección de objetivos pequeños en imágenes de polarización de defectos [21]



(c) Detección de celda solares en imágenes de electroluminiscencia fotovoltaica [22]

Fig 1.7: Tabajos relacionados con el procesamiento de imágenes en celdas fotovoltaicas.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar un prototipo para el depósito del material conductivo usado como contacto eléctrico en las celdas fotovoltaicas, mediante la implementación de un sistema automático de posición.

1.3.2. Objetivo Específicos

1. Desarrollar un sistema mecánico que permita el movimiento en diferentes planos para el depósito de un material conductor.
2. Implementar un sistema que permita identificar el patrón geométrico de una celda fotovoltaica ya fabricada.
3. Integrar el sistema de procesamiento de imágenes con el sistema mecánico y de depósito.
4. Realizar la caracterización eléctrica de la celda fotovoltaica con el material depositado.

1.3.3. Bloques funcionales del sistema

En la Fig.1.9 se presentan las etapas que integran la elaboración del sistema, comenzando con el diseño y la construcción del sistema mecánico. Esta etapa incluye la selección de los componentes que conforman las partes mecánica y electrónica del proyecto. Posteriormente, se diseño de cada componente para el ensamblaje del sistema, con el apoyo del software SolidWorks®, que permite visualizar el prototipo antes de su construcción y obtener una perspectiva de sus dimensiones.

En la fase de construcción, el sistema se ensamblo siguiendo el diseño previamente elaborado. En esta etapa se integraran todos los elementos seleccionados, así como las conexiones electrónicas necesarias para la etapa de potencia de los motores y el acoplamiento de señales de los sensores. La etapa de control se divide en tres bloques principales: el control de los motores a pasos, el procesamiento de imagen y el desarrollo de la interfaz de usuario.

En el control de actuadores, se diseñó un circuito electrónico en una placa de circuito impreso (PCB) para gestionar el movimiento de los motores. Este diseño permite ajustar las direcciones, las velocidades y las distancias que recorrerá el sistema mecánico. Además, el circuito facilita la conexión de todos los periféricos del sistema.

El funcionamiento del procesamiento de imagen en este proyecto es recopilar la información sobre la geometría de los patrones de los contactos de la celda fotovoltaica, donde

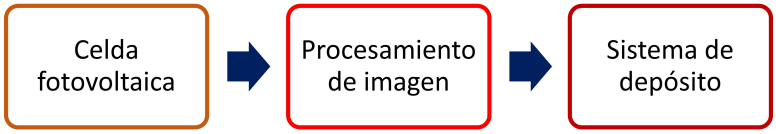


Fig 1.8: Diagrama de bloques.

se depositará el material conductor. Esta información servirá como referencia de entrada en el diagrama de bloques, como se muestra en la Fig.1.8. Con el procesamiento de imagen, se obtendrá la ubicación de los patrones, permitiendo que el sistema de depósito siga el recorrido adecuado y deposite el material en dicho patrón.

El desarrolló de la interfaz incluye el diseño de una interfaz gráfica mediante un software que ofrece diversas herramientas para facilitar el desarrollo del sistema gráfico. Esta interfaz brinda al usuario un entorno intuitivo y funcional, permitiéndole ajustar parámetros como el número de pasos del motor, la velocidad de operación, la asignación del origen del sistema mecánico y la visualización de la cámara en tiempo real, entre otras tareas.

En la etapa de integración y pruebas del sistema, se realizó el depósito del material conductor en los contactos de la celda fotovoltaica, seguido de una caracterización eléctrica. Inicialmente, se evaluó la celda sin el material conductor y, posteriormente, con el material depositado en los patrones de los contactos. Este proceso permitió comparar el rendimiento de la celda en ambas condiciones. Además, se analizó su desempeño en la generación de energía, evaluando el impacto directo de la incorporación del material conductor en los patrones de los contactos de la celda.

Fig 1.9: Diagrama de actividades del desarrollo del proyecto.

Capítulo 2

Marco teórico

2.1. Física de materiales semiconductores

Las características de los semiconductores se pueden modificar agregando impurezas al material, a este fenómeno se le denomina dopamiento. El silicio es el material más utilizado en los semiconductores, ya que se encuentra en el grupo 4 de la tabla periódica, esto hace que se pueda unir con el grupo 3 y 5.

2.1.1. Bandas de energía

La materia cuenta con un gran número de átomos que permite formar bandas continuas de energía, los electrones asociados a los átomos del material llenan en orden ascendente en una banda que contiene mayor energía, se le llama banda de valencia que están ligados los electrodos a los átomos. La siguiente banda es de conducción qué está parcialmente ocupada o vacía también nominados huecos, está desligada con los electrones y los átomos. Esta banda puede estar separada por otra banda de energía que corresponde al estado no permitido por ello se le nombra la banda prohibida.
La materia depende de la posición entre bandas para determinar las propiedades eléctricas que tiene este, así que la anchura de la banda prohibida permite clasificar el tipo de conducción que tiene el material. Un material que cuenta con una buena conductancia, este tiene la banda prohibida traslapada entre las bandas de conducción y valencia por lo cual permite que los electrones circulen fácilmente entre bandas, en cambio un material aislante necesita de una mayor cantidad de energía para poder pasar la banda prohibida, sin embargo un semiconductor la banda es baja de forma que los electrones pueden saltar la banda de conducción con un aporte de energía, por ejemplo, para el silicio $E_G = 1,12eV$.

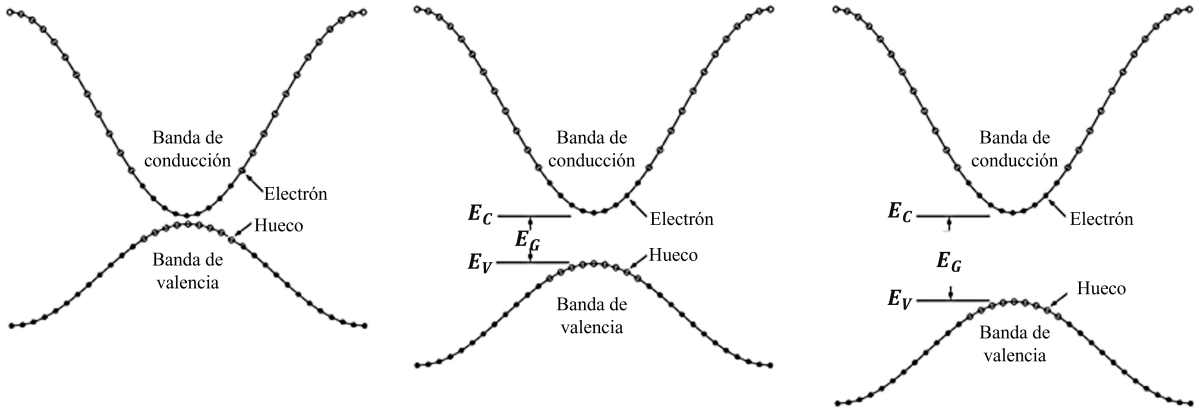


Fig 2.1: Diagrama de bandas de energía de un material conductor, semiconductor y aislante [6].

La temperatura ocupa un gran papel en este asunto, cuando los enlaces se rompen debido a la vibración térmica de los átomos en la red, creando electrones libres en el material. La energía requerida para romper los enlaces es E_G, el electrón que tiene esta energía y queda libre realiza la entrada de la banda de valencia a la banda de conducción, en esta banda los electrones libres podrán adquirir movimiento bajo la acción de un campo externo, los electrones que se encuentran en la banda de valencia también se van a desplazar ya que existe un estado libre, por la pérdida de ese electrón que saltó a la banda de conducción. El resultado de esto es el movimiento de vacantes o huecos de carga positiva, esto se forma cuando se rompe un enlace en un material semiconductor puro, con un electrón y un hueco, a lo que se identifica como portadores, estos quedan libres para moverse por el material, sin embargo, la densidad de los huecos y los electrones es idéntica, llamada densidad intrínseca, que depende de la anchura de la banda prohibida y la temperatura. La corriente eléctrica resultante no tiene una dirección definida dentro del material, por lo tanto, no es aprovechable en un circuito externo. Se restablecen enlaces electrón-hueco que liberan energía en forma de calor cada cierto tiempo, esta incidencia se llama recombinación y es a través de las impurezas existentes en el cristal. El objetivo es tener corriente eléctrica y aprovecharla externamente, para ello es necesario evitar la recombinación y dirigir el movimiento de los electrones y huecos mediante un campo eléctrico.

2.1.2. Semiconductor intrínseco

Un semiconductor intrínseco ideal es un semiconductor puro sin átomos de impurezas y sin defectos en la red cristalina, por ejemplo, silicio puro. Este tipo de semiconductor, si cuenta con una temperatura de $0°K$, todos los estados de energía en la banda de valencia

están llenos de electrones y todos los estados de energía en la banda de conducción están vacíos de electrones. A medida que la temperatura comienza a aumentar por encima de $0°K$, los electrones de valencia ganarán energía térmica. Entonces, un semiconductor intrínseco, crean pares de electrones y huecos mediante la energía térmica, de modo que el número de electrones en la banda de conducción es igual al número de huecos en la banda de valencia.

2.1.3. Semiconductor extrínseco

Se define como un semiconductor extrínseco a aquel que se le añade una cantidad controlada de dopantes específicos o átomos de impureza para que las concentraciones de electrones y huecos que están en equilibrio térmico sean diferentes en la concentración de portadores intrínsecos, esto quiere decir que un tipo de portador predominará en un extrínseco [5].

2.1.4. Unión n-p

La unión empieza a través de dos materiales extrínseco de tipo P, son los materiales que cuentan con una mayor cantidad de huecos y menor cantidad de electrones, lo cual este material está cargado positivamente, en cambio el material de tipo N, tiene la característica contraria, este tiene una mayor cantidad de electrones, en consecuencia está cargado negativamente.
Al realizar la unión físicamente de los dos tipos, se produce un desequilibrio potencial de cargas, ya que hay una diferencia de electrones y huecos en cada material, para alcanzar el equilibrio se produce la difusión de portadores mayoritarios, donde hay un movimiento de huecos desde el material tipo P al tipo N, quedando negativamente. Al mismo tiempo, hay un movimiento de electrones desde el material tipo N al tipo P.
Los iones cargados originan un campo eléctrico dirigido desde el semiconductor tipo N, que está cargado positivamente hacia el semiconductor tipo P, que está cargado negativamente. Este campo arrastra a los electrones del material P hacia el N, sacando a los huecos desde el material N hacia el P, este proceso tiene la dirección opuesta al proceso de difusión, el cual permite alcanzar el equilibrio del movimiento cuando el proceso de difusión y de arrastre se compensan. Esta recombinación se realiza en un área cercana a la unión de los materiales llamada área de carga espacial o área de agotamiento, esta área queda vacía de portadores y solo hay iones cargados que generan un campo eléctrico de arrastre en la unión, como una barrera de potencial llamada potencial termodinámico que limita el paso de portadores mayoritarios de un lado a otro, por lo cual al alcanzar un equilibrio en la

unión P-N, la corriente eléctrica es nuevamente nula [6].

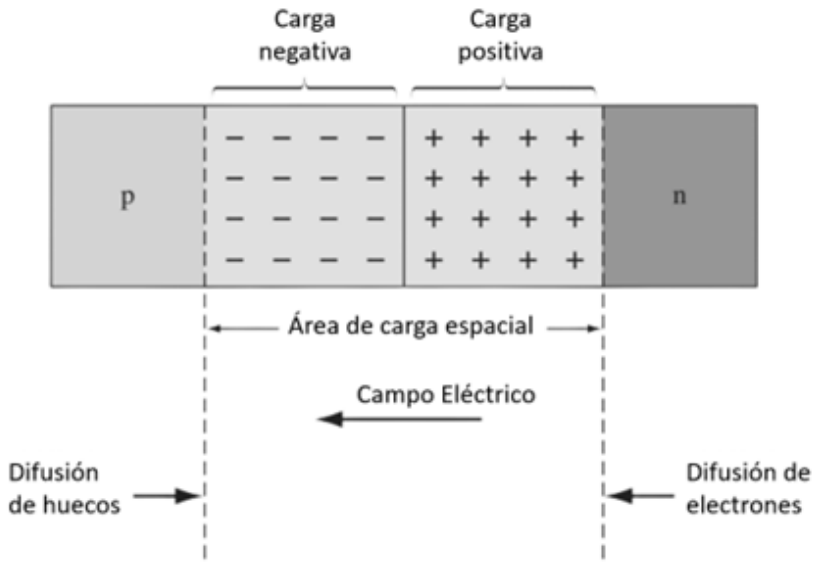


Fig 2.2: Unión n-p [6].

Para superar la barrera, es necesario reducir el valor del potencial termodinámico, una solución es polarizar la unión P-N aplicando una diferencia de potencial entre los extremos del material, de manera que el lado P se polarice positivamente con respecto al lado N, esto se conoce como polarización directa de la unión P-N. Por lo cual la corriente de arrastre disminuye y no logra compensar la corriente de difusión, impidiendo que se alcance el equilibrio. Como resultado, aparece un flujo de corriente. Los huecos de la zona P pueden atravesar la barrera de la zona de carga espacial e ingresar a la zona N, donde se convierten en portadores minoritarios. Allí se produce un proceso de difusión y recombinación, que ocurre de manera similar con los electrones provenientes de la zona N que se introducen en la zona P. Así se originan dos corrientes en sentido contrario, pero se trata de partículas de diferente signo, estas no se anulan entre ellas, sino que dan origen a una corriente que se puede aprovechar. Si la diferencia de potencial aplicado consigue que en la zona P esté a menor tensión que la zona N, la unión quedará polarizada en inversa, en este caso la barrera de potencial en la unión queda reforzada y el paso de portadores de una zona a otra queda debilitada, entonces la corriente que fluye en la unión es muy baja, como la configuración que tiene un diodo en su polarización, donde la zona P es el ánodo y la zona N es el cátodo del diodo.

2.1.5. Circuito equivalente de una celda solar

Consiste en el modelo matemático ideal de un diodo clásico de unión p-n. El circuito para analizar el funcionamiento y la conducta de las celdas solares es el modelo del diodo que se muestra en la Fig. 2.3. Utilizando las leyes Kirchhoff de voltaje y corriente en el circuito de la Fig. 2.3, se puede saber la corriente generada (I_{CEL}) por el panel solar, como se muestra en las ecuaciones (2.1) y (2.2) [8]. La fuente I_{PH} es la corriente fotogenerada a un valor fijo de radiación solar, la resistencia Shunt (R_{SH}) es la pérdida por corriente de fuga, la resistencia en serie (R_S) se utiliza para representar la caída de voltaje a la salida, aunque también se menciona que esta si afecta significativamente el comportamiento de la celda y la resistencia de carga (R_L) que representa la carga usada para la medición. Adicionalmente, I_{SH} es la corriente de fuga en la resistencia shunt, I_D es la corriente del diodo ideal, V_{CEL} es el voltaje de la celda solar, K_B es la constante de Boltzmann ($1.38x10^{-23}J/K$), I_0 es la corriente de saturación de polarización inversa del diodo, T es la temperatura de operación de la celda en Kelvin, n es el factor de idealidad y q es la carga del electrón ($1.6x10^{-19}C$) [7].

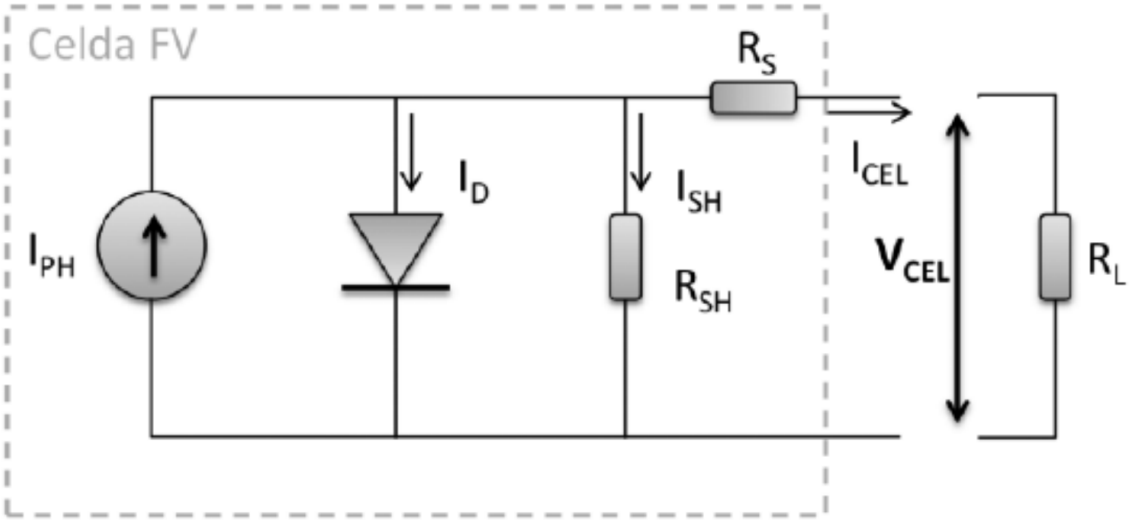


Fig 2.3: Circuito equivalente de una panel solar [7].

$$I_{CEL} = I_{PH} - I_D - I_{SH} \tag{2.1}$$

- I_{CEL}: Corriente de la celda fotovoltaica.
- I_{PH}: Corriente fotogenerada.
- I_D: Corriente de cortocircuito.

- I_{SH}: Corriente de resistencia Shunt.

$$I_{CEL} = I_{PH} - I_0 \left[e^{\frac{q\,(V_{CEL} + R_S I_{CEL})}{n K_B T}} - 1 \right] - \frac{V_{CEL} + R_S I_{CEL}}{R_{SH}} \tag{2.2}$$

- I_0: Corriente del vació.
- q: Carga de un electrón ($1.6x10^{-19}C$).
- V_{CEL}: Voltaje de la celda.
- R_S: Resistencia en serie.
- R_{Sh}: Resistencia en Shunt.
- n: Factor de idealidad del diodo.
- K_B: Constante de Boltzmann ($1.38x10^{-23}J/K$).
- T: Temperatura en Kelvin.

2.1.6. Resistencia en Serie R_S

La R_S representa la resistencia total de la celda y es un compuesto de [23]:

- El movimiento de la corriente a través del emisor y la base de la célula solar.
- La resistencia de contacto entre el contacto de metal y el silicio.
- La resistencia en los contactos superiores y traseros con el metal.

La R_{sh} representa la unión p-n de la celda solar. En conjunto R_s y R_{sh} corresponden al factor de llenado.

Existen diferentes modelos reportados en la literatura para utilizar la ecuación que representa a una celda fotovoltaica, con distintos grados de detalle y diversos tipos de pérdidas considerados. Sin embargo, estos modelos se basan en un conjunto de parámetros que no son directamente accesibles, pero que son necesarios para crear un modelo de ajuste adecuado para la celda fotovoltaica. En este artículo [24] se menciona un modelo matemático para calcular la resistencia en serie de la celda, el cual se representa en la Ecuación (2.3).

$$R_s = \frac{V_t In\left(\frac{I_{sc} - I_{max}}{I_{sc}}\right) + V_{OC} - V_{max}}{I_{max}} \tag{2.3}$$

Para la ecuación (2.3), R_{sh} se considera mínima, por lo cual no se consideró. Otra simplificación fue $I_{PH} \approx I_{sc}$, lo cual se debe a que la diferencia entre la corriente fotogenerada y la corriente de cortocircuito es muy pequeña.

- R_s: Resistencia en Serie.
- V_t: Voltaje térmico.
- I_{sh}: Corriente de cortocircuito.
- V_{oc}: Voltaje de circuito abierto.
- I_{max}: Corriente máxima para la Celda.
- V_{max}: Voltaje máximo para la celda.

Para calcular el voltaje térmico V_t se aplica la siguiente ecuación (2.4) [25].

$$V_t = \frac{nK_BT}{q} \tag{2.4}$$

- V_t: Voltaje térmico.
- n: El factor de idealidad del diodo.
- K_B: Constante de Boltzmann (1.38×10^{-23}J/K).
- T: Temperatura en Kelvin.
- q: Carga del electrón (1.6×10^{-19}C).

2.1.7. Gráfica I-V

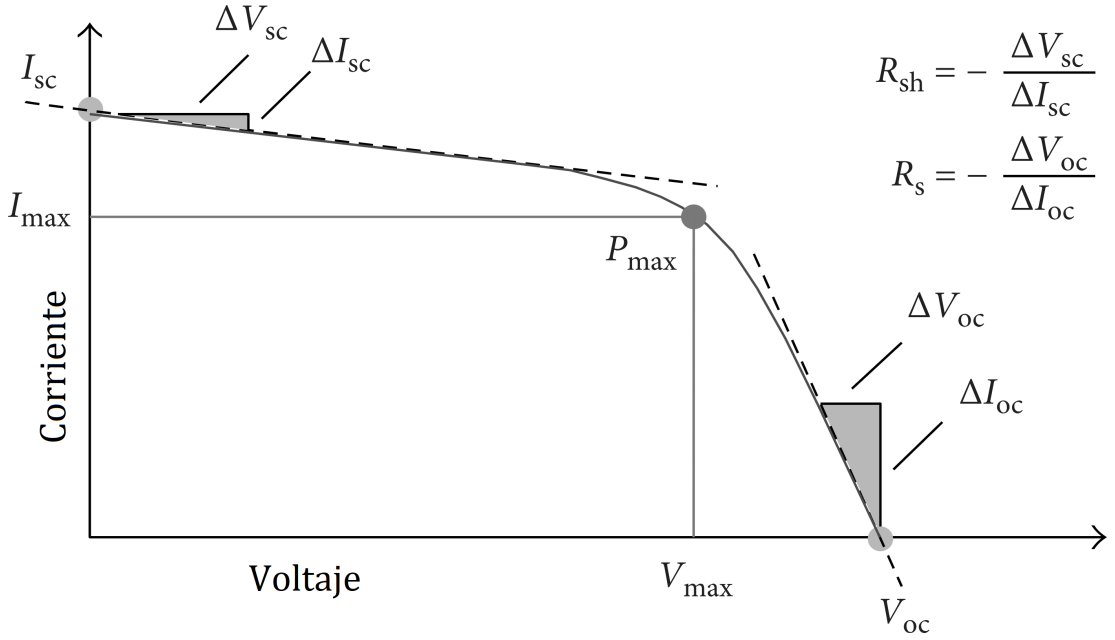


Fig 2.4: Gráfica I-V de una celda solar [8].

Ilustrando la gráfica I-V de la celda solar en la Fig. 2.4, se observan los parámetros más importantes que la conforman. En el eje horizontal se encuentran los valores del voltaje

y en relación con el eje vertical que representa las corrientes de la celda solar. La gráfica I-V es realmente útil para poder encontrar los valores de los parámetros que conforman el análisis de la celda de una forma visual. La corriente de corto circuito (I_{SC}), se localizará en el punto máximo que toca la curva en el eje vertical que representa la corriente; de forma similar, el voltaje de circuito abierto (V_{OC}), se encontrará en el punto que corta el eje horizontal del voltaje, de igual forma se define como el punto donde I_{SC} es igual a cero. De igual modo se pueden encontrar la corriente máxima (I_{max}) y el voltaje máximo (V_{max}), que están ubicados en el punto donde la curva decrece de forma casi constante o donde se puede trazar fácilmente un rectángulo como se observa en la Fig. 2.4. Tomando en cuenta que disminuyendo el valor de R_s hace que I_{SC} y V_{max} aumenten, y aumentando el valor de R_{sh} hace que I_{max} y V_{OC} aumenten, por lo que, a partir de este suceso, los investigadores desarrollan celdas solares a través de la variación de las resistencias Rs y Rsh [8].

2.1.8. El factor de llenado (FF)

Representado por sus siglas en inglés (Fill Factor), este parámetro determina la potencia máxima (P_{max}) de la celda solar, en relación con la potencia máxima calculada y el producto de la corriente de cortocircuito (I_{sc})y la tensión del circuito abierto (V_{oc}) que son los parámetros máximos que se pueden obtener en la celda solar, la fórmula para su obtención está representada mediante la ecuación (**??**). Como se observa en la Fig. 2.4, los valores del voltaje y corriente máxima, así como el voltaje de circuito abierto y la corriente de corto circuito, crean la figura de un rectángulo. Esta figura representa al factor de llenado, dado que refleja la relación entre los parámetros máximos y los parámetros reales del dispositivo, precisando qué tan cercanos se encuentran al modelo ideal; es por esto que el Factor de llenado se representa como un valor porcentual [26].

$$FF = \frac{I_{max}V_{max}}{I_{sc}V_{oc}} = \frac{P_{max}}{I_{sc}V_{oc}} \tag{2.5}$$

2.1.9. La eficiencia

Conocida también como el rendimiento, se representa mediante el símbolo η, es la relación entre la potencia máxima (P_{max})que se puede obtener de la celda entre la potencia de la radiación solar (P_{in}) que incide sobre el área (A) de la celda, es una manera de medir el aprovechamiento de los recursos que inciden en relación con los que se obtienen. La potencia de entrada de la radiación solar para cálculos de eficiencia se considera como $1\frac{kW}{m^2}$ o $100\frac{mW}{cm^2}$, la ecuación (2.6) representa la eficiencia de una celda solar.

$$\eta = \frac{I_{max}V_{max}}{P_{in}A} = \left(\frac{I_{sc}V_{oc}}{P_{in}A}\right) FF \tag{2.6}$$

- η: Eficiencia.
- P_{in}: Potencial de la radiación solar $(\frac{W}{m^2})$.
- A: El area de la celda fotovoltaica (m^2).

2.2. El procesamiento de imágenes

Es implicar cambiar la naturaleza de una imagen para

1. Mejorar su información pictórica para la interpretación humana.
2. Hacer una adecuada percepción de máquinas autónomas.

El proceso de hacer que las imágenes sean mejores; posiblemente no sea lo mejor para satisfacer la condición, pero a las personas les gusta que sus imágenes sean nítidas, claras y detalladas; al contrario, a las máquinas les gusta que su imagen sea simple y ordenada [27].

2.2.1. Iluminación

Para obtener una imagen es necesario tener una fuente de energía que ilumine el escenario, para ello es necesario tener en consideración la dirección de la iluminación para obtener una imagen brillante o una no uniforme como se muestra en la siguiente imagen. La única diferencia entre las cuatro imágenes es la dirección de la fuente de luz (una lámpara) cuando se capturaron las imágenes.

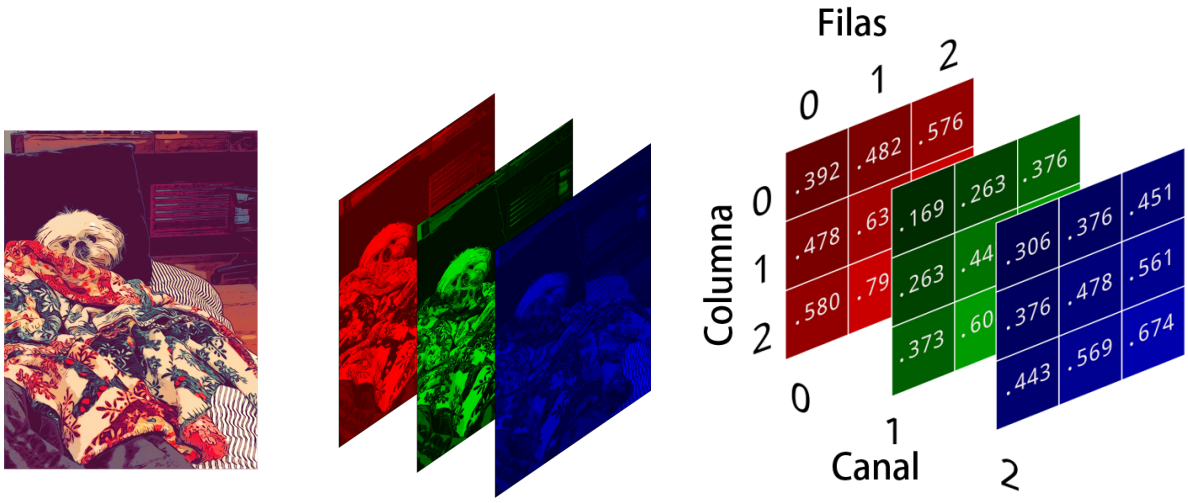
2.2.2. El sensor de imagen

Consiste en una matriz 2D de celdas denominadas píxeles, que son capaces de medir la cantidad de luz incidente y convertirla en voltaje, que a su vez se convierte en un número digital a través de un convertidor analógico a digital. Cuanta más luz haya, mayor será el valor del voltaje. Antes de obtener una imagen, todos los píxeles se vacían, lo que se refiere a que no tienen carga. Cuando se captura la imagen, se permite la entrada de luz y las cargas empiezan a acumularse en cada píxel. Después de un tiempo determinado, conocido como tiempo de exposición, si el tiempo es demasiado bajo o alto, el resultado es una imagen subexpuesta o sobreexpuesta respectivamente, esto afecta el brillo de la imagen [27], [28].

2.2.3. Resolución

Significa la cantidad de píxeles que se utilizan para capturar una imagen, una resolución alta tiene como resultado detalles finos en la imagen: una baja significa que se utiliza un número relativamente bajo de píxeles.

2.2.4. Representación de una imagen digital



(a) Imagen RGB. (b) Valor cuantitativo de la matriz.

Fig 2.5: Representación matricial de una imagen RGB.

Una imagen a color está conformada por tres matrices: una de color rojo, verde y azul, en referencia a una imagen RGB. Como se muestra en la Fig. 2.5a esta imagen se representa con una resolución de 8 bits, lo que significa que tiene 256 posibles valores para representar un color (de 0 a 255). Un ejemplo de ello se muestra en la Fig. 2.5b, donde este valor ya no es visible sino cuantitativo, ya que la computadora interpreta estos valores para ser procesados [29].

2.2.5. La ecualización del histograma

Esta es una técnica que se utiliza para mejorar el contraste de una imagen, al redistribuir las intensidades de píxeles de manera uniforme en todo el rango de valores disponibles. Funciona transformando el histograma original de la imagen en un histograma uniforme, para realizar esta técnica se ocupó un comando de la biblioteca Opencv que fue el siguiente:

$$\text{Imagen\_ecualizada} = \text{cv2.equalizeHist(imagen)}$$

2.2.6. Filtro pasa altas

Esta técnica de procesamiento de imágenes es utilizada para resaltar las características de alta frecuencia de una imagen, como los bordes, los detalles finos y las texturas, mientras suprime o elimina los componentes de baja frecuencia, como el fondo o la iluminación suave. Este filtro funciona centrándose en las diferencias locales en la intensidad de los píxeles de la imagen.

El filtro de pasa altas es una operación de diferencia móvil, que involucra el mismo conjunto de información. Un filtro se clasifica por su respuesta al impulso $f(t)$, también conocido como el núcleo (kernel) del filtro. Esta respuesta representa la señal de salida generada cuando se aplica un impulso unitario como entrada. La señal de salida $y(t)$ de una señal de entrada general $x(t)$ se puede obtener a través de la convolución de $x(t)$ y el núcleo del filtro $f(t)$ [30].

$$y(x) = x(t) * f(t) = \int_{-\infty}^{\infty} x(\tau) f(t-\tau) \mathrm{d}\tau \tag{2.7}$$

Para este caso, el kernel del filtro es el siguiente [31]:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.8}$$

2.2.7. Umbralización

Es el parámetro que se le da a un umbral t para que cuando el valor del pixel $P(i,j)$ sea menor al umbral este sea igual a cero, en cambio cuando el píxel es mayor al umbral, a este se le asignará el valor de 255 [32].

$$P(i,j) = \begin{cases} 0 & \text{si } P(i,j) < t \\ 255 & \text{si } Pi(i,j) > t \end{cases} \tag{2.9}$$

De esa forma se obtiene una imagen binaria, porque en todo el histograma solo se tienen valores de 0 o 255.

2.2.8. Método de Otsu

En la umbralización declaramos el umbral que se va a ocupar de modo arbitrario. Por el contrario, el método de Otsu evita tener que elegir un valor y lo determina automáticamente a través de la ecuación (2.10).

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{2.10}$$

Donde q_1 son los números de píxeles del lado izquierdo de t y q_2 son los píxeles restantes de la imagen, $\sigma_1^2(t)$ y $\sigma_2^2(t)$ son las variaciones de los píxeles de la izquierda y la derecha de t respectivamente [33].

2.2.9. Error cuadrático medio (MSE)

Es el error cuadrático medio entre imágenes es la suma de las diferencias al cuadrado para después sacar su medio, está dado por la siguiente fórmula (2.11), donde M y N son el ancho y el alto de las imágenes, imagen original $I(i,j)$ y la imagen modificada $K(i,j)$. i y j son los píxeles de fila y columna de ambas imágenes [34].

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I(i,j) - K(i,j)]^2 \tag{2.11}$$

Capítulo 3

Diseño del sistema

3.1. Elección de materiales

Un sistema mecánico está conformado esencialmente por componentes, dispositivos o elementos cuya función específica es convertir o transmitir el movimiento desde las fuentes que lo generan, transformándolo en diferentes formas de energía. Los componentes que se utilizaron para el diseño del sistema de depósito se eligieron de tal modo que el sistema cumpla con su funcionamiento principal, que es el depósito correcto del material conductor en los patrones de la celda fotovoltaica. La elección adecuada de estos materiales es importante, ya que su desempeño afecta directamente el funcionamiento adecuado del proyecto.

3.1.1. Materiales para el soporte y protección del sistema

Tomando en cuenta las variables externas que pueden influir negativamente en el rendimiento del proceso, como el polvo, el flujo de aire y otros factores ambientales que pueden afectar el rendimiento, es fundamental implementar medidas que minimicen su impacto. Para ello, se debe garantizar una adecuada reducción de estas variables mediante una estructura de protección y soporte que proporcione un aislamiento al sistema. Por lo tanto, se proponen los siguientes materiales para el diseño y construcción de dicha estructura:

- Acrílico transparente de 4 mm, utilizado para conformar las partes de la estructura de la caja de protección del sistema.
- Ángulo de aluminio de lados iguales 1 mm x 19 mm, empleado para sujetar las piezas de acrílico que conforman a la caja de protección.
- Lámina de aluminio H14 3003 de calibre 20, destinada a la base de la protección.

- Madera triplay de 15 mm, utilizada como soporte principal de la caja, la cual está unida al soporte inferior del sistema.
- PTR de aluminio de 40 mm x 40 mm, empleado en el armado del soporte de la mesa del sistema, el soporte del sistema de y del soporte de la pantalla táctil.

(a) Acrílico trasparente.

(b) Ángulo de aluminio.

(c) Lámina de aluminio.

(d) Madera triplay

(e) PTR de aluminio.

Fig 3.1: Materiales para el soporte y protección del sistema.

3.1.2. Actuadores

Para realizar el movimiento del sistema de depósito, es necesario tomar en cuenta los actuadores, estos juegan un papel crucial en este proceso, estos dispositivos son responsables de convertir la energía eléctrica en movimiento mecánico. Los actuadores deben cumplir con varios requisitos esenciales, en primer lugar, deben de tener precisión para garantizar que el movimiento sea exacto y repetitivo, esto es fundamental para asegurar el posicionamiento y el seguimiento adecuado, ya que si se avanza una cierta distancia con un cierto número de pasos, este debe ser capaz de repetir la misma distancia con el mismo número de pasos. En segundo lugar, la estabilidad es un aspecto crucial para mantener el rendimiento y la eficiencia del sistema a lo largo del tiempo, evitando vibraciones y movimientos no deseados que pudieran afectar la alineación y el depósito de material en los contactos.

Tomando en cuenta esta consideración se realizó un estudio sobre motores a pasos que pueden llegar a cumplir con los requerimientos necesarios para el sistema. En este caso, se seleccionó el motor a paso tipo NEMA 23, que cumple con los estándares establecidos por la National Electrical Manufacturers Association (NEMA) en cuanto a montaje y dimensiones [35]. El término NEMA 23 hace referencia a las características dimensionales del motor, en particular a un tamaño de cuadro o brida de 2.3 pulgadas.

Especificación del motor a pasos E21NRFT-LNN-NS-00

- Torque unipolar: 0.61 N/m
- Torque bipolar: 0.87 N/m
- Carga máxima de empuje: 7.975 kg
- Minimo de resistencia torque: 0.034 N/m
- Avance por paso: 1.8°

Fig 3.2: Motor a pasos E21NRFT-LNN-NS-00.

Se ocupó otro tipo de motor, este siguió con las mismas especificaciones del estándar NEMA permitiendo mantener la estandarización del sistema y facilitar la adquisición de piezas en el futuro. Se utilizó un motor NEMA 17, es más pequeño en comparación con el motor NEMA 23, pero adecuado para desempeñar una función distinta, su tarea es el movimiento del sistema de depósito de tinta conductora. A continuación, se presentan algunas de sus especificaciones técnicas.

Fig 3.3: Motor a pasos NEMA 17.

Especificación del motor a pasos NEMA 17

- Voltaje: 12V-24V
- Corriente: 1.4 A
- Avance por paso: 1.8°
- Par de retención: 420 mN/m

3.1.3. Controlador de motor a pasos

Para obtener el movimiento del motor a pasos, se requiere de una etapa de potencia que opera mediante un controlador, este funciona utilizando un tren de pulsos, el cual está directamente proporcional a la frecuencia que va relacionado a su velocidad, en otras palabras, si tiene una frecuencia elevada, se obtendrá una velocidad elevada, mientras que, si no posee una frecuencia el motor no se moverá. Otra función que realiza el controlador es la capacidad de invertir el sentido de giro del motor a pasos, esta función permitirá a que se pueda tener control en la dirección que tome el mecanismo. Adicionalmente, algunos controladores incluyen una entrada específica para activar o desactivar el controlador, con el fin de tener un paro de emergencia o para detener el motor cuando no se requiere su operación, lo que contribuye a la seguridad del sistema.

Para mover el motor a pasos E21NRFT-LNN-NS-00 de la empresa Pacific Scientific, se utilizó el controlador que el fabricante recomienda, el cual es el controlador 5230. Este controlador, de la misma marca, cuenta con un manual de operación que detalla diversas características técnicas. A continuación, se describen algunas de ellas:

Características del controlador 5230

- Corriente de entrada: 0-4 A
- Voltaje de salida: 35 V DC
- Señal de control: 5-24 v DC
- Frecuencia máxima: 20kHz
- Fuente de alimentación: Si

Fig 3.4: Controlador 5230.

Aunque inicialmente se consideró el controlador 5230, se exploraron otras opciones para accionar el motor a pasos E21NRFT-LNN-NS-00, buscando un dispositivo que cumpliera con las especificaciones requeridas. En este proceso, se encontró el controlador DM860, el cual reúne todas las características necesarias para operar el motor.

Una de las ventajas del controlador DM860 es que incluye un módulo de interruptores que permite ajustar diversas configuraciones, lo que facilita la personalización de su modo de operación para adaptarse a los requerimientos específicos del sistema.

Características del controlador DM860

- Corriente de entrada: 0 - 7.2 A
- Voltaje de entrada: 24 a 80 V DC
- Señal de control: 5-24 v DC
- Frecuencia máxima: 200kHz
- Fuente de alimentación: No

Fig 3.5: Controlador DM860.

Para controlar el motor NEMA 17 del sistema de depósito de material, se seleccionó un controlador específico que cumple con los requisitos del motor. Dado que el NEMA 17 es un motor pequeño con bajo consumo energético, se optó por el controlador A4988, que se adapta perfectamente a sus características y necesidades de operación. Este controlador permite ajustar la corriente máxima de salida mediante un potenciómetro integrado en la PCB. Además, cuenta con protección contra sobrecorriente y ofrece cinco configuraciones de resolución de pasos. Funciona con voltajes de alimentación entre 8V a 35V, suministra una corriente continua de hasta 1A y soporta picos de corriente de hasta 2A. Su funcionamiento requiere de dos pines: uno para controlar la dirección de giro y otro para los pasos. Para activar el motor, el pin Enable debe conectarse a tierra. La configuración de las resoluciones de paso se realiza mediante los pines MS1, MS2 y MS3, según la tabla del fabricante. En este caso, se utilizarán pasos completos, por lo tanto, todos los pines se conectarán a tierra.

Fig 3.6: Controlador A4988.

Características del controlador A4988

- Voltaje de alimentación-potencia: 8V-35V DC
- Señal de control: 3V-5V DC
- Corriente de salida: 1A
- 5 resoluciones de pasos: paso completo, medio paso, 1/4, 1/8 y 1/16
- Fuente de alimentación: No

3.1.4. Tornillo de bola

Fig 3.7: Tornillo de bola.

El sistema de depósito requiere de un tornillo de bola o también conocido como husillo de bola, para transformar el movimiento rotatorio del motor a pasos en un movimiento lineal, el hilo del tornillo de bola es helicoidal y está especialmente diseñado para incorporar una serie de pequeñas bolas. Estas bolas se encuentran en el espacio entre el hilo del tornillo y una tuerca especial que se desliza sobre el tornillo. Las bolas actúan como rodamientos, lo que reduce la fricción y permite un movimiento mucho más suave y eficiente.

3.1.5. Guías lineales tipo R y carro lineal de precisión

El diseño del sistema se basa en movimientos lineales, por lo que es crucial utilizar sistemas que generen este tipo de movimiento de manera precisa y con baja fricción. Una alternativa de solución es ocupar guías lineales de tipo R, acompañadas de un carro lineal de precisión. Con el fin de obtener un gran rendimiento en el movimiento lineal, evitando que se descalibre o se desgaste el sistema en un corto plazo.

Fig 3.8: Guías lineales y carro lineal.

3.1.6. Rodamiento

Fig 3.9: Rodamiento.

Para garantizar un funcionamiento adecuado en el movimiento del tornillo de bola y minimizar los efectos de fricción entre el tornillo y el soporte, se opta por el uso de un rodamiento rígido de bolas de acero inoxidable, modelo W61703 de la marca SKF, como se muestra en la Fig.3.9. Este rodamiento ha sido seleccionado debido a sus excelentes características, que aseguran un rendimiento eficiente y duradero en el sistema. Las especificaciones del rodamiento elegido son las siguientes:

Características del rodamiento W61703

- Diámetro interno: 17mm.
- Diámetro externo: 23mm.
- Altura: 4mm.
- Capacidad de carga dinámica: 0.559kN.
- Capacidad de carga estática: 0.34kN.
- Velocidad límite: 38000 RPM.
- Peso: 0.0035kg.

3.1.7. Interruptores de fin de carrera

Un final de carrera o interruptor de posición es un sensor que limita la posición de un elemento móvil mediante accionamiento mecánico, capacitivo o inductivo. En este sistema se ocupará para evitar que el mecanismo sobrepase el área de trabajo asignado. Para definir las distancias en cada eje, se seleccionó el siguiente sensor que cumplirá con la función de delimitar de manera adecuada el recorrido del sistema.

Características del Sensor GXL-8 type

- Tipo del sensor: Magnético.
- Modo de accionamiento: Inductivo.
- Tipo de salida: NPN-NA.
- Voltaje de operación: 12-24V DC.
- Ventajas: Se puede detectar metales y su dimención es reducida.
- Desventajas: Necesita una etapa de acoplamiento de señal de salida.

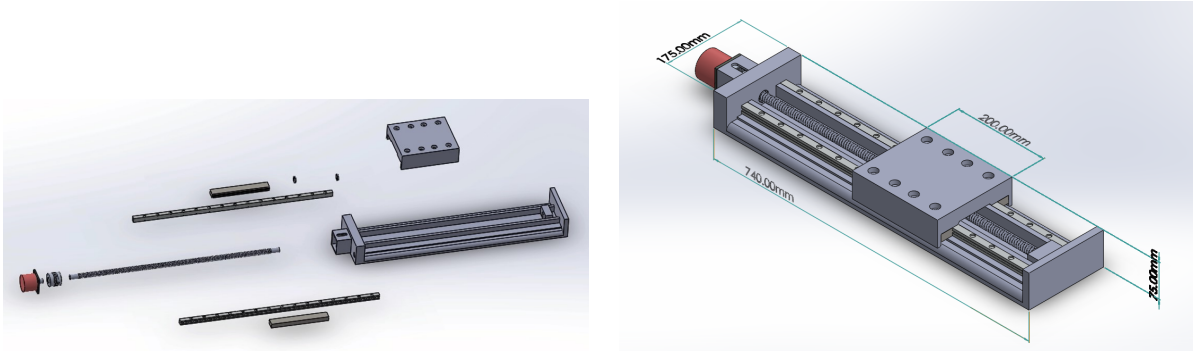
Fig 3.10: Sensor GXL-8 type.

3.2. Diseño mecánico del sistema

Se llevó a cabo el diseño detallado de cada componente mecánico que integra el sistema de depósito, con el objetivo de obtener una visualización precisa antes del ensamblaje del mecanismo. Las piezas y el proceso de ensamblaje fueron modelados en el software SolidWorks®, utilizando las herramientas de dicho programa para simular las uniones y los movimientos que el mecanismo debe realizar. Además, se determinaron las distancias necesarias para garantizar el correcto funcionamiento del sistema.

3.2.1. Diseño mecánico del movimiento en el eje XY

Se realizó primero el ensamble del sistema con el movimiento en el eje X, tal como se muestra en la Fig.3.14a. Este sistema incluye un motor a pasos NEMA23, un tornillo de bola, un acoplamiento para conectar el motor al tornillo, dos sensores de final de carrera inductivos, dos rodamientos W61703, dos rieles tipo R, un carro lineal y una base diseñada para soportar todos estos componentes, tal como se observa en la Fig.3.11b.



(a) Elementos del sistema del eje X. (b) Sistema del eje X.

Fig 3.11: Ensamble del movimiento en el eje X.

Una vez completado el ensamblaje del sistema en el eje X en SolidWorks® , se guardó el modelo y se utilizó el mismo sistema, pero con un giro de 90° para configurar el eje Y, tal como se muestra en la Fig. 3.12.

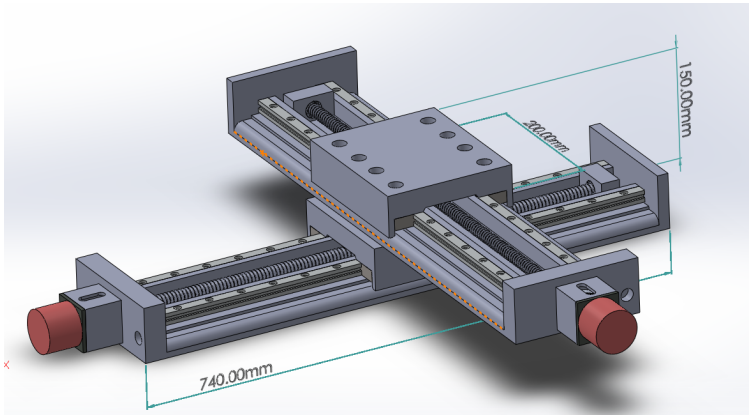


Fig 3.12: Ensamble del movimiento en el eje XY.

En la parte del armado del sistema se tomó en cuenta un sistema ya fabricado por la empresa suiza SCHNEEBERGER®, reconocida por su experiencia en la fabricación de sistemas mecánicos que pueden realizar transmisiones lineales como lo son las guías lineales tipo R, el carro lineal de precisión y el tornillo de bola. En la parte de los motores se optó

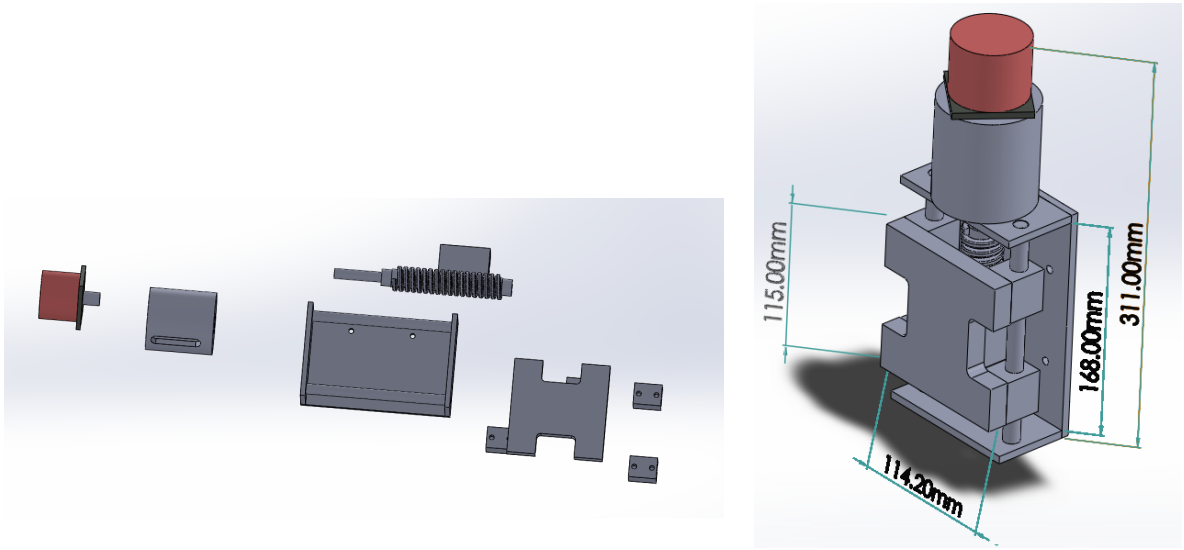
por la empresa Pacific Scientific, con ellos se obtuvo el motor a pasos y su respectivo controlador, para poder controlar el movimiento del sistema.

Fig 3.13: Sistema XY ensamblado.

3.2.2. Diseño mecánico del movimiento en el eje Z

Al igual que en el diseño del eje X, para el eje Z se implementó un sistema que convierte el movimiento rotativo del motor a pasos en un movimiento lineal. Para ello, se emplearon los mismos componentes fundamentales: un tornillo de bola, dos rodamientos, un carro lineal y un acople para conectar el motor con el tornillo. Estos elementos se ilustran en la Fig.3.14a, que muestra una vista explosionada del ensamble, y en la Fig.3.14b, donde se presenta la unión de los componentes modelada en el software SolidWorks®. Este mecanismo tiene como propósito permitir el movimiento vertical del sistema de inyección, facilitando el desplazamiento adecuado en el eje Z. Posteriormente, algunas piezas del sistema fueron fabricadas mediante el uso de una fresadora CNC. Entre los componentes mecanizados destacan las piezas del carro lineal, que incluyen dos cilindros para sujetar el carro, su base y el acople. El ensamblaje de estas piezas se realizó utilizando tornillos M2, M4 y M6, junto con sus respectivas tuercas y rondanas, con el objetivo de garantizar la estabilidad y funcionalidad del sistema.

El proceso de fabricación y ensamble requirió un tiempo considerable, ya que fue necesario ajustar ciertas medidas para asegurar que todas las piezas encajaran correctamente. En la Fig.3.15, se presenta el resultado final del mecanismo ensamblado, donde se observa la configuración completa del sistema.



(a) Elementos del sistema del eje Z. (b) Sistema del eje Z.

Fig 3.14: Ensamble del movimiento en el eje Z.

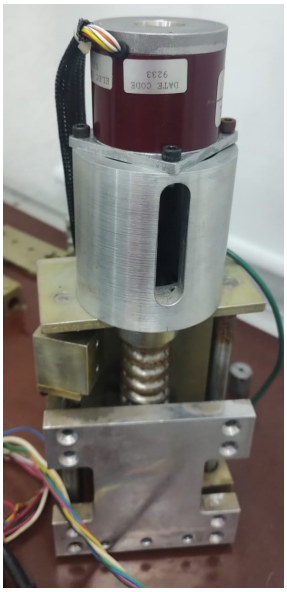
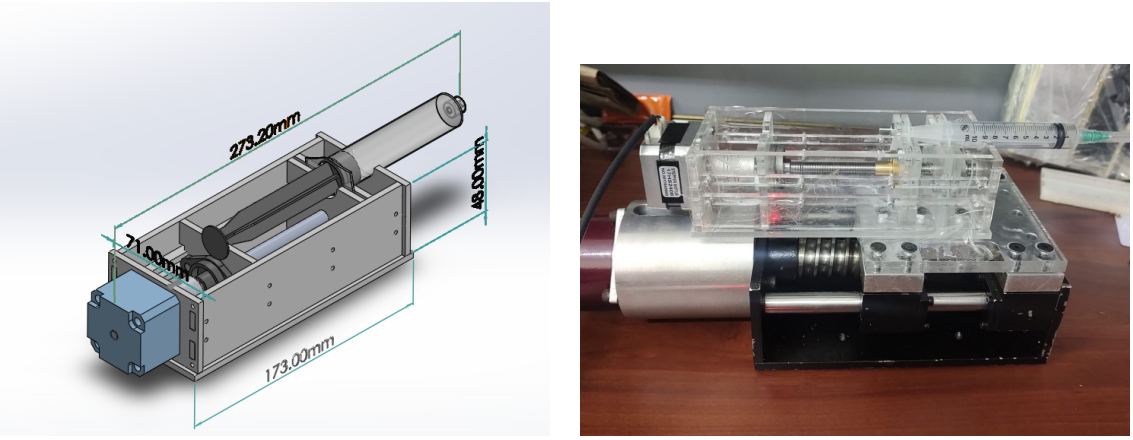


Fig 3.15: Ensamble del movimiento en el eje Z

3.2.3. Ensamble del movimiento del sistema de inyección

En esta sección se detalla el proceso de ensamblaje del sistema de inyección, cuyo propósito principal es depositar material conductor en los contactos de la celda fotovoltaica, constituyendo una de las tareas más relevantes del proyecto. Este sistema emplea un mecanismo mecánico que convierte el movimiento rotatorio del motor en un desplazamiento lineal controlado, siguiendo un principio de diseño de los demás sistemas del proyecto.

El diseño de cada pieza fue realizado en el software SolidWorks®, como se ilustra en la Fig.3.16a Este diseño tomó como referencia elementos de los otros sistemas, adaptándose



(a) Sistema de inyección. (b) Sistema de depósito armado.

Fig 3.16: Ensamble del Sistema de depósito.

para mantener la dimensión dentro del conjunto general del proyecto. Una vez finalizado el modelado, se procedió al ensamblaje en el mismo programa para verificar que las dimensiones de las piezas fueran compatibles con el sistema completo, evitando colisiones. Cabe destacar que el ensamblaje físico de este sistema se realizó al final del proceso de desarrollo, debido a su dependencia de las configuraciones de los demás subsistemas.

Tras validar que todos los parámetros cumplían con los requisitos establecidos, se procedió al corte de las piezas de acrílico mediante una cortadora láser disponible en el laboratorio de prototipado rápido de la institución. El ensamblaje físico del mecanismo se realizó utilizando pegamento instantáneo a base de cianoacrilato, complementado con tornillos M2 junto con sus respectivas tuercas y rondanas, asegurando así la estabilidad y funcionalidad del sistema. El resultado final del ensamblaje del sistema se presenta en la Fig.3.16b, donde se puede observar la unión completa con el eje Z.

3.2.4. Brazo de soporte del eje Z

El objetivo de esta pieza es proporcionar soporte y altura al sistema Z, con el fin de que permita el movimiento ascendente y descendente de su riel. Este riel sostendrá al sistema de depósito encargado de depositar material sobre la celda fotovoltaica. Se tomó en consideración durante el diseño la altura que tiene entre el sistema Z y el sistema XY, a su vez se consideró la longitud del brazo de soporte. Esta longitud se determinó a la mitad del eje X, garantizando que el trabajo se realice en el centro del sistema. Esto contribuye a la estabilidad del movimiento y evita desequilibrios que podrían afectar su funcionamiento. Como resultado tenemos la figura que se muestra en Fig. 3.17.

En la imagen se muestra el diseño del brazo de soporte creado en SolidWorks®, poste-

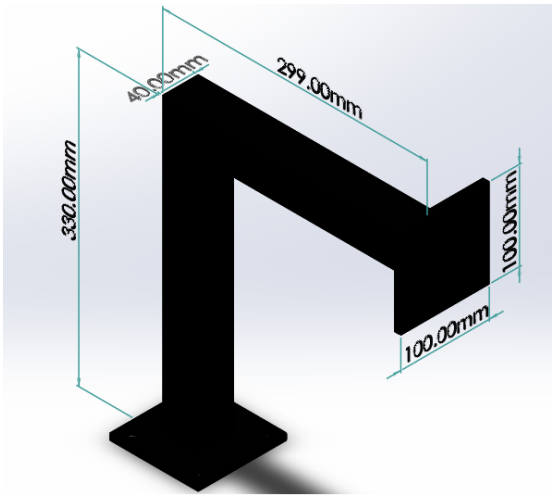


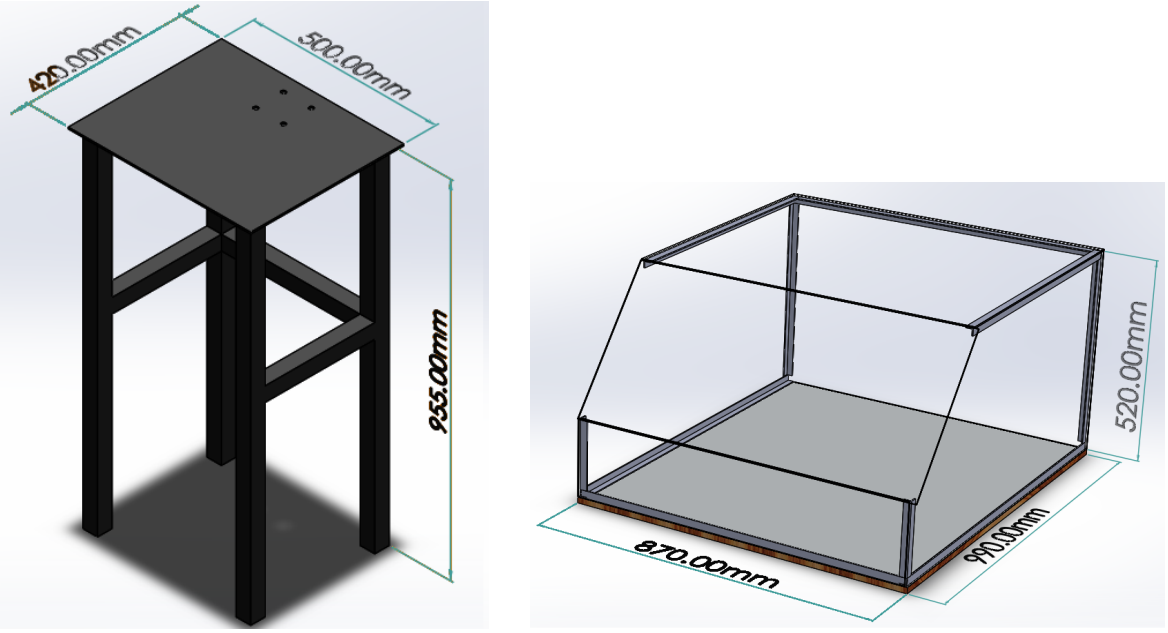
Fig 3.17: Brazo de soporte.

riormente se integró al ensamblaje general del proyecto para verificar su correcta posición en el centroide del sistema. Este proceso permitió asegurar el cumplimiento de las condiciones estructurales establecidas. Una vez validado, el diseño se llevó a la fabricación, cortando el PTR de acuerdo con las dimensiones especificadas y ensamblando las piezas mediante soldadura con electrodos 6013 de 3/32". El resultado es una estructura robusta capaz de soportar todo el peso del eje Z y el sistema de inyección.

3.2.5. Ensamble del soporte del sistema

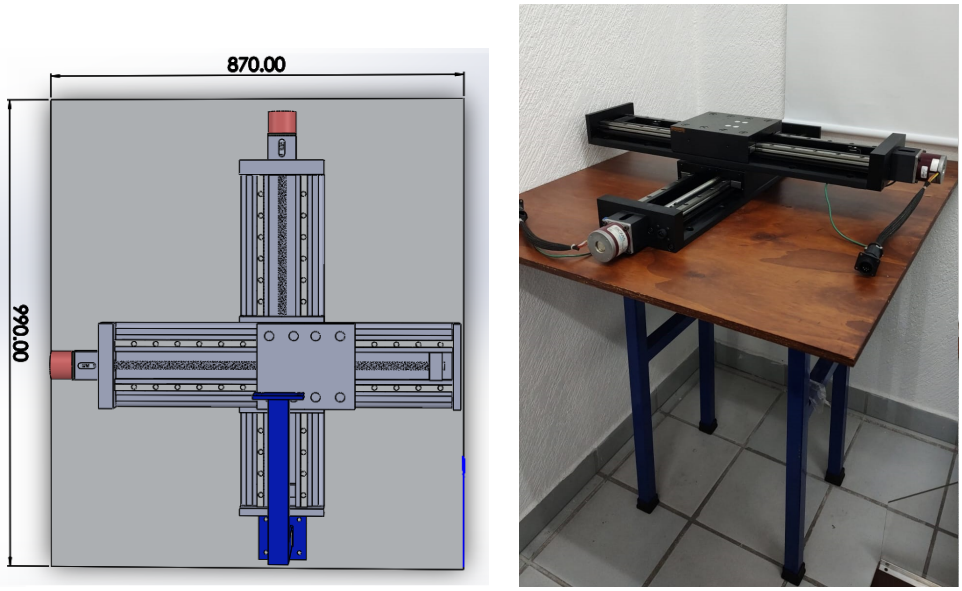
Se diseñó un soporte para alojar el sistema de depósito, con el objetivo de aislarlo de manera efectiva dentro de una caja de acrílico. Este aislamiento es importante para proteger el sistema de las variables externas que podrían comprometer su funcionamiento y las propiedades resistivas de los contactos, como cambios en la temperatura, humedad o la presencia de partículas en el ambiente, enviadas por el flujo de aire que se encuentre en el medio. Además, contar con un entorno controlado asegura una mayor seguridad en el proceso. Para garantizar el correcto diseño de la caja de acrílico, se llevó a cabo una visualización detallada del ensamblaje del proyecto, lo que permitió definir las piezas necesarias para su fabricación a través del programa de SolidWorks®.

Como se muestra en la Fig.3.18b, este ensamblaje está compuesto por ángulos de aluminio y tornillos M2 acompañados de sus respectivas tuercas y rondanas de acero inoxidable. Estos elementos proporcionan un soporte robusto para las piezas de acrílico. Además, se añadió una solera en la parte superior de la caja para evitar que el techo de acrílico se deforme. Para asegurar un acabado estético, se realizaron avellanados en el acrílico, permitiendo que los tornillos queden al ras de la superficie y mejorando la



(a) Parte inferior del soporte. (b) Caja de acrílico.

Fig 3.18: Diseño del soporte del sistema.



(a) Medidas de la base. (b) Sistema XY en la base.

Fig 3.19: Medidas de la base.

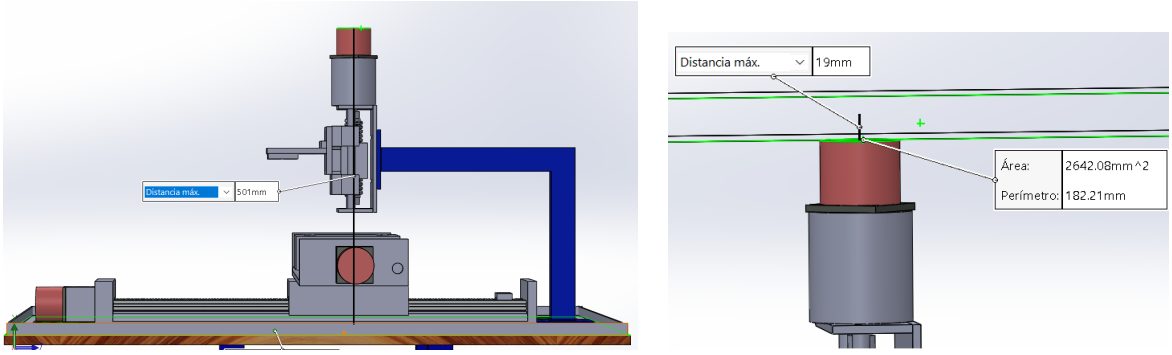
apariencia final del ensamblaje de la caja.

La parte inferior del proyecto se encuentra el soporte que está formado por cuatro tubos de PTR, unidos mediante una lámina de aluminio y tres cortes más pequeños de PTR, los cuales han sido soldados con electrodos 6013 de 3/32. Este soporte tiene la función de

cargar todo el sistema y proporcionar la altura necesaria para que el usuario pueda tener una visualización directa del mismo. El resultado del diseño de este soporte se presenta en la Fig.3.18a.

Para la base de la caja, se calcularon la longitud y el ancho necesarios para garantizar el espacio de movimiento del sistema en los ejes XY, teniendo en cuenta también el espacio requerido por los laterales de la caja de acrílico y los ángulos de aluminio. Como resultado, se obtuvo una base de 990 mm x 870 mm, como se muestra en la Fig. 3.19a. Esta base está construida con madera triplay de 15 mm, seleccionada por su rigidez para evitar deformaciones. Además, para cumplir con los requisitos de aislamiento del sistema, la madera fue recubierta con una lámina de aluminio H14 3003 de calibre 20.

A continuación, se procedió a calcular la altura de la caja de acrílico, considerando la altura del brazo y la del sistema Z en su conjunto, como se muestra en la Fig. 3.20a. El cálculo inicial resultó en una altura de 501 mm, a la cual se añadieron 19 mm adicionales para garantizar una holgura adecuada en la parte superior del sistema, tal como se observa en la Fig. 3.20b. De este modo, la altura final de la caja quedó definida en 520 mm. Con estas dimensiones establecidas, se dio inicio al diseño detallado de la caja de acrílico.



(a) Altura del sistema.

(b) Holgura de la altura del sistema con la altura de la caja.

Fig 3.20: Medidas de la altura.

Una vez finalizado el diseño, las piezas fueron exportadas a un archivo en formato DXF, un tipo de archivo vectorial ampliamente utilizado en aplicaciones CAD debido a su compatibilidad universal y naturaleza de código abierto. Este formato permitió que las dimensiones de las piezas se transfirieran a una cortadora láser. Después del proceso de corte, se procedió al ensamblaje de las piezas. Para garantizar una unión firme y estable, se emplearon ángulos de aluminio y tornillos M2, este proporcionó robustez a la estructura. Como resultado, se obtuvo el diseño final de la caja de acrílico, que se muestra en la Fig. 3.21.

(a) Caja de acrílico en su primera etapa de construcción.

(b) Caja de acrílico en su etapa final de construcción.

Fig 3.21: Caja de acrílico estapas de construcción.

3.2.6. Brazo de soporte de la pantalla táctil, protección para pantalla táctil y cámara.

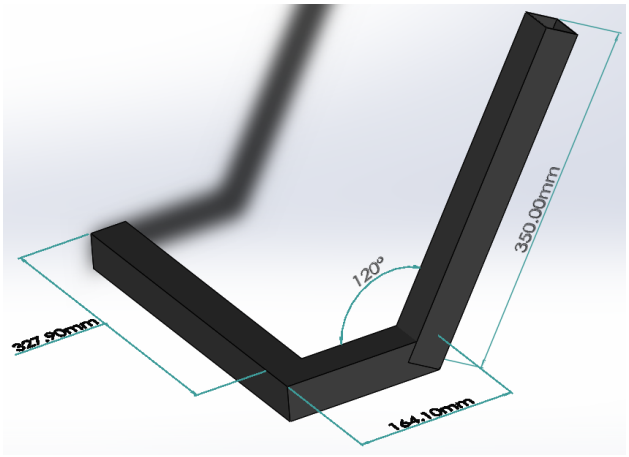
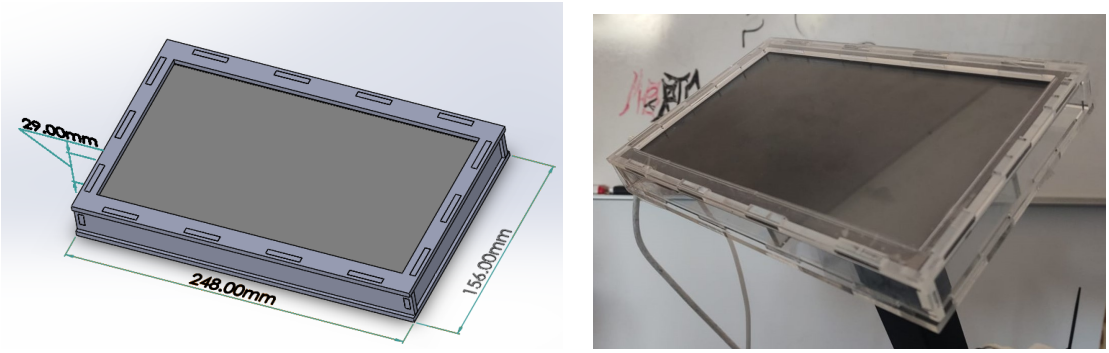


Fig 3.22: Soporte de la pantalla táctil.

La función de esta pieza es proporcionar soporte y altura a la pantalla táctil, permitiendo sujetarla de forma estable y segura para mostrar la interfaz gráfica del sistema. Para su diseño, se tomó en cuenta la altura del soporte de la Fig. 3.18a, que sostiene toda la estructura. En una de las uniones se ha previsto un espacio para atornillar y conectar ambos soportes, asegurando así la estabilidad del conjunto. Además, se ha considerado la altura ideal de la pantalla para que el usuario pueda visualizarla cómodamente durante la operación del sistema. Como resultado, el diseño final del brazo de soporte de la pantalla,

modelado en SolidWorks®, se presenta en la Fig. 3.22

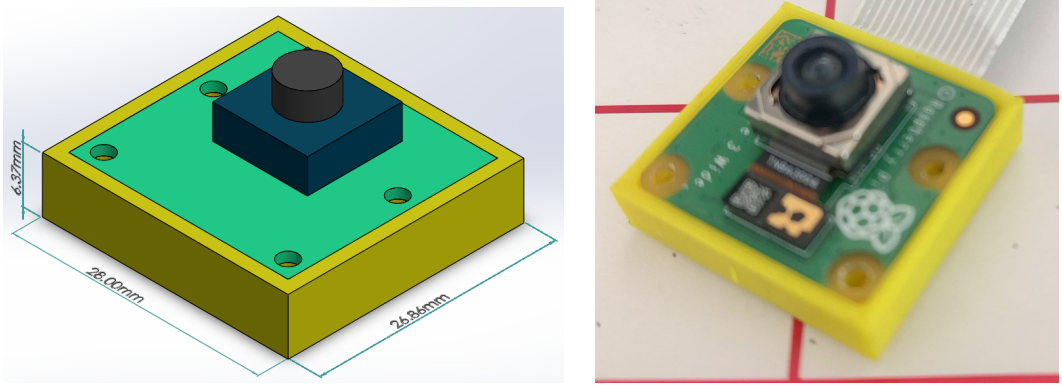
Para proteger la pantalla táctil de 10.1 pulgadas de la marca HMTECH, se consideraron las dimensiones especificadas por el fabricante. Se diseñaron bases para evitar que la tarjeta principal de la pantalla estuviera en contacto con el acrílico, facilitando la conexión de los puertos HDMI y micro USB, utilizados para alimentación y transferencia de datos. El diseño se realizó en SolidWorks®, como se muestra en la Fig.3.23, donde se presentan las dimensiones y el resultado final del proceso.



(a) Diseño de caja de acrílico para pantalla táctil. (b) Caja de acrílico armada para la pantalla táctil.

Fig 3.23: Protección de acrílico para pantalla táctil.

Para proteger la electrónica de la cámara modelo V3 de la empresa Raspberry Pi, se diseñó una carcasa en SolidWorks® considerando las dimensiones proporcionadas por el fabricante. Se dejó un espacio para atornillar la placa y una apertura para el cable flexible. El resultado de este proceso se muestra en la Fig.3.24, junto con las dimensiones y la impresión 3D del diseño.

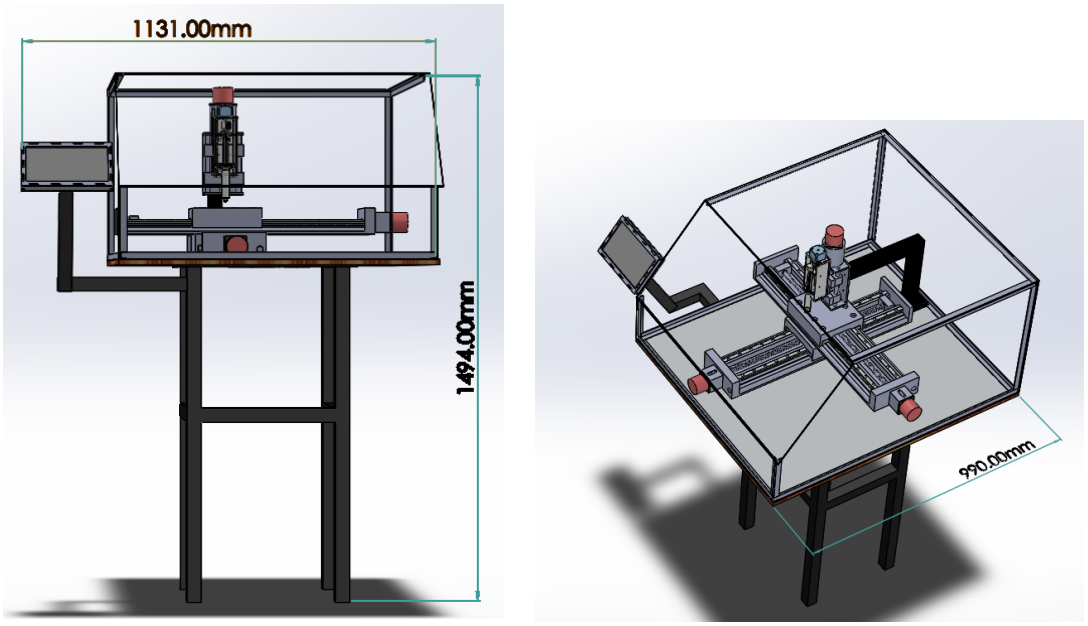


(a) Diseño de carcasa para la cámara. (b) Impresión 3D de la carcasa para la cámara.

Fig 3.24: La carcasa para la cámara modelo V3 de la empresa Raspberry Pi.

3.2.7. Ensamble completo del sistema de depósito

Al tener todas las piezas que conforman el sistema diseñadas en CAD, se pudo realizar un dimensionado del resultado final del proyecto, como se muestra en la Fig. 3.25. Este modelo representa el diseño completo del proyecto, incluyendo todos los subsistemas que lo integran. Para ello, se comenzó a armar las piezas disponibles, como se observa en la Fig. 3.26, ensamblando progresivamente cada uno de los sistemas que forman parte del proyecto.



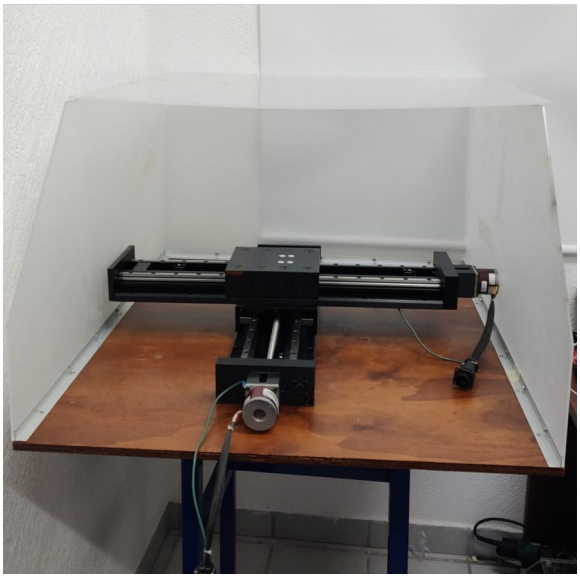
(a) Vista frontal del ensamble. (b) Vista lateral superior del ensamble.

Fig 3.25: Ensamble completo del sistema de depósito.

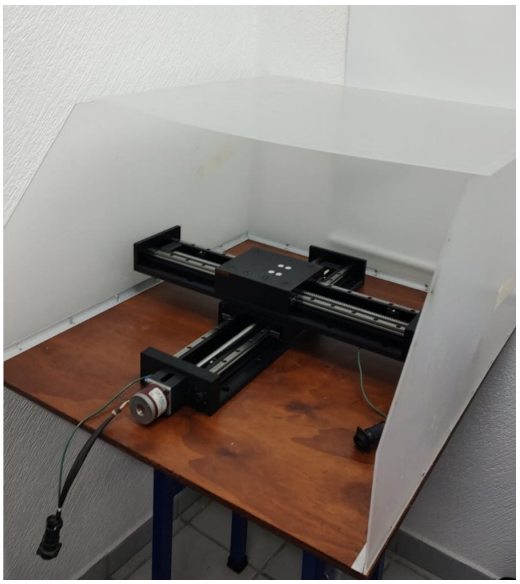
A medida que avanzaba el diseño y construcción de las piezas, se evaluaron cosas clave como la ubicación de la cámara y la posición de los arneses. Como resultado final, se obtuvo el sistema completamente armado, representado en la Fig. 3.27. Esta imagen muestra la integración de los diferentes subsistemas.

Una mejora importante en el diseño fue la incorporación de bisagras y una manija, lo que permite abrir y cerrar fácilmente la puerta de la caja de acrílico. Este material se seleccionó porque es transparente, lo que facilita la observación, mientras que la madera, recubierta de lámina de aluminio, minimiza la reflexión de luz directa, esto evita interferencias en el procesamiento de imágenes de la celda fotovoltaica y mejora el rendimiento del sistema.

Este sistema conserva el protector plástico del acrílico debido a su transparencia. Además, la madera está recubierta con una lámina de aluminio, que podría reflejar la luz del entorno directamente sobre la superficie. Este fenómeno podría generar interferencias en

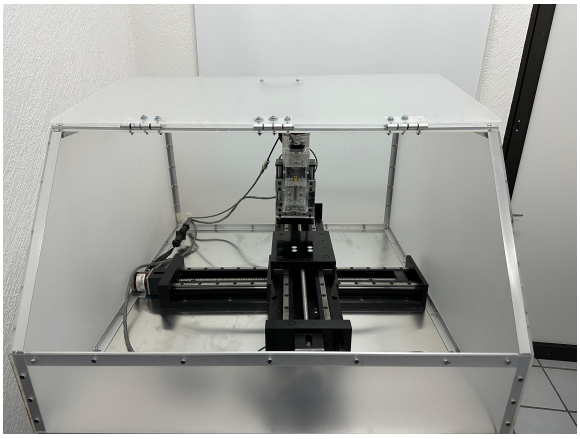


(a) Vista frontal del sistema en proceso

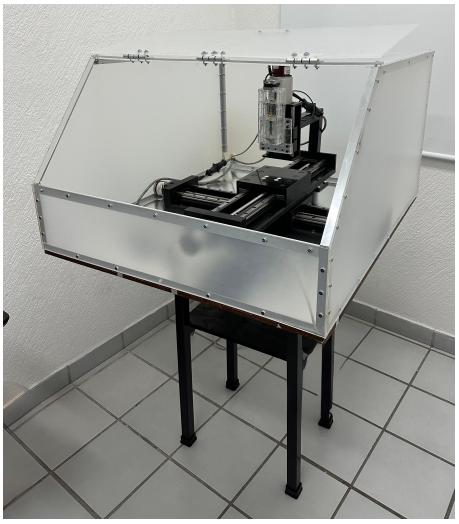


(b) Vista lateral superior del sistema en proceso

Fig 3.26: Sistema de depósito armado en proceso



(a) Vista frontal del sistema armado completo



(b) Vista lateral superior del sistema armado completo

Fig 3.27: Sistema de depósito armado completo

el procesamiento de imágenes de la celda fotovoltaica, por lo que se conservará el protector para mitigar este efecto y garantizar un mejor rendimiento en el procesamiento de las imágenes.

Capítulo 4

Control del sistema

En esta fase del proyecto, se aborda el diseño, configuración y puesta en marcha del sistema, con el fin de llevar a cabo las diferentes tareas del sistema. Un aspecto fundamental de esta etapa es el diseño y ensamblaje de arneses eléctricos, los cuales son clave para establecer las conexiones electrónicas que integran todos los componentes del proyecto. Asimismo, se desarrollaron diferentes prototipos de PCB para conectar los diversos componentes que conforman el sistema y que serán controlados por un microcontrolador. Este proceso garantiza el correcto funcionamiento del sistema y facilita su mantenimiento.

Además, se detalla el desarrollo del procesamiento de imágenes utilizado para obtener el patrón de los bordes de los contactos en las celdas fotovoltaicas. Este avance fue logrado mediante métodos de análisis visual, cuyo resultado involucra el desempeño que va a tener el sistema.

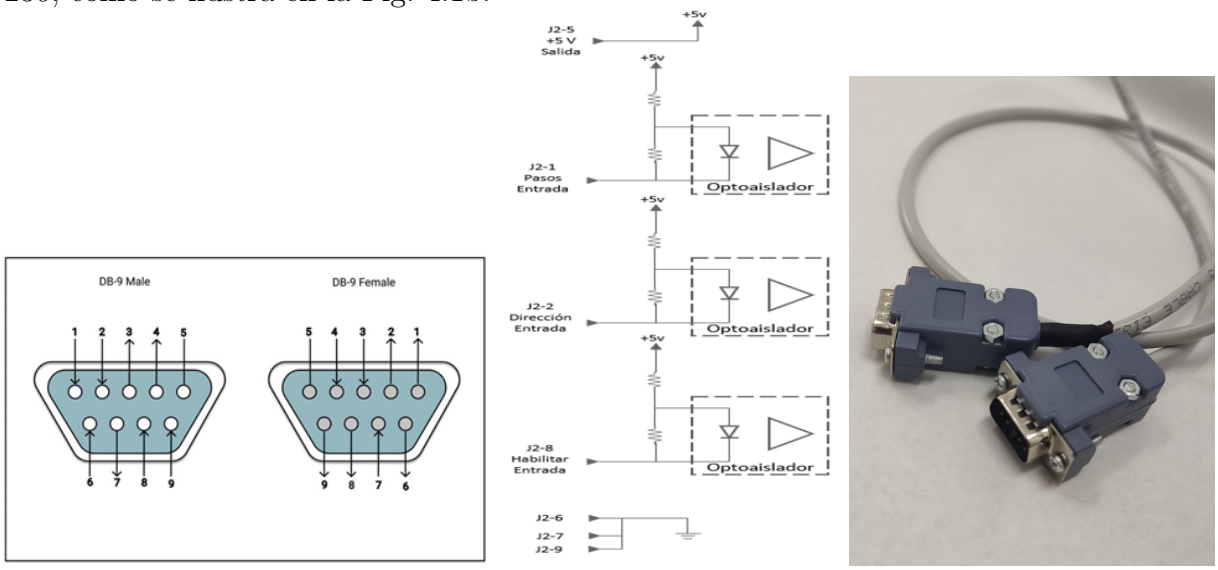
Finalmente, este capítulo aborda el diseño e implementación de una interfaz gráfica de usuario (GUI). Esta herramienta permite al usuario controlar de forma intuitiva el movimiento del sistema de depósito, cerrando así el ciclo de control del proyecto. Con estas acciones, se concluye un paso esencial hacia la integración total y la operación autónoma del sistema.

4.1. Conexiones

Un diseño y ensamblaje adecuado de los arneses eléctricos es fundamental, pues asegura el correcto funcionamiento del sistema. Los arneses no solo sirven como medio para conectar los diferentes componentes electrónicos en la PCB o en las fuentes de alimentación, sino que también son esenciales en la organización y protección de las conexiones dentro del sistema. En conjunto, un diseño eficiente de los arneses eléctricos contribuye a la durabilidad del sistema y a la seguridad durante su operación.

4.1.1. Interfaz del Controlador 5230

La conexión de la interfaz del controlador se realiza a través de un conector DB9, ampliamente utilizado en la comunicación serial RS232. Sin embargo, esto no implica que se emplee dicho protocolo de comunicación para enviar los comandos al controlador, el conector se utiliza exclusivamente para transmitir los pulsos y la dirección deseada al motor. En el diagrama de la Fig. 4.1a se detalla el orden de los pines del conector DB9, sirviendo como guía para la correcta conexión de los pines en la interfaz del controlador 5230, como se ilustra en la Fig. 4.1b.

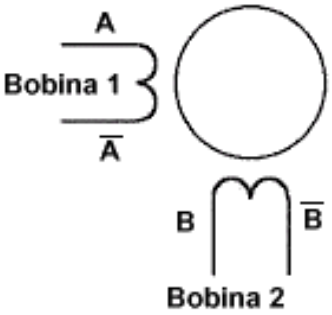


(a) Esquema de conexión de DB-9. (b) Esquema de la interfaz. (c) Cable con conexiones DB-9.

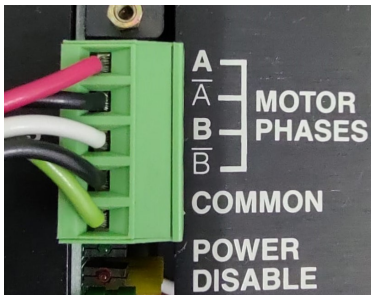
Fig 4.1: Conexión de la interfaz del controlador 5230.

4.1.2. Conexión del motor a pasos bipolar

Los motores a pasos bipolares requieren la aplicación de impulsos en una secuencia específica para generar movimiento. Esta secuencia de pulsos es generada externamente mediante un controlador electrónico. Los motores bipolares se pueden hacer avanzar a frecuencias establecidas por el fabricante, en este caso es de 20 kHz, lo que les permite girar, tomando como referencia el controlador 5230 se estableció la tabla de la Fig. 4.2b que marca la conexión de las bobinas que tiene el motor, como se muestra en el esquema de un motor bipolar Fig.4.3a, con ello se puede hacer arrancar y detener en cualquier instante y en una posición determinada.



Motor	Pin
1	A
2	$\bar{A}$
3	B
4	$\bar{B}$
5	GND



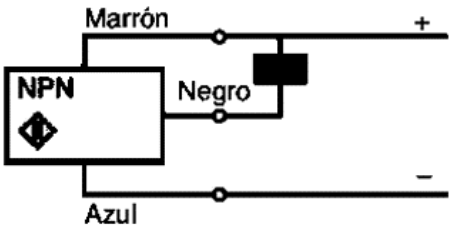
(a) Esquema de motor bipolar. (b) Tabla de conexión del motor. (c) Conexión de motor al controlador.

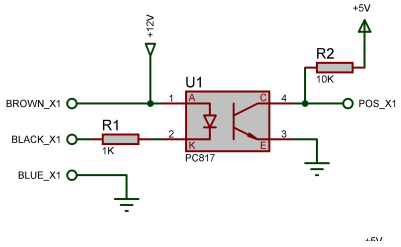
Fig 4.2: Conexión del motor a pasos bipolar.

4.1.3. Conexión del sensor de final de carrera

Para realizar el acoplamiento de señal que tiene el sensor de final de carrera con la parte de control del sistema que se le accionará la tarea a un microcontrolador, se tiene que realizar una etapa previa para que no afecte al dispositivo electrónico y éste pueda leer la señal que tenga el sensor, por lo cual, con las características que tiene el sensor se realizó un circuito de aislamiento con un optoacoplador, que funciona como un interruptor activado mediante la luz emitida por un diodo que satura un componente opto electrónico, en este caso un fototransistor.

Dado el caso que el sensor GXL-8 tiene una salida normalmente abierta de tipo NPN, es decir, todo el tiempo el sensor va a mandar una señal en alto (12 v), hasta que el sensor detecte un objeto va a mandar una señal en bajo (0 v), por lo cual se diseñó el siguiente esquemático que se encuentra en la Fig. 4.3b que muestra que el diodo del optoacoplador va a encender solo cuando la señal de salida este en bajo, después este activa el fototransistor que mandará una señal que sea adecuada para que no afecte al microcontrolador.





(a) Esquema de sensor GXL-8 type. (b) Circuito de acoplamiento del sensor NPN.

Fig 4.3: Conexión del sensor de final de carrera.

4.2. Diseño del circuito electrónico

Una vez definidas las conexiones del controlador, se procedió al diseño de un circuito electrónico que integrara todas las conexiones necesarias para garantizar el correcto funcionamiento del sistema. Este circuito incluye la generación de un tren de pulsos, realizado mediante una tarjeta de desarrollo ESP32 de la marca Espressif, seleccionada por sus características que son las siguientes:

Características de la tarjeta de desarrollo Esp32 devkit V1

- Microprocesador Tensilica LX6 a 240MHz de doble núcleo.
- Wifi y Bluetooth.
- Memoria Flash de 4MB.
- Oscilador a 32kHz.
- Dimensiones: 51mm×23mm×8mm sin pines.
- Peso: 0.007kg.

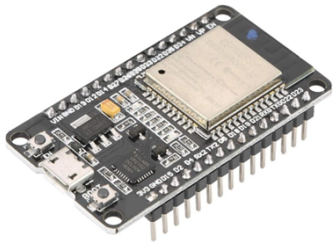


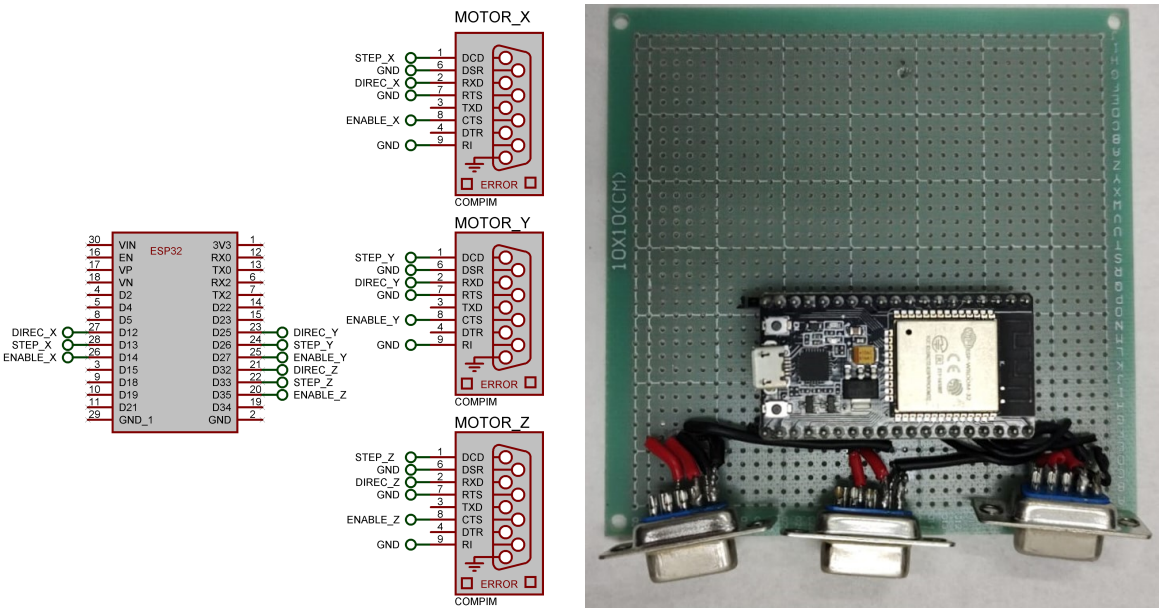
Fig 4.4: Tarjeta de desarrollo Esp32 Devkit V1.

Además, el circuito incorpora la configuración de los sensores de final de carrera, elementos para la seguridad del sistema. Estos sensores están conectados a los pines de interrupción de la ESP32, aprovechando su capacidad de respuesta rápida. Su función principal es notificar al microcontrolador de manera inmediata cuando el sistema excede las áreas de trabajo definidas, evitando daños al equipo, pues éste puede apagar o redireccionar el movimiento del sistema.

4.2.1. Prototipo V1

La primera versión del diseño del circuito electrónico para controlar el movimiento de los motores a pasos se diseñó utilizando una placa fenólica perforada. Este enfoque permitió crear un prototipo funcional que facilitara las conexiones con los conectores DB9, los cuales no podían integrarse en una tabla experimental (protoboard) debido a sus dimensiones, ya que solo es posible soldarlos mediante cables, como se ilustra en la Fig.4.5b .

En esta versión inicial, no se incluyó la etapa de acoplamiento de señal para el sensor de final de carrera, ya que el objetivo principal era centrarse exclusivamente en el control del movimiento del motor, por lo que en el esquema de la Fig.4.5a solo se muestran las conexiones necesarias para controlar el motor a pasos. Este enfoque permitió avanzar de manera escalonada en el desarrollo del sistema.



(a) Esquema de circuito electrónico de primer prototipo.

(b) Primer prototipo.

Fig 4.5: Prototipo V1.

4.2.2. Prototipo V2

El diseño del circuito electrónico se desarrolló utilizando Flux, una plataforma en línea disponible en www.flux.ai. Este programa es ampliamente utilizado en la comunidad electrónica, destaca por ser de código abierto, y también ofrece opciones premium para proteger proyectos electrónicos privados. Flux está consolidándose como una herramienta esencial debido a su gran variedad de bibliotecas de componentes electrónicos y su capacidad para generar diseños 3D realistas, como se muestra en la Fig. 4.7b. Estas características facilitaron el diseño y permitieron visualizar el proyecto antes de la fabricación.

Tras finalizar el diseño del PCB, se utilizó el servicio de JLCPCB, reconocido por su calidad, para imprimir las pistas del circuito y realizar las perforaciones necesarias para las terminales. Esta empresa te da la opción de hacer tus diseños con dos capas al mismo precio de hacerlo todo en la misma capa, esta ventaja ayudó mucho a reducir el tamaño de la PCB. El resultado final fue una PCB profesional y funcional, como se observa en la Fig. 4.7c, con todos los componentes soldados de forma adecuada y lista para integrarse al sistema. Este logro marcó un paso importante hacia el desarrollo del proyecto.

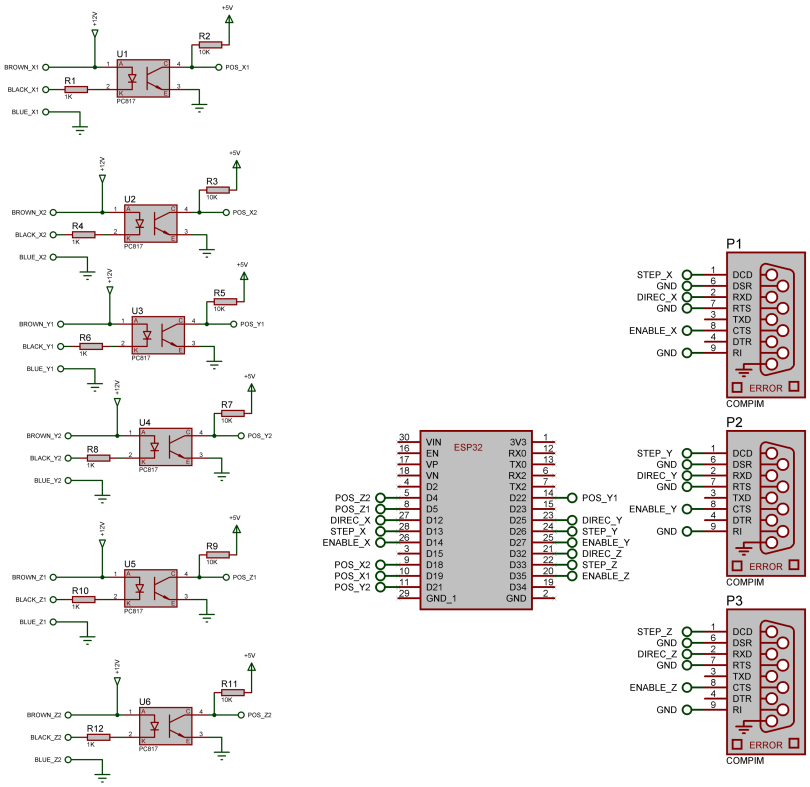
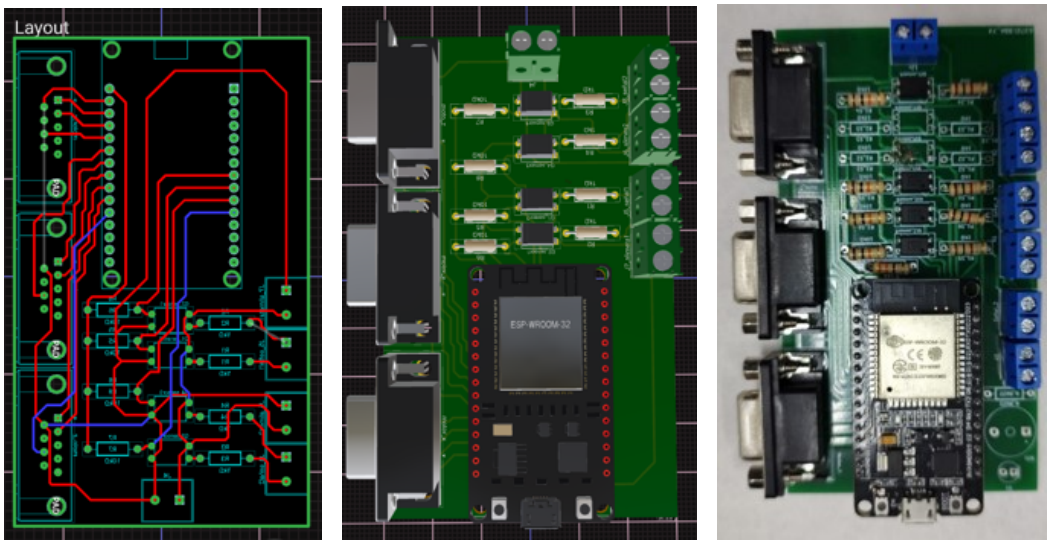


Fig 4.6: Esquema de circuito electrónico del segundo prototipo.



(a) Diseño de pistas del circuito electrónico.

(b) Diseño 3D del circuito electrónico.

(c) Segundo Prototipo.

Fig 4.7: Prototipo V2.

4.3. Implementación del circuito electrónico final

En el producto final se integraron los dos prototipos previos desarrollados durante la fase de diseño del circuito electrónico para el proyecto. En esta etapa final, se implementó el control del motor NEMA17 mediante el controlador A4988. Además, se incorporaron dos sensores adicionales para delimitar el área de movimiento del sistema de depósito, asegurando que no exceda los límites establecidos y evitando posibles colisiones con otros mecanismos que se encuentren a su alrededor.

Otra mejora añadida fue agregar un transistor PNP para activar las entradas ENABLE del controlador 5230. Esto permite optimizar el uso de las entradas de la ESP32, dejando más disponibles para gestionar las interrupciones de los sensores. Como parte de las mejoras en la conectividad y usabilidad, se decidió cambiar el tipo de conectores, utilizando clemas de conexión rápida para que el usuario pueda conectar y desconectar los cables de forma sencilla y sin problemas en futuras conexiones. Estos conectores pueden observarse en la Fig. 4.8.

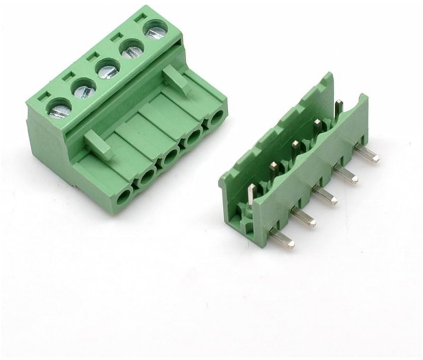
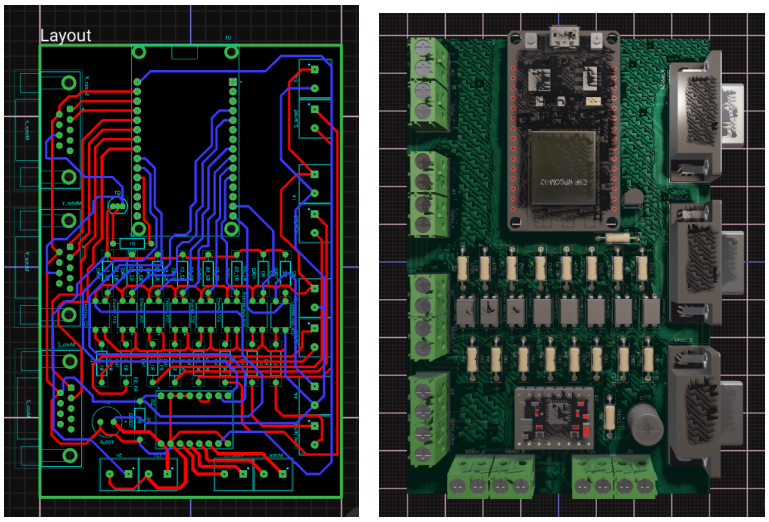


Fig 4.8: Bloque de terminales 5.08mm.

El circuito final del proyecto se diseñó siguiendo las características del segundo prototipo, utilizando la plataforma en línea Flux.ai para desarrollar el esquema electrónico y obtener un renderizado en 3D, como se muestra en la Fig.4.29b. Este circuito cuenta con una estructura de doble capa, lo que facilitó la conexión entre el microcontrolador y los periféricos. Esta configuración fue fundamental para lograr un diseño compacto y eficiente, que se muestra a continuación en Fig.4.29a

La fabricación del PCB se realizó a través de la empresa JLCPCB, cuyo resultado final se muestra en la Fig. 4.10. Al optar por una empresa especializada, se logró obtener un PCB de alta calidad y precisión, lo cual fue fundamental para asegurar la confiabilidad y durabilidad del circuito en condiciones de uso. Además, la impresión profesional facilitó el ensamblaje y soldado de los componentes necesarios para controlar el movimiento del sistema del proyecto mediante la ESP32, asegurando una mejor integración y reduciendo los posibles errores en la conexión de los componentes.



(a) Diseño de pistas del circuito de la PCB final.

(b) Diseño 3D del circuito final

Fig 4.9: Prototipo V2.

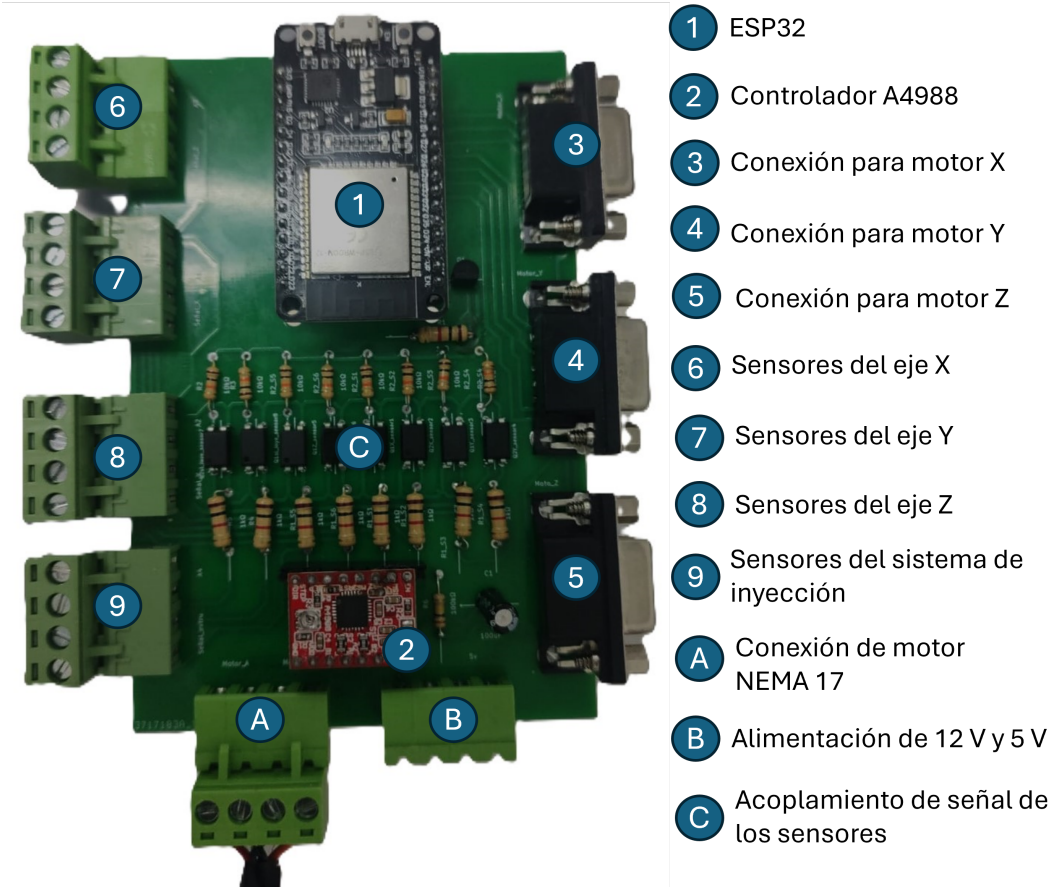


Fig 4.10: PCB con los componentes electrónicos soldados.

4.4. Control de Motores

Para implementar el movimiento de los motores, se tomó como referencia los manuales de los controladores 5230, DM860 y A4988, los cuales especifican la forma de conectar los motores bipolares al controlador. A su vez, detallan el rango de frecuencia de operación y las distintas configuraciones para los pasos de los motores. Basado en estas especificaciones, se integró la placa PCB diseñada que incluye todo el circuito electrónico y los periféricos necesarios, como se ilustra en la Fig. 4.11, se muestran los elementos requeridos para el movimiento de los motores. Entre estos, se incluye una fuente de 12 V y un regulador que proporciona 5 V, utilizados para alimentar el motor NEMA 17. Para el motor controlado por el DM860, se empleó la fuente de alimentación integrada en los controladores 5230. De esta manera, se asegura la alimentación para todos los motores.

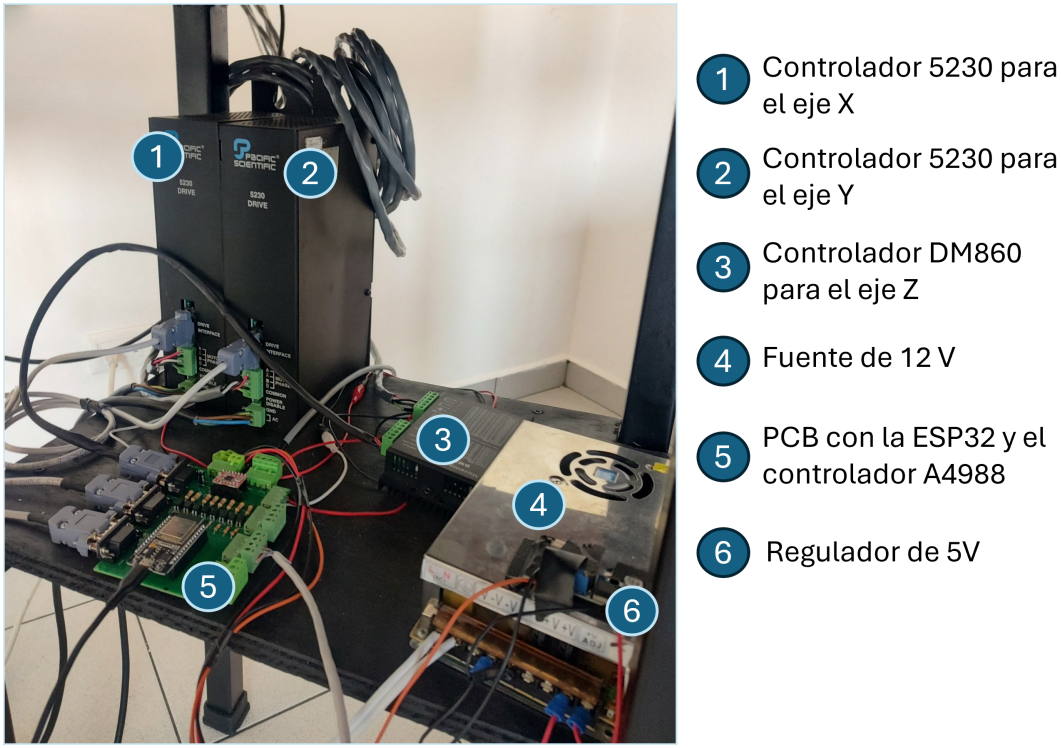


Fig 4.11: Elementos requeridos para el movimiento de los motores.

Se diseñaron los diagramas de flujo para guiar el funcionamiento del sistema, los cuales se presentan en la Fig. 4.12 y la Fig. 4.13. Estos diagramas representan la lógica del control del movimiento, destacando los pasos necesarios para configurar el microcontrolador y las interacciones entre sus entradas y salidas clave.

El proceso comienza con la declaración de las entradas y salidas que utilizará el microcontrolador. Las salidas incluyen la dirección de giro, los pulsos y el habilitador, mientras que las entradas corresponden a la señal proveniente del sensor de final de carrera. Posteriormente, se definieron las variables, enfocándose principalmente en aquellas relacionadas

con el tiempo y los pasos.

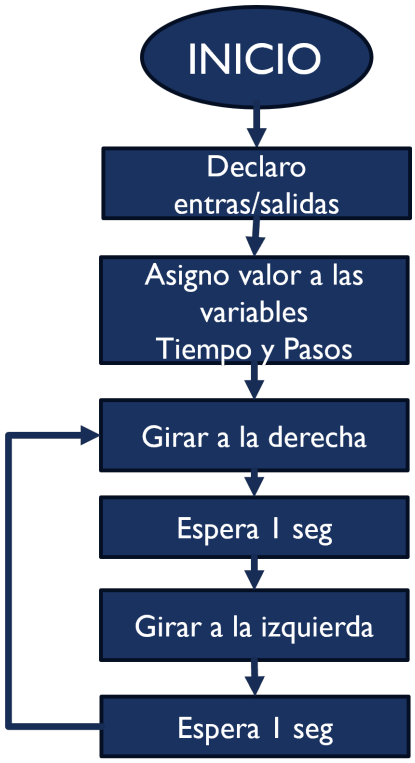


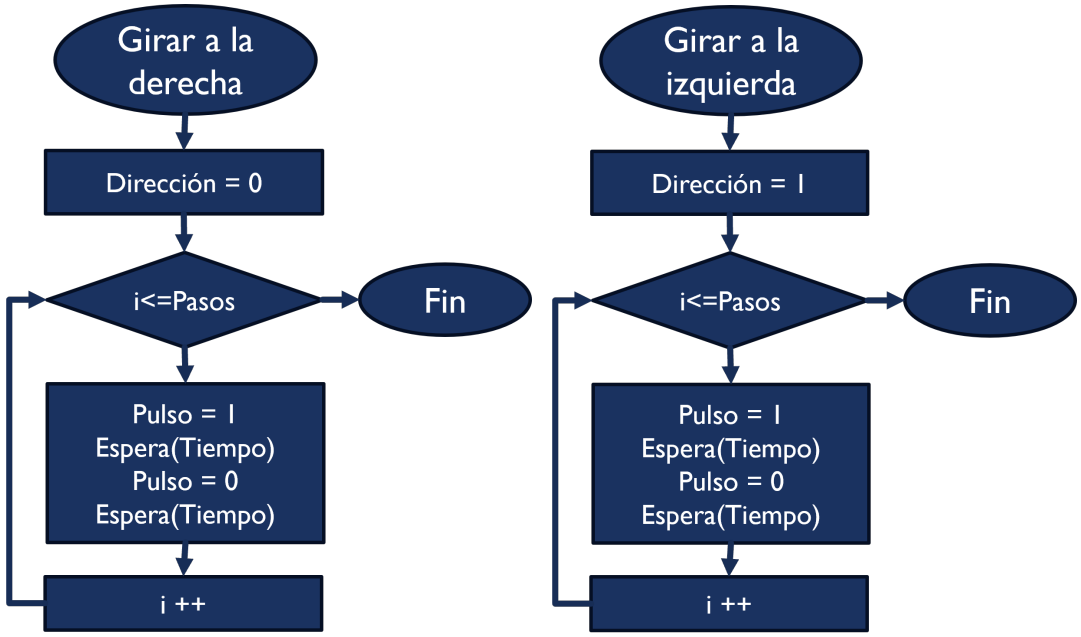
Fig 4.12: Diagrama de flujo para el control del motor a pasos.

La variable del tiempo almacena la duración de los estados alto y bajo del pulso, garantizando una señal cuadrada simétrica. Por su parte, la variable de pasos define la cantidad de movimientos que realizará el motor. El programa sigue un ciclo continuo en el que, primero, ejecuta la función "Girar a la derecha", espera un segundo, y luego invoca la función Girar a la izquierda. Estas funciones se realizan de manera consecutiva, formando un bucle infinito que se mantiene activo hasta que el microcontrolador se desconecta de su fuente de alimentación. La función "Girar a la derecha", representada en la Fig. 4.13a, asigna un valor de cero a la salida Dirección, lo que indica al controlador que active las bobinas del motor en la secuencia correspondiente para desplazar el carro lineal hacia la derecha. Dentro de esta función, un bucle for recorre un número de iteraciones igual al valor de la variable de pasos. En cada iteración, se genera un estado alto y bajo en la salida de pulsos, cuya duración está determinada por el valor de la variable tiempo. De manera similar, la función Girar a la izquierda, descrita en el diagrama de flujo de la Fig. 4.13b, realiza el mismo procedimiento, pero asignando un valor de uno a la salida Dirección. Este cambio ajusta la secuencia de energización de las bobinas, provocando que el carro lineal se desplace hacia la izquierda.

El mismo enfoque lógico se aplicó al motor NEMA 17 y su controlador correspondiente, adaptándolo para optimizar su desempeño en movimiento. Además, se aprovecharon las experiencias previas con los motores NEMA 23, integrando los trabajos ya hechos para mejorar el funcionamiento en este nuevo motor a pasos.

4.5. Movimiento del sistema

Se llevó a cabo la caracterización del sistema controlando el movimiento de los motores. Para ello, se tomaron medidas con diferentes cantidades de pasos, con el objetivo de determinar la unidad más pequeña que el sistema puede realizar. Aunque un método sencillo sería pedir al driver que mueva el motor un solo paso, esto resultaría muy difícil de medir



(a) Diagrama de flujo de la función Girar a la derecha.

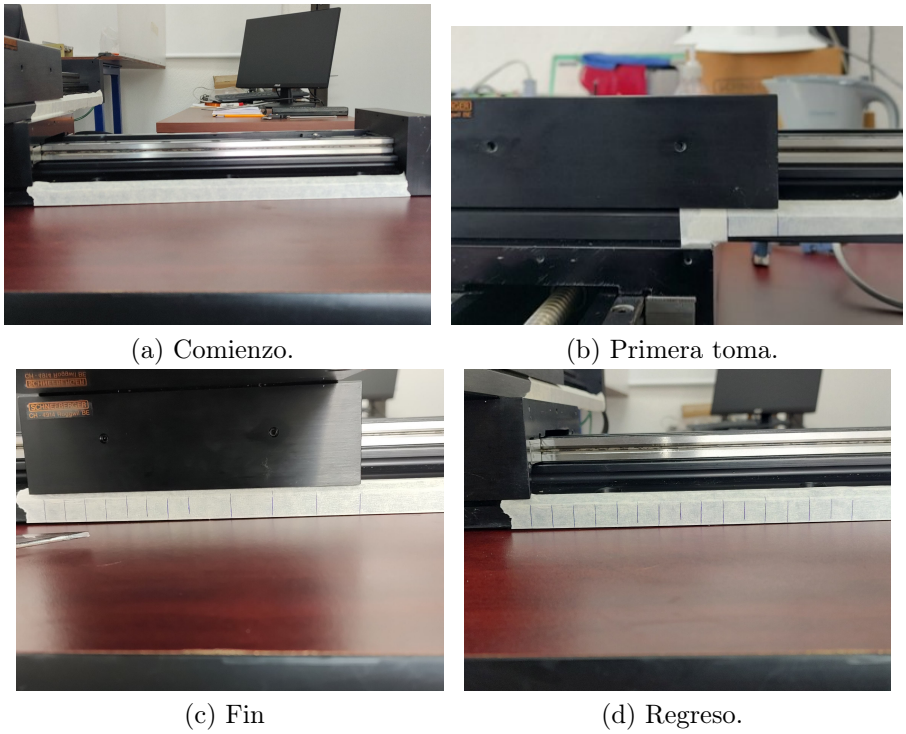
(b) Diagrama de flujo de la función Girar a la izquierda.

Fig 4.13: Diagrama de flujo de las funciones.

con un vernier u otro instrumento de medición. Por ello, se optó por un enfoque similar pero más práctico: en lugar de mover un solo paso, se solicitó al sistema que ejecutara una cantidad mayor de pasos, para luego analizar el resultado.

En este proceso se comenzó con cantidades pequeñas, en el orden de las decenas, pero estas seguían siendo muy difíciles de medir. Por ello, se optó por trabajar con cantidades mayores, en el rango de las milésimas. Se inició con 1000 pasos y se repitió este movimiento 7 veces consecutivas. Al finalizar la última repetición, se solicitó al driver que retrocediera la misma cantidad de pasos, con el objetivo de verificar la repetibilidad y exactitud mediante operaciones opuestas. Este procedimiento se repitió en los demás casos, hasta que el sistema regresó al punto de origen. Esto puede observarse en las figuras siguientes: la Fig. 4.14a muestra el inicio, la Fig. 4.14b representa la primera medición, la Fig. 4.14c corresponde al final del recorrido, y la Fig. 4.14d ilustra el retorno al origen del sistema.

El procedimiento anterior se repitió con pasos de 2000, 3000, 4000 y de 5000 con el objetivo de ver el movimiento del sistema con diferentes cantidades con múltiplos de mil para que así consideran en las medidas marcadas anteriormente como se observa en la Fig. 4.14 y se obtuvo una tabla con las diferentes medidas obtenidas como se tiene en la Tabla 4.1, que representa las medidas del sistema Y en milímetros y la Tabla 4.2 contiene las medidas del sistema X en milímetros, con estas mediciones se podrá calcular el paso mínimo que puede realizar el motor, con la ecuación (4.1) que representa la media, después



(a) Comienzo. (b) Primera toma.

(c) Fin (d) Regreso.

Fig 4.14: Secuencia de toma de medidas de 1000 pasos.

se vuelve a obtener la media de los resultados obtenidos con la misma ecuación (4.1) y como resultado final se obtuvo que la medida de un paso es de 0.01265 milímetros para los dos sistemas XY.

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N} \tag{4.1}$$

Dónde:
N es el número de medidas
X_i representa la i-ésima medida
$\bar{X}$ es la media de las medidas

Con el sistema Y se realizó el proceso de obtener la medida de un paso del motor, de una forma similar al proceso anterior, tan solo que este sistema ocupa un tornillo sin fin lo que provoca que el movimiento de un paso sea mayor a comparación del movimiento de un paso con un tornillo de bola, para ello sé probo con pequeños paso y se observó que con 100 pasos se podía contener una medida, pero era muy pequeña entonces se tomaron medidas con múltiplos de doscientos,cada medida se hizo con marcas de 200 pasos, como se observa en la Fig. 4.16a es el comienzo, Fig. 4.16b la mitad del trayecto y Fig. 4.16c es el final de la trayectoria, se capturaron todas la medidas y se representaron en la Tabla

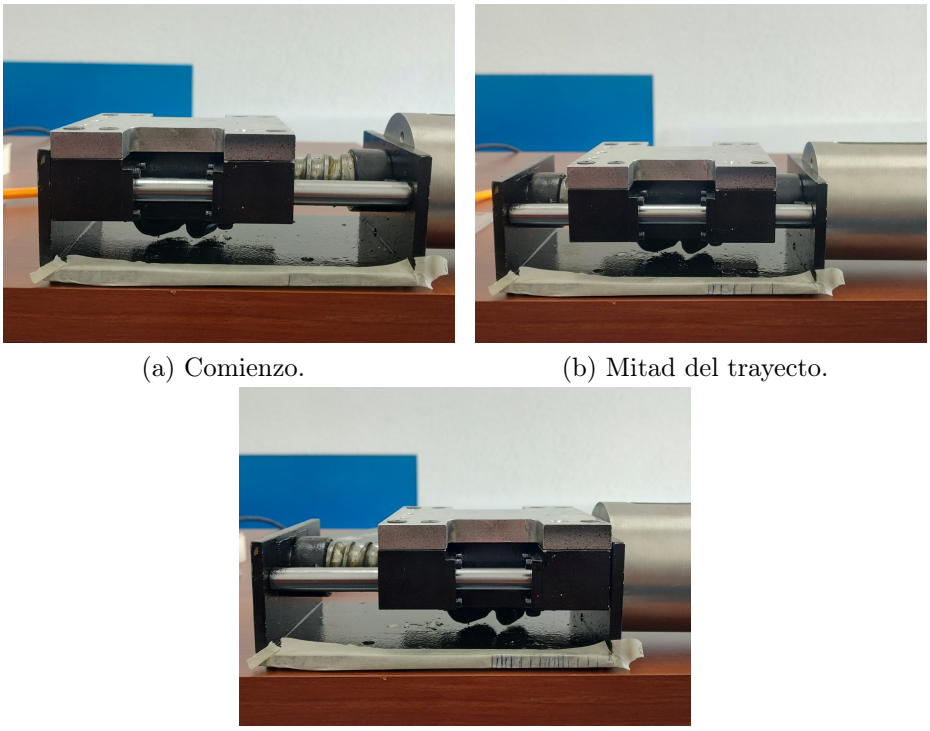
Tabla 4.1: Tabla de medidas del sistema Y en milímetros (mm).

Tomas	1000 pasos	2000 pasos	3000 pasos	4000 pasos	5000 pasos
1	11.53	25.58	37.46	51.27	63.67
2	11.89	25.30	38.11	50.46	63.00
3	13.03	25.35	38.81	51.58	64.15
4	12.99	25.19	38.89	51.09	63.48
5	12.83	25.11	37.99	50.80	63.51
6	12.51	25.57	38.00	50.68	63.26
7	12.80	25.05	38.56	50.66	63.30
8	13.17	25.43	37.87	50.59	63.37
9	12.51	25.08	37.95	50.51	63.13
10	12.44	25.00	37.66	50.77	63.36
11	12.64	25.13	37.54	50.12	63.50
12	12.49	25.17	38.04	50.10	63.57
13	12.54	25.97	37.85	50.45	63.02
14	12.25	25.26	37.69	51.00	63.44
15	12.28	25.01	37.92	50.20	63.61
Sumatoria	187.9	379.2	570.34	760.28	951.37
Media	12.52666667	25.28	38.02266667	50.68533333	63.42466667
Paso	0.012526667	0.01264	0.012674222	0.012671333	0.012684933

Tabla 4.2: Tabla de medidas del sistema X en milímetros (mm).

Tomas	1000 pasos	2000 pasos	3000 pasos	4000 pasos	5000 pasos
1	12.30	25.15	38.00	51.25	62.82
2	12.51	25.19	38.43	50.57	62.85
3	12.36	25.21	37.14	51.22	63.33
4	13.08	24.84	37.56	50.81	62.74
5	11.56	24.66	37.25	49.61	63.74
6	12.89	25.13	37.54	50.60	63.34
7	12.79	25.59	37.94	50.76	63.11
8	12.53	25.48	37.87	49.77	63.60
9	12.49	24.71	37.91	51.47	63.67
10	12.17	25.70	38.00	50.68	63.84
11	11.99	25.10	37.86	51.00	63.80
12	12.57	25.05	37.50	50.55	63.44
13	12.75	24.69	37.83	50.58	63.48
14	12.28	24.86	37.52	50.86	63.13
15	12.23	25.53	37.67	50.44	63.50
Sumatoria	186.5	376.89	566.02	760.17	950.39
Media	12.43333333	25.126	37.73466667	50.678	63.35933333
Paso	0.012433333	0.012563	0.012578222	0.0126695	0.012671867

4.4 estas medidas están en milímetros. Posteriormente, se calculó la media con la ecuación (4.1) y se obtuvo el paso que es de 0.02392 milímetros.



(a) Comienzo. (b) Mitad del trayecto.

(c) Fin.

Fig 4.15: Secuencia de toma de medidas de 200 pasos.

Tabla 4.3: Tabla de medidas del sistema Y en milímetros (mm).

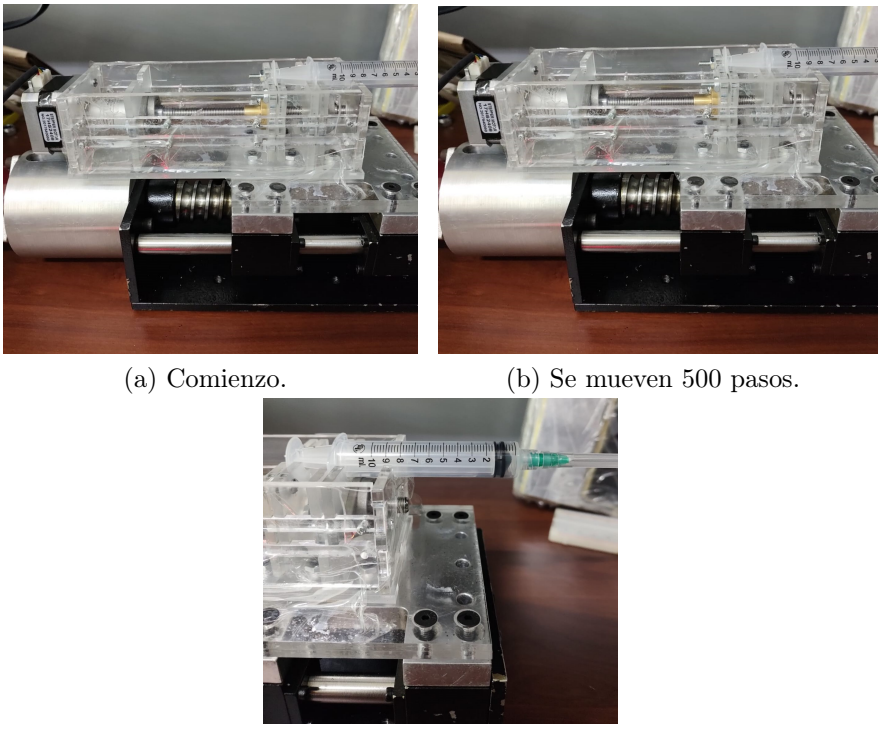
Tomas	100 pasos	200 pasos	400 pasos	600 pasos	800 pasos
1	2.48	5.00	10.12	15.07	19.35
2	2.73	5.00	9.46	14.26	19.28
3	2.44	4.87	9.16	14.46	18.80
4	2.25	4.85	9.75	14.13	18.59
5	2.43	4.53	9.45	14.31	18.39
6	2.64	4.57	9.54	14.09	19.13
7	2.35	4.80	9.70	14.61	19.47
8	2.10	4.59	9.41	14.10	19.01
9	2.10	4.60	9.01	14.15	19.43
10	2.74	5.02	9.58	14.67	18.63
11	2.49	4.65	9.47	14.22	18.42
Sumatoria	26.75	52.48	104.65	158.07	208.5
Media	2.43181818	4.77090909	9.51363636	14.37	18.95454545
Paso	0.02431818	0.02385455	0.023784091	0.02395	0.023693182

Finalmente, en el apartado de movimiento del sistema, se caracterizó el funcionamiento

del motor paso a paso para determinar la distancia mínima que el sistema puede recorrer con la jeringa. Para ello, se elaboró una tabla con los datos obtenidos, teniendo en cuenta el desgaste de las llantas, lo que permitió evaluar con precisión los desplazamientos mínimos posibles y optimizar el desempeño del sistema.

Tabla 4.4: Tabla de medidas del sistema de depósito en milímetros (mm).

Tomas	500 pasos	1000 pasos	2000 pasos
1	1.76	3.68	8.73
2	2.34	4.57	9.85
3	2.51	4.86	10.38
4	1.76	5.06	–
5	2.45	5.71	–
6	2.44	5.28	–
7	2.61	4.63	–
8	2.36	4.84	–
9	2.45	4.63	–
10	2.69	4.84	–
11	2.03	5.21	–
Sumatoria	28.17	42.84	29.96
Media	2.3475	4.76	9.9886
Paso	0.004695	0.00476	0.004993



(a) Comienzo.

(b) Se mueven 500 pasos.

(c) Distancia recorida por la jeringa.

Fig 4.16: Secuencia de toma de medidas de 500 pasos.

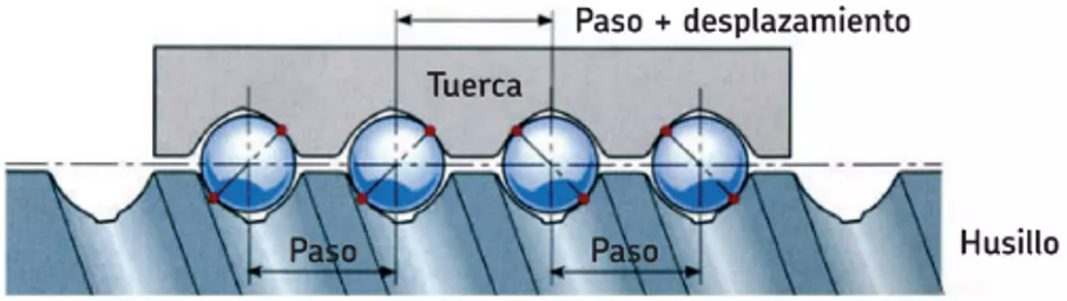


Fig 4.17: Partes que lo componen al husillo.

Ahora, con las medidas obtenidas mediante la caracterización del sistema, se verifican los resultados utilizando la ecuación (4.2) para determinar la distancia de un paso. Para entender esto, se analiza cómo el husillo cumple su función. En la Fig. 4.17, se ilustran las partes que lo componen [36].

$$\delta = \frac{P}{\theta} \tag{4.2}$$

Dónde:
δ Un paso del motor.
P El paso del husillo.
θ El número de pasos por una revolución

Tabla 4.5: Tabla resultados de distancia por un paso.

Tipo de motor	Eje	P	θ	δ	Caracterización
NEMA 23	X	200 Pasos	2.5 mm	0.0125 mm	0.012639 mm
	Y				0.012583 mm
	Z		4.76 mm	0.0238 mm	0.02392 mm
NEMA 17	Depósito		1mm	0.005 mm	0.00481 mm

En la Tabla 4.5 se muestran los resultados obtenidos con la ecuación (4.2) y con las variables que representan a cada eje de movimiento, con el resultado de la caracterización del sistema.

4.6. Procesamiento de imagen

Para desarrollar esta estapa del proyecto se realizaron diferentes partes para tener un detector de bordes para la celda fotovoltaica, este proceso se representa en el diagrama de la Fig. 4.18, el cual se dividió en 3 partes que fueron los siguientes: la elección de la cámara, la ecualización de la imagen y la elección del detector de bordes.

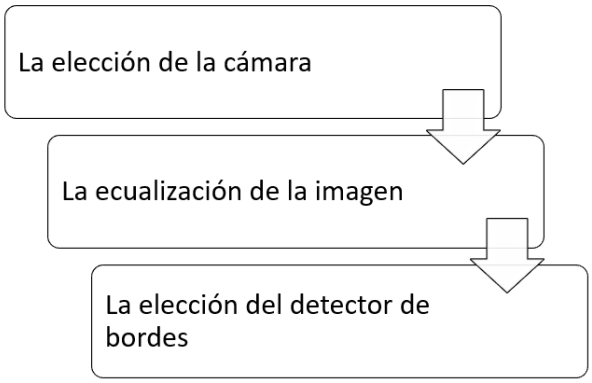


Fig 4.18: Diagrama de bloques para realizar el trabajo del procesamiento de imagen.

4.6.1. Elección de la cámara

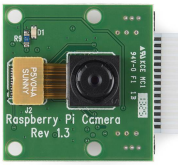
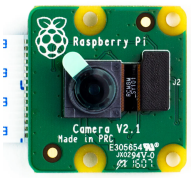
Para seleccionar la cámara adecuada, se realizaron diversas pruebas con los modelos listados en la Tabla 4.6, teniendo en cuenta algunas de sus características específicas.

Para evaluar las cámaras, se consideraron la resolución de imagen y el efecto focal del lente. Este último es especialmente relevante debido a la necesidad de detectar los patrones de los contactos de la celda. Las distorsiones comunes de los lentes, como la distorsión en barril y la distorsión en cojín, generan curvaturas en las líneas que deberían ser rectas. Estas distorsiones pueden generar ruido y afectar la interpretación de los resultados.

Para detectar tales distorsiones, se utilizó una hoja tamaño carta con varias líneas rojas, como se muestra en la Fig. 4.19. Este patrón ayudó a alinear de manera centrada la celda fotovoltaica, permitiendo verificar si los contornos de la imagen presentaban alguna curvatura en las líneas.

Se diseñó un diagrama de trabajo como se muestra en Fig.5.3a,con el software de diseño SolidWorks®, mostrando los elementos necesarios para obtener la adquisición de imagen, como guía para poder armar un diseño que ayude a tomar las imágenes de la celda. Con esto se obtuvo un espacio de trabajo con los siguientes elementos: un tripié, un aro de luz, un soporte flexible para celular, el cual ayudó a colocar las diferentes cámaras y la hoja de referencia, como se observa en la Fig.5.3b.

Tabla 4.6: Información general de las cámaras.

Foto	Nombre	Sensor	Pixeles	MP
	Perfect Choice	—	1920 x 1080	+ 3 MP
	Cámara de Aliexpress	OV5647	2592 X 1944	5 MP
	Módulo V1.3	OV5647	2592 x 1944	5 MP
	Módulo V2.1	IMX219	3280 x 2464	8 MP
	Módulo V3	IMX708	4608 x 2592	12 MP

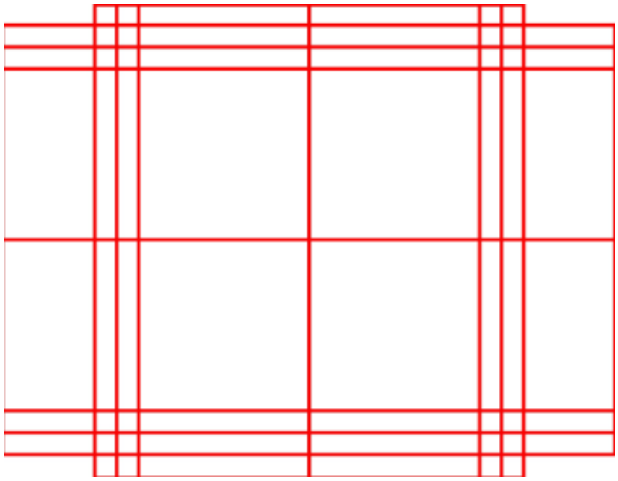


Fig 4.19: Hoja de referencia para la adquisición de imagen.

Se tomó en cuenta la distancia de la cámara a la hoja de referencia, con el criterio de que se muestre toda la hoja de referencia para evaluar su distorsión, a su vez se utilizó una tarjeta de desarrollo de la empresa Raspberry Pi, modelo Zero 2 W, con las siguientes características [37]:

Especificación de la Rasberry Pi Zero 2 W

- LAN inalámbrica 802.11 b/g/n,
- Bluetooth 4.2 de baja consumo energético (BLE)
- CPU de 4 nucleos Cortex-A53 de 64 bit a 1GHz
- 512 MB de RAM
- Puerto mini HDMI® y puerto micro USB On-The-Go (OTG)
- cuenta con cabezal de 40 pines compatible con HAT
- VideoCore IV
- Conector de cámara CSI.

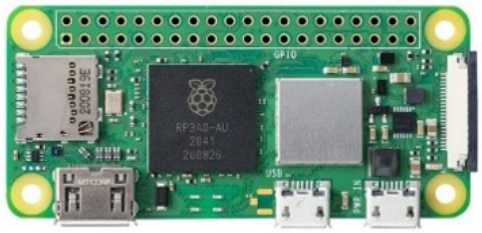
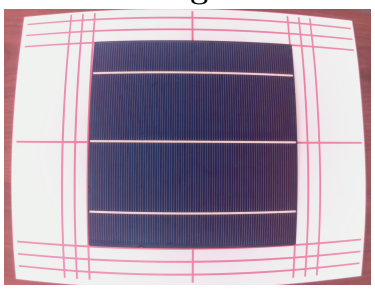
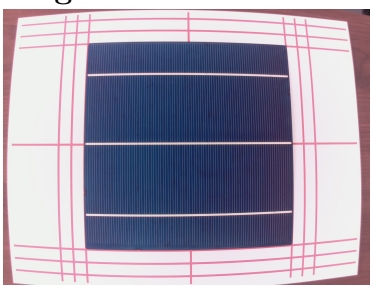
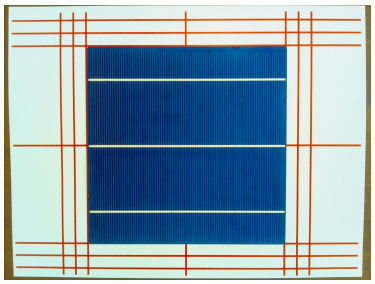
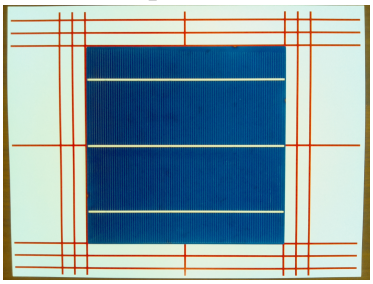
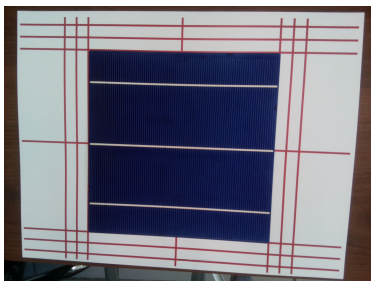
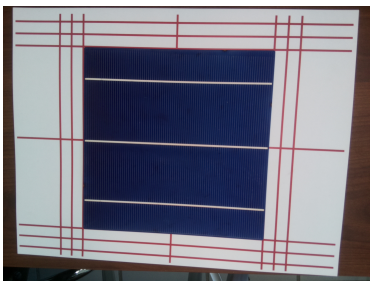
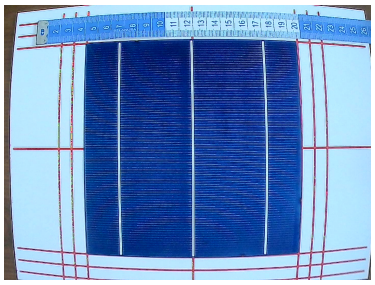
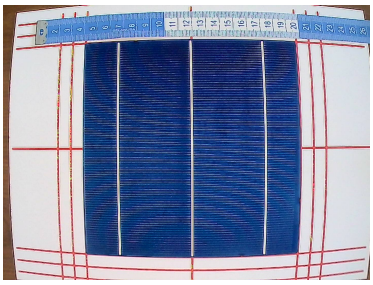
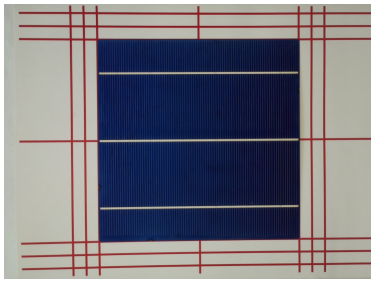
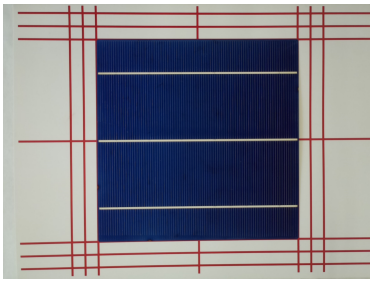
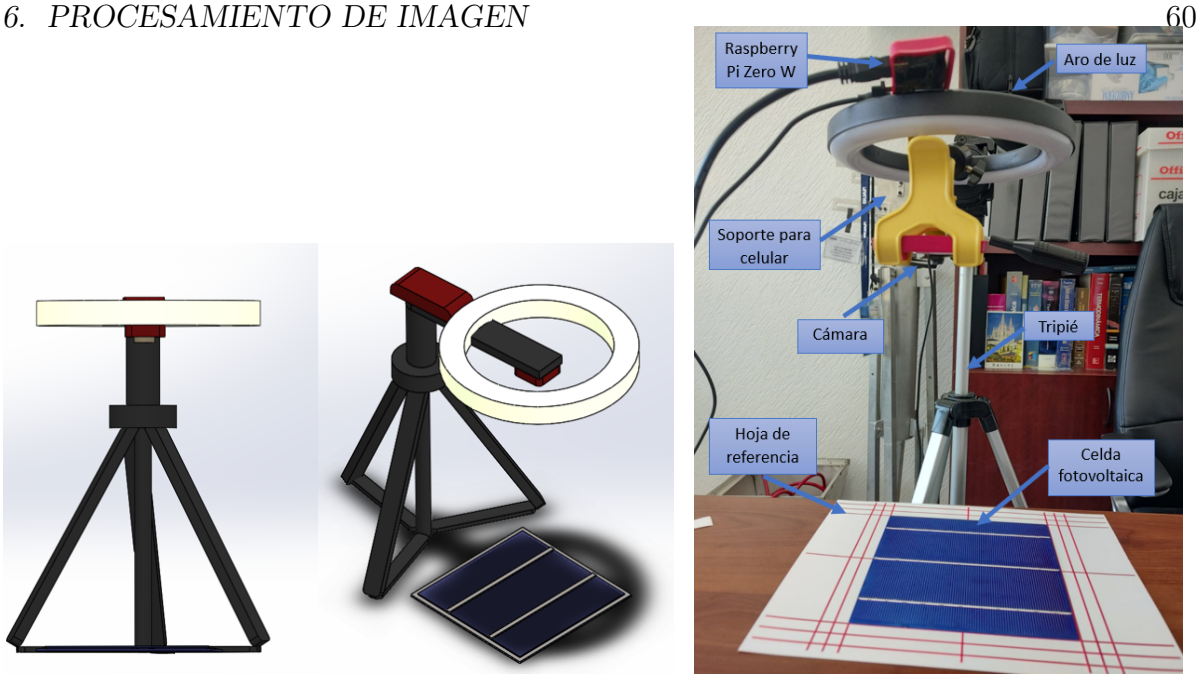


Fig 4.20: Rasberry Pi Zero 2 W

Para la elección de la cámara se tomó la resolución UXGA que se representa con 1600 x 1200 de ancho y altura correspondientemente, desafortunadamente, la cámara Perfect Choice no pudo alcazar esta resolución y se escogió una SXG que tiene una resolución de 1280 x 1024. Al comparar los resultados obtenidos, que se encuentran en la Tabla. 4.7, se observa que se puede descartar la cámara Perfect Choice, por tener una resolución menor, lo que limita tener una toma de foto en la celda solar con la misma resolución que las demás cámaras y que cuenta con una distorsión de barril en los contornos de la imagen.

Tabla 4.7: Tabla de Imágenes

Imagen	Imagen con iluminación	Altura
		**28 cm**
Cámara de Aliexpress		
		**28 cm**
Módulo de Cámara V1.3		
		**28 cm**
Módulo de Cámara V2.1		
		**28 cm**
Perfect Choice		
		**29 cm**
Módulo de Cámara V3		



(a) Diagrama de trabajo en SolidWorks®. (b) Resultado del diagrama de trabajo.

Fig 4.21: Espacio de trabajo

El segundo descartado sería la cámara de aliexpress de visión nocturna, al ser una cámara económica, se pudo tener imágenes con la misma resolución que la cámara Módulo V1.3, pues ambas son de 5 megapíxeles y con la ayuda de la luz pudo mejorar la visualización de los patrones de los contactos de la celda, este criterio solo se observaba a simple vista, pero también se eliminó porque representaba una distorsión en el contorno de la imagen debido a la forma de la lente de la cámara. Dicho esto, las tres opciones posibles son las versiones 1.3, 2.1 y 3 del módulo de cámara Raspberry, que no tienen problemas importantes con los bordes de la imagen en comparación con los dos anteriores.

4.6.2. La ecualización de la imagen

En esta segunda parte del trabajo, es mejorar la imagen antes de ocupar un detector de bordes, la forma en que se realizó esta parte se muestra en el diagrama de flujo representado en la Fig 4.22.

Las librerías que se ocuparon para el procesamiento de imagen, fueron de Opencv que ha tenido un desarrollo importante y tiene un libre acceso para realizar proyectos, se pueden realizar con diferentes tipos de lenguaje, como lo es en JavaScript, C++ y Python. En este trabajo se ocupó el lenguaje de Python, ya que también tiene un constante desarrollo y cuenta con una numerosa cantidad de librerías y funciones matemáticas que ayudarán a realizar el procesamiento de imagen.

OpenCv ayudó a poder insertar y visualizar la imagen, al ingresar la imagen, se tiene que convertir a escala de grises, con la finalidad de no tener muchos datos que procesar,

ya que una imagen de color cuenta con tres matrices una de color rojo, una color verde y otra de color azul, en cambio una imagen de escala de grises solo cuenta con una matriz, al realizar esta convención se creó el histograma de la imagen de escala de grises, esta gráfica indica la frecuencia que se repite ese valor de pixel en la imagen, posteriormente se aplicas una ecualización que puede ser un filtro Gaussiano, una ecualizado del histograma o un filtro pasa altas, de cada opción de ecualización se obtuvo su histograma, para poder comparar y tomar una decisión de cual sería el adecuado para la detección de bordes.

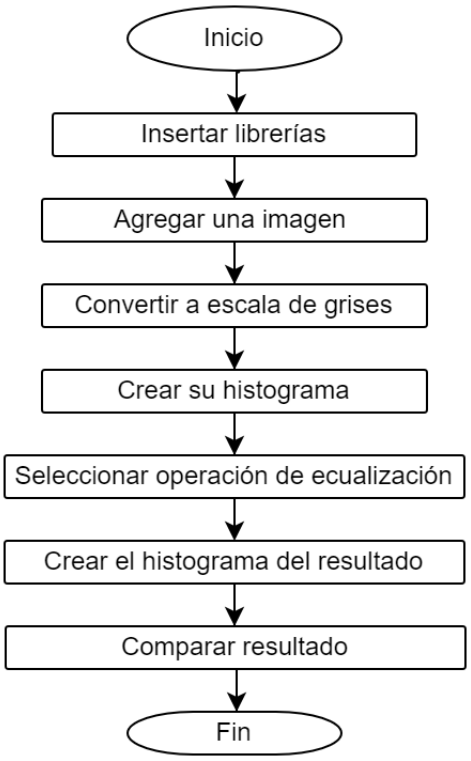


Fig 4.22: Diagrama de flujo de la ecualización de imagen.

El resultado de las ecualizaciones se tiene en la Tabla .4.8, donde se observa que la ecualización del histograma solo reparte todos los píxeles en el rango de la imagen, por lo cual ocasiona un aumento de brillo en la imagen, por lo que no se tomó en cuenta para la segunda parte, en cambio el filtro gaussiano se tenía que modificar la signa dependiendo de la imagen esta se tenía, en cambio el filtro pasa altas se escogió porque resalta los bordes contra el fondo de la imagen, esto ayudó para que se pudiera escoger un detector de bordes adecuado.

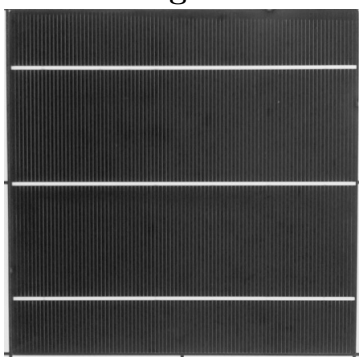
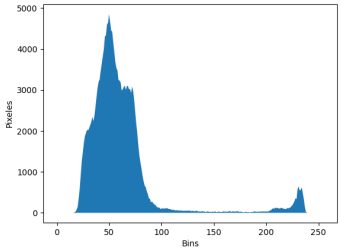
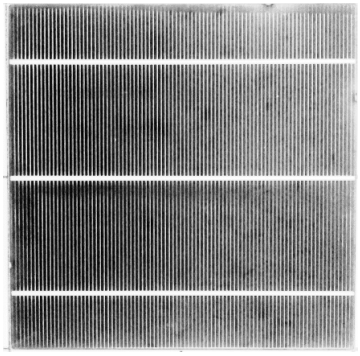
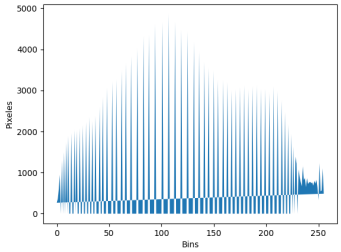
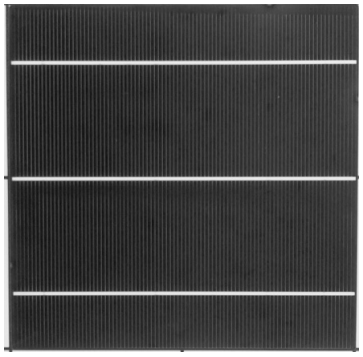
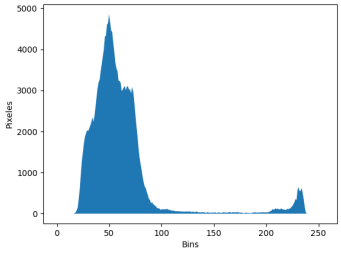
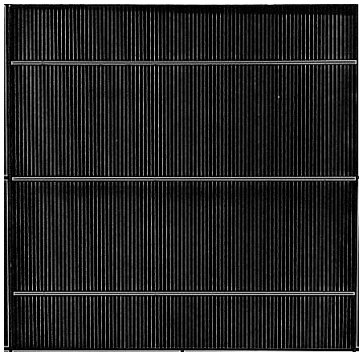
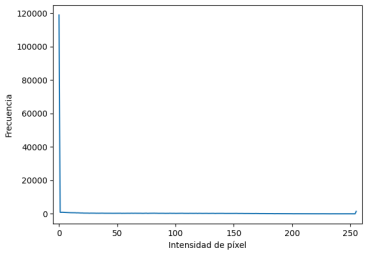
Tras varias pruebas se recurrió a realizar un ROI (Región de interés) que es una segmentación de la imagen en la que esta interesado en trabajar con eso pixeles, se le aplicó este método a la celda y la función de ecualización del histograma, que causó un problema porque esta función llena todo el histograma y esto ocasiona que ilumine zonas que afectarán en un futuro, para poder detectar los bordes, por lo que se descartará ocupar la ecualización del histograma y se ocupará el filtro pasa altas.

4.6.3. La elección de detector de bordes

En la detección de bordes se ocupó el módulo Camera V3, en que se realizaron varias tomas de fotos, con diferentes ángulos, resoluciones y modulación de luz, en el cual se ocupó la imagen de la Fig.4.23, aun así se realizó un ROI para poder tener un procesamiento de imagen específico.

En el ROI de la Fig.4.23 se realizó la ecualización del histograma, el filtro gaussiano

Tabla 4.8: Tabla de imágenes del módulo cámara V1.3.

Imagenes	Histogramas
	
Escala a grises	
	
Ecualizado del histograma	
	
Filtro Gaussiano	
	
Filtro pasa altas	

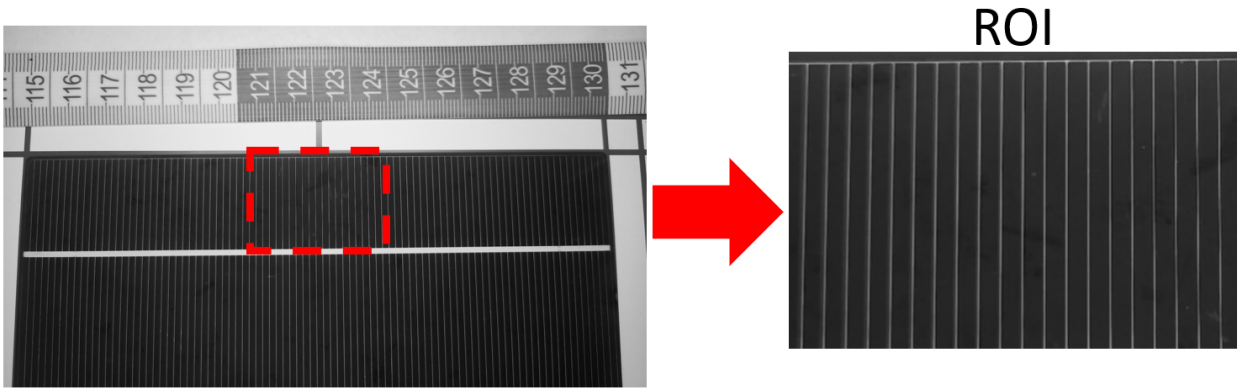


Fig 4.23: ROI de la celda. Autoría propia.

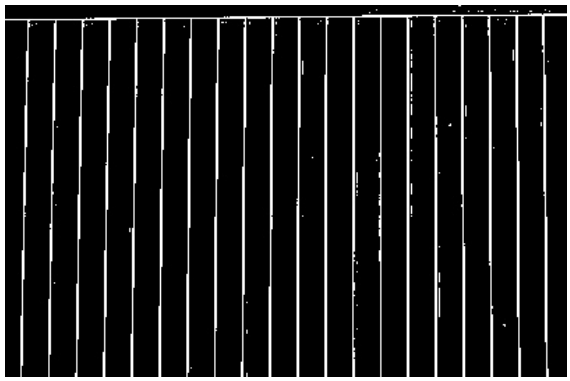
y el filtro pasa altas, posteriormente se analizaron varios tipos de detectores de bordes como el Prewitt, Roberts, Sobel, Canny, la umbralización y el método de Otsu, los cuales fueron analizados a través de su histograma y la relación con la imagen original, por lo cual se utilizó la umbraliazación y el método de Otsu. Posteriormente, se calculó el error cuadrático medio, que ayudó a comparar las diferencias entre el filtro pasa altas y los diferentes valores de umbralización que se propusieron.

4.7. Resultados

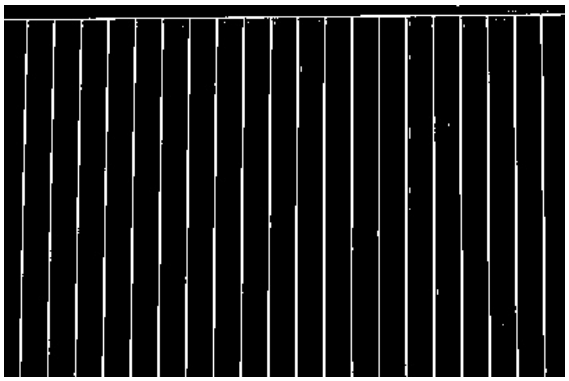
Para obtener los resultados de la tabla 4.10, se realizó primero el método de Otsu que dio un parámetro inicial de umbral, que se calculó con la ecuación (2.10), se obtuvo un umbral igual a 94, es decir que todos los píxeles que tuvieran un valor menor a 94 se convirtieron en 0 y los píxeles que tuvieron un valor más alto se convertían en 255, por ello en la tabla se compara el aumento o la disminución de los píxeles que tiene el valor de 0 o 255.

Para tener certeza de que el umbral obtenido es el adecuado, se realizó un barrido de umbrales de 50 a 120 y se calculó su error cuadrático medio con la ecuación (2.11), donde $I(i,j)$ es la imagen umbralizada y $K(i,j)$ es la imagen filtrada, este criterio se tomó ya que con la vista, no se puede dar un resultado cuantitativo real, en la tabla 4.9 que contiene los resultados del barrido de umbrales, no se puede notar una diferencia entre ellas, en cambio el MSE es la comparación cuantitativa del valor del píxel en la misma ubicación, si el error es igual a cero, entonces se tiene el mismo valor, por consiguiente se calcularon todo los MSE de cada umbral y se obtuvo que el umbral con menor error es de 74, esto se refiere a que el umbral más parecido a la ecualización de la imagen de la Fig 4.23 obtuvo un error

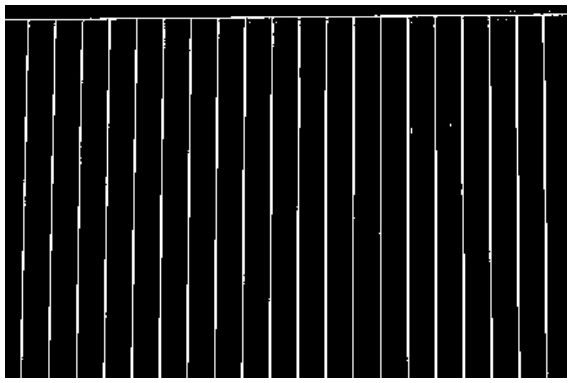
Tabla 4.9: Tabla de imágenes con diferentes valores umbrales.



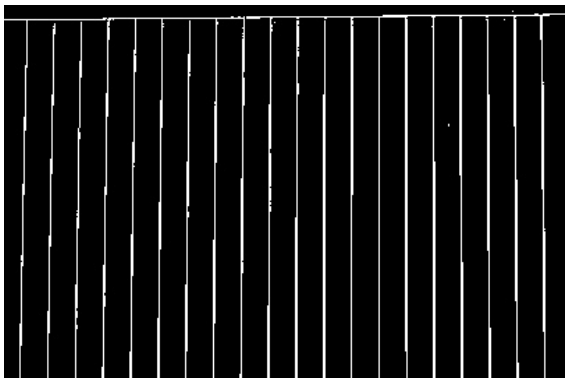
Umbral = 50



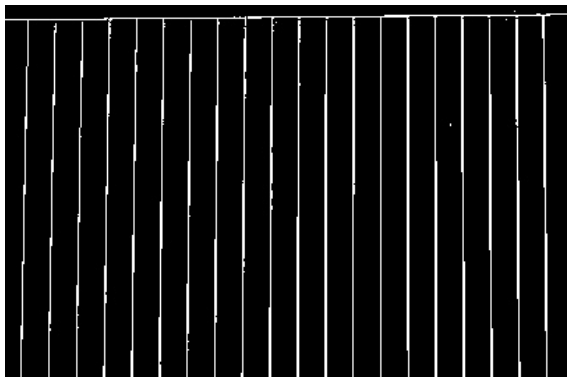
Umbral = 60



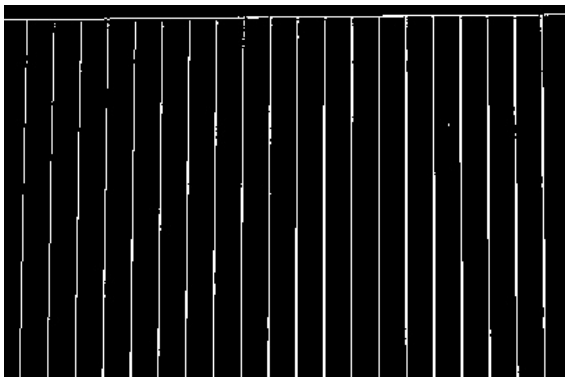
Umbral = 74



Umbral = 94



Umbral = 100



Umbral = 120

igual a 25.4088438, como se observa en la tabla 4.10 el error entre umbrales es cercano, debido a que se está ocupando el mismo ROI y el mismo método de ecualización, para ello se basó de la gráfica de la Fig. 4.24 que fundamentó el valor del error más pequeño entre los umbrales.

Tabla 4.10: Tabla del error cuadrático medio (MSE).

Umbral	Pixel = 0	Pixel = 255	MSE
50	98156	9445	25.5047351
60	98648	8953	25.4484438
70	98948	8653	25.4234068
72	99005	8596	25.4244942
73	99040	8561	25.4239366
74	99070	8531	25.4088438
75	99094	8507	25.4099125
76	99128	8473	25.4519382
80	99234	8367	25.4596426
90	99498	8103	25.49112
92	99556	8045	25.4771889
94	99600	8001	25.4724027
96	99656	7945	25.4752
98	99709	7892	25.4769101
100	99763	7838	25.4467152
110	100084	7517	25.5015288
120	100491	7110	25.5190286

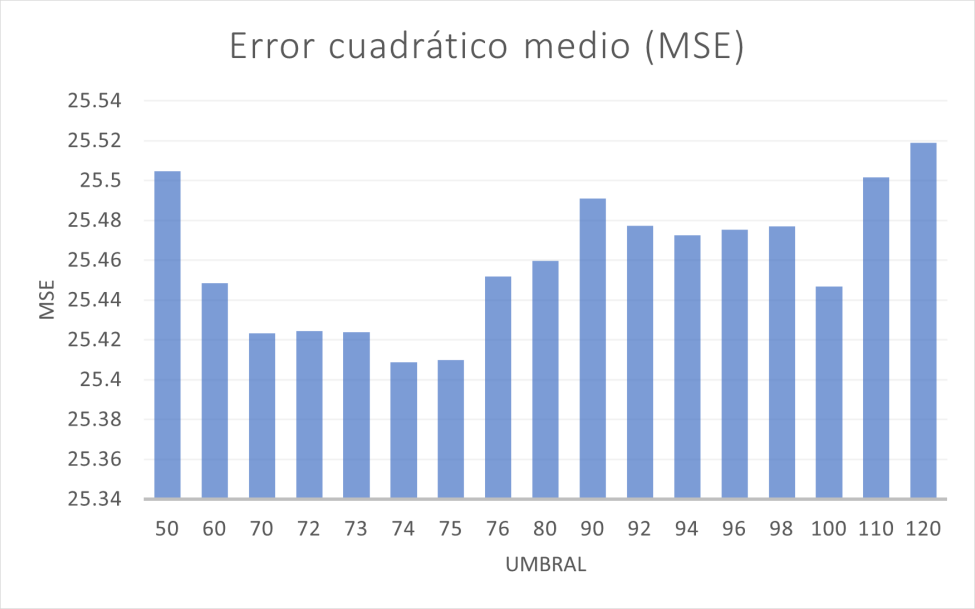


Fig 4.24: Gráfica del error cuadrático medio (MSE) del ROI de la Fig. 4.23.

Posteriormente se realizó una segmentación de diferentes ROI de la Fig. 4.23, en total fueron seis imágenes diferentes que pasaron por el mismo proceso que la primera segmentación y se obtuvieron los resultados mostrados en la Tabla 4.11

Tabla 4.11: Tabla segmentación de la Fig. 4.23.

ROI	Método de Otsu	MSE	Umbral	MSE
Superior izquierdo	98	22.4735551	100	22.4718544
Superior derecho	94	21.398997	74	21.2532297
Inferior izquierdo	103	20.7639241	80	20.693172
Inferior derecho	95	19.6838609	74	19.5772057
Superior central	94	25.4724027	74	25.4088438
Inferior central	120	16.5696127	112	16.5520953

Para finalizar, se hizo un ROI de las imágenes de la Tabla 4.7 y se realizó el mismo procedimiento de ecualización, con el filtro pasa altas y el método de Otsu para verificar el tipo de umbral, que se requiere para obtener los bordes de los patrones de los contactos de la celda, los resultados se muestran en la Tabla 4.12 en esta no se encuentran resultados de la cámara web Perfect Choice, porque no se pudo ecualizar la imagen con el filtro pasa altas, se tenía que realizar modificaciones en el espacio de trabajo para poder mejorar la adquisición de imagen de la celda.

Tabla 4.12: Tabla de resultados del método de Otus y la umbralización de las imágenes de la Tabla 4.7.

Camara	Método de Otsu	MSE	Umbral	MSE
Aire express	98	42.9773397	88	42.833083
Modulo v1.3	44	59.2603194	42	59.2017468
Modulo v2.1	26	55.0473856	26	55.0473856
Modulo v3	94	21.398997	74	21.2532297

4.8. Dimensiones del píxel

La dimensión de un pixel tiene un valor establecido, para definirlo hay que recordar que las imágenes son matrices, como lo representa la Fig. 4.25, que muestra que cada píxel es un valor de la matriz, por lo cual también tiene un espacio en ella.

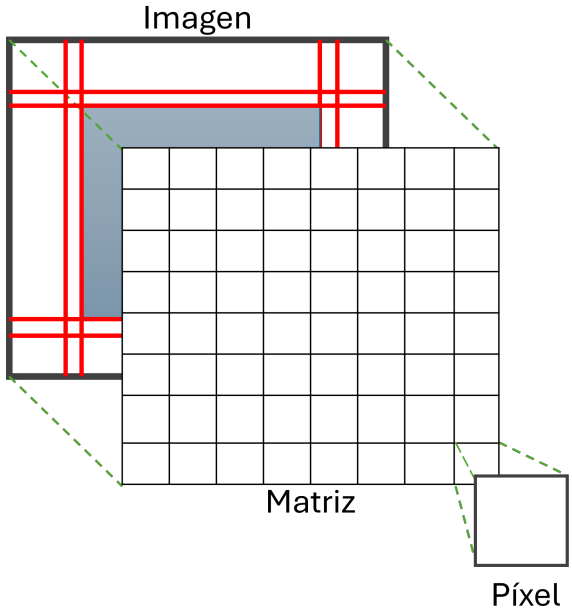


Fig 4.25: Imagen, Matriz y píxel.

Al obtener las dimensiones se tiene que considerar la altura donde está colocada la cámara y el ángulo que esta tiene, como se muestra en el esquema de la Fig.4.26, en donde se observan las diferentes variables que se involucran para tener el dimensionamiento de la imagen.

Estas variables dependen totalmente de la altura de la cámara, y lo que se pretende realizar es parametrizar los demás valores, para ello se siguieron los siguientes pasos.

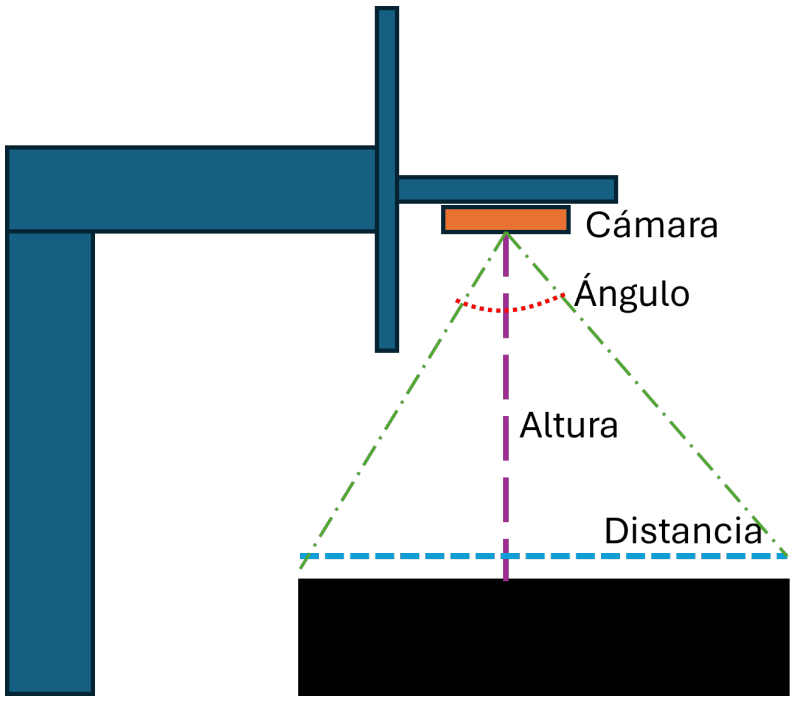


Fig 4.26: Esquema de la captura de imagen.

El primer paso es saber los píxeles de la cámara, en este caso la cámara que se ocupó es el módulo de cámara V3 de la empresa Raspberry Pi de 12 MP que equivale a 4608 x 2592 píxeles en la matriz con un ángulo de 66 grados.

El segundo paso es redimensionar la captura de imagen, ya que se está la pantalla del ordenador que se está ocupando para realizar el trabajo es de una resolución de 1920 x 1080, por lo cual la imagen que se captura con la Raspberry Pi tiene que ser de la misma resolución o más pequeña para poder ser observada en la pantalla del ordenador, por ello se ocupó una resolución UXGA que tiene 1600 x 1200.

El tercer paso es realizar un ROI con un objeto cuya dimensión se conozca, en este caso se empezó haciendo el ROI de la hoja del patrón para identificar deformidades de la cámara, al ser una hoja tamaño carta cuenta con las siguientes dimensiones 21.59 x 27.94 cm.

El cuarto paso es comparar los píxeles obtenidos con el tamaño de la imagen, en este caso fue 1577 x 1192.

Después se realizó el paso tres y cuatro, haciendo otra comparación con otro objeto que se conoce las dimensiones reales, en este caso fue la celda solar, que cuenta con 156 x 156 mm y los píxeles que abarca la celda fueron de 442 x 434.

Todo este proceso de dimensionar se puede ilustrar en la Fig. 4.27, que muestra los pasos que se realizaron para obtener las medidas de un píxel, a su vez se obtuvo la Tabla 4.13 que contiene las medidas reales de los objetos en milímetros y su representación en píxeles, estas medidas se tomaron con una imagen que se captura a 290 mm de altura.

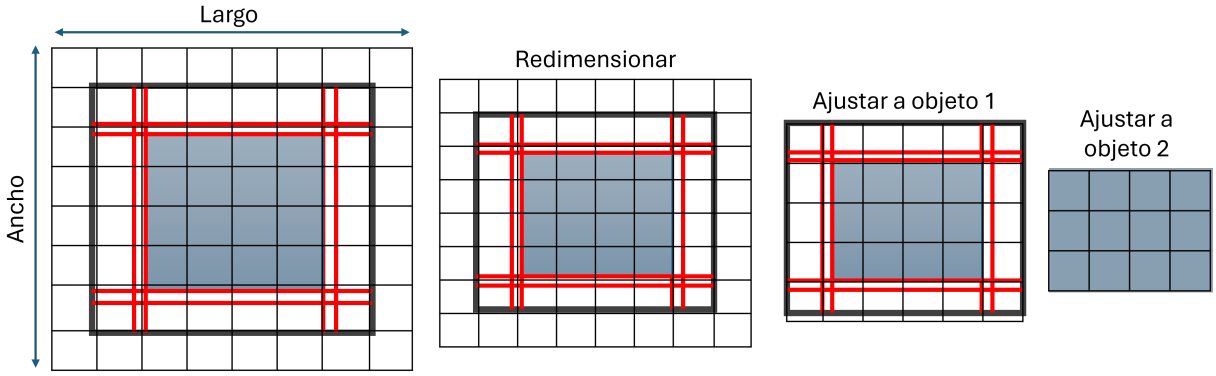
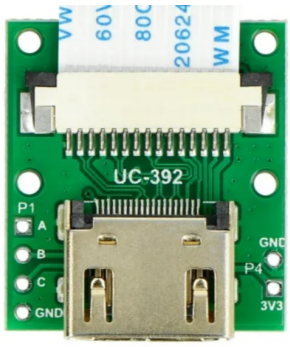


Fig 4.27: Esquema de la captura de imagen.

Tabla 4.13: Tabla de dimensiones de una hoja tamaño carta y una celda solar.

Objeto	Altura	Medidas reales (mm)	Píxeles	Valor del píxel (mm)
hoja tamaño carta	290 mm	279.4 x 215.9 mm	1577 x 1192	0.1771 x 0.1811 mm
Celda Solar	290 mm	156 x 156 mm	884 x 868	0.1764 x 0.1797 mm

Para realizar pruebas con el sistema completo, se llevó a cabo un estudio y se determinó que no existía un cable flexible lo suficientemente largo para conectar directamente la cámara a la Raspberry Pi 5. Por ello, se procedió a buscar una solución para este problema. Una alternativa fue adquirir un adaptador de CSI a HDMI. Este dispositivo, mostrado en la Fig. 4.28, es de la empresa ArduCam y cuenta con las siguientes características [38]:



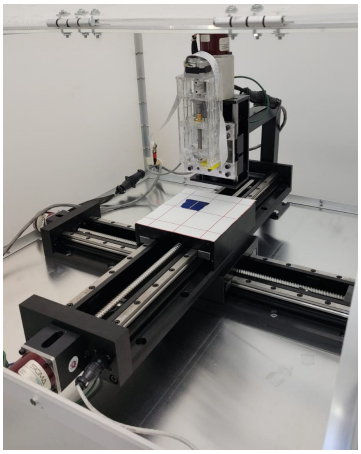
Adaptador de CSI a HDMI.

- Compatible con NVIDIA Jetson y Raspberry Pi.
- No se necesita software ni configuración adicional.
- El cables HDMI pueden tener una longitud de hasta 3 a 5 metros.
- Conector de micrófono y auriculares reservado para señales de audio.
- Tamaño: 27,63 x 25,00 mm

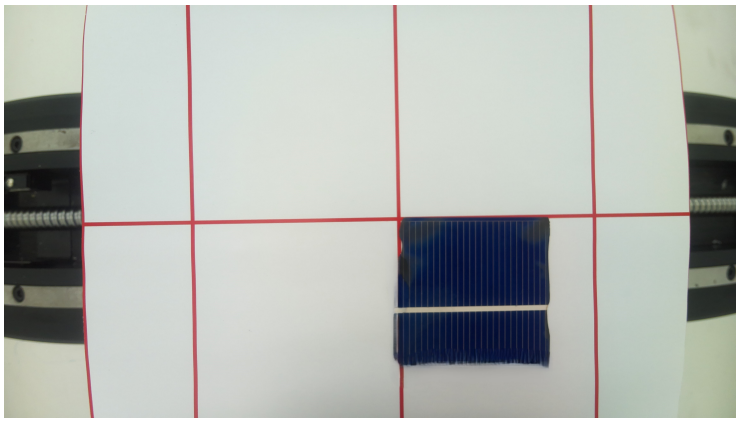
Fig 4.28: Adaptador de CSI a HDMI.

Gracias a este dispositivo, es posible conectar un cable HDMI y ubicar la Raspberry Pi fuera del sistema, permitiendo su conexión con los demás dispositivos que interactúan con ella. Importante remarcar que el cable HDMI no tiene que superar los 4 metros y este tiene que ser de tipo 2.1, de lo contrario este no trasmitirá la imagen y la Raspberry Pi 5 no detectará que está conectada la cámara al dispositivo.

La Fig.4.29 muestra la estructura final de cómo se adquiere la imagen de la celda solar con todo el sistema armado. Este puede cambiar de altura dependiendo de la posición en la que se encuentre el sistema Z que carga el sistema de depósito.



(a) Ejemplo de adquisición de imagen en el sistema completo



(b) Adquisición de imagen con la cámara V3.

Fig 4.29: Adquisición de imagen con el sistema completo.

4.9. Interfaz gráfica

Al crear una interfaz de usuario, no se trata del mismo problema que crear un programa para ejecutarse en una computadora, ya que existen varias herramientas de depuración que pueden encontrar errores en el código. En cambio, los diseñadores de interfaz no pueden anticipar todas las necesidades y preferencias del usuario en detalle, por lo que desarrollar un prototipo de la interfaz es una solución para detectar los errores que puede tener al ser utilizada por el usuario, sin embargo, la seguridad de los datos de los usuarios es importante y no debe pasarse por alto. Este componente es decisivo para la creación de una interfaz. Por lo tanto, es importante lograr un equilibrio con procedimientos formales para garantizar la precisión y la ausencia de errores. Esto se debe a que las pruebas de usabilidad deben usarse para detectar los detalles de la interfaz de usuario e identificar las clases que deben probarse para detectar defectos.

Los diseñadores de interfaz mencionan que los métodos formales son inaccesibles e inaplicables, y en otro caso, los expertos en métodos creen que las interfaces de usuario son triviales, lo cual es una consecuencia irónica de su facilidad de uso, debido a que parecen tan simples que es obviamente funcional [39], sin embargo, según la norma internacional ISO 9241 [40], que establece las prácticas de las interfaces de usuario deben implementarse, probarse con los usuarios, corregir los errores y luego probarse nuevamente, realizando un ciclo de pruebas para garantizar una confiabilidad.

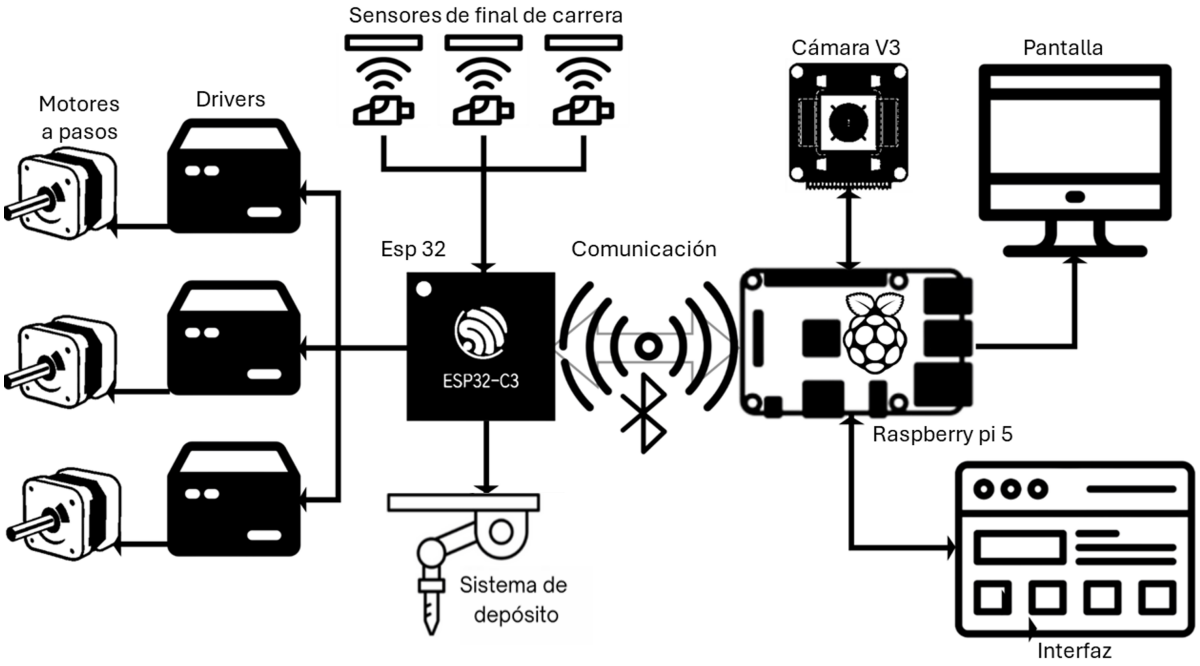


Fig 4.30: Esquema generar de las diferentes etapas que componen al proyecto.

En la Fig. 4.30 muestran las distintas etapas que cuenta el proyecto: control del movimiento de los motores, el procesamiento de imagen y el control del sistema de depósito. Cada una de estas etapas se enfoca en sus diferentes procesos, el objetivo es establecer un vínculo entre estas etapas y la interfaz, como una herramienta que pueda controlar el movimiento del sistema y a la vez poder capturar una imagen. Cada proceso se está realizando con diferente tarjeta de desarrollo, para ello es necesario comunicar a los dos microcontroladores de forma continua y eficiente. Actualmente, existen diferentes formas de poder cumplir con este requerimiento, correspondientemente a cada hoja de datos [41] [42], se puede utilizar Wi-Fi, UART (RS232) y Bluetooth entre otras, para realizar pruebas con el sistema se basará entre la UART y Bluetooth.

4.9.1. Materiales para la interfaz

Para realizar la parte de la interfaz, se requirieron ciertos materiales, para que pueda desarrollar su papel, en donde se desarrolló todo el programa de la interfaz fue en una tarjeta de desarrollo, la cual fue Raspberry Pi 5 que cuenta con las siguientes características [42] :

Especificación de la tarjeta Raspberry Pi 5.

- Procesador Arm Cortex-A76 de cuatro núcleos de 64 bits de 2.4 GHz.
- GPU VideoCore VII de 800 MHz.
- Wifi Dual-band.
- Bluetooth 5.0.
- 2 × puertos USB 3.0.
- 2 × puertos USB 2.0.
- Conector de cámara CSI.
- 8 GB de RAM.
- Alimentación de 5V/5A DC.

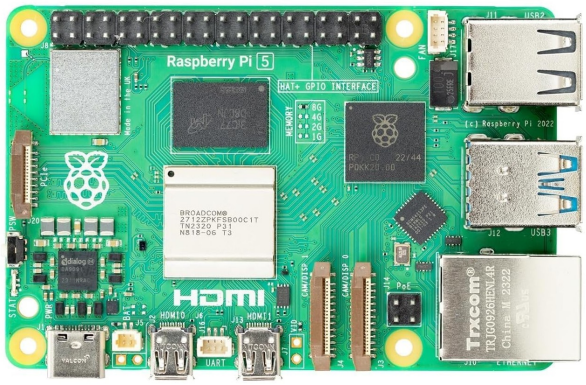


Fig 4.31: Raspberry Pi 5

Para visualizar y programar la interfaz, se requiere una pantalla con entrada HDMI (High-Definition Multimedia Interface), con el fin de observar los programas necesarios para el desarrollo de la interfaz. Por ello, se utilizó una pantalla táctil de 10.1 pulgadas de la empresa HMTECH, la cual se muestra en la Fig. 4.32 con las siguientes características: [43].

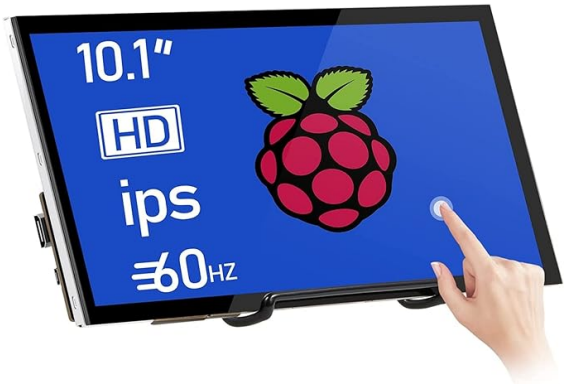


Fig 4.32: Pantalla con entrada HDMI.

Especificación de la pantalla táctil.

- Resolución: 1024X600 (HD).
- Terminales: USB y HDMI.
- Dimensiones: 235 x 143 x 7 mm.
- Tipo de pantalla: Táctil capacitivo de un solo toque.
- Alimentación: 5 V, 1 A.
- Frecuencia de actualización: 60 Hz.

4.9.2. Desarrollo de la interfaz

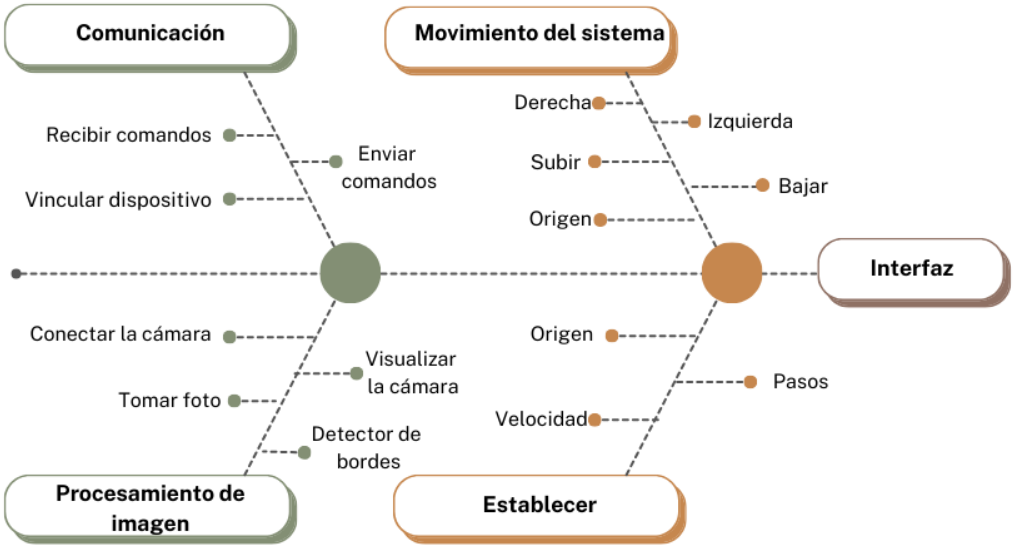


Fig 4.33: Diagrama de Ishikawa de las tareas que realizará la interfaz

Se pretende proporcionar un entorno de comunicación efectivo entre las personas y las máquinas, permitiendo a los usuarios controlar, supervisar y comprender el funcionamiento de un sistema de depósito de material de manera intuitiva y sencilla. El objetivo principal es simplificar las operaciones y el control del sistema. Los usuarios podrán ajustar parámetros, iniciar y detener procesos, y recibir retroalimentación en tiempo real. Además, se permitirá la visualización de datos y del estado del sistema, lo cual es crucial para este proyecto, ya que la monitorización de variables como las posiciones y el estado de los sensores de final de carrera resulta fundamental para garantizar un funcionamiento seguro y eficiente. La interfaz también podrá adaptarse a las preferencias del usuario, permitiendo establecer el origen de cada eje del sistema. Antes de iniciar la escritura de cualquier código o desarrollo del proyecto, es recomendable contar con una guía clara de las tareas a realizar. Por esta

razón, se elaboró un diagrama de Ishikawa que detalla las funciones de la interfaz. Dicho diagrama se presenta en la Fig. 4.33.

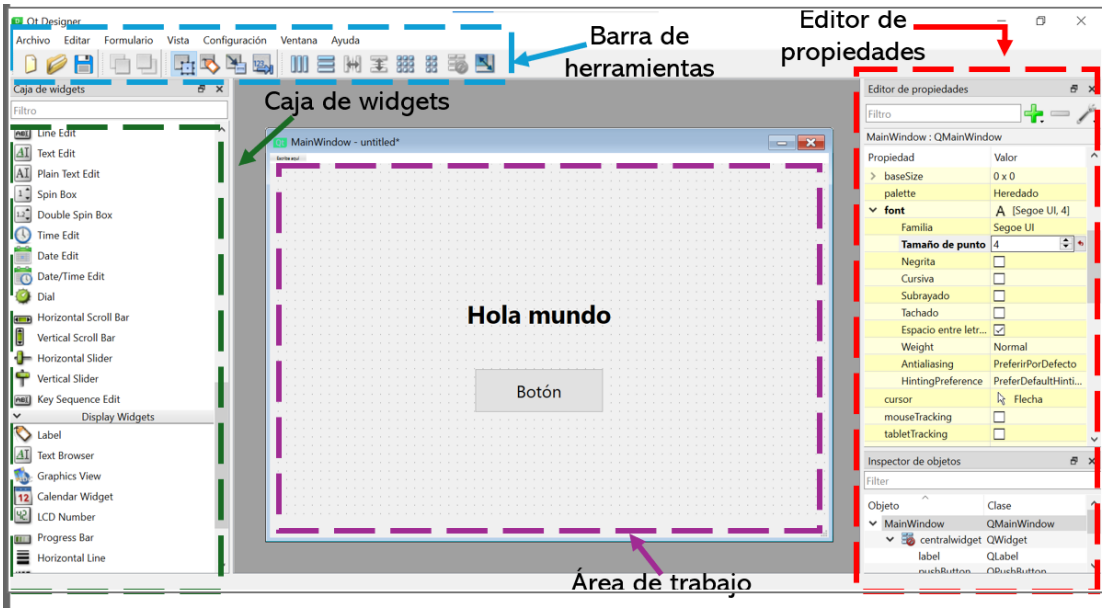


Fig 4.34: Herramientas de QT Designer.

Para poder realizar la interfaz con las tareas requeridas, se ocupó el software de QT y el programa QT Designer de la misma empresa, con este se puede crear una interfaz de una manera intuitiva, arrastrando y soltando elementos gráficos, esto ayuda a planificar los elementos requeridos, como lo son botones, cuadros de texto, menús desplegables, etiquetas y otros elementos, sin la necesidad de ser escritos en código de forma manual. Otra ventaja que se tiene sobre este programa es su integración con el entorno de desarrollo integrado (IDE) Qt Creator, al mismo tiempo facilita la creación y edición de la interfaz mientras se escribe el código de la aplicación [44].

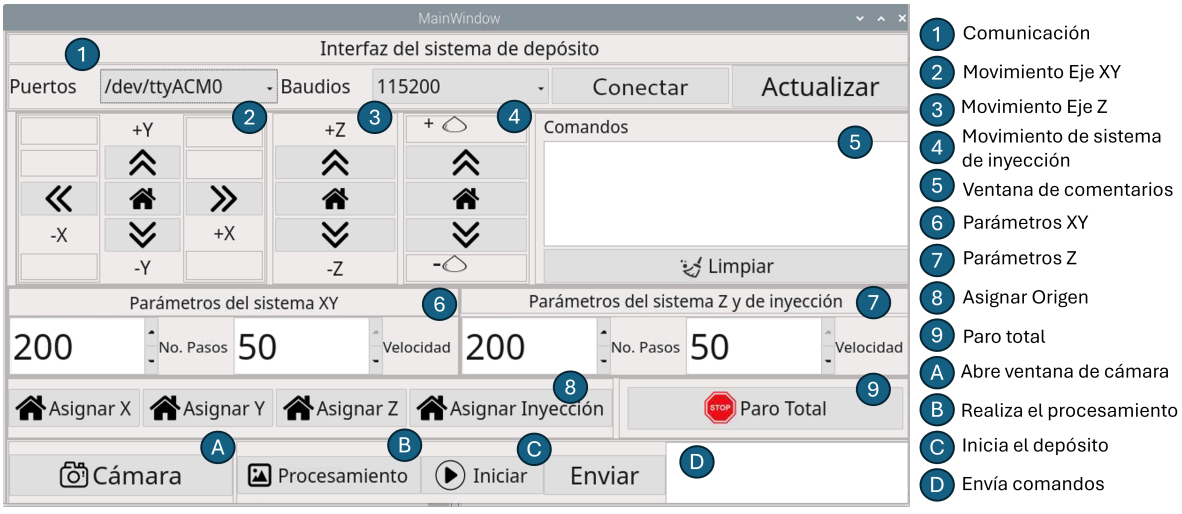


Fig 4.35: Interfaz ventana de inicio.

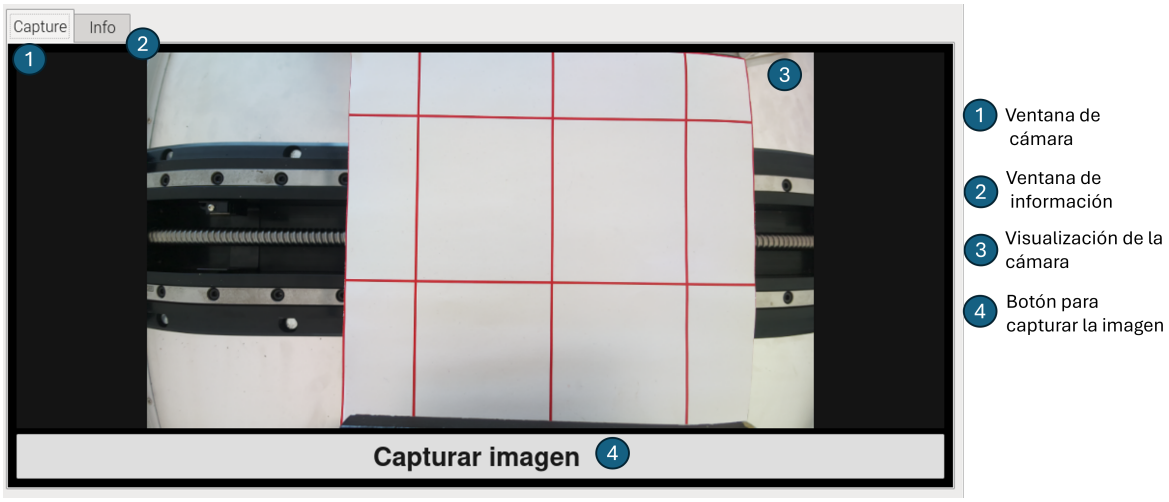


Fig 4.36: Interfaz ventana de la cámara.

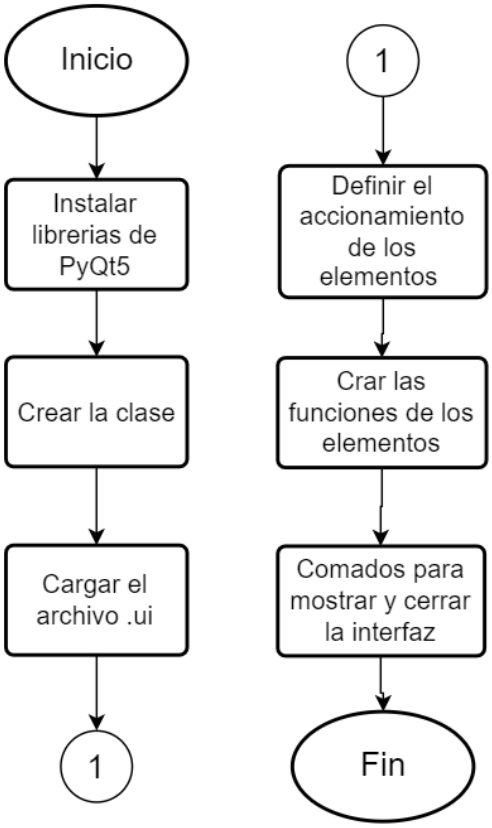


Fig 4.37: Diagrama de flujo del desarrollo de la interfaz.

A continuación, en la Fig. 4.34 se muestran las diferentes herramientas que ofrece el programa. La barra de herramientas incluye todas las opciones de trabajo disponibles, mientras que la caja de widgets contiene los elementos que se pueden utilizar en la interfaz, como botones, etiquetas, cuadros de texto, imágenes, temporizadores, editores de texto y barras de desplazamiento, entre otros componentes útiles para crear la interfaz.

El editor de propiedades permite modificar las características de los widgets, como su tamaño, ubicación, color, estilo y otros atributos específicos. Finalmente, el área de trabajo es el espacio donde se colocan y organizan los widgets que conformarán la interfaz.

Al tener la idea de cómo es el funcionamiento del programa, se comenzó a insertar los diferentes widgets necesarios para conformar la interfaz, con el fin de cumplir la lista de tareas establecidas. Como resultado, se obtuvo la interfaz mostrada en la Fig. 4.35 y la Fig. 4.36 en esta se indica el funcionamiento de cada elemento que lo conforma. Una vez diseñada la interfaz, el archivo se guarda en formato .ui, el cual se utilizará posterior-

mente para implementar el código que dará funcionalidad a la interfaz. Cabe destacar que Qt permite trabajar con diversos lenguajes de programación, como C++ y Python. En este proyecto, dado que la parte del procesamiento de imágenes se está desarrollando en Python, esto facilita la integración entre la interfaz y las funciones de procesamiento de imágenes.

Para iniciar con el código en Python, se siguió con el diagrama de flujo que se muestra en la Fig. 4.37 se instalaron las diferentes librerías que conforman a PyQt5, después se crea la clase que va a hacer la interfaz (main window) que se creó en QT Designer, para ello se necesita cargar el archivo de la interfaz con el loadUi, es un comando que indica que archivo ui se va a ocupar como clase.

```
        super(MyApp, self).__init__()
        loadUi('interfaz_prueba.ui', self)
```

Después se definen los accionamientos de todos los elementos que están en la interfaz con una función, por ejemplo:

```
        self.Button_menos_Y.clicked.connect(self.bajar_y)
```

Donde "*Button_menos_Y*" es el nombre del botón, clicked es el acción y *bajar_y* es la fución que se va a hacer cuando se de un click en el botón. Un ejemplo de una función es la siguiente.

```
        def bajar_y(self):
                pasos = self.spinBox_Pasos_XY.value()
                velocidad = self.spinBox_Velocidad_XY.value()
                lista_pasos = "b" + str(pasos)
                lista_velocidad = "v" + str(velocidad)
                self.serial.send_data(lista_velocidad)
                self.serial.send_data(lista_pasos)
```

"*pasos*" es la variable que guarda el valor definido en el editor de texto llamado "*spinBox_Pasos_XY.value*" y "*velocidad*" es la variable que guarda el valor de editor de texto llamado "*spinBox_Velocidad_XY.value*", al tener estas variables se les agregará una letra para que el otro microcontolador identifique que operación va a realizar y después la cantidad numérica de la variable. Para finalizar se llama la función "*serial.send_data*" que se encarga de enviar los valores obtenidos en la interfaz al otro microcontolador a través de una comunicación serial. Este proceso se repite en todos los elementos que conforman a la interfaz dependiendo a las tareas que van a realizar en el sistema.

Al tener todas las funciones definidas se concluye el código con los siguientes comandos que ayudan a mostrar y a cerrar la interfaz.

```
app = QApplication(sys.argv)
main = MyApp()
main.show()
sys.exit(app.exec_())
```

4.9.3. Comunicación entre microcontroladores

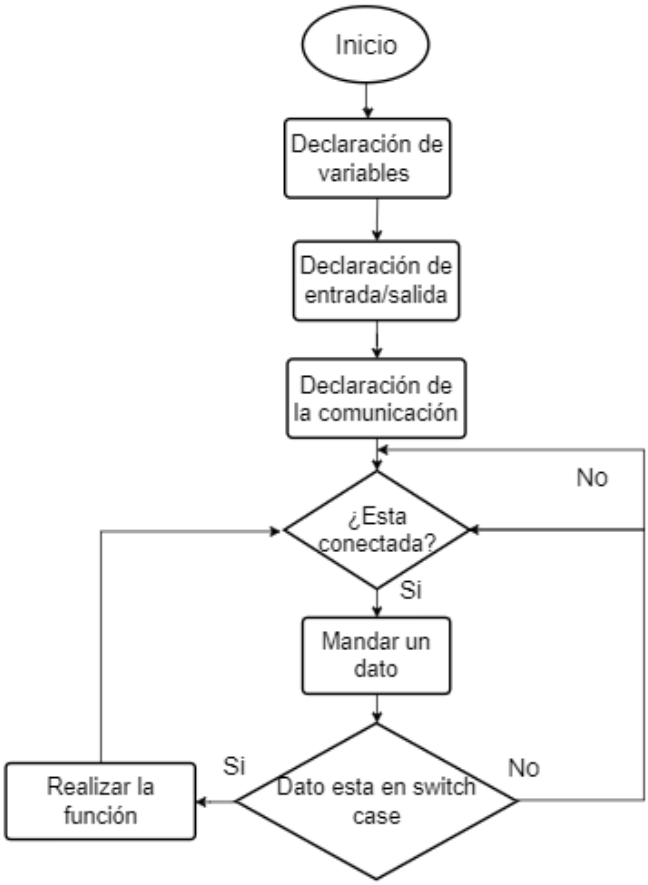


Fig 4.38: Diagrama de flujo del desarrollo de la interfaz.

La siguiente parte es desarrolla el código para recibir los datos enviados por la interfaz, para ello se siguió el diagrama de flujo de la Fig. 4.38, donde muestra la secuencia que se tiene el código del microcontrolador, iniciando declarando todas las variables que se ocuparán, después se declararon los puertos de entrada y salida del sistema, para finalizar con la declaración de la comunicación serial, esto conforma la primera parte del código. La segunda parte es un bucle de una estructura switch case, este iniciará inmediatamente cuando comience la comunicación entre la Raspberry Pi 5 y la Esp32, se repetirá todo el tiempo la estructura hasta que se desconecten los microcontroladores o la interfaz se cierre.

Cuando recibe un dato la Esp32 este se evalúa si se encuentra entre las opciones que tiene la estructura, si en este caso no hay dato que coincida, entonces no realizará nada el microcontrolador con el sistema, pero si coincide el dato enviado, este se evaluará entre las opciones que tiene y realizará la actividad pertinente en el sistema. Un ejemplo de dicha operación es el siguiente.

```
    if (SerialBT.available())
cmd = 0;
cmd = SerialBT.read();
```

```
    if (cmd > 31)
    {
      switch(cmd)
      {
                case 'r' : pasos_x   = SerialBT.parseFloat(); pasos_x_total
                   = pasos_x_total + pasos_x; digitalWrite(motor1EnablePin,
                   HIGH); SerialBT.print("Motor avanca en el eje x.");
                   SerialBT.println(pasos_x_total);  break;
                case 'd' : pasos_y   = SerialBT.parseFloat(); pasos_y_total
                   = pasos_y_total + pasos_y; digitalWrite(motor2EnablePin,
                   HIGH); SerialBT.print("Motor avanca en el eje y.");
                   SerialBT.println(pasos_y_total); break;
      }
    }
  }
```

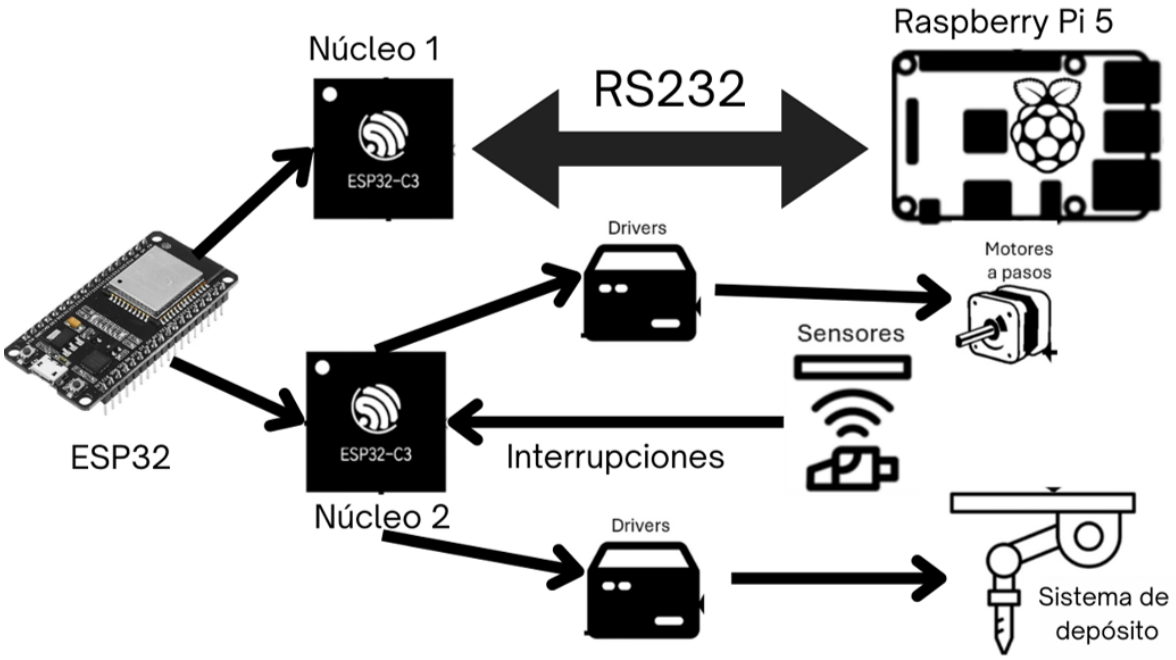


Fig 4.39: Esquema del trabajo que realiza la ESP32.

Capítulo 5

Integración y pruebas del sistema

En este último capítulo se anexarán todos los resultados obtenidos durante la implementación del sistema completo, trabajando de forma sincronizada cada componente para alcanzar el objetivo, el cual es el depósito de material en el patrón de contacto de la celda fotovoltaica. Para ello, se realizó una caracterización previa de las celdas sin ningún tipo de material agregado. Posteriormente, se depositó el material en el contacto con el fin de evaluar el rendimiento de la celda en cada caso.

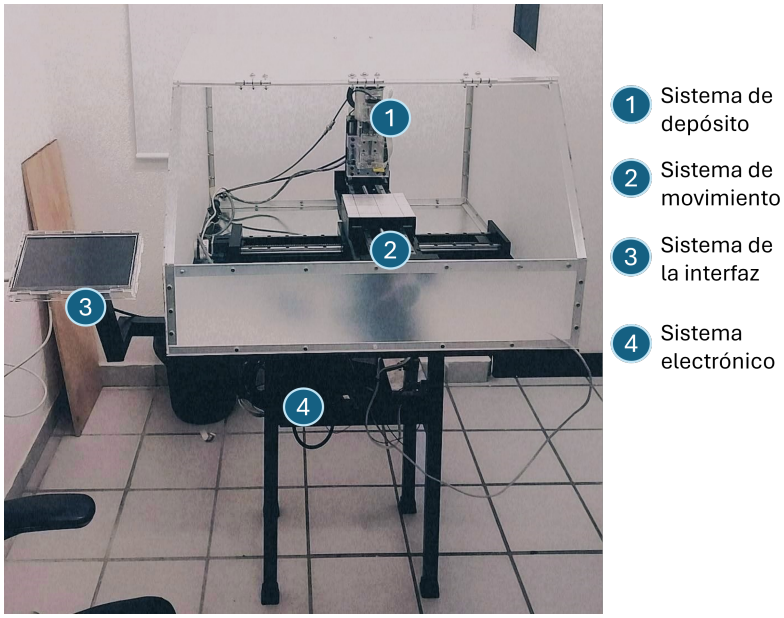
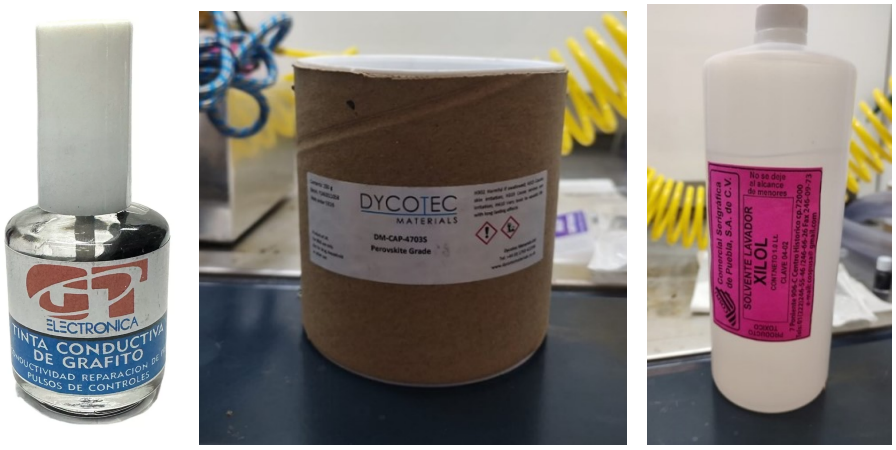


Fig 5.1: Elementos de la implementación.

5.1. Depósito del material

Al tener la parte del desarrollo concluida con la interfaz en la Raspberry Pi 5 y el funcionamiento del sistema con la Esp32, procedió al depósito de material, para ello se conectó la pantalla a la Raspberry Pi 5 y también en el puerto USB la Esp32 para realizar

la comunicación serial RS232, como se muestra en la Fig. 5.1, en donde se observan los diferentes elementos que se ocuparán en la implementación. Para realizar las pruebas del depósito de material se ocuparon dos materiales, los cuales fueron:



(a) Tinta Conductora base de Grafito.

(b) Pasta de carbono DM-CAP-4703S.

(c) Solvente xilol.

Fig 5.2: Elementos para hacer tinta de carbono

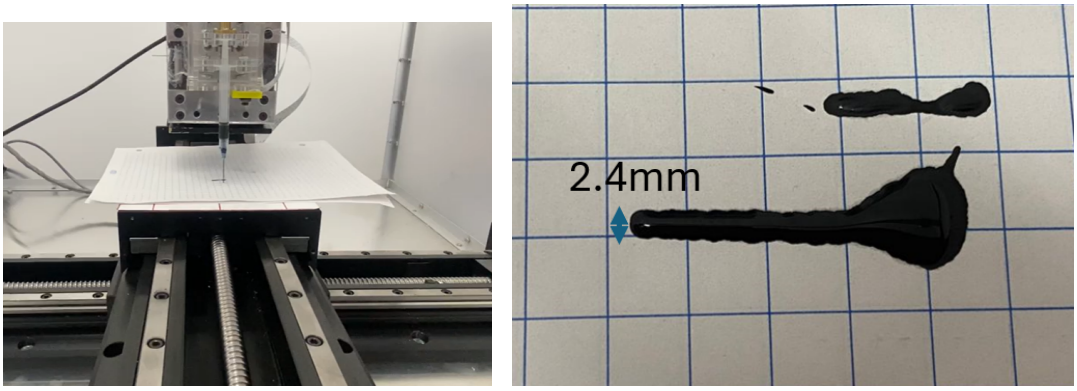
- Pintura conductora de grafito.

De uso versátil, elaborada a partir de una dispersión de grafito micronizado con una estructura molecular plana y hexagonal. Las partículas de grafito se adhieren a diversos tipos de sustratos una vez que el solvente se evapora tras su aplicación. Al secarse, el grafito queda firmemente adherido a la superficie, manteniendo las propiedades de flexibilidad de la pintura [45].

- Pasta de carbono DM-CAP-4703S.

Esta pasta es especialmente formulada para celdas solares de perovskita con el fin de producir electrodos de alta conductividad. La Dycotec DM-CAP-4703S es una pasta termoplástica diseñada para aplicaciones de serigrafía y recubrimiento mediante cuchillas, permitiendo un secado rápido a bajas temperaturas. Esta pasta incorpora aditivos patentados que incrementan la función de trabajo y mejoran la transferencia de carga desde las capas de perovskita.

Esta solución representa una alternativa estable y económica al recubrimiento con oro [46]. Para poder disolver la pasta se ocupó el Solvente Xilol, también conocido como Dimetilbenceno, es un solvente ampliamente utilizado en diversas industrias debido a su alta capacidad para disolver grasas, resinas, pinturas y otros compuestos orgánicos. Se emplea principalmente en la imprenta y en las industrias del caucho y del cuero [47].



(a) Sistema depositando tinta de Grafito. (b) Ancho final del depósito de tinta.

Fig 5.3: Prueba de depósito de tinta de Grafito en una hoja cuadriculada de 7mm

El sistema de depósito fue diseñado para permitir el uso de jeringas de diferentes tipos, para lograr un ancho reducido, se utilizó una jeringa pediátrica de 3 mL con una aguja de 23G x 25 mm ("G.^{es} de Gauge haciendo referencia al calibre de la aguja). Este tipo de jeringa tiene un diámetro exterior de 0.6 mm, conforme a lo establecido por la Norma Oficial Mexicana NOM-133-SSA1-1995 [48].
Se realizaron pruebas en una hoja de papel cuadriculada de 7 mm para verificar el ancho final de la inyección, obteniendo un resultado de 2.4 mm, como se ilustra en la Fig. 5.3.

5.2. Caracterización eléctrica de la celda fotovoltaica

Antes de aplicar el material en los patrones de los contactos de la celda fotovoltaica, se llevará a cabo la caracterización de la celda utilizando un instrumento especializado en generar gráficas IV. Este dispositivo corresponde al modelo 2450 de la empresa Keithley Instruments, el cual se muestra en la Fig. 5.4 y posee las siguientes características:[49].

Características del modelo 2450.

- Corriente máxima de fuente y rango de medición: 1 A.
- Voltaje máxima de fuente y rango de medición: 200 V.
- Resolución en corriente: 10 fA (femto amperes 10^{-15}).
- Resolución en voltaje: 10 nV.
- Potencia: 20 W.

Fig 5.4: Instrumento SMU modelo 2450 de Keithley.

Asimismo, se utilizó un piranómetro para me-

dir la radiación a la que estaba siendo sometida la celda fotovoltaica en ese momento, y con ello obtener esta variable que influye en la caracterización eléctrica de los contactos. Este dispositivo se muestra en la Fig. 5.5 y, a continuación, se presentan algunas de sus características: [50].

Fig 5.5: Detector de radiación solar PCE-SPM 1.

Características del piranómetro PCE-SPM 1.

- Rango de medición: 0 - 2000 W/m^2
- Resolución: 1 W/m^2
- Precisión: $\pm$10 W/m^2 o $\pm$5 %
- Rango de temperatura ambiente: 0 - 50 °C

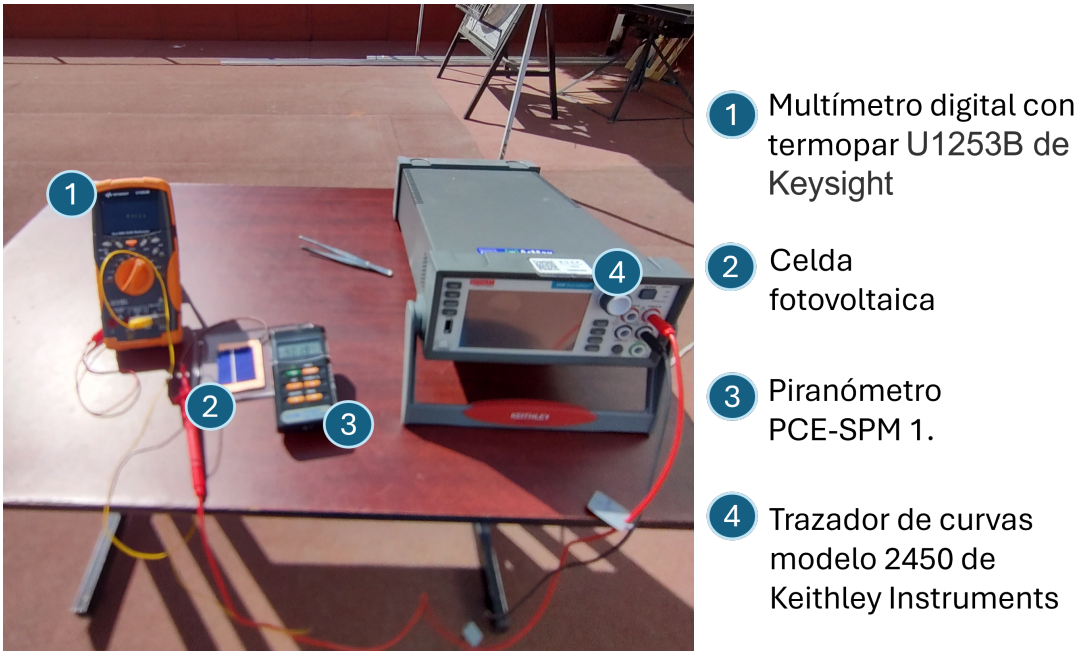


Fig 5.6: Arreglo experimental.

En la Fig. 5.6 se observa el arreglo experimental utilizado para obtener la gráfica IV de la celda, antes y después de la colocación del material. Además, se utilizó un multímetro digital con termopar U1253B de la empresa Keysight, para medir la temperatura a la que se encontraba sometida la celda fotovoltaica en el momento de la medición.

5.2.1. Gráfica IV

En esta sección se presentan las gráficas IV obtenidas utilizando el trazador de curvas modelo 4550, como se muestra en la Fig.5.11. Estas gráficas se obtuvieron mediante un barrido de tensión de 0 a 0.6 V, el cual es el voltaje necesario para activar el diodo que representa la unión N-P de la celda fotovoltaica, con una corriente limitada a 0.49 A. Este valor depende del nivel de radiación aplicado a la celda. A partir de la gráfica mostrada en la Fig.5.11, se determinaron las variables necesarias para caracterizar eléctricamente la celda. El voltaje de circuito abierto (V_{oc}) corresponde al valor más alto registrado en la gráfica y ocurre cuando la corriente es cero. Por otro lado, la corriente de cortocircuito (I_{sc}) representa la corriente más alta registrada, aunque no es aprovechable, ya que el voltaje es nulo en ese punto. Para calcular el voltaje máximo (V_{max}) y la corriente máxima (I_{max}) que se puede aprovechar, se graficó la potencia de la celda, definida como el producto de la corriente y el voltaje, como se muestra en la Fig.5.12. En esta representación se puede visualizar el punto más alto de la curva que corresponde a la potencia máxima (P_{max}) de la celda fotovoltaica. Este valor fue obtenido a partir de los datos del trazado de la gráfica IV. El proceso de caracterización eléctrica comenzó midiendo la celda sin ningún material conductor depositado. Posteriormente, se trazó nuevamente la curva después de aplicar el material conductor. En el caso de la celda 1, se utilizó pasta de carbono disuelta en xilol, mientras que en la celda 2 se aplicó tinta de grafito. A continuación, se detallan los parámetros registrados durante las mediciones IV de cada celda, como la temperatura, la radiación y el área de la celda, que son parámetros necesarios para obtener los resultados de la caracterización.

Fig 5.7: Celda 1 sin material

Celda 1 sin material

- Temperatura 46°C
- Radiación 925 W/m².
- Área 42.75 x 43.63mm

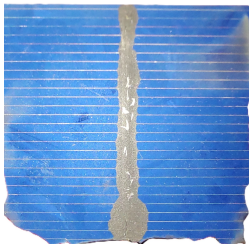


Fig 5.8: Celda 1 con material

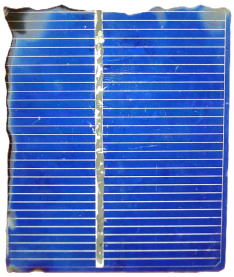
Celda 1 con pasta de carbono.

- Temperatura 36°C
- Radiación 990 W/m².
- Área 42.75 x 43.63mm

Celda 2 sin material.

- Temperatura 42°C
- Radiación 853 W/m².
- Área 46 x 54.7mm

Fig 5.9: Celda 2 sin material.



Celda 2 con tinta de grafito.

- Temperatura 38°C
- Radiación 850 W/m².
- Área 46 x 54.7mm

Fig 5.10: Celda 2 con material

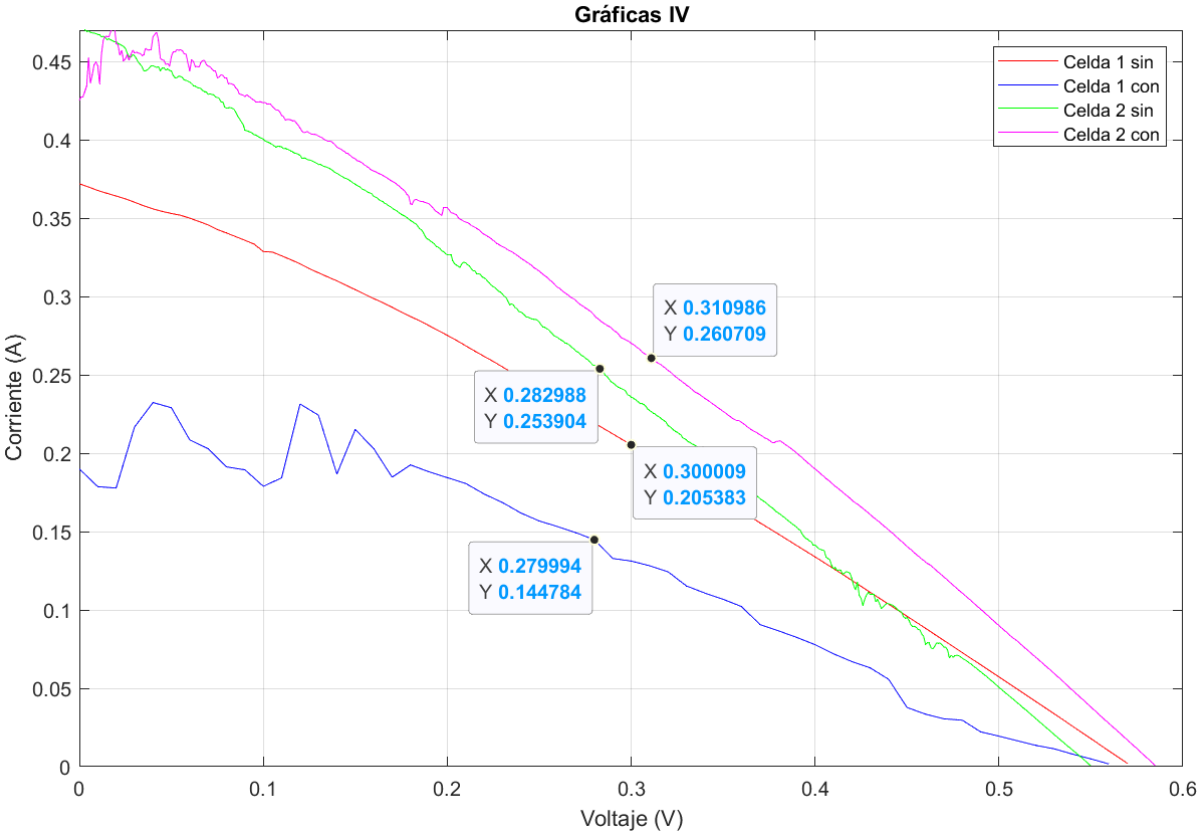


Fig 5.11: Gráficas IV.

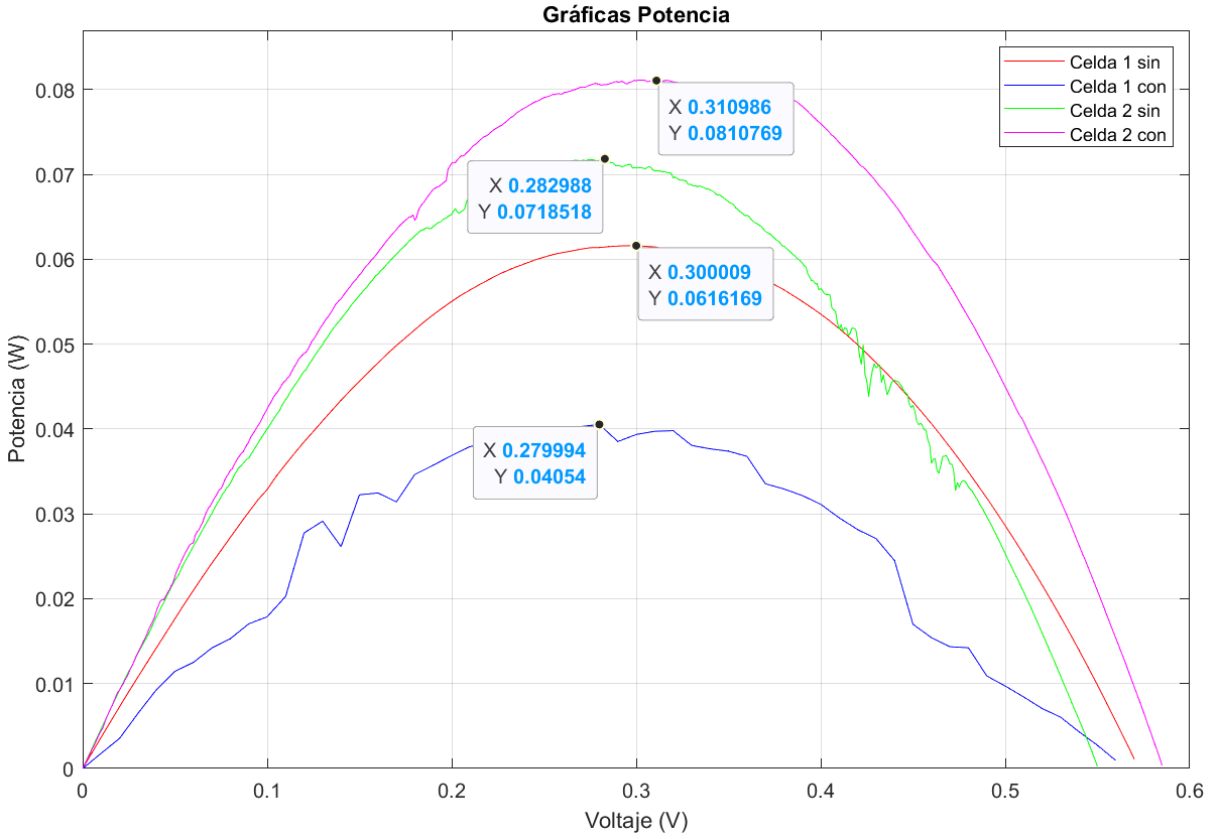


Fig 5.12: Gráficas de potencia.

5.2.2. Resultados

Con base en las gráficas presentadas en la Fig.5.11 y la Fig.5.12, se obtuvieron los valores de las variables necesarias para calcular el voltaje térmico (V_t), la resistencia en serie (R_s), el factor de llenado (FF) y la eficiencia (η) de las celdas fotovoltaicas. Estos cálculos se realizaron utilizando los valores presentados en la Tabla 5.1 y las ecuaciones (2.3), (2.4), (2.5) y (2.6). El parámetro principal en esta caracterización es la resistencia en serie (R_s), que representa la resistencia de unión del contacto con el silicio.

Debido a que el objetivo de este proyecto es el depósito de material en los contactos de la celda, es fundamental analizar cómo este afecta su funcionalidad. Para ello, también se evaluaron el factor de llenado (FF), que mide qué tan cerca está la celda de operar en sus condiciones ideales de corriente y voltaje maximizando la potencia generada, también se mide la eficiencia (η), que indica el aprovechamiento de los recursos de la celda. La Tabla 5.2 presenta los resultados de la caracterización eléctrica de las celdas fotovoltaicas.

En el análisis se determinó que la Celda 1 con pasta de carbono presentó la mayor resistencia en serie, con un valor de 1.5659Ω calculada con la ecuación (2.3), y el factor de llenado obtenido con la ecuación (2.5), que fue el más alto de 0.3916 % con respecto a la

gráfica de factor de llanado que se encuentra en la Fig. 5.13. Sin embargo, en condiciones de operación real, esta celda verá afectada su eficiencia debido a su elevada resistencia en serie, la cual limita su capacidad para suministrar corriente a cargas externas y obtuvo un valor de 0.0163 % que se calculo con la ecuación (2.6).

En el otro caso la Celda 2 con tinta de grafito mostró la menor resistencia en serie, con un valor de 0.9689Ω, calculada con la ecuación (2.3), y un factor de llenado de 0.2924 %, estimado con le ecuación (2.5), que se encuentra dentro del rango de las celdas sin material conductor depositado, esto se observa en la gráfica de factor de llenado que se encuentra en la Fig.5.13. Aunque la diferencia de la resistencia en serie respecto a las celdas sin material es pequeña, tiene un impacto significativo en la eficiencia, ya que alcanzó un valor de 0.0511 % de acuerdo con la ecuación (2.6), siendo la más alta registrada en la caracterización eléctrica de las celdas, se observa en la grafica de eficiencia de la Fig. 5.13.

Tabla 5.1: Tabla de variables de las celdas.

Celda	I_{MAX}	V_{MAX}	I_{SC}	V_{OC}
1 sin material	0.205383 A	0.300009 V	0.371972 A	0.565037 V
1 con pasta de carbono	0.144784 A	0.279994 V	0.189966 A	0.545002 V
2 sin material	0.253904 A	0.282988 V	0.471986 A	0.550015 V
2 con tinta de grafito	0.260709 A	0.310986 V	0.474013 A	0.585028 V

Tabla 5.2: Tabla de resultados de la caracterización eléctrica de las celdas.

Celda	$T(K)$	Vt	R_s	FF	η
1 sin material	319.15 °K	0.0275 V	1.1827 Ω	0.2932	0.0265
1 con pasta de carbono	309.15 °K	0.0267 V	1.5659 Ω	0.3916	0.0163
2 sin material	315.15 °K	0.0272 V	0.9690 Ω	0.2768	0.0452
2 con tinta de grafito	311.15 °K	0.0268 V	0.9689 Ω	0.2924	0.0511

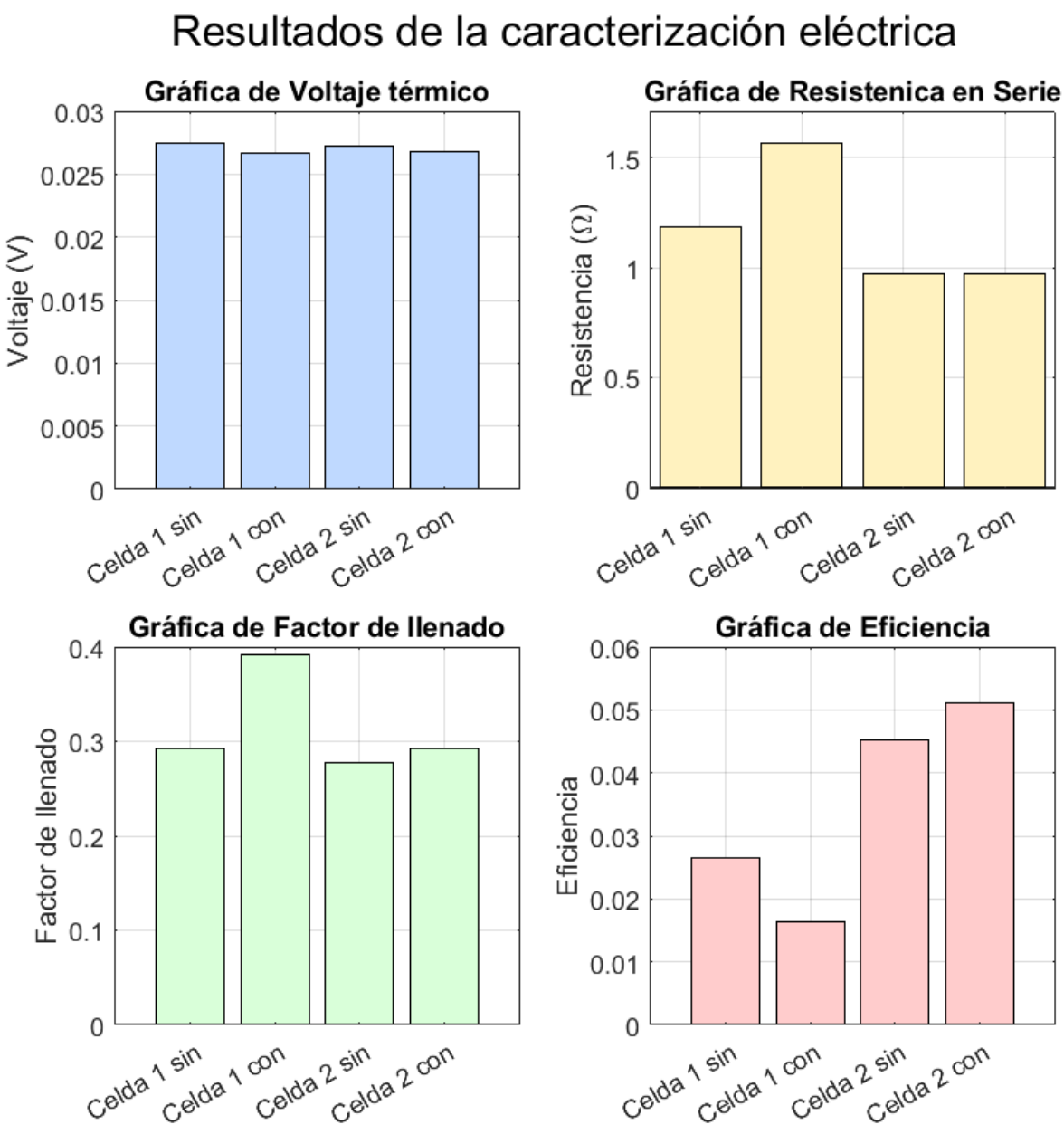


Fig 5.13: Gráfica de los resultados de la caracterización eléctrica de las celdas.

Capítulo 6

Conclusiones

El desarrollo e implementación de este proyecto permitió alcanzar los objetivos planteados, enfrentando diversos retos técnicos y estructurales. A continuación, se presentan las principales conclusiones obtenidas:

Diseño y Construcción del Sistema Mecánico. La idea inicial del desarrollo del mecanismo implicó un proceso que demandó un análisis constantemente detallado de las dimensiones, materiales y configuración del sistema. Aunque, el depósito del material en una línea sobre la celda fotovoltaica pueda parecer una tarea sencilla, su implementación requirió un esfuerzo en diseño y ensamblaje para garantizar un movimiento en diferentes planos. Este resultado destaca la importancia de abordar el proyecto de manera estructurada, avanzando por etapas y resolviendo los desafíos técnicos que surgieron en cada fase.

Integración de los Sistemas Eléctrico y Electrónico. El desarrollo de los sistemas eléctrico y electrónico se llevó a cabo de manera paralela a la construcción del sistema mecánico. Cada avance en la estructura demandó ajustes en los componentes electrónicos, lo que contribuyó a un diseño más eficiente y funcional. La optimización de la conexión de los arneses facilitó el ensamblaje y minimizo significativamente el margen de error durante la instalación en la placa PCB diseñada. Los prototipos previos permitieron mejorar la funcionalidad y la confiabilidad, asegurando un sistema integral y robusto.

Procesamiento de Imágenes. La implementación del procesamiento de imágenes resultó fundamental para la detección de los patrones geométricos de los contactos eléctricos en las celdas fotovoltaicas. Este objetivo se logró mediante la selección de una cámara adecuada y su integración con un microcontrolador capaz de realizar el procesamiento necesario. Las pruebas iniciales, enfocadas en ajustar los parámetros de detección, revelaron la complejidad del problema como el potencial del sistema para aplicaciones futuras en automatización y caracterización."

Desafíos en el Desarrollo de la Interfaz Gráfica. La implementación de la interfaz gráfica se llevó a cabo utilizando la Raspberry Pi 5, una tecnología recientemente introducida al mercado. Este desafío, lejos de ser una limitación, sirvió como una oportunidad para innovar y explorar capacidades avanzadas de hardware y software. La integración de la cámara con la interfaz requirió un esfuerzo considerable para lograr un acoplamiento funcional, lo que permitió establecer las bases sólidas para futuros desarrollos y optimizaciones.

Caracterización Eléctrica de las Celdas Fotovoltaicas. Aunque el objetivo principal del proyecto no fue la optimización del desempeño energético de las celdas, se llevó a cabo un análisis detallado de su comportamiento eléctrico. Este estudio permitió identificar y evaluar el impacto del material conductor depositado sobre los patrones geométricos Los resultados obtenidos proporcionan información importante para establecer bases sólidas en futuros trabajos enfocados a la mejora de la eficiencia energética en la celda fotovoltaica.

Finalmente, se logró integrar un sistema mecánico, electrónico y de procesamiento de imágenes, capaz de detectar patrones geométricos y realizar depósitos de material conductor sobre celdas fotovoltaicas. Los desafíos superados durante el desarrollo y las soluciones implementadas han establecido una base para el avance en la tecnología de manufactura y caracterización fotovoltaica

6.1. Trabajo futuro

Al concluir este trabajo, se encontraron varias mejoras que se podrían realizar en el futuro, como las siguientes:

- Investigar un mecanismo de depósito que pueda cumplir con los requerimientos necesarios para depositar el material en anchos más reducidos y que tenga un mejor control.
- Diseñar un prototipo de depósito que cumpla con la resolución requerida para el patrón de los contactos.
- Implementar una tinta conductiva o material equivalente que mejore la eficiencia de la generación eléctrica y sea adecuado para el depósito.
- Realizar la calibración a través del procesamiento de imagen.
- Mejorar la velocidad y la estabilidad del mecanismo.
- Incluir diferentes herramientas en la interfaz para ecualizar la imagen en el momento.

Bibliografía

[1] SENER, "Balance nacional de energía 2023," 2024. [En línea]. Disponible: https://www.gob.mx/cms/uploads/attachment/file/947752/BALANCE_NACIONAL_DE_ENERG_A_PRELIMINAR_2023.pdf

[2] Secretaría de energía, (2024). Reporte mensual de productores independientes de energía eléctrica, Sistema de información energética. [En línea]. Disponible: https://sie.energia.gob.mx

[3] A. Dziedzic, J. Nijs, and J. Szlufcik, "Thick-film fine-line fabrication techniques—application to front metallisation of solar cells," *Microelectronics International*, vol. 10, no. 1, pp. 18–26, 1993.

[4] A. Mette, P. L. Richter, M. Hörteis, and S. W. Glunz, "Metal aerosol jet printing for solar cell metallization," *Progress in Photovoltaics: Research and Applications*, vol. 15, no. 7, pp. 621–627, 2007. [En línea]. Disponible: https://onlinelibrary.wiley.com/doi/abs/10.1002/pip.759

[5] P. Mariani, L. Vesce, and A. D. Carlo, "The role of printing techniques for large-area dye sensitized solar cells," *Semiconductor Science and Technology*, vol. 30, no. 10, p. 104003, sep 2015. [En línea]. Disponible: https://dx.doi.org/10.1088/0268-1242/30/10/104003

[6] D. Neamen, *Semiconductor Physics And Devices*, ser. McGraw-Hill Series in Electrical and Computer Engineering. McGraw-Hill Education, 2003. [En línea]. Disponible: https://books.google.com.mx/books?id=TPE9AQAAIAAJ

[7] M. Diantoro, T. Suprayogi, A. Hidayat, A. Taufiq, A. Fuad, R. Suryana *et al.*, "Shockley's equation fit analyses for solar cell parameters from iv curves," *International Journal of Photoenergy*, vol. 2018, 2018.

[8] M. Diantoro and Suprayogi., "Shockley's equation fit analyses for solar cell parameters from iv curves," *International Journal of Photoenergy*, vol. 2018, 2018.

[9] W. Junchao and Z. Chang, "Defect detection on solar cells using mathematical morphology and fuzzy logic techniques," *Journal of Optics*, pp. 1–11, 2023.

[10] E. E. Rodríguez Suárez, J. Santiago, "La industria solar fotovoltaica y fototérmica en méxico," 2017.

[11] Solargis. (2018) Download solar resource maps and gis data. [En línea]. Disponible: https://solargis.com/maps-and-gis-data/download

[12] CONAHCYT. Energía y cambio climático. [En línea]. Disponible: https://conahcyt.mx/pronaces/pronaces-energia-y-cambio-climatico/

[13] D. G. Murias Figueroa, "Estudio de pasivación de sustratos de c-si y texturizado por plasma para aplicaciones en celdas solares de heterounión," 2016.

[14] E. O. Duran, "Historia, desarrollo y actualidad de las celdas solares."

[15] G. Schubert, "Thick film metallisation of crystalline silicon solar cells: Mechanisms, models and applications," Ph.D. dissertation, 2006.

[16] M. OHRING, *Materials Science of Thin Films: Depositon and Structure*. Academic Press, 2002. [En línea]. Disponible: https://researchs.bibliotecabuap.elogim.com/linkprocessor/plink?id=4878d005-ea5f-3604-bd68-a4e157ced946.

[17] J. M. A. Martín, *Láminas delgadas y recubrimientos: preparación, propiedades y aplicaciones*. Editorial CSIC-CSIC Press, 2003, vol. 11.

[18] A. H. Alami, S. Alasad, H. Aljaghoub, M. Ayoub, A. Alashkar, A. Mdallal, and R. Hasan, *Manufacturing of Silicon Solar Cells and Modules*. Cham: Springer International Publishing, 2023, pp. 45–63. [En línea]. Disponible: https://doi.org/10.1007/978-3-031-31349-3_5

[19] S. Angrist, *Direct Energy Conversion*, ser. Allyn and Bacon series in mechanical engineering and applied mechanics. Allyn and Bacon, 1982. [En línea]. Disponible: https://books.google.com.mx/books?id=SO1SAAAAMAAJ

[20] X. Hu, Z. Du, and F. Wang, "Research on detection method of photovoltaic cell surface dirt based on image processing technology," *Scientific Reports*, vol. 14, no. 1, p. 16842, 2024.

[21] H. Wang, F. Wang, X. Gong, D. Zhu, R. Wang, and P. Wang, "Detection of small targets in photovoltaic cell defect polarization imaging based on improved yolov7," *Applied Sciences*, vol. 14, no. 9, 2024. [En línea]. Disponible: https://www.mdpi.com/2076-3417/14/9/3899

[22] E. Sovetkin and A. Steland, "Automatic processing and solar cell detection in photovoltaic electroluminescence images," *Integrated Computer-Aided Engineering*, vol. 26, no. 2, pp. 123–137, 2019.

[23] Pveducation.org., "Resistencia en serie," 2024. [En línea]. Disponible: https://www.pveducation.org/es/fotovoltaica/4-operaci%C3%B3n-de-c%C3%A9lula-solar/resistencia-en-serie

[24] D. Sera, R. Teodorescu, and P. Rodriguez, "Photovoltaic module diagnostics by series resistance monitoring and temperature and rated power estimation," pp. 2195–2199, 2008.

[25] R. F. Farfán, L. T. Villa, and C. A. Cadena, "Método para la determinación del voltaje térmico de paneles fotovoltaicos bajo condiciones estánda," 2012.

[26] A. Luque and S. Hegedus, *Handbook of photovoltaic science and engineering*. John Wiley & Sons, 2011.

[27] T. B. Moeslund, *Introduction to video and image processing: Building real systems and applications*. Springer Science & Business Media, 2012.

[28] B. Jähne, *Applications and Tools*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 3–27. [En línea]. Disponible: https://doi.org/10.1007/978-3-662-04781-1_1

[29] B. Rohrer., "How to convert an rgb image to grayscale." November 14, 2019. [En línea]. Disponible: https://e2eml.school/convert_rgb_to_grayscale

[30] Y. You, J.-S. Chen, and H. Lu, "Filters, reproducing kernel, and adaptive meshfree method," *Computational mechanics*, vol. 31, no. 3, pp. 316–326, 2003.

[31] R. C. Gonzalez, *Digital Image Processing Using MATLAB*. Pearson education india, 2004.

[32] N. Senthilkumaran and S. Vaithegi, "Image segmentation by using thresholding techniques for medical images," *Computer Science & Engineering: An International Journal*, vol. 6, no. 1, pp. 1–13, 2016.

[33] Opencv, "Opencv: Image thresholding," (12 Septiembre 2023) [Online]. Available:, https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html.

[34] D. Asamoah, E. Ofori, S. Opoku, and J. Danso, "Measuring the performance of image contrast enhancement technique," *International Journal of Computer Applications*, vol. 181, no. 22, pp. 6–13, 2018.

[35] S. L. Herman and W. N. Alerich, *Industrial motor control*. Delmar Cengage Learning, 2010.

[36] S. Catálogo, "Movimiento lineal: Guías y husillos de bolas y rodillos," 2008.

[37] MCI, "Raspberry pi zero 2 w," 2023. [En línea]. Disponible: https://raspberrypi.cl/producto/raspberry-pi-zero-2-w/

[38] ArduCam, "Csi-to-hdmi adapter set for raspberry pi cameras," 2024. [En línea]. Disponible: https://www.arducam.com/product/arducam-csi-hdmi-cable-extension-module-15pin-60mm-fpc-cable-raspberry-pi

[39] H. Thimbleby, "Safer user interfaces: A case study in improving number entry," *IEEE Transactions on Software Engineering*, vol. 41, no. 7, pp. 711–729, 2015.

[40] I. DIS, "9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems," *International Standardization Organization (ISO). Switzerland*, 2009.

[41] E. Systems, "Esp32 series datashee," 2024. [En línea]. Disponible: https://www.espressif.com/sites/default/filesdocumentationesp32_datasheet_en.pdf

[42] R. Pi, "Raspberry pi 5," Obtubre 2023. [En línea]. Disponible: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf

[43] Sz-htc.com, "10.1 inch 1024x600 capactive touch hdmi display(hcik101v.cc)," 2024. [En línea]. Disponible: https://www.sz-htc.com/products_details/24.html

[44] B. M. Harwani, *Qt5 Python GUI Programming Cookbook: Building responsive and powerful cross-platform applications with PyQt*. Packt Publishing Ltd, 2018.

[45] S. S. y componentes, "Pintura conductora base de grafito." [En línea]. Disponible: www.sycelectronica.com.ar

[46] D. Materials., "High electrical conductivity carbon paste for use on perovskite solar cells." 2024. [En línea]. Disponible: https://www.dycotecmaterials.com/product/carbon-paste-perovskite/

[47] ALVEG., "EspecificaciÓn de venta xilol," 2024. [En línea]. Disponible: https://www.alveg.com.mx//

[48] N. O. Mexicana., "Nom-133-ssa1-1995, que establece las especificaciones sanitarias de las agujas hipodérmicas desechables." 29 de cotubre de 1998. [En línea]. Disponible: https://salud.gob.mx/unidades/cdi/nom/133ssa15.html

[49] TEKTRONIX, "2400 graphical series smu," 2024. [En línea]. Disponible: https://www.tek.com/en/products/keithley/source-measure-units/2400-graphical-series-sourcemeter.

[50] P. A. I. T. Instruments, "Solar radiation detector pce-spm 1." 2024. [En línea]. Disponible: https://www.pce-instruments.com/english/measuring-instruments/test-meters/radiation-detector-pce-instruments-solar-radiation-detector-pce-spm-1-det_62040.htm

Anexo

Código de la tarjeta de desarrollo Esp32

Listing 6.1: Comunicación y movimiento del sistema mecánico con la Esp32

```
#include <Arduino.h>
#include "BluetoothSerial.h"
// Define los pines para el motor 1 en el eje x
#define motor1StepPin 13
#define motor1DirPin 12
#define motor1EnablePin 35
// Define los pines para el motor 2 en eje y
#define motor2StepPin 14
#define motor2DirPin 27
#define motor2EnablePin 35
/// Define los pines para el motor 3 en eje z
#define motor3StepPin 26
#define motor3DirPin 25
#define motor3EnablePin 35
/// Define los pines para el motor 4 en sistema de inyeccion
#define motor4StepPin 33
#define motor4DirPin 32
#define motor4EnablePin 35
///Variables
byte        cmd       =  0;
byte        smd       =  0;
int         pasos_x   = 0, pasos_x_total = 0;
int         pasos_y   = 0, pasos_y_total = 0;
int         pasos_z   = 0, pasos_z_total = 0;
int         pasos_j   = 0, pasos_j_total = 0;
long pasos_x_original = 0, pasos_y_original = 0, pasos_z_original = 0,
    pasos_j_original = 0;
int         step_x    = 0;
int         step_y    = 0;
```

```
int         step_z    = 0;
int         step_j    = 0;
int         speed_x    = 50;
int         speed_y    = 50;
int         speed_z    = 50;
int         speed_j    = 50;
bool activar_x = false, activar_y = false;
bool activar_z = false, activar_j = false;
bool mover_x_der= false, mover_y_der= false, mover_z_der= false, mover_j_der
    = false;

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable
    it
#endif

BluetoothSerial SerialBT;

void setup() {
  // Inicializaci n del puerto serie
  Serial.begin(115200);
  pinMode(motor1StepPin, OUTPUT);
  pinMode(motor1DirPin, OUTPUT);
  pinMode(motor1EnablePin, OUTPUT);
  pinMode(motor2StepPin, OUTPUT);
  pinMode(motor2DirPin, OUTPUT);
  pinMode(motor2EnablePin, OUTPUT);
  pinMode(motor3StepPin, OUTPUT);
  pinMode(motor3DirPin, OUTPUT);
  pinMode(motor3EnablePin, OUTPUT);
  pinMode(motor4StepPin, OUTPUT);
  pinMode(motor4DirPin, OUTPUT);
  pinMode(motor4EnablePin, OUTPUT);

  // Crear tarea en N cleo 0
  xTaskCreatePinnedToCore(
    taskCore0,      // Funci n de la tarea
    "TaskCore0",    // Nombre de la tarea
    1000,           // Tama o de la pila
    NULL,           // Par metro para la tarea
    1,              // Prioridad
    NULL,           // Manejo de la tarea (opcional)
    0               // N cleo 0
```

```
  );

  // Crear tarea en N cleo 1
  xTaskCreatePinnedToCore(
    taskCore1,      // Funci n de la tarea
    "TaskCore1",    // Nombre de la tarea
    1000,           // Tama o de la pila
    NULL,           // Par metro para la tarea
    1,              // Prioridad
    NULL,           // Manejo de la tarea (opcional)
    1               // N cleo 1
  );

  SerialBT.begin("ESP32_INT"); //Nombre del dispositivo
  delay(500);
  Serial.println("El dispositivo se inici , Ahora puedes emparejarlo con
     bluetooth!");
}

int moveMotorDer(int stepPin, int dirPin, int& steps, int speed_motor) {
  digitalWrite(dirPin, HIGH); // Configura la direcci n del motor (cambia
     HIGH por LOW seg n la direcci n deseada)
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(speed_motor);  // Puedes ajustar este valor seg n la
     velocidad del motor
  digitalWrite(stepPin, LOW);
  delayMicroseconds(speed_motor);  // Puedes ajustar este valor seg n la
     velocidad del motor
  steps++;
  return steps;
}

int moveMotorIzq(int stepPin, int dirPin, int& steps, int speed_motor) {
  digitalWrite(dirPin, LOW); // Configura la direcci n del motor (cambia
     HIGH por LOW seg n la direcci n deseada)
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(speed_motor);  // Puedes ajustar este valor seg n la
     velocidad del motor
  digitalWrite(stepPin, LOW);
  delayMicroseconds(speed_motor);  // Puedes ajustar este valor seg n la
     velocidad del motor
  steps--;
  return steps;
```

```
}

void guardar(){
  digitalWrite(motor1EnablePin, LOW);
  digitalWrite(motor2EnablePin, LOW);
  digitalWrite(motor3EnablePin, LOW);
  digitalWrite(motor4EnablePin, LOW);
  pasos_x_original = -step_x;
  pasos_y_original = -step_y;
  pasos_z_original = -step_z;
  pasos_j_original = -step_j;
  Serial.println("Se guardo los valorese.");
  pasos_x_total = 0;
  pasos_y_total = 0;
  pasos_z_total = 0;
  pasos_j_total = 0;
  Serial.println("Pasos totales a cero");
}
// Funci n de la tarea 1 (N cleo 0)
void taskCore1(void *parameter) {
while (true) {
    if (SerialBT.available()) {
    cmd = 0;
    cmd = SerialBT.read();
    if (cmd > 31)
    {
      switch(cmd)
      {
        //Conexion exitoda
        case 'a' : SerialBT.print("controlarme."); Serial.println(""); break
            ;
        case 'A' : SerialBT.print("controlarme."); Serial.println(""); break
            ;
        //motor x
        case 'r' : pasos_x   = SerialBT.parseFloat(); pasos_x_total =
            pasos_x_total + pasos_x; digitalWrite(motor1EnablePin, HIGH);
            SerialBT.print("Motor avanca en el eje x."); SerialBT.println(
            pasos_x_total);  break;
        case 'l' : pasos_x   = SerialBT.parseFloat(); pasos_x_total =
            pasos_x_total - pasos_x; digitalWrite(motor1EnablePin, HIGH);
            SerialBT.print("Motor avanca en el eje x."); SerialBT.println(
            pasos_x_total);  break;
        //motor y
```

```
      case 'b' : pasos_y   = SerialBT.parseFloat(); pasos_y_total =
         pasos_y_total - pasos_y; digitalWrite(motor2EnablePin, HIGH);
         SerialBT.print("Motor avanca en el eje y."); SerialBT.println(
         pasos_y_total); break;
      case 'd' : pasos_y   = SerialBT.parseFloat(); pasos_y_total =
         pasos_y_total + pasos_y; digitalWrite(motor2EnablePin, HIGH);
         SerialBT.print("Motor avanca en el eje y."); SerialBT.println(
         pasos_y_total); break;
      //motor z
      case 'n' : pasos_z   = SerialBT.parseFloat(); pasos_z_total =
         pasos_z_total - pasos_z; digitalWrite(motor3EnablePin, HIGH);
         SerialBT.print("Motor avanca en el eje z."); SerialBT.println(
         pasos_z_total); break;
      case 'm' : pasos_z   = SerialBT.parseFloat(); pasos_z_total =
         pasos_z_total + pasos_z; digitalWrite(motor3EnablePin, HIGH);
         SerialBT.print("Motor avanca en el eje z."); SerialBT.println(
         pasos_z_total); break;
      case 'v' : pasos_j   = SerialBT.parseFloat(); pasos_j_total =
         pasos_j_total - pasos_j; digitalWrite(motor4EnablePin, HIGH);
         SerialBT.print("Motor avanca en el sistema de inyecci n.");
         SerialBT.println(pasos_j_total); break;
      case 'w' : pasos_j   = SerialBT.parseFloat(); pasos_j_total =
         pasos_j_total + pasos_j; digitalWrite(motor4EnablePin, HIGH);
         SerialBT.print("Motor avanca en el sistema de inyecci n.");
         SerialBT.println(pasos_j_total); break;
      case 'h' : guardar();  SerialBT.print("Ir a casa"); break;
      case 'c' : step_x = 0; break;
      case 'k' : step_y = 0; break;
      case 'u' : step_z = 0; break;
      case 'j' : step_j = 0; break;
      case 's' : speed_x = SerialBT.parseFloat();  break;
      case 'f' : speed_y = SerialBT.parseFloat();  break;
      case 'q' : speed_z = SerialBT.parseFloat();  break;
      case 'o' : speed_j = SerialBT.parseFloat();  break;
      case 'p' : guardar(); Serial.println("Paro total"); SerialBT.println
         ("Motor paro."); step_z = 0; break;
    }
   }
  }
//////////////Final de comunicaci n bluethooth////////////////
    if (Serial.available()) {
    smd = 0;
    smd = Serial.read();
```

```
if (smd > 31)
  {
   //Serial.println(smd);
   switch(smd)
  {
   case 'a' : Serial.print("Conexi n exitosa."); break;
   case 'A' : SerialBT.print("Sistema desconectado."); Serial.println("
      "); break;
   //motor x
   case 'r' : pasos_x   = Serial.parseFloat(); pasos_x_total =
      pasos_x_total + pasos_x; digitalWrite(motor1EnablePin, HIGH);
      Serial.print("Motor avanca en el eje x."); Serial.println(
      pasos_x_total);  break;
   case 'l' : pasos_x   = Serial.parseFloat(); pasos_x_total =
      pasos_x_total - pasos_x; digitalWrite(motor1EnablePin, HIGH);
      Serial.print("Motor avanca en el eje x."); Serial.println(
      pasos_x_total);  break;
   //motor y
   case 'b' : pasos_y   = Serial.parseFloat(); pasos_y_total =
      pasos_y_total - pasos_y; digitalWrite(motor2EnablePin, HIGH);
      Serial.print("Motor avanca en el eje y."); Serial.println(
      pasos_y_total); break;
   case 'd' : pasos_y   = Serial.parseFloat(); pasos_y_total =
      pasos_y_total + pasos_y; digitalWrite(motor2EnablePin, HIGH);
      Serial.print("Motor avanca en el eje y."); Serial.println(
      pasos_y_total); break;
   //motor z
   case 'm' : pasos_z   = Serial.parseFloat(); pasos_z_total =
      pasos_z_total - pasos_z; digitalWrite(motor3EnablePin, HIGH);
      Serial.print("Motor avanca en el eje z."); Serial.println(
      pasos_z_total); break;
   case 'n' : pasos_z   = Serial.parseFloat(); pasos_z_total =
      pasos_z_total + pasos_z; digitalWrite(motor3EnablePin, HIGH);
      Serial.print("Motor avanca en el eje z."); Serial.println(
      pasos_z_total); break;
   case 'w' : pasos_j   = Serial.parseFloat(); pasos_j_total =
      pasos_j_total - pasos_j; digitalWrite(motor4EnablePin, HIGH);
      Serial.print("Motor avanca en el sistema de inyecci n."); Serial
      .println(pasos_j_total); break;
   case 'v' : pasos_j   = Serial.parseFloat(); pasos_j_total =
      pasos_j_total + pasos_j; digitalWrite(motor4EnablePin, HIGH);
      Serial.print("Motor avanca en el sistema de inyecci n."); Serial
      .println(pasos_j_total); break;
```

```
        case 'h' : pasos_x_total = pasos_x_original; pasos_y_total =
            pasos_y_original;  pasos_z_total = pasos_z_original;
            pasos_j_total = pasos_j_original; digitalWrite(motor1EnablePin,
            HIGH); digitalWrite(motor2EnablePin, HIGH); Serial.print("Ir a
            casa"); break;
        case 'c' : pasos_x_total =0;  pasos_x_original= -step_x; step_x = 0;
             break;
        case 'k' : pasos_y_total =0;  pasos_y_original= -step_y; step_y = 0;
             break;
        case 'u' : pasos_z_total =0;  pasos_z_original= -step_z; step_z = 0;
             break;
        case 'j' : pasos_j_total =0;  pasos_j_original= -step_j; step_x = 0;
             break;
        //Procesamiento de imagen
        case 'F' :  Serial.println("La imagen se a procesado correctamente."
            ); break;
        case 'T' :  Serial.println("Comenzar el depsoito de material.");
            break;
        case 's' : speed_x = Serial.parseFloat();  Serial.print("Velocidad
            eje X: "); Serial.println(pasos_j_total); break;
        case 'f' : speed_y = Serial.parseFloat();  Serial.print("Velocidad
            eje Y: "); Serial.println(pasos_j_total); break;
        case 'q' : speed_z = Serial.parseFloat();  Serial.print("Velocidad
            eje Z: "); Serial.println(pasos_j_total); break;
        case 'o' : speed_j = Serial.parseFloat();  Serial.print("Velocidad
            del sistema de inyecci n: "); Serial.println(pasos_j_total);
            break;
        case 'p' : guardar(); SerialBT.println("Motor paro."); step_z = 0;
            Serial.println("Alto total");break;
      }
     }
    }
  }//final de while(true)
}
// Funci n de la tarea 2 (N cleo 1)
void taskCore0(void *parameter) {
  while (true) {
   if (pasos_x_total == 0 && pasos_y_total == 0 && pasos_z_total == 0 &&
       pasos_j_total == 0) {
      delay(20);
      continue;
    }
 ////////////////////motor en eje x//////////////////////
```

```
  if (pasos_x_total > step_x)
{
  step_x = moveMotorDer(motor1StepPin, motor1DirPin, step_x, speed_x);
  if (step_x >= pasos_x_total) {
      digitalWrite(motor1EnablePin, LOW);
      Serial.println("Motor alcanz el n mero de pasos, se detiene 6.");
      SerialBT.println("Motor alcanz el n mero de pasos, se detiene 8."
          );
      mover_x_der = false;
  }
}
else if (pasos_x_total < step_x)
{
  step_x = moveMotorIzq(motor1StepPin, motor1DirPin, step_x, speed_x);
  if (step_x <= pasos_x_total) {
      digitalWrite(motor1EnablePin, LOW);
      Serial.println("Motor alcanz el n mero de pasos, se detiene. 9");
      SerialBT.println("Motor alcanz el n mero de pasos, se detiene. 10
          ");
      mover_x_der = false;
  }
}
////////////////////motor en eje y//////////////////////////
if (pasos_y_total > step_y)
{
  step_y = moveMotorDer(motor2StepPin, motor2DirPin, step_y, speed_y);
  if (step_y >= pasos_y_total) {
      digitalWrite(motor2EnablePin, LOW);
      Serial.println("Motor alcanz el n mero de pasos, se detiene.");
      SerialBT.println("Motor alcanz el n mero de pasos, se detiene.");
      mover_y_der = false;
  }
}
else if (pasos_y_total < step_y)
{
  step_y = moveMotorIzq(motor2StepPin, motor2DirPin, step_y, speed_y);
  if (step_y <= pasos_y_total) {
      digitalWrite(motor2EnablePin, LOW);
      Serial.println("Motor alcanz el n mero de pasos, se detiene.");
      SerialBT.println("Motor alcanz el n mero de pasos, se detiene.");
      mover_y_der = false;
  }
}
```

```
//////////////////////////motor en el eje z ////////////////////////////////
  if (pasos_z_total > step_z)
  {
    step_z = moveMotorDer(motor3StepPin, motor3DirPin, step_z, speed_z);
    if (step_z >= pasos_z_total) {
        digitalWrite(motor3EnablePin, LOW);
        Serial.println("Motor alcanz el n mero de pasos, se detiene.");
        SerialBT.println("Motor alcanz el n mero de pasos, se detiene.");
        mover_z_der = false;
    }
  }
  else if (pasos_z_total < step_z)
  {
    step_z = moveMotorIzq(motor3StepPin, motor3DirPin, step_z, speed_z);
    if (step_z <= pasos_z_total) {
        digitalWrite(motor3EnablePin, LOW);
        Serial.println("Motor alcanz el n mero de pasos, se detiene.");
        SerialBT.println("Motor alcanz el n mero de pasos, se detiene.");
        mover_z_der = false;
    }
  }
//////////////////////////motor del sistema de inyecci n
   ////////////////////
  if (pasos_j_total > step_j)
  {
    step_j = moveMotorDer(motor4StepPin, motor4DirPin, step_j, speed_j);
    if (step_j >= pasos_j_total) {
        digitalWrite(motor4EnablePin, LOW);
        Serial.println("Motor alcanz el n mero de pasos, se detiene.");
        SerialBT.println("Motor alcanz el n mero de pasos, se detiene.");
        mover_j_der = false;
    }
  }
  else if (pasos_j_total < step_j)
  {
    step_j = moveMotorIzq(motor4StepPin, motor4DirPin, step_j, speed_j);
    if (step_j <= pasos_j_total) {
        digitalWrite(motor4EnablePin, LOW);
        Serial.println("Motor alcanz el n mero de pasos, se detiene.");
        SerialBT.println("Motor alcanz el n mero de pasos, se detiene.");
        mover_j_der = false;
    }
  }
```

```
////////////////////////////////////////////////////////////////
  delay(20);
  }
}
void loop() {
  // El loop se ejecuta en el n cleo por defecto (generalmente N cleo 1)
}
```

Código de la tarjeta de desarrollo Raspberry Pi 5

Listing 6.2: Código Main de la interfaz, la comunicación y el procesamiento de imagen

```
from PyQt5 import QtWidgets, uic,  QtCore, QtGui

import sys, platform, os
from PyQt5.QtCore import *
from PyQt5.QtWidgets import QMainWindow, QApplication, QPushButton, QLabel,
    QWidget, QTabWidget, QVBoxLayout, QGridLayout
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QFileDialog
from PyQt5.QtGui import *
from picamera2 import Picamera2
from picamera2.previews.qt import QGlPicamera2
from picamera2 import __name__ as picamera2_name
from libcamera import controls
from function import Function_UI
from segunda import Ui_segunda
import serial, serial.tools.list_ports
#import imutils
import cv2
import sys
import numpy as np
import subprocess
import time
from importlib.metadata import version
os.environ["LIBCAMERA_LOG_LEVELS"] = "3"
picam2 = Picamera2()
#=============================================
preview_width= 900
preview_height = int(picam2.sensor_resolution[1] * preview_width/picam2.
    sensor_resolution[0])
preview_config_raw = picam2.create_preview_configuration(main={"size": (
```

```
    preview_width, preview_height)},
                                                    raw={"size": picam2
                                                        .
                                                        sensor_resolution
                                                        })
picam2.configure(preview_config_raw)
#=============================================
picam2.set_controls({"AfMode": controls.AfModeEnum.Continuous})
#=============================================
try:
    import cv2
    cv_present = True
except ImportError:
    cv_present = False
    print("OpenCV not found - HDR not available")
import threading

import numpy as np

class MyApp(QMainWindow):
        def __init__(self):
                super(MyApp, self).__init__()
                loadUi('inter_faz_12.ui', self)

                self.image_label = QLabel(self)
                self.image_label.setGeometry(10, 500, 640, 480)  # Ajusta
                    las dimensiones seg n necesites
                self.serial = Function_UI()
                self.serialPort = serial.Serial()
                self.baud_List.addItems(self.serial.baudList)
                self.baud_List.setCurrentText('115200')
                self.update_ports()
                self.connect_Button.clicked.connect(self.connect_serial)
                self.send_Button.clicked.connect(self.send_data)
                self.clear_Button.clicked.connect(self.clear)
                self.update_Button.clicked.connect(self.update_ports)
                self.serial.data_available.connect(self.update_view)
                ################Botones de la interfaz
                self.Button_mas_Y.clicked.connect(self.subir_y)
                self.Button_menos_Y.clicked.connect(self.bajar_y)
                self.Button_mas_X.clicked.connect(self.subir_x)
                self.Button_menos_X.clicked.connect(self.bajar_x)
                self.Button_mas_Z.clicked.connect(self.subir_z)
```

```
        self.Button_menos_Z.clicked.connect(self.bajar_z)
        self.Button_PARO.clicked.connect(self.paro_total)
        self.Button_home_XY.clicked.connect(self.ir_a_casa)
        self.Button_camara.clicked.connect(self.segunda_ventana)
        self.Button_mas_J.clicked.connect(self.subir_j)
        self.Button_menos_J.clicked.connect(self.bajar_j)
        self.Button_home_Z.clicked.connect(self.ir_a_casa_z)
        self.Button_home_J.clicked.connect(self.ir_a_casa_j)
        self.Button_asignar_home_X.clicked.connect(self.new_casa_x)
        self.Button_asignar_home_Y.clicked.connect(self.new_casa_y)
        self.Button_asignar_home_Z.clicked.connect(self.new_casa_z)
        self.Button_asignar_home_J.clicked.connect(self.new_casa_j)
        self.Button_procesamiento.clicked.connect(self.
            procesamiento_imagen)
        self.Button_Inicio.clicked.connect(self.comenzar)

def update_view(self,data):
        print(data)
        self.textBrowser.append(data)

def connect_serial(self):
        if(self.connect_Button.isChecked()):
                port = self.port_List.currentText()
                baud = self.baud_List.currentText()
                self.serial.serialPort.port = port
                self.serial.serialPort.baudrate = baud
                self.serial.connect_serial()
                if(self.serial.serialPort.is_open):
                        self.connect_Button.setText("Desconectar")
                        self.serial.send_data("a")
        else:
                self.serial.disconnect_serial()
                self.connect_Button.setText("Conectar")
                self.serial.send_data("A")

def send_data(self):
        data_send = self.send_Text.toPlainText()
        self.serial.send_data(data_send)

def update_ports(self):
        self.serial.update_port()
        self.port_List.clear()
        self.port_List.addItems(self.serial.portList)
```

```
def clear (self):
        self.textBrowser.clear()
# def show(self):
#       # command to runta
#       self.main_win.show()

def subir_y(self):
        pasos = self.spinBox_Pasos_XY.value()
        velocidad = self.spinBox_Velocidad_XY.value()
        lista_pasos = "d" + str(pasos)
        #lista_pasos = "d200"
        #lista_velocidad = "v" + str(velocidad)
        #self.serial.send_data(lista_velocidad)
        self.serial.send_data(lista_pasos)

def bajar_y(self):
        pasos = self.spinBox_Pasos_XY.value()
        velocidad = self.spinBox_Velocidad_XY.value()
        lista_pasos = "b" + str(pasos)
        #lista_pasos = "b200"
        #lista_velocidad = "v" + str(velocidad)
        #self.serial.send_data(lista_velocidad)
        self.serial.send_data(lista_pasos)

def subir_x(self):
        pasos = self.spinBox_Pasos_XY.value()
        #velocidad = self.spinBox_Velocidad_XY.value()
        lista_pasos = "r" + str(pasos)
        #lista_pasos = "r200"
        #lista_velocidad = "s" + str(velocidad)
        #self.serial.send_data(lista_velocidad)
        self.serial.send_data(lista_pasos)

def bajar_x(self):
        pasos = self.spinBox_Pasos_XY.value()
        #velocidad = self.spinBox_Velocidad_XY.value()
        lista_pasos = "l" + str(pasos)
        #lista_pasos = "l200"
        #lista_velocidad = "s" + str(velocidad)
        #self.serial.send_data(lista_velocidad)
        self.serial.send_data(lista_pasos)
```

```
def subir_z(self):
        pasos = self.spinBox_Pasos_Z.value()
        lista_pasos = "n" + str(pasos)
        #data_send = "n200"
        self.serial.send_data(lista_pasos)

def bajar_z(self):
        pasos = self.spinBox_Pasos_Z.value()
        lista_pasos = "m" + str(pasos)
        #data_send = "n200"
        self.serial.send_data(lista_pasos)

def subir_j(self):
        pasos = self.spinBox_Pasos_Z.value()
        velocidad = self.spinBox_Velocidad_XY.value()
        lista_pasos = "v" + str(pasos)
        #lista_pasos = "v200"
        #lista_velocidad = "v" + str(velocidad)
        #self.serial.send_data(lista_velocidad)
        self.serial.send_data(lista_pasos)

def bajar_j(self):
        pasos = self.spinBox_Pasos_Z.value()
        velocidad = self.spinBox_Velocidad_XY.value()
        lista_pasos = "w" + str(pasos)
        #lista_pasos = "w200"
        #lista_velocidad = "v" + str(velocidad)
        #self.serial.send_data(lista_velocidad)
        self.serial.send_data(lista_pasos)

def paro_total(self):
        data_send = "p"
        self.serial.send_data(data_send)

def ir_a_casa(self):
        data_send = "h"
        self.serial.send_data(data_send)

def ir_a_casa_j(self):
        data_send = "h"
        self.serial.send_data(data_send)

def ir_a_casa_z(self):
```

```
        data_send = "h"
        self.serial.send_data(data_send)

def new_casa_x(self):
        data_send = "c"
        self.serial.send_data(data_send)

def new_casa_y(self):
        data_send = "k"
        self.serial.send_data(data_send)

def new_casa_z(self):
        data_send = "u"
        self.serial.send_data(data_send)

def new_casa_j(self):
        data_send = "j"
        self.serial.send_data(data_send)

def procesamiento_imagen(self):
        data_send = "F"
        self.serial.send_data(data_send)
        """Captura una imagen y la muestra en la interfaz principal.
            """
        cfg = picam2.create_still_configuration()
         # Generar la ruta con el timestamp
        timestamp = time.strftime("%Y%m%d-%H%M%S")
        image_path = f"/home/pi5/project/camara/img_{timestamp}.jpg"
        picam2.switch_mode_and_capture_file(cfg, image_path)
        pixmap = QPixmap(image_path)
        self.image_label.setPixmap(pixmap.scaled(self.image_label.
            size(), QtCore.Qt.KeepAspectRatio))
        print(f"Imagen capturada y mostrada: {image_path}")

def comenzar(self):
        data_send = "T"
        self.serial.send_data(data_send)

def segunda_ventana(self):

        self.otraventana = SegundaVentana(self)

        self.otraventana.show()
```

```
class SegundaVentana(QMainWindow):
        def __init__(self, parent = None):
                super(SegundaVentana, self).__init__(parent)
                self.main_window = None
                self.title = __file__
                self.left = 0
                self.top = 0
                self.setWindowTitle(self.title)
                self.setGeometry(self.left, self.top, 1000, 514)
                self.main_widget = MyMainWidget(self)
                self.setCentralWidget(self.main_widget)
                self.show()

class MyMainWidget(QWidget):

    def read_f(self, file):
        with open(file, encoding='UTF-8') as reader:
            content = reader.read()
        return content

    def read_pretty_name(self):
        with open("/etc/os-release") as f:
            os_release = {}
            for line in f:
                k,v = line.rstrip().split("=")
                os_release[k] = v.strip('"')
        return os_release['PRETTY_NAME']

    def on_Capture_Clicked(self):
        global picam2
        self.btnCapture.setEnabled(False)
        cfg = picam2.create_still_configuration()
        timeStamp = time.strftime("%Y%m%d-%H%M%S")
        targetPath="/home/pi5/project/camara/img_"+timeStamp+".jpg"
        print("- Capture image:", targetPath)
        gruardar = True
        picam2.switch_mode_and_capture_file(cfg, targetPath,signal_function=
            self.qpicamera2.signal_done)

    def capture_done(self, job):
        global picam2
        result = picam2.wait(job)
```

```
        self.btnCapture.setEnabled(True)
        print("- capture_done.")

    def __init__(self, parent):
        super(QWidget, self).__init__(parent)
        self.layout = QVBoxLayout()
        # Initialize tab screen
        self.tabs = QTabWidget()
        self.tabCapture = QWidget()
        self.tabInfo = QWidget()
        # Add tabs
        self.tabs.addTab(self.tabCapture,"Capture")
        self.tabs.addTab(self.tabInfo,"  Info  ")
        #=== Tab Capture ===
        # Create first tab
        self.tabCapture.layout = QVBoxLayout()
        self.tabCapture.setAutoFillBackground(True)
        palette = self.palette()
        palette.setColor(QPalette.Window, QColor('black'))
        self.tabCapture.setPalette(palette)
        #Prepare Preview
        self.qpicamera2 = QGlPicamera2(picam2,
                          width=preview_width, height=preview_height,
                          keep_ar=True)
        self.tabCapture.layout.addWidget(self.qpicamera2)
        self.qpicamera2.done_signal.connect(self.capture_done)
        self.btnCapture = QPushButton("Capturar imagen ")
        self.btnCapture.setFont(QFont("Helvetica", 20, QFont.Bold))
        self.btnCapture.clicked.connect(self.on_Capture_Clicked)
        self.tabCapture.layout.addWidget(self.btnCapture)
        self.tabCapture.setLayout(self.tabCapture.layout)
        #=== Tab Info ===
        self.tabInfo.layout = QVBoxLayout()
        infoGridLayout = QGridLayout()
        rowSpan = 1
        columnSpan0 = 1
        columnSpan1 = 5
        infoGridLayout.addWidget(QLabel('Python', self), 0, 0, rowSpan,
            columnSpan0)
        infoGridLayout.addWidget(QLabel(platform.python_version(), self), 0,
             1, rowSpan, columnSpan1)
        infoGridLayout.addWidget(QLabel(picamera2_name, self), 1, 0, rowSpan
            , columnSpan0)
```

```
infoGridLayout.addWidget(QLabel(version(picamera2_name), self), 1,
    1, rowSpan, columnSpan1)
infoGridLayout.addWidget(QLabel('␣', self), 2, 0, rowSpan,
    columnSpan0)
infoGridLayout.addWidget(QLabel('Camera␣Module:', self), 3, 0,
    rowSpan, columnSpan0)
cam_properties = picam2.camera_properties
cam_Model = cam_properties['Model']
infoGridLayout.addWidget(QLabel('Model', self), 4, 0, rowSpan,
    columnSpan0)
infoGridLayout.addWidget(QLabel(cam_Model, self), 4, 1, rowSpan,
    columnSpan1)
cam_PixelArraySize = str(cam_properties['PixelArraySize'][0]) + "␣x␣
    " + str(cam_properties['PixelArraySize'][1])
infoGridLayout.addWidget(QLabel('PixelArraySize', self), 5, 0,
    rowSpan, columnSpan0)
infoGridLayout.addWidget(QLabel(cam_PixelArraySize, self), 5, 1,
    rowSpan, columnSpan1)
infoGridLayout.addWidget(QLabel('␣', self), 6, 0, rowSpan,
    columnSpan0)
infoGridLayout.addWidget(QLabel('Machine:', self), 7, 0, rowSpan,
    columnSpan0)
infoGridLayout.addWidget(QLabel('Board', self), 8, 0, rowSpan,
    columnSpan0, Qt.AlignTop)
board_def = "/proc/device-tree/model"
board_info = self.read_f("/proc/device-tree/model") +"\n(" +
    board_def +")"
infoGridLayout.addWidget(QLabel(board_info, self), 8, 1, rowSpan,
    columnSpan0)
infoGridLayout.addWidget(QLabel('OS', self), 9, 0, rowSpan,
    columnSpan0, Qt.AlignTop)
os_info = self.read_pretty_name() + "\n" + os.uname()[3]
infoGridLayout.addWidget(QLabel(os_info, self), 9, 1, rowSpan,
    columnSpan1)
self.tabInfo.layout.addLayout(infoGridLayout)
self.tabInfo.layout.addStretch()
self.tabInfo.setLayout(self.tabInfo.layout)
#=============================
# Add tabs to widget
self.layout.addWidget(self.tabs)
self.setLayout(self.layout)
picam2.start()
```

```
app = QApplication(sys.argv)

main = MyApp()
main.show()
sys.exit(app.exec_())
# if __name__ == '__main__':
#         app = QApplication(sys.argv)
#         my_app = MyApp()
#         my_app.show()
#         sys.exit(app.exec_())
```

Código de la interfaz de Qt

Listing 6.3: Código de la interfaz de Qt en pytho

```
#
    ################################################################################

## Form generated from reading UI file 'inter_faz_12wSzXjl.ui'
##
## Created by: Qt User Interface Compiler version 6.6.0
##
## WARNING! All changes made in this file will be lost when recompiling UI
    file!
#
    ################################################################################


from PySide6.QtCore import (QCoreApplication, QDate, QDateTime, QLocale,
    QMetaObject, QObject, QPoint, QRect,
    QSize, QTime, QUrl, Qt)
from PySide6.QtGui import (QBrush, QColor, QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage, QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient, QTransform)
from PySide6.QtWidgets import (QApplication, QComboBox, QFrame, QGridLayout,
    QHBoxLayout, QLabel, QMainWindow, QPushButton,
    QSizePolicy, QSpacerItem, QSpinBox, QTextBrowser,
    QTextEdit, QVBoxLayout, QWidget)
```

```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        if not MainWindow.objectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.setEnabled(True)
        MainWindow.resize(1030, 500)
        font = QFont()
        font.setPointSize(16)
        font.setBold(False)
        font.setHintingPreference(QFont.PreferFullHinting)
        MainWindow.setFont(font)
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.verticalLayout_4 = QVBoxLayout(self.centralwidget)
        self.verticalLayout_4.setSpacing(0)
        self.verticalLayout_4.setObjectName(u"verticalLayout_4")
        self.verticalLayout_4.setContentsMargins(0, 0, 0, 0)
        self.frame = QFrame(self.centralwidget)
        self.frame.setObjectName(u"frame")
        self.frame.setFrameShape(QFrame.StyledPanel)
        self.frame.setFrameShadow(QFrame.Raised)
        self.verticalLayout_5 = QVBoxLayout(self.frame)
        self.verticalLayout_5.setSpacing(0)
        self.verticalLayout_5.setObjectName(u"verticalLayout_5")
        self.verticalLayout_5.setContentsMargins(0, 0, 0, 0)
        self.frame_2 = QFrame(self.frame)
        self.frame_2.setObjectName(u"frame_2")
        self.frame_2.setMaximumSize(QSize(16767214, 16767215))
        font1 = QFont()
        font1.setFamilies([u"SolidWorks_GDT"])
        font1.setPointSize(16)
        font1.setBold(False)
        font1.setHintingPreference(QFont.PreferFullHinting)
        self.frame_2.setFont(font1)
        self.frame_2.setFrameShape(QFrame.StyledPanel)
        self.frame_2.setFrameShadow(QFrame.Raised)
        self.verticalLayout = QVBoxLayout(self.frame_2)
        self.verticalLayout.setSpacing(0)
        self.verticalLayout.setObjectName(u"verticalLayout")
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.frame_28 = QFrame(self.frame_2)
        self.frame_28.setObjectName(u"frame_28")
        self.frame_28.setFrameShape(QFrame.StyledPanel)
```

```
self.frame_28.setFrameShadow(QFrame.Raised)

self.verticalLayout.addWidget(self.frame_28)

self.frame_4 = QFrame(self.frame_2)
self.frame_4.setObjectName(u"frame_4")
self.frame_4.setFrameShape(QFrame.StyledPanel)
self.frame_4.setFrameShadow(QFrame.Raised)
self.verticalLayout_2 = QVBoxLayout(self.frame_4)
self.verticalLayout_2.setSpacing(1)
self.verticalLayout_2.setObjectName(u"verticalLayout_2")
self.verticalLayout_2.setContentsMargins(1, 1, 1, 1)
self.frame_6 = QFrame(self.frame_4)
self.frame_6.setObjectName(u"frame_6")
self.frame_6.setFrameShape(QFrame.StyledPanel)
self.frame_6.setFrameShadow(QFrame.Raised)
self.horizontalLayout_4 = QHBoxLayout(self.frame_6)
self.horizontalLayout_4.setSpacing(0)
self.horizontalLayout_4.setObjectName(u"horizontalLayout_4")
self.horizontalLayout_4.setContentsMargins(0, 0, 0, 1)
self.horizontalSpacer = QSpacerItem(206, 20, QSizePolicy.Expanding,
    QSizePolicy.Minimum)

self.horizontalLayout_4.addItem(self.horizontalSpacer)

self.label = QLabel(self.frame_6)
self.label.setObjectName(u"label")

self.horizontalLayout_4.addWidget(self.label)

self.horizontalSpacer_2 = QSpacerItem(205, 20, QSizePolicy.Expanding
    , QSizePolicy.Minimum)

self.horizontalLayout_4.addItem(self.horizontalSpacer_2)


self.verticalLayout_2.addWidget(self.frame_6)

self.frame_7 = QFrame(self.frame_4)
self.frame_7.setObjectName(u"frame_7")
self.frame_7.setEnabled(True)
self.frame_7.setFrameShape(QFrame.StyledPanel)
self.frame_7.setFrameShadow(QFrame.Raised)
```

```
        self.horizontalLayout_3 = QHBoxLayout(self.frame_7)
        self.horizontalLayout_3.setSpacing(3)
        self.horizontalLayout_3.setObjectName(u"horizontalLayout_3")
        self.horizontalLayout_3.setContentsMargins(0, 0, 1, 1)
        self.label_2 = QLabel(self.frame_7)
        self.label_2.setObjectName(u"label_2")
        self.label_2.setFont(font1)

        self.horizontalLayout_3.addWidget(self.label_2)

        self.port_List = QComboBox(self.frame_7)
        self.port_List.setObjectName(u"port_List")

        self.horizontalLayout_3.addWidget(self.port_List)

        self.label_3 = QLabel(self.frame_7)
        self.label_3.setObjectName(u"label_3")
        self.label_3.setFont(font1)

        self.horizontalLayout_3.addWidget(self.label_3)

        self.baud_List = QComboBox(self.frame_7)
        self.baud_List.setObjectName(u"baud_List")

        self.horizontalLayout_3.addWidget(self.baud_List)

        self.connect_Button = QPushButton(self.frame_7)
        self.connect_Button.setObjectName(u"connect_Button")
        font2 = QFont()
        font2.setFamilies([u"Segoe UI"])
        font2.setPointSize(20)
        font2.setBold(False)
        font2.setHintingPreference(QFont.PreferFullHinting)
        self.connect_Button.setFont(font2)
        self.connect_Button.setStyleSheet(u"#connect_Button:checked{\n"
"background-color: rgb(0, 255, 127);\n"
"}")
        self.connect_Button.setCheckable(True)

        self.horizontalLayout_3.addWidget(self.connect_Button)

        self.update_Button = QPushButton(self.frame_7)
        self.update_Button.setObjectName(u"update_Button")
```

```
font3 = QFont()
font3.setFamilies([u"SolidWorks_GDT"])
font3.setPointSize(22)
font3.setBold(False)
font3.setHintingPreference(QFont.PreferFullHinting)
self.update_Button.setFont(font3)

self.horizontalLayout_3.addWidget(self.update_Button)

self.horizontalLayout_3.setStretch(0, 1)
self.horizontalLayout_3.setStretch(1, 2)
self.horizontalLayout_3.setStretch(2, 1)
self.horizontalLayout_3.setStretch(3, 2)
self.horizontalLayout_3.setStretch(4, 2)
self.horizontalLayout_3.setStretch(5, 2)

self.verticalLayout_2.addWidget(self.frame_7)


self.verticalLayout.addWidget(self.frame_4)
```