



DIVISIÓN DE ESTUDIOS POSGRADO E INVESTIGACIÓN

Opción 1.- TESIS

TRABAJO PROFESIONAL

**PROTOTIPO DE POSICIONADOR DE INVERNADERO VERTICAL
CON SISTEMA DE PROCESAMIENTO DE IMÁGENES PARA
COLOCACIÓN DE SUSTRATO CON SEMILLA PARA LA
OBTENCIÓN DE PLANTULA**

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN
INGENIERÍA ELECTRÓNICA**

PRESENTA:

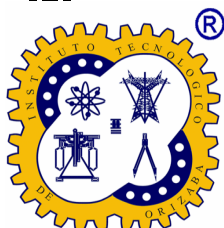
ING. BRANDON FALCO GARCÍA PEÑUELA

DIRECTOR DE TESIS:

DR. IGNACIO HERRERA AGUILAR

CO DIRECTOR DE TESIS:

DR. GERARDO AGUILA RODRIGUEZ





Instituto Tecnológico de Orizaba
Departamento de División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, **3/febrero/2026**
Dependencia: **División de Estudios de
Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: I

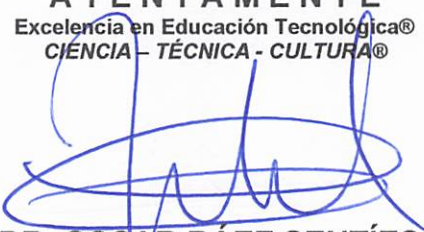
C. BRANDON FALCO GARCÍA PEÑUELA
Candidato(a) a Grado de Maestro(a) en:
INGENIERÍA ELECTRÓNICA
P R E S E N T E.-

De acuerdo con el Reglamento de Titulación vigente de los Centros e Institutos Tecnológicos Federales del Tecnológico Nacional de México, de la Secretaría de Educación Pública, y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

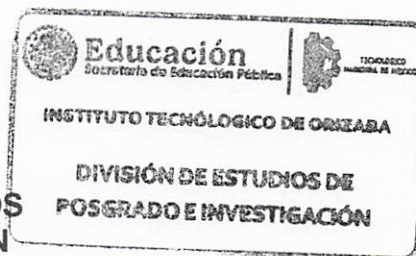
" PROTOTIPO DE POSICIONADOR DE INVERNADERO VERTICAL CON SISTEMA DE PROCESAMIENTO DE IMÁGENES PARA COLOCACIÓN DE SUSTRATO CON SEMILLA PARA LA OBTENCIÓN DE PLANTULA QUÍMICAS".

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®



DR. OSCAR BÁEZ SENTÍES
**JEFE DE LA DIVISIÓN DE ESTUDIOS DE
DE POSGRADO E INVESTIGACIÓN**



OG-13-F06



2026
año de
**Margarita
Maza**





Educación

Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Orizaba

Departamento de División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, 8/enero/2026

Asunto: **Revisión de trabajo escrito**

C. OFELIA LANDETA ESCAMILLA
JEFA DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
PRESENTE.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

BRANDON FALCO GARCÍA PEÑUELA

La cual lleva el título de:

**“PROTOTIPO DE POSICIONADOR DE INVERNADERO VERTICAL CON SISTEMA DE PROCESAMIENTO
DE IMÁGENES PARA COLOCACIÓN DE SUSTRATO CON SEMILLA PARA LA OBTENCIÓN DE
PLANTULA”**

Y concluyen que se acepta.

ATENTAMENTE

Excelencia en Educación Tecnológica®
CIENCIA – TÉCNICA - CULTURA®

PRESIDENTE: DR. IGNACIO HERRERA AGUILAR

FIRMA

SECRETARIO: DR. GERARDO ÁGUILA RODRÍGUEZ

FIRMA

VOCAL: M.C. ENRIQUE ALEJANDRO CUELLAR
CORTÉS

FIRMA

VOCAL SUP.: DR. JOSÉ PASTOR RODRÍGUEZ JARQUIN

FIRMA

TA-09-18



2026
año de
**Margarita
Maza**



CONTENIDO

AGRADECIMIENTOS	9
RESUMEN	10
ABSTRACT	12
INTRODUCCIÓN	13
PLANTEAMIENTO DEL PROBLEMA	15
JUSTIFICACIÓN	17
HIPÓTESIS	19
OBJETIVOS	20
OBJETIVOS ESPECÍFICOS	20
CONTRIBUCIÓN AL CONOCIMIENTO	21
CAPÍTULO 1: ANTECEDENTES	22
CAPÍTULO 2: ESTADO DEL ARTE	27
CAPÍTULO 3: ESTADO DEL CAMPO	39
CAPÍTULO 4: METODOLOGÍA	41
CAPÍTULO 5: RESULTADOS	91
CONCLUSIONES	96
RECOMENDACIONES	97
REFERENCIAS BIBLIOGRÁFICAS	101
PRODUCTOS ACADÉMICOS	102
ANEXOS	103

LISTA DE FIGURAS

Figura 1.	Estructura tradicional de un invernadero.....	23
Figura 2.	Sistema de invernaderos verticales con torres de invernadero.....	24
Figura 3.	Detección de plántulas y raíz mediante visión por computadora.....	25
Figura 4.	Modelo de color RGB.....	26
Figura 5.	Sistema automatizado de cultivo NFT de lechuga.....	28
Figura 6.	Invernadero vertical con sistema hidropónico	29
Figura 7.	Diseño del sistema de subida de la solución nutritiva	29
Figura 8.	Instrumentación electrónica para lector de ph	30
Figura 9.	Invernadero vertical en modelo de torre.....	30
Figura 10.	Charola de 98 cavidades	32
Figura 11.	Esquema del prototipo	32
Figura 12.	Temperaturas anuales.....	32
Figura 13.	Temperaturas mensuales.....	33
Figura 14.	Rele inteligente modular Zelio Logic modelo SR3B261BD.....	34
Figura 15.	Imagen del software (Zeliosoft 2) empleado para la programación del PLC.....	34
Figura 16.	Instalación típica de un sistema de nebulización.....	35
Figura 17.	Ejemplo de sistema de nebulización aplicado a un semillero.....	35
Figura 18.	Estructura del cultivo hidropónico.....	36
Figura 19.	Entorno de invernadero automatizado para cultivo hidropónico.....	36
Figura 20.	Interfaz de Aplicación móvil de invernadero.....	37
Figura 21.	Pasos para obtención de características de plantas mediante procesamiento de imágenes.....	38
Figura 22.	Capa única de crecimiento.....	39
Figura 23.	Pila de 3 capas de crecimiento.....	39
Figura 24.	Mapa de nitrógeno de un cultivo mediante procesamiento de imágenes.....	40
Figura 25.	Diseño de partida del mecanismo del prototipo que cuenta con un motor reductor de 12 V, un husillo, un cople y una base para una torre de invernadero a escala..	43
Figura 26.	Diseño final y simulación del prototipo con componentes estructurales y dispositivos a utilizar.....	44
Figura 27.	Torre de invernadero de tamaño real.....	45
Figura 28.	Diseño de torre de invernadero a escala en fusion 360.....	45

Figura 29.	Torre de invernadero impresa en 3D.....	46
Figura 30.	Cople rígido para impresora 3D.	47
Figura 31.	Diseño de base para motor DC.....	48
Figura 32.	Motor de engranajes de 12 a 24 V	49
Figura 33.	Diseño de base para motor DC.....	50
Figura 34.	Diseño de base para motor DC con varilla.....	50
Figura 35.	Diseño de acople de riel	51
Figura 36.	Base avance/retroceso en 3D (vista frontal).	52
Figura 37.	Base avance/retroceso en 3D (vista posterior).	52
Figura 38.	Base avance/retroceso impresa en 3D.	52
Figura 39.	Tuerca combrida de varilla enroscada.	52
Figura 40.	Base avance/retroceso acoplada con tuerca combrida (vista frontal).	53
Figura 41.	Base avance/retroceso acoplada con combrida (vista posterior).	53
Figura 42.	motor a pasos 28byj-48	53
Figura 43.	Cople motor a pasos para torre de invernadero (vista frontal).	54
Figura 44.	Cople motor a pasos para torre de invernadero (vista posterior).	54
Figura 45.	Estructura básica del prototipo.....	55
Figura 46.	Estructura básica del prototipo con torre de invernadero	55
Figura 47.	Diseño de soporte de rodamiento.....	56
Figura 48.	Soporte de rodamiento impreso en 3D.....	56
Figura 49.	Balero de giro.....	57
Figura 50.	Soporte con rodamiento de bolas.....	57
Figura 51.	Base para soporte de rodamiento.....	58
Figura 52.	Estructura final del prototipo	59
Figura 53.	Módulo driver L298N	62
Figura 54.	Módulo driver ULN2003	63
Figura 55.	Diagrama básico de conexión de motores.....	64
Figura 56.	Tarjeta de desarrollo Raspberry pi 4.....	64
Figura 57.	Arquitectura de Raspberry pi 4	66
Figura 58.	Pines GPIO de Raspberry pi 4	66
Figura 59.	Diagrama del diseño electrónico del sistema.....	67
Figura 60.	Representación gráfica de la proyección de movimiento de la torre de invernadero	70

Figura 61.	Torre de invernadero con espirales de insertos diferenciados por colores.	72
Figura 62.	Cámara USB utilizada en el prototipo.....	73
Figura 63.	Base para cuerpo de cámara USB.	74
Figura 64.	Base para cámara USB.	75
Figura 65.	Prototipo con cámara USB.	76
Figura 66.	Frame mostrando torre de invernadero.....	77
Figura 67.	Frame con ROI programado.	78
Figura 68.	Inserto detectado en ROI.....	79
Figura 69.	Cálculo de error de posicionamiento de ROI.....	80
Figura 70.	Ajuste de desfaseamiento de inserto negro.	81
Figura 71.	Inserto negro centrado.....	82
Figura 72.	Ajuste de desfaseamiento de inserto azul.	83
Figura 73.	Inserto azul centrado.....	83
Figura 74.	Ajuste de desfaseamiento de inserto verde.	83
Figura 75.	Inserto verde centrado.	83
Figura 76.	Ajuste de desfaseamiento de inserto rojo.	84
Figura 77.	Inserto rojo centrado.	84
Figura 78.	Ajuste de desfaseamiento de inserto fucsia.....	84
Figura 79.	Inserto fucsia centrado.	84
Figura 80.	Ajuste de desfaseamiento de inserto café.....	85
Figura 81.	Inserto café centrado.....	85
Figura 82.	Ajuste de desfaseamiento de inserto morado.	85
Figura 83.	Inserto morado centrado.	85
Figura 84.	Ajuste de desfaseamiento de inserto naranja.....	86
Figura 85.	Inserto naranja centrado.....	86
Figura 86.	Placa fenólica con circuito de interrupción con push button.	89
Figura 87.	Circuito electrónico de sistema completo.	90

LISTA DE TABLAS

Tabla 1.	Fases y actividades del proyecto.....	42
Tabla 2.	Comparación general entre un motor DC pequeño, servomotor y motor a pasos 28BYJ-48. [1]	61
Tabla 3.	Comparación de técnicas de detección de objetos en imágenes. [2].	71
Tabla 4.	Comparación de tiempo de operación con sistema de detección con tiempo de operación manual.	92
Tabla 5.	Resultados de detección y error en 30 pruebas	93

LISTA DE ECUACIONES

Ecuación 1.	Ecuación de cálculo de error de posicionamiento (px)	93
Ecuación 2.	Ecuación de cálculo de error de posicionamiento real (mm).....	93
Ecuación 3.	Ecuación de cálculo de varianza (mm)	94
Ecuación 4.	Ecuación de cálculo de Desviación estándar (mm).....	94
Ecuación 5.	Ecuación de cálculo de error estándar (mm)	95
Ecuación 6.	Ecuación de cálculo de intervalo de confianza del 95 % (mm)	95
Ecuación 7.	Ecuación de cálculo de eficiencia basado en error.....	95
Ecuación 8.	Ecuación de cálculo de eficiencia basado en tiempo	95
Ecuación 9.	Ecuación de cálculo de eficiencia global	95

List of Algorithms

1	Prueba de control de Motor DC con Raspberry Pi y L298N	68
2	Prueba de control de motor 28BYJ-48 con ULN2003 en Raspberry Pi	69
3	Sistema de procesamiento de imágenes para sistema de corrección de posicionamiento de insertos de torre de invernadero	87

Lista de códigos

1	Código de control de motores DC con L298N en python	103
2	Código de control de motor a pasos con ULN2003 en python	104
3	Sistema de procesamiento de imágenes para sistema de corrección de posicionamiento de insertos de torre de invernadero	106

AGRADECIMIENTOS

A mis padres...

Tanya Yamel Peñuela Morán y Adolfo Rodríguez Parada, por siempre brindarme su apoyo y amor pero principalmente, por brindarme las herramientas necesarias y mostrarme el camino para afrontar la vida. Por motivarme a afrontar nuevos retos, a formar mi carácter y coraje para no retroceder ante las adversidades. Por alimentar mi madurez para poder crecer y forjar mi camino sin depender de nadie. Les dedico este logro, ya que ellos han sido parte vital en mi vida y en mi crecimiento como persona y como profesional, ya que he llegado a donde estoy gracias a ellos.

A mi director de Tesis...

Dr. Ignacio Herrera Aguilar, por la oportunidad de trabajar a su lado, por brindarme la confianza necesaria de sobrellevar un proyecto importante, por sus consejos que me permitieron entender la vida desde otra perspectiva. Le agradezco por este logro ya que me abrió las puertas desde la primera vez que ingresé al programa; por lo que, más que un maestro, lo considero un amigo.

A mis compañeros...

No solo le agradezco a mis compañeros de generación, sino a todos los que formaron parte del programa, ya que me he llevado una parte de cada uno de ellos, ellos han formado parte de mi crecimiento y formación académica; por lo que, más que su conocimiento, me llevo su amistad.

A mis maestros...

A todos los doctores y maestros que conforman al programa de posgrado, les agradezco por brindarme sus conocimientos, sus aportes a mi aprendizaje y desarrollo académico han sido de tal importancia para poder tener herramientas para desenvolverme en el mundo laboral.

RESUMEN

En el presente trabajo titulado “Prototipo de posicionador de invernadero vertical con sistema de procesamiento de imágenes para colocación de sustrato con semilla para la obtención de plántula” se desarrolla un prototipo de un posicionador automático de insertos de una torre de invernadero con el fin de proyectar una idea tecnológica e innovadora para el sector agrícola, específicamente para la germinación de caña de azúcar mediante la obtención de plántula. Este prototipo consiste en un sistema mecatrónico cuyo mecanismo acelerará la colocación de sustrato, así como el aumento en la ergonomía del proceso mediante un sistema de procesamiento de imágenes.

La literatura ha mostrado una reducción en la producción agrícola en los últimos años siendo este sector uno de los de mayor potencial y crecimiento, aunado a esa problemática, las investigaciones han demostrado una falta de implementación tecnológica en procesos pre-germinación, lo cual limita la eficiencia y la calidad en la producción de plántulas, por lo cual, este prototipo busca brindar una alternativa para mejorar los procesos que llevan a cabo los invernaderos verticales.

La metodología se compone de 3 etapas principales: diseño estructural y mecánico del prototipo (modelado, diseñado en software 3D a tamaño a escala e impreso en 3D), diseño electrónico (incorporado con actuadores como un motor DC y un motor a pasos con sus respectivos controladores, ambos controlados por una raspberry pi 4) y desarrollo del software de control (algoritmo de procesamiento de imágenes).

Los resultados obtenidos validan el desempeño del sistema del prototipo comparando el tiempo del proceso del sistema con el tiempo de colocación manual de sustrato. Además, este proyecto se muestra como una alternativa tecnológica óptima para la mejora en la eficiencia agrícola obteniendo resultados cuantitativos sobre la eficiencia del prototipo con respecto al error promedio del proceso.

El prototipo sienta las bases para la implementación en etapas tempranas a la producción de cultivos y su adaptabilidad a otros procesos industriales.

Palabras Clave: Invernadero vertical, colocación de sustrato, procesamiento de imágenes, ergonomía.

ABSTRACT

In this study, titled "Prototype of a Vertical Greenhouse Positioner with an Image Processing System for Placing a Seed-Containing Substrate to Obtain Seedlings," a prototype of an automatic positioner for greenhouse tower inserts is developed with the aim of projecting a technological and innovative idea for the agricultural sector, specifically for sugarcane germination through seedling production. This prototype consists of a mechatronic system whose mechanism will accelerate substrate placement, as well as improve the ergonomics of the process through an image processing system.

The literature has shown a reduction in agricultural production in recent years, despite this sector being one of the most promising and fastest-growing. Adding to this problem, research has demonstrated a lack of technological implementation in pre-germination processes, which limits the efficiency and quality of seedling production. Therefore, this prototype seeks to provide an alternative to improve the processes carried out in vertical greenhouses.

The methodology consists of 3 main stages: structural and mechanical design of the prototype (modeled, designed in 3D software to scale and 3D printed), electronic design (incorporating actuators such as a DC motor and a stepper motor with their respective controllers, both controlled by a Raspberry Pi 4) and development of the control software (image processing algorithm).

The results obtained validate the performance of the prototype system by comparing the system's process time with the time required for manual substrate placement. Furthermore, this project demonstrates itself as an optimal technological alternative for improving agricultural efficiency, obtaining quantitative results on the prototype's efficiency with respect to the average process error.

The prototype lays the groundwork for early-stage implementation in crop production and its adaptability to other industrial processes.

Keywords: Vertical greenhouse, substrate placement, image processing, ergonomics.

INTRODUCCIÓN

En el transcurso de los años, la tecnología ha evolucionado de una manera acelerada y perceptible, lo cual se han creado herramientas que se han vuelto indispensables para la mayoría de las actividades de la sociedad, especialmente en la vida laboral, de ahí la dependencia de sistemas automatizados o inteligentes en los procesos. Esta idea del desarrollo tecnológico está ligada al área de control y automatización, que pretende incorporar diversas tecnologías a la industria con el fin de agilizar diversos procesos y crear dispositivos inteligentes que se llevan a cabo en el día a día.

La agricultura juega un papel importante en el crecimiento y desarrollo de un país. Las industrias y empresas que se enfocan en la agricultura le permiten a una nación producir sus propios alimentos y otros bienes naturales. La agricultura, como actividad básica para la supervivencia humana durante siglos, ha sufrido cambios drásticos a lo largo de la historia. La agricultura siempre ha sido un pilar importante del desarrollo social, desde los métodos agrícolas básicos más antiguos hasta las actuales innovaciones modernas. Inicialmente, la agricultura se basaba en técnicas tradicionales y rudimentarias como la siembra manual y el arado animal. Conforme pasa el tiempo, la necesidad de aumentar la productividad y alimentar a una población en crecimiento exponencial impulsó la búsqueda de innovación en esta área. Hoy en día, la agricultura enfrenta muchos desafíos globales como el cambio climático, la escasez de recursos y la necesidad de producir alimentos de manera sustentable.

Para darle solución a estas problemáticas se introdujeron innovaciones tecnológicas, como los invernaderos, que implicaban un control más preciso de las condiciones ambientales mediante la organización de los cultivos. Los invernaderos son parte esencial del desarrollo y evolución agrícola, ya que proporcionan un entorno controlado que permite que las plantas crezcan de manera normal, aun cuando estén fuera de su temporada de crecimiento natural. Con el tiempo, la creación de los invernaderos verticales trascendió como una forma revolucionaria de optimizar los espacios limitados y aumentar la eficiencia de la producción de alimentos.

Los invernaderos tradicionales se han utilizado durante siglos para aprovechar la energía solar y regular la temperatura, la humedad y la ventilación, pero con la creciente presión sobre los

recursos y la necesidad de producir alimentos de manera más sostenible, los invernaderos verticales son cada vez más importantes. En las torres de invernadero, los cultivos se colocan verticalmente en lugar de horizontalmente. Esta posición vertical no sólo aumenta el espacio disponible, sino que también permite un control más preciso de las condiciones ambientales de cada planta. La iluminación artificial, los sistemas de riego automático y la tecnología de seguimiento ayudan a optimizar eficazmente el crecimiento de las plantas.

Los invernaderos verticales se crearon como una respuesta ingeniosa a los desafíos de la urbanización y las limitaciones de tierras agrícolas. Estos sistemas no sólo son adecuados para entornos urbanos donde el espacio es limitado y escaso, sino que también son más resistentes a las inclemencias del tiempo y al cambio climático.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la agricultura mexicana constituye uno de los pilares fundamentales de la economía del país, pues se beneficia de una de las mayores biodiversidades a nivel mundial, además de condiciones climáticas propicias y una extensa superficie territorial. Estos elementos le permiten posicionarse como productor de una amplia gama de cultivos. [3]

De acuerdo con [4], la economía mundial sufrió un fuerte impacto en los últimos años debido a la pandemia de COVID-19, y el sector agrícola mexicano no fue ajeno a esta situación. Sin embargo, su dinámica frente a la crisis generó un crecimiento positivo, superando incluso a otros sectores de la economía nacional y consolidándose en 2020 como un importante proveedor de alimentos a nivel global. La dedicación de los productores y la articulación de las cadenas productivas hicieron posible alcanzar un superávit de 12,347 millones de dólares, con un incremento anual del 39.92 %. Este comportamiento evidencia que la agricultura posee una mayor capacidad de resistencia en escenarios de crisis, favorecida además por las variaciones en el tipo de cambio que compensaron la baja productividad de la industria manufacturera.

Aunque el sector agrícola ha sido de los menos impactados dentro de la economía nacional, su limitada dimensión lo hace estructuralmente vulnerable frente a crisis de gran magnitud. Factores como la topografía accidentada, la calidad de los suelos y las condiciones climáticas restringen de manera natural la disponibilidad de áreas aptas para el desarrollo agrícola, lo que evidencia sus debilidades frente a escenarios de choque económico o ambiental.

La agricultura vertical resuelve los desafíos de la producción de alimentos mediante la introducción de invernaderos verticales, aumentando así la seguridad alimentaria y beneficiando a la población mundial. La agricultura convencional suele estar asociada con el uso intensivo de recursos y la degradación del suelo.

Los invernaderos verticales permiten una agricultura más sostenible al reducir la necesidad de expansión de la tierra y utilizar sistemas de cultivo más eficientes en el uso de recursos. Aunque los invernaderos verticales tienen muchas ventajas, el tiempo de producción de alimentos sigue siendo el mismo debido al retraso en la colocación del sustrato en la torre del invernadero, por lo que este proyecto pretende reducir este tiempo, agilizar y simplificar el proceso de

colocación del sustrato, aumentando así el rendimiento de la germinación de la plántula de caña de azúcar.

JUSTIFICACIÓN

El proyecto surge de la creciente necesidad de desarrollar tecnologías agrícolas eficientes y sostenibles para abordar los desafíos actuales de la producción de alimentos. Los invernaderos verticales representan una innovadora solución agrícola que optimiza y utiliza espacios reducidos y recursos limitados, pero su enorme potencial sólo puede cumplirse mediante el uso de sistemas de automatización avanzados.

La implementación de tecnologías agrícolas puede optimizar el uso de recursos como el agua, energía y/o fertilizantes. Los sistemas de riego automatizados, los sensores de humedad del suelo y de temperatura y los controladores climáticos inteligentes pueden gestionar estos recursos de una forma más eficiente, reduciendo costos y el impacto ambiental. También ayudan a mejorar la calidad de los productos cultivados en invernaderos. La capacidad de controlar con precisión el entorno de cultivo puede influir positivamente en aspectos como el sabor, el tamaño y la apariencia de un producto, aumentando así su valor del mercado. Las tecnologías utilizadas para los invernaderos verticales pueden incluir sistemas inteligentes de ahorro y reciclaje de agua, que pueden reducir significativamente el consumo de agua en comparación con los métodos de cultivos tradicionales. Esto es especialmente valioso en áreas donde el agua es escasa.

Las innovaciones técnicas en invernaderos verticales no solo abordan los desafíos particularmente asociados con este método de cultivo, sino que también brindan soluciones eficientes y sustentables para la producción de alimentos en entornos urbanos con recursos limitados y espacios reducidos. Esto ocasiona que los programas de innovación tecnológica en invernaderos verticales sean esenciales para avanzar hacia un sistema alimentario más sostenible y sólido. [5]

La automatización de invernaderos verticales busca mejorar la eficiencia operativa al reducir la intervención humana, pero en este caso, buscamos que esa intervención sea ergonómica para optimizar y acelerar los procesos agrícolas.

Actualmente, los sustratos, que es un material base que da soporte físico a las plantas y les proporciona un medio para que sus raíces se anclen, absorban nutrientes y agua, representan

una pieza importante en la agricultura moderna, específicamente en la agricultura llevada a cabo a través de sistemas hidropónicos. La mejora de la colocación de sustrato ergonómicamente correcta implica optimizar el proceso para reducir la tensión y el riesgo de lesiones en los trabajadores. Con este proyecto, al ajustar la altura y ángulo de trabajo, se evitará que los trabajadores tengan que agacharse o estirarse en exceso a la hora del acomodo de los sustratos. Además, se diseñará un área de trabajo con suficiente espacio para que los trabajadores se muevan cómodamente al manipular la torre y el sustrato, evitando posturas incómodas y/o forzadas.

La colocación del sustrato en un invernadero vertical es un factor crítico para el éxito del cultivo. Dado que estos sistemas permiten apilar el cultivo verticalmente, el sustrato utilizado debe proporcionar suficiente soporte a las plantas y garantizar una nutrición y retención de humedad óptimas. El sustrato debe distribuirse uniformemente en cada recipiente para garantizar que todas las plantas reciban los mismos nutrientes y condiciones de crecimiento.

HIPÓTESIS

Es posible automatizar la corrección posicional de insertos en una torre de invernadero mediante un prototipo de posicionador vertical con un sistema de procesamiento de imágenes, acelerando la colocación de sustrato con semilla y mejorando la ergonomía del operador.

OBJETIVOS

Crear un prototipo de posicionador de invernadero vertical que le permita al usuario colocar sustratos con semilla de caña de manera acelerada y ergonómica a una torre de invernadero para acelerar la obtención de plántula.

OBJETIVOS ESPECÍFICOS

- Automatizar el movimiento de avance/retroceso y de rotación de una torre de invernadero en un grado de inclinación de 45/50°.
- Desarrollar un programa de detección de insertos mediante procesamiento de imágenes.
- Desarrollar un algoritmo de corrección de posición de insertos en una torre de invernadero.
- Validar la eficiencia funcional del prototipo.

CONTRIBUCIÓN AL CONOCIMIENTO

Este trabajo contribuye al área de la ingeniería mecatrónica enfocada a la automatización de procesos en el sector agrícola, a través del desarrollo de un prototipo a escala cuyo propósito es mostrar una alternativa para agilizar y ergonomizar la colocación de sustrato en invernaderos verticales para aumentar y optimizar la germinación de caña de azúcar.

El aporte principal radica en la implementación de un sistema de control mecánico-electrónico eficiente, que integre principios de procesamiento de imágenes, automatización y ergonomía para resolver un problema recurrente en la producción agrícola: la necesidad de reducir esfuerzo físico, tiempos y errores humanos en la etapa de siembra.

Ademas de las mencionadas, este trabajo contribuye en:

- Generar conocimiento de carácter interdisciplinario, cuando se conecta la automatización con la mecatrónica agricultura de precisión y ergonomía.
- Fortalece la perspectiva ergonómica, Al sugerir un sistema que reduzca la carga física de los operadores y optimización de la eficacia del procedimiento.
- Contribuye al estado del arte en tema de invernaderos verticales, mostrando como la automatización puede integrarse en etapas tempranas al ciclo de germinado o productivo.
- Introduce la visión por computadora como herramienta de apoyo, sentando bases para futuros nuevos sistemas de clasificación y posicionamiento automático en la obtención de plántulas.
- Aporta un modelo experimental adaptable y replicable hacia otros procesos industriales.

CAPÍTULO 1: ANTECEDENTES

En este primer capítulo, se expondrán los antecedentes de este proyecto, es decir, el conjunto de investigaciones, artículos, tesis y trabajos previos relacionados con este proyecto, así como los conceptos e ideas que se utilizarán en este trabajo para contextualizar justificar el tema principal de esta investigación. Esta capítulo recoge los estudios previos que han tratado el tema de este proyecto desde sus orígenes, esto con el objetivo de mostrar cómo se ha desarrollado la investigación del tema planteado a lo largo del tiempo.

1.1 ¿Que es la ergonomía?

Este concepto se expone en [6] que analiza el papel de la ergonomía en la utilización eficiente de la mano de obra agrícola. La ergonomía es la ciencia que estudia cómo adecuar la relación del ser humano con su entorno, y en este caso los problemas que representan un riesgo para la salud, relacionados con la fase operativa en la industria. El objetivo de este trabajo fue mostrar cómo la ergonomía puede mejorar la eficiencia del trabajo agrícola, tanto en tareas manuales como en la operación de maquinaria.

Entre los temas tratados en este trabajo, se encuentran vibraciones y posturas en la operación de maquinaria agrícola y la implementación de cabinas y asientos diseñados ergonómicamente para reducir fatiga y accidentes.

1.2 ¿Que es un invernadero?

Este concepto se expone en [7], donde se analiza la importancia de la tecnología en invernaderos. Un invernadero es un conjunto de marcos de estructura inflada cubiertos con un material transparente, el cual los cultivos crecen bajo condiciones de ambientes controlados.

Además de la importancia de la tecnología en invernaderos, esta literatura trata un panorama histórico sobre el desarrollo de los invernaderos y su impacto en la productividad agrícola, incluye detalles sobre diseños, materiales, estructuras y métodos de construcción de invernaderos, así como análisis de errores y limitaciones.



Figura 1: Estructura tradicional de un invernadero.

1.2.1 Invernaderos verticales

La agricultura vertical consiste en un método de apilación y riego estratégico de cultivos mediante torres cilíndricas que están conformados con bandejas para macetas (insertos) para la optimización de crecimiento de cultivos.

El cultivo vertical enfrenta un reto técnico importante, ya que al separarse las plantas del suelo se requiere el uso de bandejas y sustratos alternativos en lugar de tierra. Por ello, quienes practican esta modalidad suelen complementar con técnicas tecnológicas como la hidroponía o la aeroponía, que permiten mantener un sistema productivo eficiente sin depender del suelo o la tierra.

En [8] introduce la idea de una granja vertical como solución a los problemas de seguridad alimentaria, urbanización y sostenibilidad. El concepto de invernadero vertical no había tenido relevancia hasta hace pocos años, ya que esta técnica innovadora en el proceso de generación de cultivo ha sido implementada recientemente, así como la implementación de nuevas tecnologías para el mejoramiento de los procesos en la agricultura vertical y este trabajo se muestra como pionero de las investigaciones que mencionan este concepto de invernaderos

verticales.



Figura 2: Sistema de invernaderos verticales con torres de invernadero.

1.3 Procesamiento de imágenes

En [9] se desarrolló una técnica de visión por computadora para la medición morfológica de las características de 3 plántulas de árbol para su clasificación.

La visión por computadora es un campo de la Inteligencia Artificial (IA) que se encarga de interpretar y analizar datos visuales mediante imágenes y videos. En este trabajo se utilizó una técnica de calibración, detección de plántulas, localización de raíz y clasificación de las plántulas.

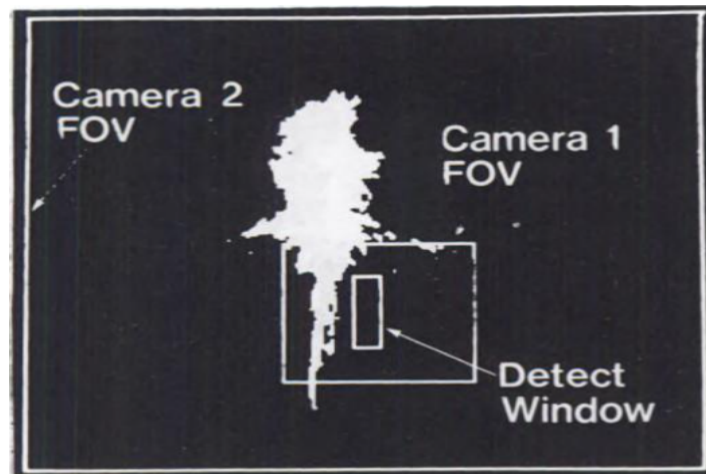


Figura 3: Detección de plántulas y raíz mediante visión por computadora.

1.3.1 OpenCV

De acuerdo con [10], OpenCV es una biblioteca/librería multiplataforma escrita principalmente en C/C++ con interfaces para lenguaje python y Java. Su función principal es proporcionar herramientas eficientes para el análisis de imágenes y videos, facilitando la implementación de algoritmos de visión por computadora en tiempo real.

En las características principales de OpenCV se encuentran:

- Procesamiento de imágenes (filtros, transformacione, segmentación).
- Detección de características (bordes, esquinas, contornos).
- Reconocimiento de patrones y objetos.
- Seguimiento de movimiento en secuencias de video.
- Integración con librerías de aprendizaje automático.

Entre sus aplicaciones prácticas se encuentran:

- Detección de color y segmentación de regiones en imágenes.
- Reconocimiento facial y biometría.

- Inspección industrial automatizada.
- Vehículos autónomos y robótica.
- OCR (reconocimiento óptico de caracteres) con integración de Tesseract.

En [11] se implementó un algoritmo de detección de colores usando OpenCV. Al usar OpenCV, la detección de color ofrece numerosas ventajas. Por ejemplo, permite detectar un color específico en una transmisión de video en vivo. El color es una característica ampliamente explotada en el procesamiento de imágenes digitales, ya que es una herramienta poderosa que facilita la clasificación e identificación de objetos, que pueden discriminarse en función de la gran cantidad de tonos de color apreciables.

Una imagen es básicamente una matriz de valores de píxeles. Una imagen se puede representar mediante diversos modelos de color (RGB, HSV), escala de grises, etc. El modelo RGB se utiliza para detectar los colores en una imagen, mientras que el modelo HSV se utiliza para detectar los componentes de dichos colores. El modelo de color RGB se basa en un sistema aditivo en el que las luces roja, verde y azul se combinan en distintas proporciones para generar una amplia variedad de tonalidades. Este espacio de color se emplea de manera habitual en procesos de detección cromática, dado que las cámaras capturan y almacenan la información visual en formato RGB.

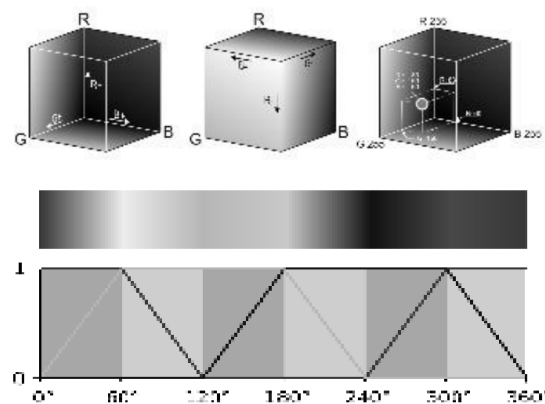


Figura 4: Modelo de color RGB.

CAPÍTULO 2: ESTADO DEL ARTE

A continuación, se analizarán las principales investigaciones que circunscriben el estado del arte de este proyecto.

En [12] se estudió el diseño de un sistema automatizado de cultivo hidropónico destinado a la producción de forraje verde, cuyo objetivo fue la germinación, desarrollo y cosecha en bandejas con sustrato, empleando fertirriego como método de fertilización y manejo hídrico. El sistema se implementó dentro de un invernadero, donde se controlaron las principales variables ambientales (temperatura, humedad, concentración de CO_2) y radiación solar mediante un conjunto de tecnologías específicas. Entre ellas se incluyó un sistema de ventilación cenital para la renovación del aire y regulación del CO_2 , un sistema de calefacción para mantener la temperatura adecuada y un mecanismo de microaspersión complementario al fertirriego para optimizar el crecimiento del forraje verde hidropónico.

Asimismo, se abordaron técnicas de cultivo sin sustrato, desarrolladas como alternativa frente a la escasez o elevado costo del suelo. Estos métodos, basados en la recirculación de soluciones nutritivas, permiten un uso más eficiente del espacio de cultivo tanto en sentido horizontal como vertical. Dentro de estas innovaciones se analizó particularmente la técnica de raíz flotante, diseñada para la producción de hortalizas de hoja como lechuga, acelga, espinaca y repollo.

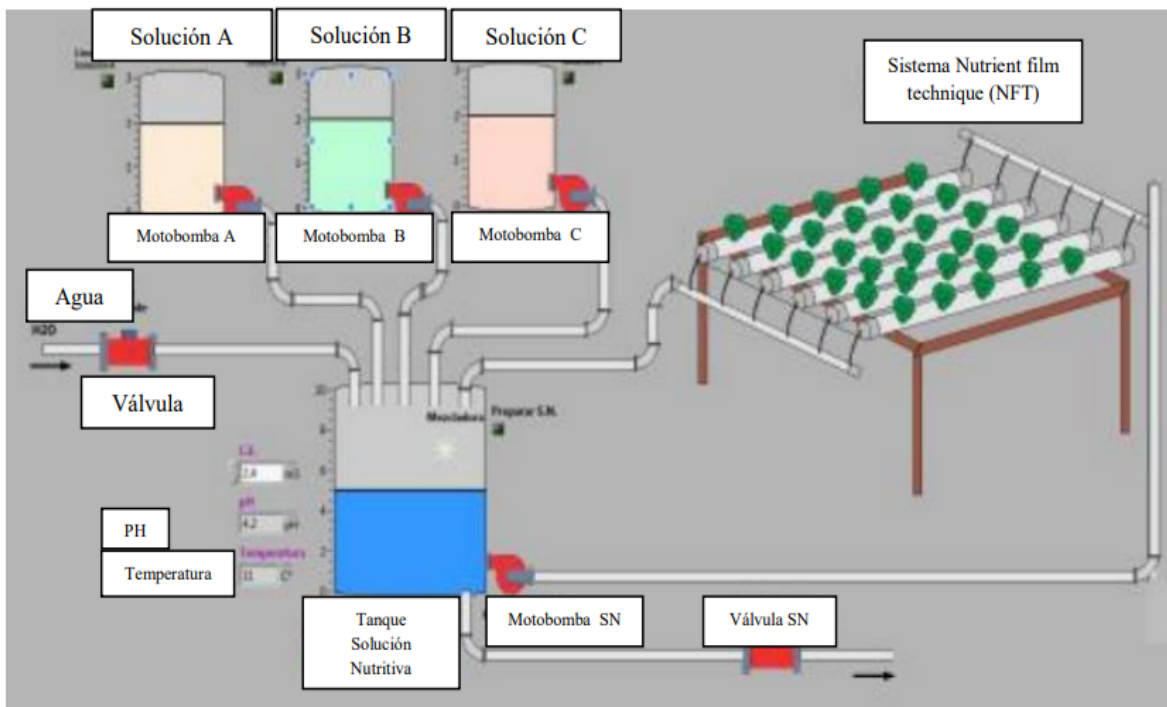


Figura 5: Sistema automatizado de cultivo NFT de lechuga

En [13] se presenta el desarrollo y construcción de un prototipo operativo de huerto hidropónico vertical automatizado, diseñado para funcionar tanto en espacios interiores como exteriores. El sistema incorpora un panel de control que facilita al usuario la supervisión y gestión remota del huerto, además de emitir alertas cuando se requiere una intervención manual. El prototipo cuenta con autonomía energética gracias al uso de paneles solares y baterías, y fue elaborado priorizando el empleo de materiales reutilizables y la optimización de los recursos adquiridos. En cuanto a su diseño electrónico, se integraron diversos sensores y actuadores, entre ellos medidores de pH, conductividad eléctrica (EC), temperatura de la solución nutritiva, ultrasonidos, así como bombas de agua, puentes H y relevadores— todos gestionados mediante microcontroladores instalados en las unidades correspondientes.

Una vez hecho el diseño electrónico, se realizó el software de control que contiene la arquitectura de control entre dispositivos, para la comunicación de estos se utilizaron ciertas entidades como controladores ESP32, Arduino Nano, Node-Red como herramienta para programar la conexión de los dispositivos que se utilizaron mediante IoT y un ESP32 para tomar fotografías. Todos los dispositivos se conectaron mediante el protocolo MQTT, una arquitectura basada en

eventos. En su defecto, el sistema trabaja de manera manual, cuando el usuario activa el modo automático será Node-RED la entidad encargada de gestionar cuando se realiza cada acción de control sobre la solución nutritiva mientras se está operando en modo automático.

Una vez establecida la comunicación entre los dispositivos, se procedió al análisis del control de la solución nutritiva. Para garantizar una adecuada disolución de los nutrientes, resulta indispensable supervisar y regular de manera precisa tres parámetros fundamentales: la temperatura de la mezcla, el nivel de pH y la electroconductividad, que refleja la concentración de nutrientes presentes.



Figura 6: Invernadero vertical con sistema hidropónico

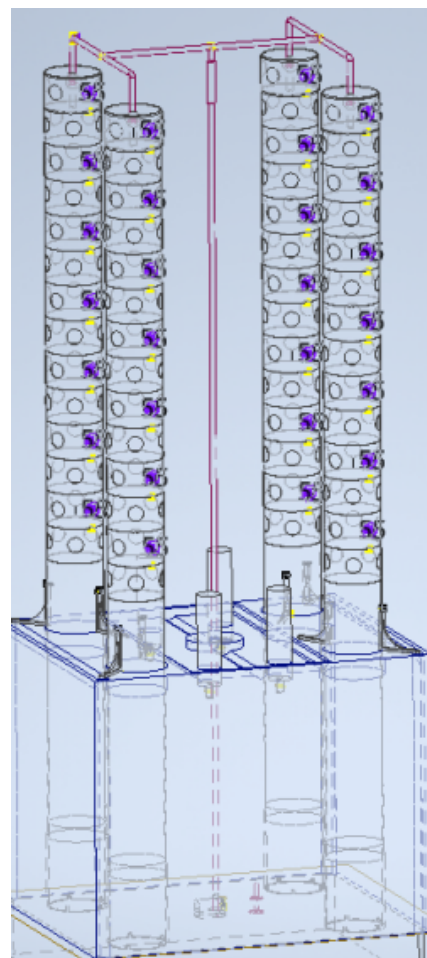


Figura 7: Diseño del sistema de subida de la solución nutritiva

En [14] se desarrolló un dispositivo destinado a la producción de vegetales a pequeña escala dentro de un invernadero, el cual permitió identificar los factores esenciales para el monitoreo

de variables químicas y físicas mediante instrumentación electrónica. El sistema incorporó sensores capaces de medir la conductividad eléctrica, el pH, la temperatura, la humedad, el nivel por desplazamiento y la iluminación. El diseño del invernadero vertical se estructuró en dos secciones: una destinada a alojar la electrónica y el reservorio hidropónico, y otra conformada por una torre que albergaba las plantas con una distribución que favorecía su crecimiento.

La investigación demostró que la incorporación de instrumentación electrónica en invernaderos ofrece ventajas frente a la agricultura convencional. En el desarrollo de las tarjetas electrónicas se logró cumplir con los requisitos de diseño, estandarizar conectores para reducir errores, emplear cables diferenciados por colores para separar alimentación y señales de muestreo, además de incluir puntos de prueba que facilitaron la calibración. Como resultado, las tarjetas pueden retirarse del sistema para mantenimiento sin arrastrar cableado, lo que optimiza su manipulación.

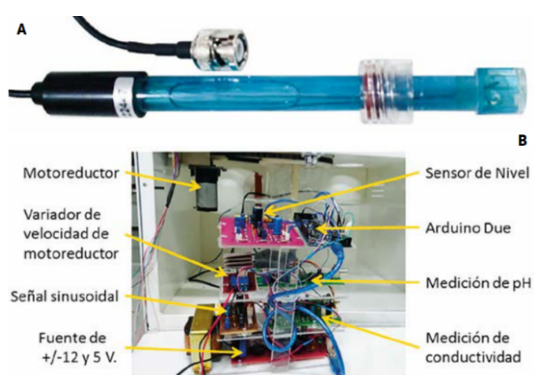


Figura 8: Instrumentación electrónica para lector de ph



Figura 9: Invernadero vertical en modelo de torre

En [15] se desarrolló un prototipo de máquina sembradora semiautomática destinada al uso en charolas de invernadero, con el propósito de disminuir tanto el tiempo como la mano de obra requeridos en esta tarea.

La implementación de un mecanismo semiautomático para el traslado y siembra de semillas en charolas permite disminuir significativamente el tiempo requerido en esta operación, que de

manera manual demanda la mayor inversión de esfuerzo. Gracias a su diseño, el sistema solo requiere la intervención de un operario, lo que reduce la necesidad de personal en comparación con los métodos tradicionales de semillado.

La charola funciona como un contenedor colectivo para plántulas, disponible en distintas dimensiones, lo que contribuye a la uniformidad de las labores de cultivo y al aprovechamiento eficiente del espacio dentro del invernadero. Su diseño varía según el tipo de cultivo, el fabricante y el material empleado en su fabricación.

Las charolas utilizadas en invernaderos suelen fabricarse principalmente con poliestireno de alto impacto moldeado al vacío o con poliestireno expandido mediante inyección, lo que les confiere rigidez estructural. Sin embargo, también existen alternativas flexibles como las denominadas paperpot, que funcionan inicialmente como charolas pero pueden separarse en macetas individuales para facilitar el manejo de las plántulas.

El prototipo incorpora una banda transportadora integrada por cuatro estaciones, cuya operación se coordina mediante un microcontrolador. En la primera de ellas, la charola se llena con sustrato proveniente de una tolva con capacidad suficiente para almacenar un costal.

En la segunda estación se retira el exceso de sustrato y se uniformiza su distribución en las cavidades mediante un cepillo. Posteriormente, la tercera estación incorpora un sistema de semillado que, utilizando la succión generada por una aspiradora, recoge las semillas y las deposita en el sustrato a la profundidad establecida.

En la última estación, la charola recibe un riego por aspersión tras haber pasado por las fases anteriores, lo que permite retirarla al concluir el proceso y dejarla lista para la germinación de las semillas sembradas en su interior.



Figura 10: Charola de 98 cavidades

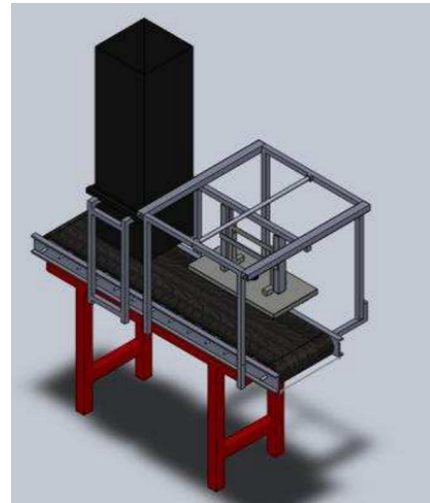


Figura 11: Esquema del prototipo

En [16] Se planteó el diseño de un semillero automatizado orientado al cultivo de especies que enfrentan serias dificultades para desarrollarse al aire libre en regiones con escasas precipitaciones, temperaturas extremas (ya sean altas o bajas) o marcadas variaciones térmicas. Con el objetivo de atender esta problemática sin necesidad de trasladar los cultivos a largas distancias, se optó por la creación de un semillero de bajo costo capaz de satisfacer las necesidades de los productores que trabajan en terrenos con dichas condiciones.

Para la propiedad con más importancia, se hizo un breve estudio climático de la zona en la que se realizó la instalación mediante una serie de tablas que permitieron observar con mayor facilidad los datos registrados.

	Temperatura media (°C)	Temperatura máxima (°C)	Temperatura mínima (°C)	Temperatura media máxima (°C)	Temperatura media mínima (°C)
2009	15,4	41,3	-7,2	23,1	8,3
2010	14,2	41,4	-6,8	21,5	7,7
2011	15,2	40,3	-9,4	23,1	8,4
2012	15,2	44,3	-7,4	23,3	7,9
2013	14,6	38,4	-5,8	22,5	7,7
2014	15,9	40,3	-5,5	23,8	9,0
2015	15,4	42,9	-6,3	23,9	8,3
2016	15,6	41,2	-7,6	23,4	8,6

Figura 12: Temperaturas anuales

De la tabla mostrada se deduce que, entre los años 2009 y 2016, la temperatura media anual fue de 15.2°C , con una variación aproximada de $\pm 1^{\circ}\text{C}$ durante dicho intervalo. Este comportamiento evidencia la consistencia de los datos, lo que los convierte en una referencia confiable para proyectar la evolución de la temperatura en los años posteriores.

	Temperatura media ($^{\circ}\text{C}$)	Temperatura máxima ($^{\circ}\text{C}$)	Temperatura mínima ($^{\circ}\text{C}$)	Temperatura media máxima ($^{\circ}\text{C}$)	Temperatura media mínima ($^{\circ}\text{C}$)
Enero	6,9	21,3	-9,4	13,2	1,8
Febrero	7,1	23,9	-7,6	13,7	1,4
Marzo	9,8	29,3	-4,9	17,1	3,3
Abril	13,4	31,9	-1,4	21,3	6,5
Mayo	17,2	40,9	1,4	26,0	9,0
Junio	22,1	40,7	6,4	31,3	13,1
Julio	25,5	42,9	10,5	35,2	16,1
Agosto	25,1	44,3	10,9	34,4	16,6
Septiembre	20,4	40,3	6,2	28,9	13,4
Octubre	16,4	32,9	-0,2	24,3	10,0
Noviembre	10,5	27,7	-4,5	16,9	5,3
Diciembre	7,2	24,1	-7,2	14,1	1,9

Figura 13: Temperaturas mensuales

Para llevar a cabo las etapas de germinación y desarrollo de una amplia variedad de cultivos, resulta necesario implementar un sistema de control capaz de mantener distintos rangos de humedad y temperatura, además de regular la cantidad de agua suministrada mediante el riego. Por ello se incorporó un sistema de riego mediante nebulización (fog system), empleado en espacios cerrados y basado en la pulverización de agua. Este mecanismo asegura el suministro hídrico a los cultivos, mantiene niveles de humedad elevados —cercaos al 90 %, óptimos para la fase de germinación— y, en determinadas circunstancias, contribuye a prevenir incrementos excesivos de temperatura.

Se optó por implementar un control integral del sistema de riego del invernadero, junto con aquellos subsistemas que inciden directamente en su funcionamiento, como el mezclado de nutrientes en el agua, incluyendo fertilizantes y reguladores de pH, además de las condiciones ambientales de temperatura y humedad. Estos factores resultan esenciales para asegurar el desarrollo óptimo de las plantas.

El control fue implementado mediante un PLC, seleccionado por la solidez y confiabilidad que

ofrece frente a las condiciones ambientales propias de este tipo de instalaciones. El programa se diseñó para alinearse estrechamente con los objetivos establecidos, optimizando el código con el fin de hacerlo compatible con un PLC de tamaño compacto.



Figura 14: Rele inteligente modular ZeLIO Logic modelo SR3B261BD.

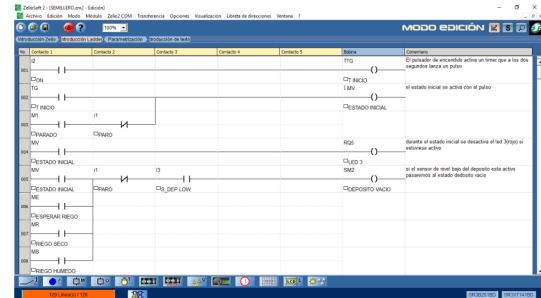


Figura 15: Imagen del software (Zeliosoft 2) empleado para la programación del PLC.

Los sistemas de nebulización constituyen una solución altamente eficaz para la regulación de las condiciones climáticas dentro de los invernaderos. Dicho control se logra mediante el intercambio de energía entre el aire caliente y las diminutas gotas generadas por los nebulizadores.

Al liberarse, las diminutas gotas de agua se transforman rápidamente de estado líquido a vapor debido a su reducido tamaño y a las elevadas temperaturas presentes en el invernadero. En este proceso, absorben calor del aire a razón de 590 calorías por gramo (16.726 cal/oz) de agua evaporada, lo que contribuye a disminuir la temperatura ambiente y a incrementar la humedad interna en un rango aproximado de 4 a 6 °C.

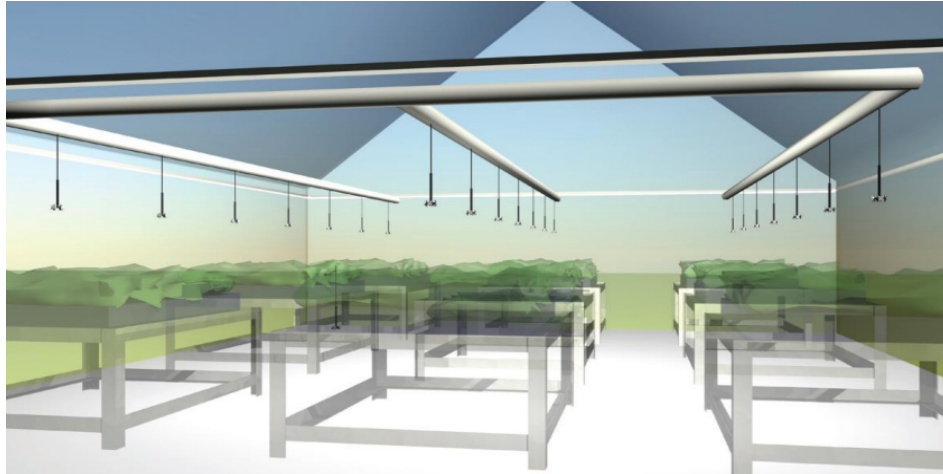


Figura 16: Instalación típica de un sistema de nebulización.



Figura 17: Ejemplo de sistema de nebulización aplicado a un semillero.

En [17] se desarrolló e implementó un invernadero automatizado con sistema de cultivo hidropónico y una aplicación móvil destinada al monitoreo de datos. El objetivo principal fue diseñar una estructura que garantizara el adecuado funcionamiento del invernadero y promoviera el uso eficiente de los recursos disponibles.

El diseño del invernadero se concibe con el propósito de generar un entorno que integre una fuente de iluminación artificial para favorecer el crecimiento de los cultivos, una estructura destinada al sistema hidropónico, además de incorporar ventilación, humidificación, riego y la correspondiente instrumentación electrónica.



Figura 18: Estructura del cultivo hidropónico.

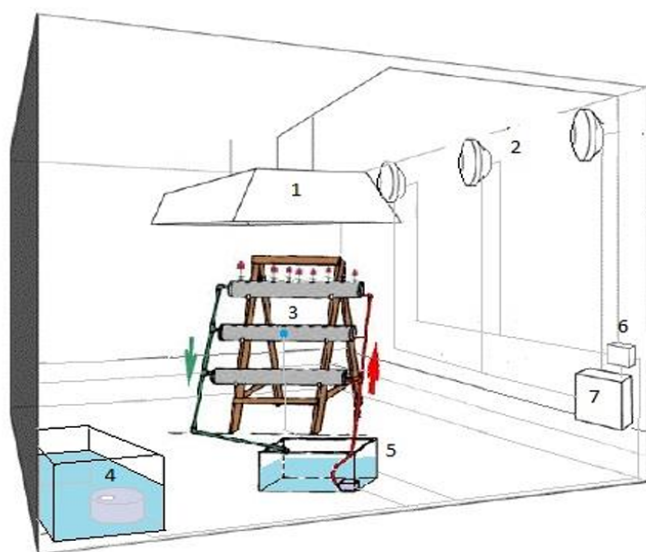


Figura 19: Entorno de invernadero automatizado para cultivo hidropónico.

Para el seguimiento de los parámetros físicos del invernadero se diseñó una aplicación móvil, desarrollada mediante la plataforma AppInventor 2 del Massachusetts Institute of Technology. Dicha aplicación permite visualizar tanto el estado de los actuadores como la información generada por los distintos sensores.

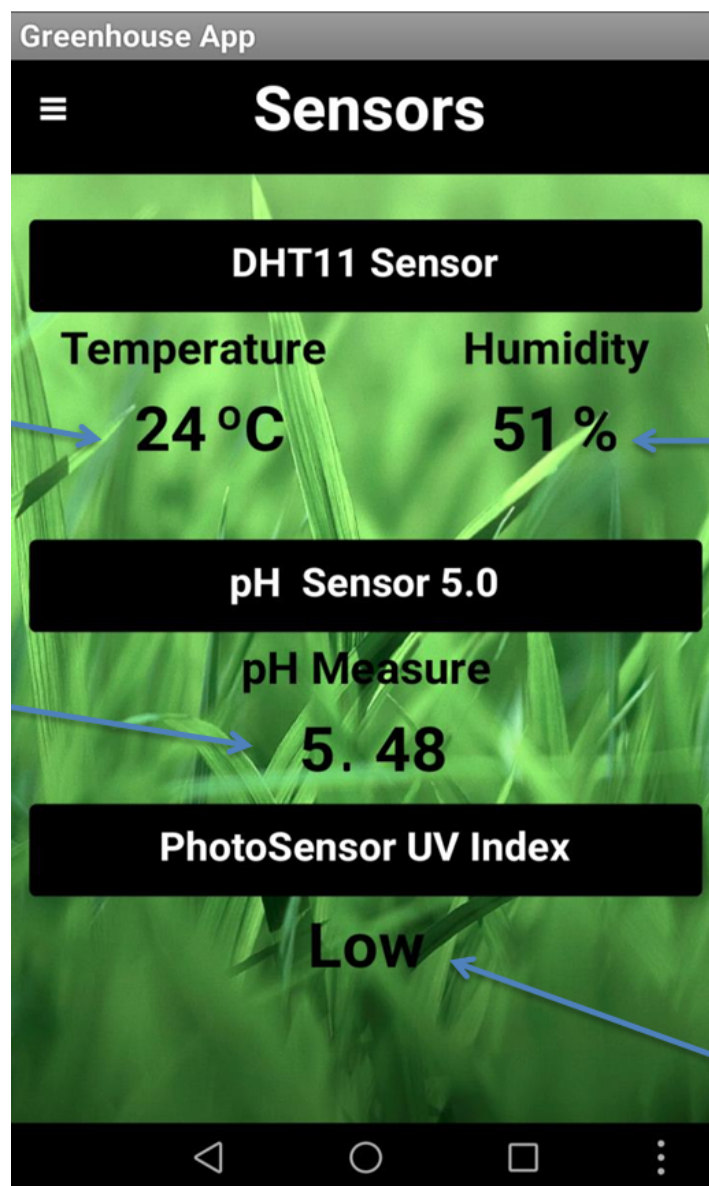


Figura 20: Interfaz de Aplicación móvil de invernadero.

En [18] se presenta un panorama amplio sobre el papel del procesamiento digital de imágenes en la agricultura de precisión. La revisión destaca cómo la visión computacional se ha convertido en una herramienta fundamental para la identificación temprana de plagas y

enfermedades, permitiendo reducir pérdidas y optimizar la gestión de cultivos. Se analizan enfoques basados en segmentación de imágenes, extracción de características de color, textura y forma, así como métodos de clasificación automática mediante redes neuronales y máquinas de soporte vectorial. El trabajo subraya que, aunque estas técnicas han demostrado eficacia en la detección de síntomas en hojas, tallos y frutos, aún existen desafíos relacionados con la variabilidad ambiental y la necesidad de bases de datos robustas que permitan entrenar modelos más generalizables. En conclusión, el artículo plantea que la integración del procesamiento digital de imágenes con inteligencia artificial y sensores especializados constituye una vía prometedora para avanzar hacia sistemas agrícolas más eficientes y sostenibles.

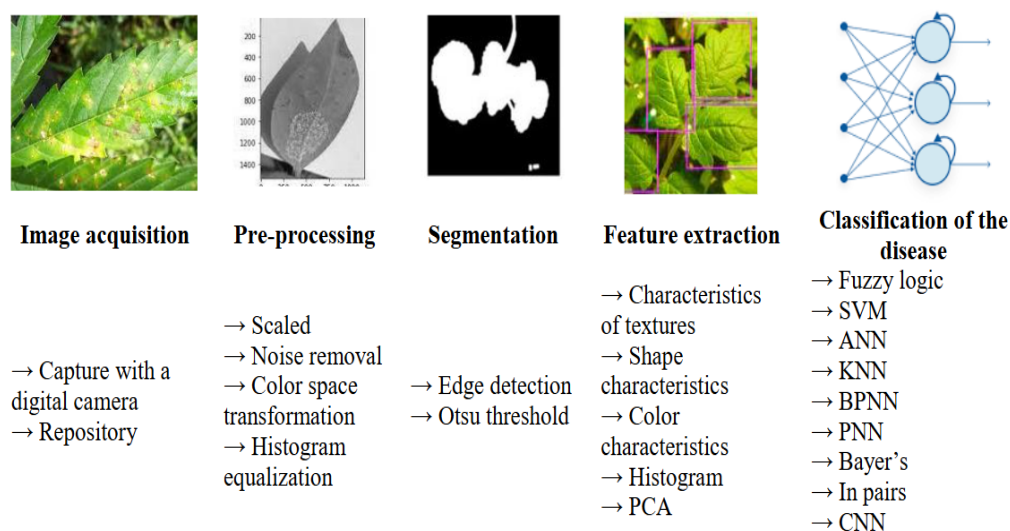


Figura 21: Pasos para obtención de características de plantas mediante procesamiento de imágenes.

CAPÍTULO 3: ESTADO DEL CAMPO

En este apartado se analizarán las principales patentes que circunscriben el estado del campo de este proyecto.

En [19] se desarrolló un aparato de cultivo automatizado de alta densidad, donde se reemplaza la rejilla rectangular, estática en las lacas de plantas cultivadas en las macetas con un espacio de cultivo anular que gira diariamente. En el interior de este espacio, las condiciones ambientales varían en función de la distancia desde el centro y el ángulo del anillo.

Para la inserción de las plántulas recién germinadas en el nivel del borde interno de una zona de cultivo anular, el espacio entre las plantas se encuentra al mínimo. Esto aumenta el rendimiento por metro cuadrado y reduce los costos de iluminación y calefacción. A medida que las plantas crecen, son empujadas hacia el borde exterior del toroide aumentando gradualmente la separación entre ellos a medida que crece su follaje. Luego, las plantas se cosechan desde un único punto en el borde exterior.

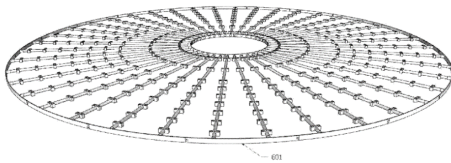


Figura 22: Capa única de crecimiento.

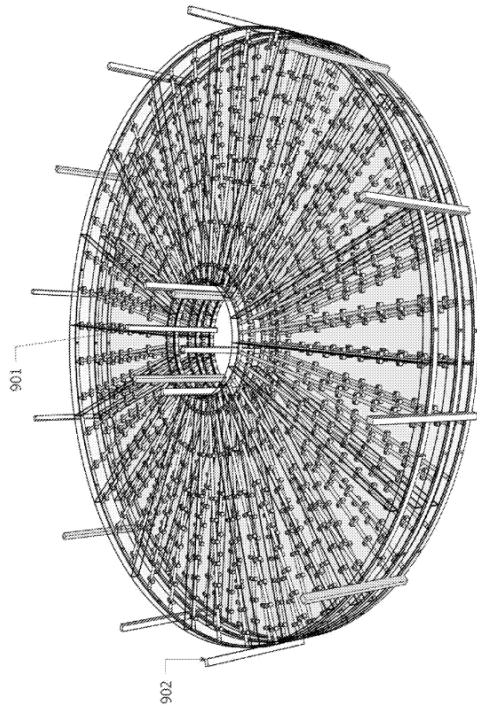


Figura 23: Pila de 3 capas de crecimiento.

En [20], registrada en 2015 y publicada en 2017, se describe un sistema de procesamiento de imágenes multiespectrales e hiperespectrales orientado a la agricultura de precisión. Su propuesta se centra en la captura de datos mediante sensores ópticos avanzados y su posterior análisis para generar mapas georreferenciados que permiten evaluar el estado fisiológico de los cultivos. Este desarrollo tecnológico busca optimizar la detección de anomalías como estrés hídrico, deficiencias nutricionales o presencia de plagas, ofreciendo una herramienta más precisa que los métodos tradicionales de monitoreo agrícola. En el contexto del estado del arte, esta patente representa un avance significativo al integrar técnicas de visión computarizada con sistemas de información geográfica, consolidando la tendencia hacia una agricultura más automatizada, eficiente y sostenible. Además, su impacto se refleja en investigaciones científicas posteriores que han adoptado tecnologías similares para el análisis multiespectral con drones y plataformas terrestres, reforzando la importancia de la innovación patentada como base para el desarrollo académico y aplicado en el campo de la agricultura digital.



Figura 24: Mapa de nitrógeno de un cultivo mediante procesamiento de imágenes.

CAPÍTULO 4: METODOLOGÍA

En este capítulo se expondrá la metodología utilizada para llevar a cabo este proyecto, así como el desarrollo del mismo.

Como se muestra en la tabla 1, este proyecto se llevó a cabo en 6 fases.

Fases	Actividades
Recopilación de información	Búsqueda de información acerca de procesos mecánicos referentes al tema central.
	Búsqueda de información acerca de proyectos de procesamiento de imágenes aplicados a la agricultura.
Propuestas de solución	Búsqueda de propuestas que cumplan con los objetivos del proyecto en función de la hipótesis.
	Determinar y seleccionar las ideas que cumplan con los requisitos del proyecto y que al mismo tiempo sean innovadoras.
Diseño del prototipo	Desarrollo de bocetos y bosquejos sobre posibles propuestas de la estructura final.
	Desarrollo de simulación en software de diseño 3D sobre el mecanismo final del prototipo.
	Diseño e impresión 3D de piezas, bases y soportes para la estructura del prototipo.
Diseño electrónico	Selección de dispositivos para la implementación electrónica en el prototipo.
	Unificar la parte mecánica y electrónica del sistema.
	Diseño del circuito electrónico para el funcionamiento del mecanismo.

Fases	Actividades
Software de control	Desarrollo del sistema de procesamiento de imágenes para el control del mecanismo final en lenguaje Python.
Experimentación	Análisis y recopilación de datos y tiempo del sistema.
	Validación funcional del sistema del prototipo.

Tabla 1: Fases y actividades del proyecto

A continuación se desglosarán los puntos importantes que refieren el esquema anterior sobre la metodología del desarrollo del prototipo.

PROPUESTAS DE SOLUCIÓN

Con respecto a la hipótesis, se presentaron propuestas que aportarán soluciones en la resolución de la problemática principal de este trabajo.

La lenta colocación de sustrato ralentiza el proceso de obtención de plántula, por lo que una posible solución propuesta para esta problemática es crear un sistema de posición de torre de invernadero cuyo diseño sea basado en el mecanismo de un guía de transmisión de movimiento lineal, el mecanismo estará ligado a un sistema de corrección de insertos de torres de invernadero mediante un algoritmo de procesamiento de imágenes.

DISEÑO DEL PROTOTIPO

DESARROLLO DE BOCETO

La ergonomía en este contexto de procesos agrícolas es esencial, por lo que se consideraron posibles soluciones en cuanto a la resolución de esta problemática.

Se propuso una estructura mecánica, que le permitiera al usuario una intervención manual ergonómica, por lo que se tomó de referencia el diseño y mecanismo de una guía de movimiento lineal, como anteriormente se menciona

Para el inicio del desarrollo del proyecto, se realizaron bocetos y bosquejos sobre posibles diseños y/o mecanismos que llevaría el prototipo, mediante el software de diseño fusion 360 de AutoDesk.

Para el mecanismo del prototipo, se tomó de referencia una guía de movimiento lineal de husillo de bolas, cuyos componentes principales son un motor de corriente directa (DC) y un husillo, que realizan un movimiento de transmisión lineal para el transporte de materiales. Tomando en cuenta esto, se realizó un primer bosquejo para el mecanismo del prototipo, el cual se compone de un motor de engranajes de 24 V, un husillo, un cople rígido y la torre de invernadero a escala.

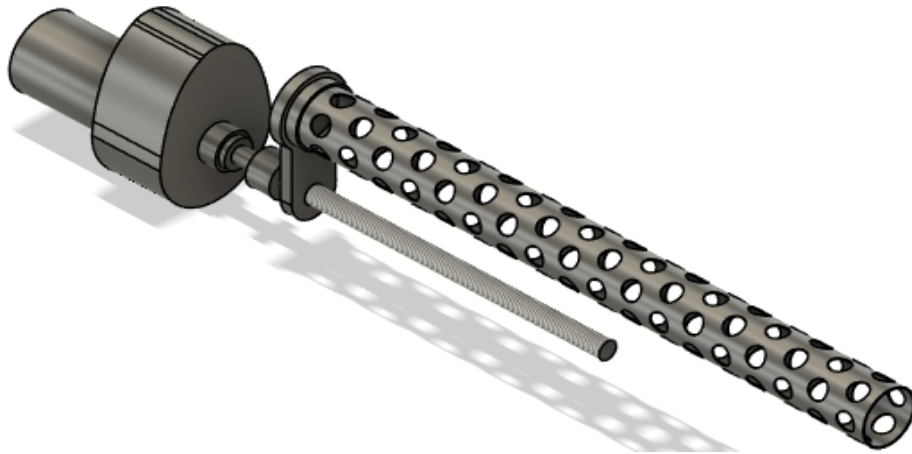


Figura 25: Diseño de partida del mecanismo del prototipo que cuenta con un motor reductor de 12 V, un husillo, un cople y una base para una torre de invernadero a escala.

El mecanismo de un husillo consiste en la transmisión de fuerza y movimiento lineal mediante un tornillo sin fin. El mecanismo final del prototipo combina el mecanismo de un husillo (transmisión de movimiento lineal), con la transmisión de movimiento rotacional, en este caso, de la torre de invernadero. Por consiguiente, se utilizó un motor a pasos para el movimiento rotacional.

SIMULACIÓN DE MECANISMO

Posterior a validar los posibles componentes a utilizar, se obtuvo un bosquejo final del prototipo y se realizó una simulación del mecanismo final que tendrá el mismo, como se muestra en [26](#)

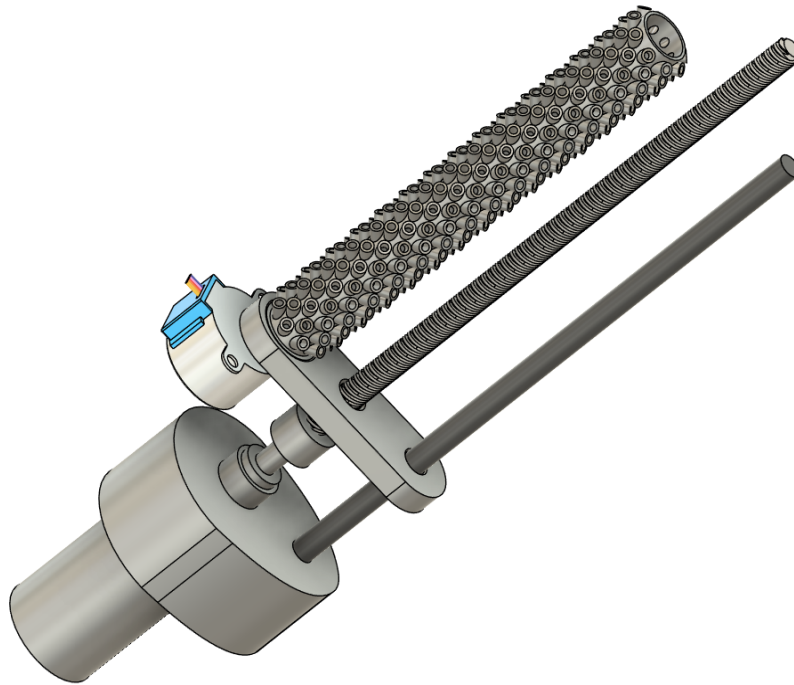


Figura 26: Diseño final y simulación del prototipo con componentes estructurales y dispositivos a utilizar.

Como se muestra en la figura [26](#) y se mencionó en el punto previo, se analizaron y validaron posibles componentes que realizarán el mecanismo final, en donde más adelante se explicará su funcionamiento y la razón de su selección con más detalle. Para el mecanismo de transmisión lineal, se añadió un cople rígido para el acoplamiento del tornillo sin fin con el motor DC, para el mecanismo de movimiento rotacional de la torre se consideró utilizar un motor a pasos, por otra parte, se diseñó una base, acoplada al motor a pasos, que funciona como elevador de la torre y por último se añadió una varilla para facilitar la transmisión lineal del husillo.

DISEÑO E IMPRESIÓN 3D

Una vez obtenido el diseño final y la simulación como referencia del mecanismo final del prototipo, se procedió con el diseño de las piezas, bases y soportes de la estructura del prototipo para su ensamblaje.

En primera estancia se obtuvieron las medidas de una torre de invernadero de tamaño real hecho a base de material de PVC; Esto con el objetivo de diseñar una torre de invernadero a escala e imprimirlo en 3D a base de material de PLA.



Figura 27: Torre de invernadero de tamaño real.

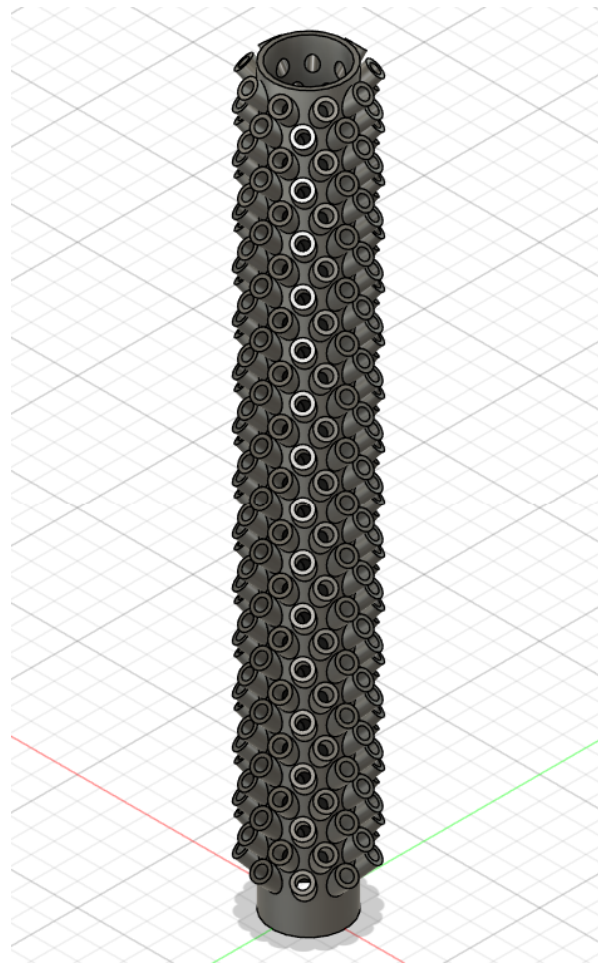


Figura 28: Diseño de torre de invernadero a escala en fusion 360.

Las dimensiones de la torre de invernadero de tamaño real (figura 27) consisten en 1.50 m

de altura con 16 cm de diámetro y cuenta con 270 insertos. En la figura 28 se aprecia el diseño de la torre de invernadero a escala, la escala utilizada fue de 1:10, esto para lograr una congruencia con las dimensiones de la torre y la cantidad de insertos.



Figura 29: Torre de invernadero impresa en 3D.

Con respecto a la figura 26, y como se mencionó anteriormente, para el acoplamiento del motor DC con el tornillo sin fin, se utilizó un cople rígido para impresoras 3D, como se muestra en la siguiente figura



Figura 30: Cople rígido para impresora 3D.

Sus dimensiones consisten en un acoplamiento de 6-8 mm, con 2.5 cm de altura y 1.3 cm de ancho.

Generalmente los insertos de las torres verticales de invernadero están colocados en una orientación de 45° y para ergonomizar el proceso de inserción de sustrato, se propuso orientar la torre a 45° colocando el motor DC a 45° y con eso posicionar los insertos a 90° . Por lo tanto, se diseñó e imprimió una base inclinada a 45° para el motor DC.

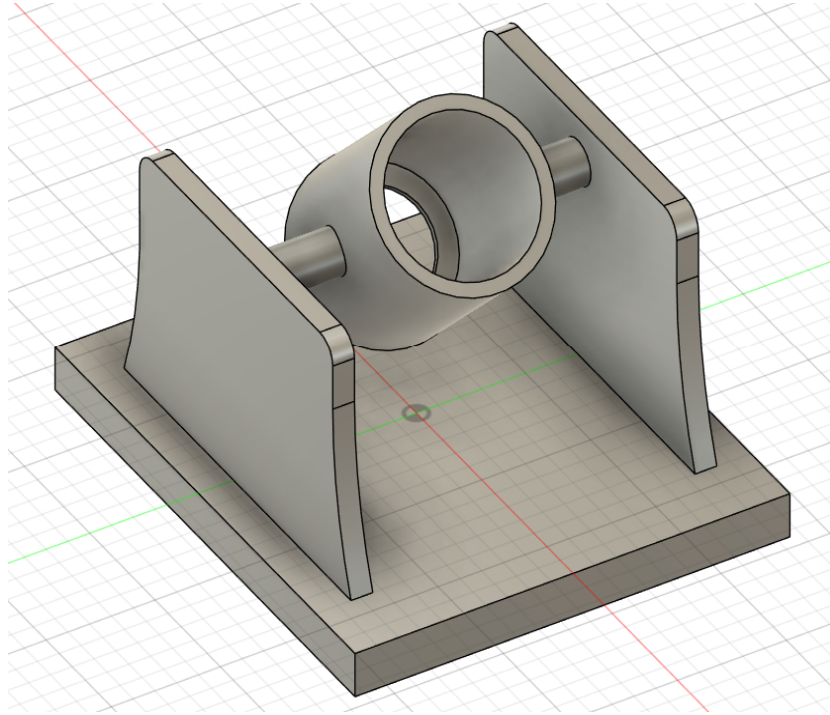


Figura 31: Diseño de base para motor DC

Una vez diseñada la base, se acopló el cople rígido al rotor del motor DC, al mismo tiempo que se colocó el motor en la base inclinada.

Como actuador para el mecanismo de ascenso/descenso de la torre de invernadero, se utilizó un motor de engranajes de 12 a 24 V.



Figura 32: Motor de engranajes de 12 a 24 V

La elección de un motor de engranajes de 12-24V se basa de las necesidades de torque, estabilidad y continuidad de funcionamiento del sistema. Los motores con engranajes convierten la alta velocidad del rotor en baja velocidad y alto par, necesaria para accionar mecanismos de carga media o para desplazamientos lentos y precisos.

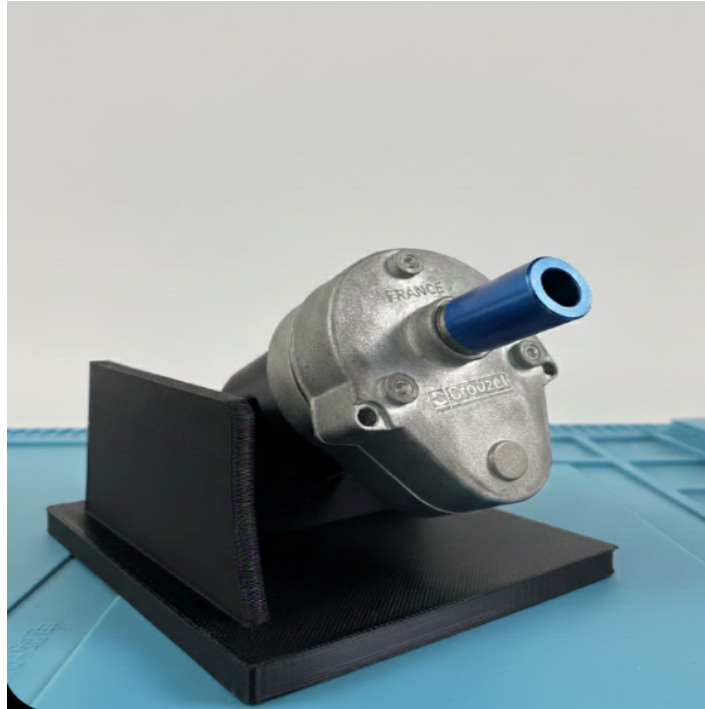


Figura 33: Diseño de base para motor DC

Al tener posicionado el motor DC, se acopló la varilla enroscada al rotor del motor por medio del cople rígido.

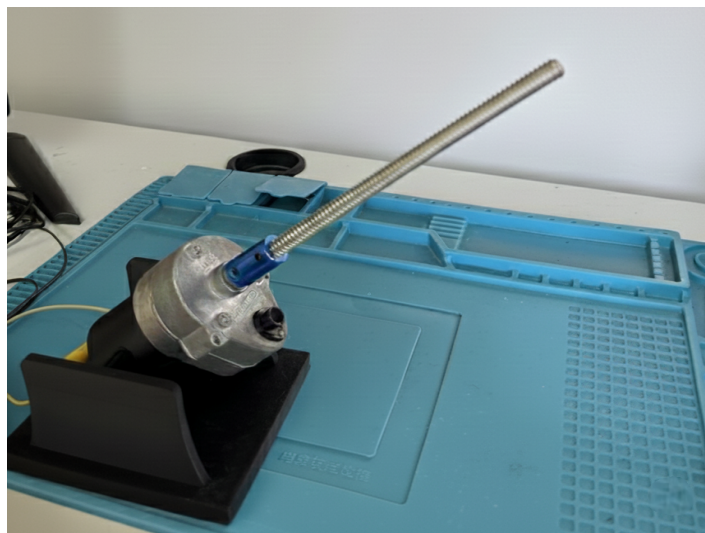


Figura 34: Diseño de base para motor DC con varilla

Las dimensiones de la varilla enroscada consiste en 20 cm de largo, 8 mm de diámetro y un

paso de rosca de 2 mm.

como se muestra en la figura 34, se colocó una pieza para acoplar una varilla al motor que cumplirá la función de riel.

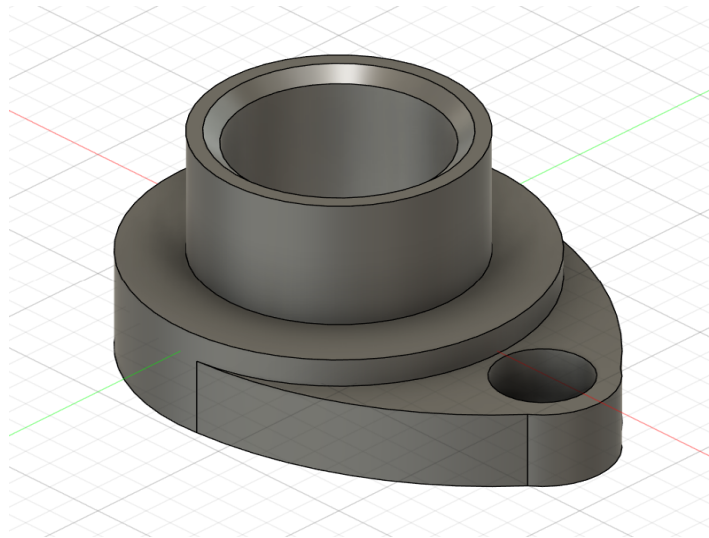


Figura 35: Diseño de acople de riel

El diseño de este cople se realizó con una dimensión de 7 mm de diámetro

Posteriormente, se procedió a diseñar las bases y soportes para la torre de invernadero a escala.

Se diseñó una base que sostendrá la torre de invernadero y realizará la función de elevación y retroceso de la misma. El diseño de la base consistió en una base en forma de óvalo con 3 ranuras, en donde una de ellas es donde quedará acoplada la torre, la ranura de en medio es donde pasará la varilla enroscada para el mecanismo de elevación y la tercera ranura funciona para el paso del riel, la función del riel consiste en evitar el giro rotacional del elevador de la torre de invernadero.

El diseño de la base se realizó para el acoplamiento a una tuerca combrida, que es la encargada de la elevación de la base y de la torre de invernadero. El mecanismo de elevación consiste en la rotación de la varilla enroscada mediante el motor, esa rotación le permite a la tuerca combrida desplazarse linealmente debido a su rosca interna.

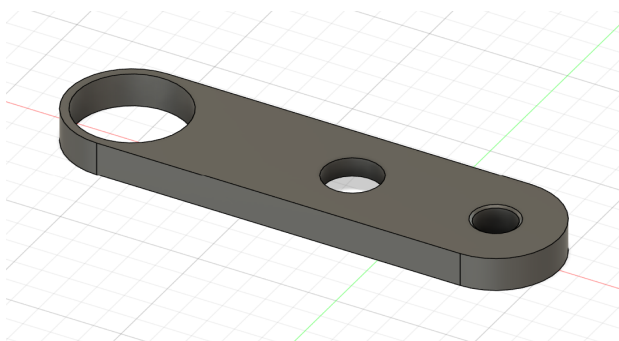


Figura 36: Base avance/retroceso en 3D (vista frontal).

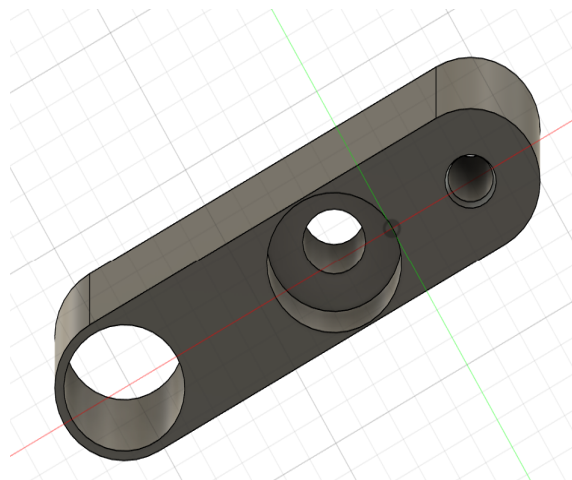


Figura 37: Base avance/retroceso en 3D (vista posterior).

Las dimensiones de la base consisten en 6.6 cm de largo, ranura superior de 20 mm, ranura media de 10.40 mm y ranura lateral de 7.5 mm.



Figura 38: Base avance/retroceso impresa en 3D.



Figura 39: Tuerca combrida de varilla enroscada.



Figura 40: Base avance/retroceso acoplada con tuerca combrida (vista frontal).



Figura 41: Base avance/retroceso acoplada con combrida (vista posterior).

Una vez diseñada e impresa la base que facilitará el movimiento de ascenso y descenso de la torre, se contempló añadir el movimiento rotacional de la misma, por lo que se pensó en utilizar un motor a pasos 28byj-48, debido a que este motor eléctrico, nos permite programar movimientos mecánicos discretos precisos llamados pasos mediante pulsos digitales, con esto es posible modificar su velocidad.



Figura 42: motor a pasos 28byj-48

Tomando esto en cuenta, se diseñó un cople para el motor a pasos donde se colocará la torre de invernadero.

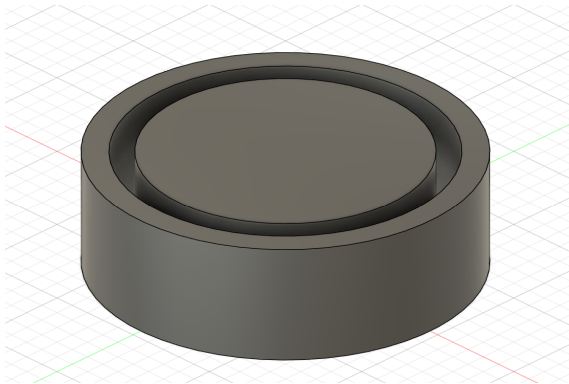


Figura 43: Cople motor a pasos para torre de invernadero (vista frontal).

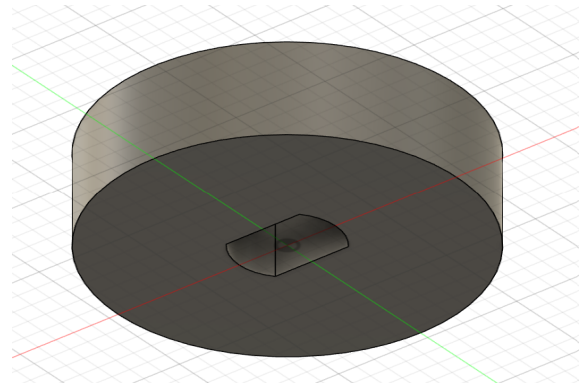


Figura 44: Cople motor a pasos para torre de invernadero (vista posterior).

Como se observa en la figura 43 y en la figura 44, el diseño de esta pieza consistió en una pieza circular, con una ranura por donde se acoplará el motor a pasos, de igual manera se diseñó un molde, donde será insertada la torre de invernadero. Las dimensiones de la pieza consisten en un diámetro de 19 mm con 7.5 mm de altura. El molde fue diseñado de acuerdo a las dimensiones de la torre de invernadero.

Adicionalmente, se añadió una varilla sin rosca. Esta varilla cumple la función de riel, evitando la rotación innecesaria a la base diseñada para la elevación de la torre de invernadero, como se observa en la siguiente figura.

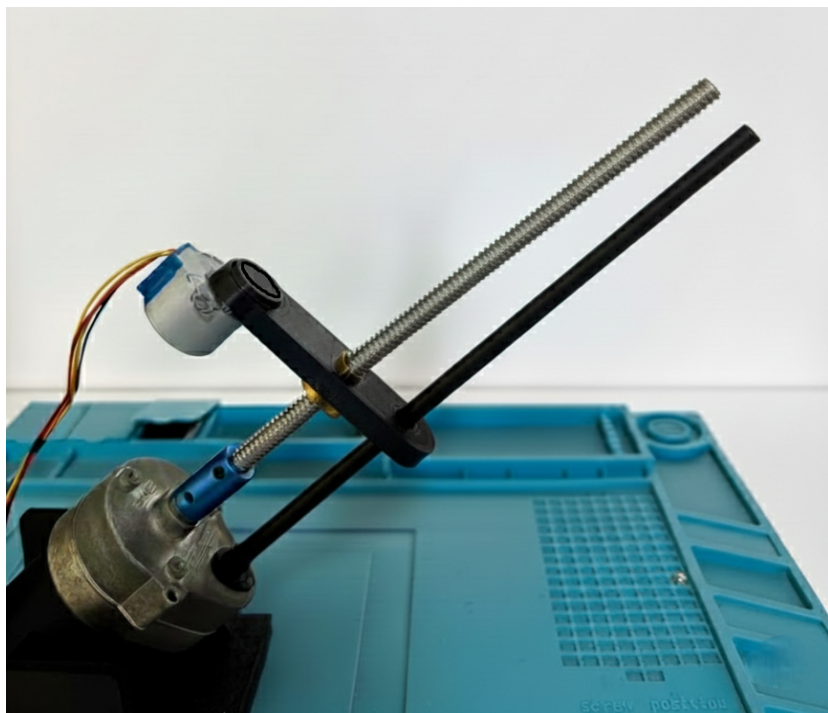


Figura 45: Estructura básica del prototipo

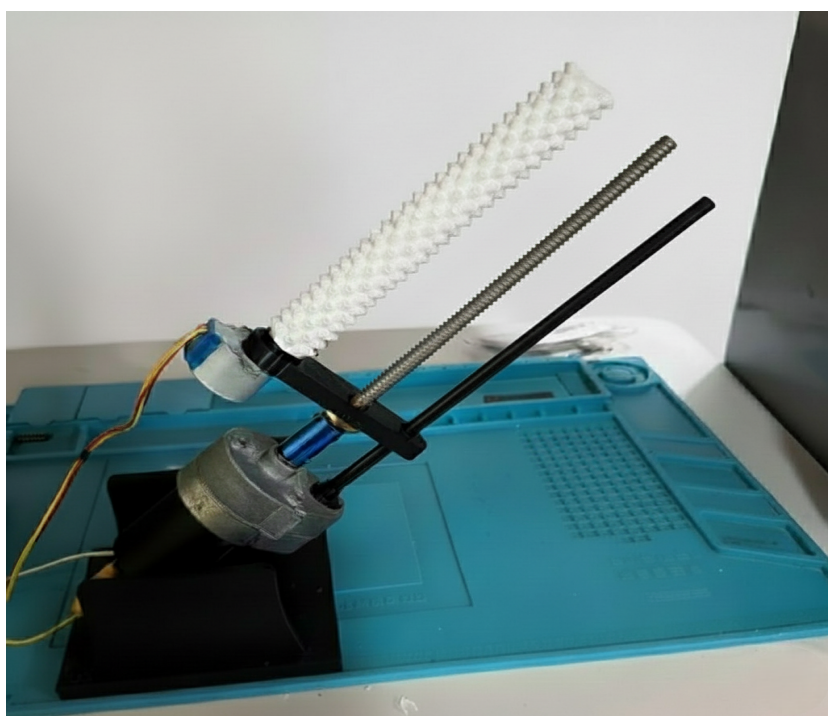


Figura 46: Estructura básica del prototipo con torre de invernadero

Como se ha explicado el proceso de diseño y ensamblaje la figura 46 se muestra la estructura básica del prototipo que incluye una base inclinada para el motor DC, un cople rígido para el acople de la varilla enroscada con el roto del motor DC, una base para el riel acoplado al motor, una base donde se alojará la torre de invernadero para el mecanismo de avance y retroceso y a su vez acoplada a la varilla enroscada y al riel, la torre de invernadero a escala que está unida al motor a pasos mediante un cople circular.

Para reducir cualquier tipo de vibración en la estructura, se utilizó un rodamiento/balero de bolas. En mecanismos, como una guía de movimiento lineal con husillo de bolas, hacen uso de rodamientos para evitar cualquier desalineación o vibración en la transmisión de movimiento.

Se diseñó un soporte para alojar el rodamiento que consistió en 2 partes: La pieza de alojamiento del rodamiento y una base que sostendrá al mismo para evitar movimiento.

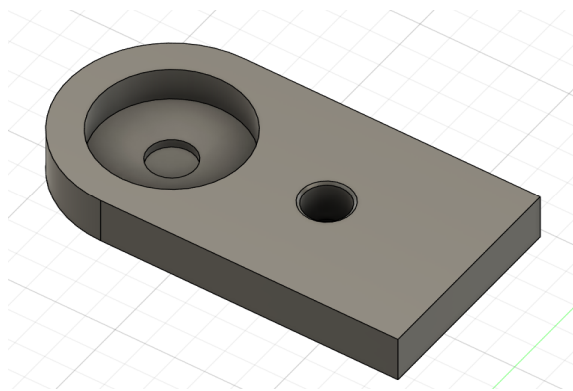


Figura 47: Diseño de soporte de rodamiento.



Figura 48: Soporte de rodamiento impreso en 3D.

Como se muestra en las figuras 47 y 48, el diseño del rodamiento cuenta con una espacio en el que se insertará un balero que le permitirá a la varilla enroscada girar libremente sin desbalanceo.

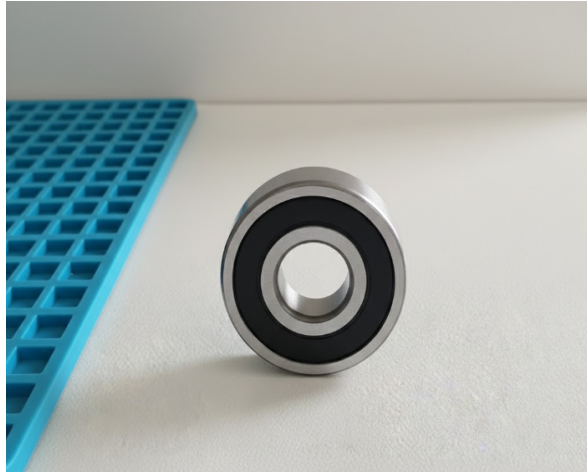


Figura 49: Balero de giro



Figura 50: Soporte con rodamiento de bolas

Con el soporte del rodamiento diseñado e impreso en 3d, se procedió con el diseño de la base que sostendrá al soporte.

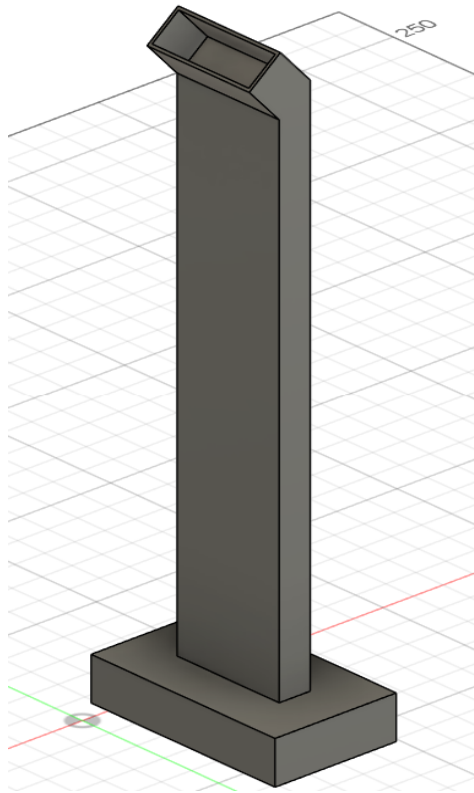


Figura 51: Base para soporte de rodamiento

Como se puede observar en la figura 51, La base tiene la función de mantener firme y estable al rodamiento, por lo que de esta manera se reducen vibraciones, balanceos o cualquier irregularidad en el mecanismo final.

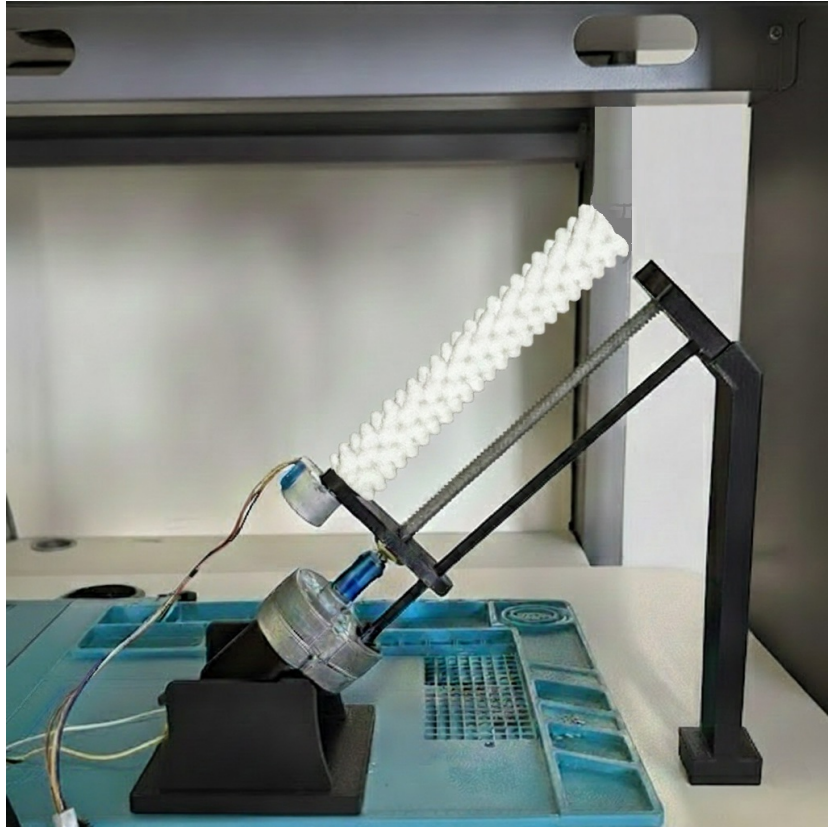


Figura 52: Estructura final del prototipo

En la figura 52 se observa la estructura final del prototipo con todos los componentes que lo componen, incluyendo el último descrito, el soporte del rodamiento con su base vertical.

Una vez ensamblada la estructura, se procedió con las pruebas preliminares de los mecanismos de elevación/retroceso y de rotación de la torre de invernadero para implementar el mecanismo final.

DISEÑO ELECTRÓNICO

En este apartado se aborda la elección, justificación y configuración de los elementos electrónicos (sensores y actuadores) que integran al sistema, en términos de eficiencia, exactitud, seguridad y compatibilidad con la arquitectura general del proyecto.

Como se mencionó anteriormente, para el diseño electrónico del sistema se seleccionaron

dispositivos específicos, entre ellos un motor de engranajes de 12–24 V y un motor a pasos.

MOTOR A PASOS

Respecto al motor paso a paso, como se ha indicado anteriormente, se decidió emplear un motor a pasos 28BYJ-48 es un motor paso a paso de tipo unipolar, utilizado en aplicaciones de control de posición y automatización. Este dispositivo se caracteriza por su bajo consumo de energía y su capacidad para proporcionar un control preciso del movimiento. Su diseño compacto lo hace adecuado para diversas aplicaciones, incluyendo proyectos de robótica y sistemas de control de movimiento en general. La decisión de emplear este motor responde a las necesidades específicas del sistema: lograr una buena precisión, asegurar la repetibilidad en los movimientos y contar con un control angular fino. Estas cualidades resultan difíciles de alcanzar con motores DC o servomotores convencionales, lo que convierte al 28BYJ-48 en una alternativa adecuada para este proyecto. A continuación, se presenta la comparación técnica que sustenta esta elección:

Característica	Motor DC pequeño	Servomotor (SG90/MG995)	Motor a pasos 28BYJ-48
Control del ángulo / Rango de movimiento	No posee; requiere encoders adicionales.	Normalmente limitado (180° o 270°).	Giro continuo en pasos, puede alcanzar 360° o más.
Velocidad	Alta, pero difícil de regular con exactitud.	Controlada mediante PWM, velocidad moderada.	Baja-media, altamente controlable mediante pasos.
Precisión	Baja sin retroalimentación externa.	Buena, depende del potenciómetro interno.	Alta, con control directo del avance por pasos.
Repetibilidad / Estabilidad en posición	Limitada sin retroalimentación.	Puede presentar juego mecánico (backlash).	Excelente, cada paso garantiza repetibilidad y estabilidad.

Característica	Motor DC pequeño	Servomotor (SG90/MG995)	Motor a pasos 28BYJ-48
Control	Requiere PWM y, para precisión, un encoder.	Señal PWM específica (modulada en ancho).	Secuencia digital directa con ULN2003.
Torque a baja velocidad	Bajo, generalmente pierde fuerza.	Alto, debido a la reducción interna del motor.	Muy bueno debido a la reducción interna del motor.
Aplicaciones típicas	Ventilación, bombas, giros continuos.	Robótica ligera, control angular fijo, articulaciones.	Mecanismos de posicionamiento y automatización ligera.

Tabla 2: Comparación general entre un motor DC pequeño, servomotor y motor a pasos 28BYJ-48. [1]

Cada uno de estos actuadores (motor dc y motor a pasos) requiere la incorporación de controladores electrónicos (drivers) que permitan su operación adecuada dentro del sistema. Si bien ambos motores pueden ser energizados únicamente mediante una fuente de alimentación directa, esta configuración únicamente posibilita su funcionamiento básico. Para realizar modulación de velocidad, control del sentido de giro, secuencia de fases o generación de señales de conmutación, es indispensable el uso de drivers diseñados para gestionar la corriente, el voltaje y los patrones de activación requeridos por cada tipo de motor.

Por lo tanto, los controladores no solo habilitan el control dinámico y seguro de los actuadores, sino que además garantizan la protección del microcontrolador y la estabilidad operativa del sistema mecatrónico.

Los controladores en cuestión son el módulo driver L298N para el control del motor DC y el módulo ULN2003 para el control del motor a pasos.

DRIVER L298N

El L298N es un driver para motores DC y bipolares que funciona como interruptor de potencia (puente H). Un puente H es un circuito electrónico que permite invertir la polaridad de tensión aplicada a una carga. Su arquitectura de puente H permite:

- Control de dirección (sentido de giro) mediante señales lógicas.
- Control de velocidad mediante PWM en los pines Enable.
- Frenado activo al activar ambas salidas iguales.

Las especificaciones de este módulo controlador son:

- Tipo de controlador: Puente H dual.
- Voltaje de alimentación: +5V a +35V.
- Corriente máxima: 2A por puente (pico 3A).
- Regulación: Disipación por calor.
- Característica: Incluye diodos de protección y enable por puente.



Figura 53: Módulo driver L298N

DRIVER ULN2003

El módulo driver ULN2003 es un arreglo de interruptores de alta ganancia que funciona como amplificador de corriente. El driver ULN2003 permite amplificar señales PWM y digitales de tarjetas de desarrollo a voltajes y corrientes de un valor óptimo para controlar un motor a pasos de 4 fases.

- Voltaje de alimentación: 5V.
- Corriente de salida por canal: 500 mA.
- Temperatura de almacenamiento: -65°C a +150 °C

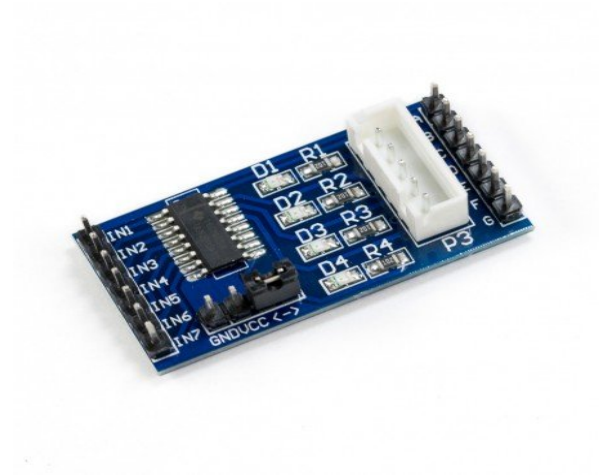


Figura 54: Módulo driver ULN2003

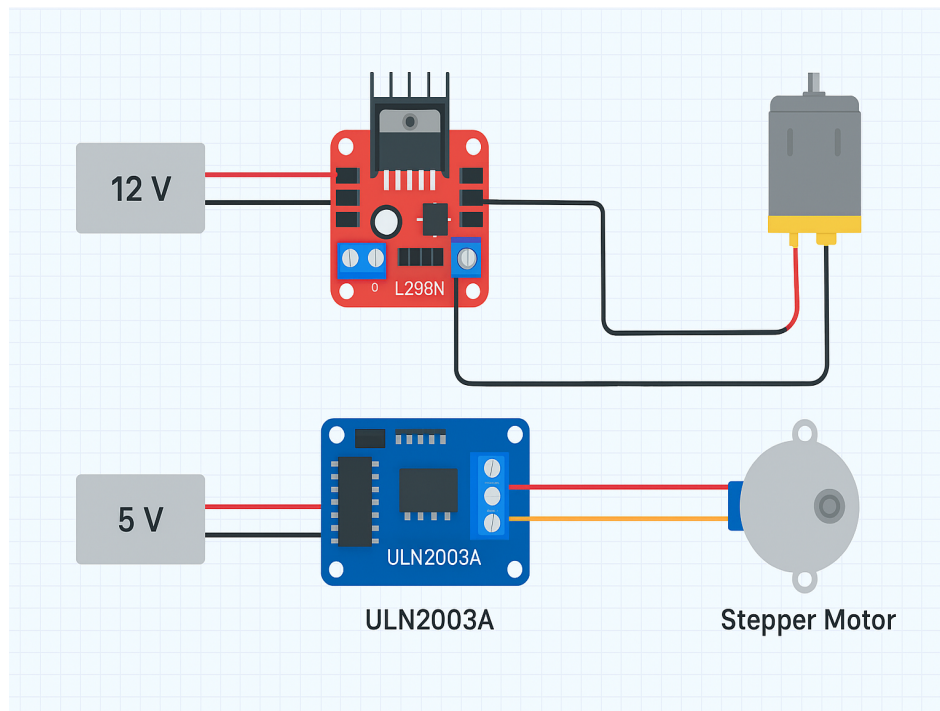


Figura 55: Diagrama básico de conexión de motores

Ambos drivers son controlados mediante una tarjeta raspberry pi 4 como tarjeta de desarrollo.



Figura 56: Tarjeta de desarrollo Raspberry pi 4

RASPBERRY PI 4

De acuerdo con el fabricante, la Raspberry Pi es una microcomputadora de bajo costo y de alto rendimiento que concentra en un solo dispositivo los elementos fundamentales de un sistema de cómputo: procesador, memoria RAM, puertos USB para periféricos, interfaz de red, salidas de video y pines GPIO (Entrada/Salida de propósito general), los cuales facilitan la interacción directa con sensores, actuadores y componentes de control. Su operación se basa en un sistema operativo derivado de Linux, que posibilita el desarrollo, ejecución y gestión de aplicaciones de manera local. Además, los pines GPIO permiten la conexión con dispositivos externos mediante protocolos de comunicación como I2C, SPI y UART, ampliamente utilizados en sistemas electrónicos.

Algunas de las principales especificaciones de la tarjeta se muestran a continuación, junto con la arquitectura general de la misma en la figura [57](#).

- Procesador Broadcom BCM2711, SoC de cuatro núcleos Cortex-A72 (ARM v8) de 64 bits a 1,8 GHz.
- Memoria RAM de 1 GB, 2 GB, 4 GB u 8 GB LPDDR4-3200 SDRAM (según el modelo).
- Conexión inalámbrica IEEE 802.11ac de 2,4 GHz y 5,0 GHz, Bluetooth 5.0.
- Conexión alámbrica Gigabit Ethernet.
- 2 puertos USB 3.0; 2 puertos USB 2.0.
- Conector GPIO de 40 pines estándar para Raspberry Pi (totalmente compatible con placas anteriores).
- 2 puertos micro-HDMI® (compatibles con hasta 4kp60).
- Puerto de cámara CSI MIPI de 2 carriles.
- Puerto de audio estéreo y vídeo compuesto de 4 polos.
- Ranura para tarjeta Micro-SD para cargar el sistema operativo y almacenar datos.

- Alimentación de 5 V CC a través del conector USB-C (mínimo 3 A).
- Alimentación de 5 V CC a través del conector GPIO (mínimo 3 A).

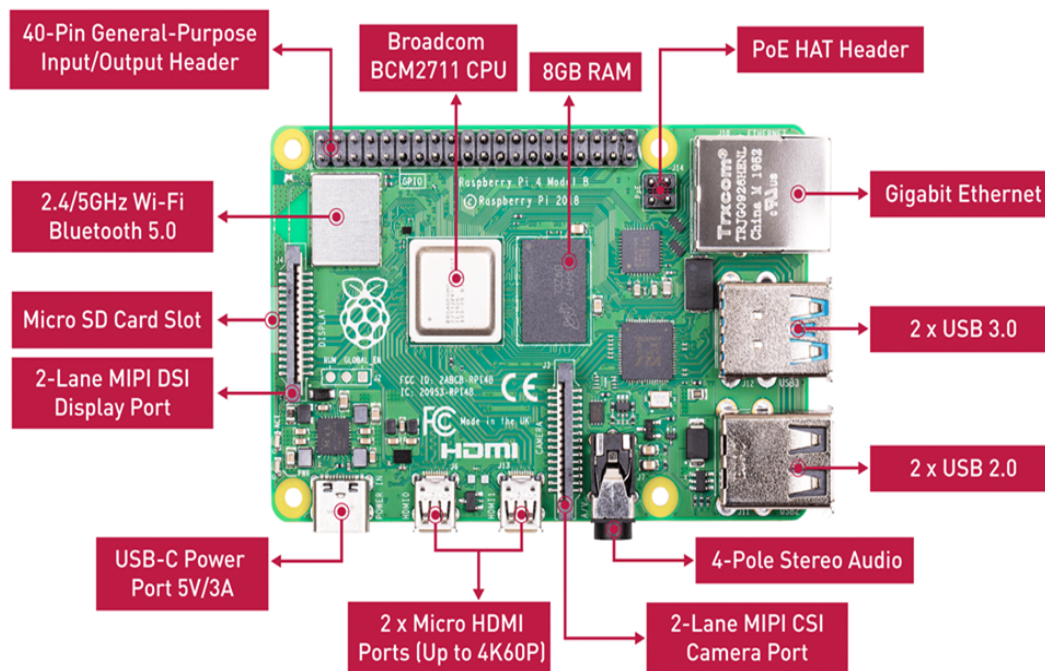


Figura 57: Arquitectura de Raspberry pi 4

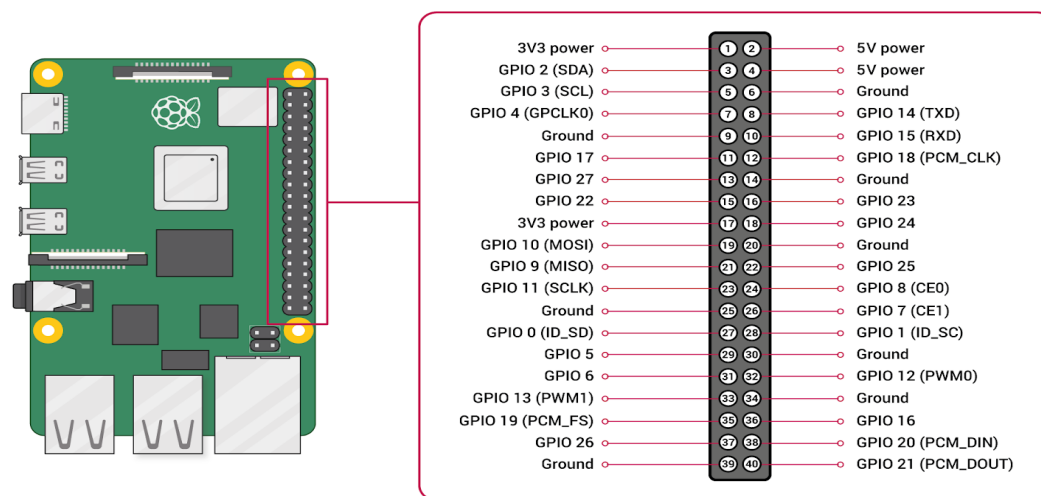


Figura 58: Pines GPIO de Raspberry pi 4

Como anteriormente se mencionó, se utilizó la tarjeta raspberry pi 4 como hardware de control del sistema, incluyendo los actuadores, como se muestra en la siguiente figura, por lo que se hicieron las pruebas preliminares de cada actuador, para posteriormente desarrollar el software de control de todo el sistema. la Raspberry Pi ofrece una arquitectura más robusta para la manipulación de imágenes en tiempo real, soporte nativo para interfaces de cámara y comunicación por red, así como una gestión más eficiente de memoria y datos. Estas características la convierten en una opción óptima para el desarrollo de sistemas de control automatizado asistidos por visión artificial en entornos experimentales o de prototipado.

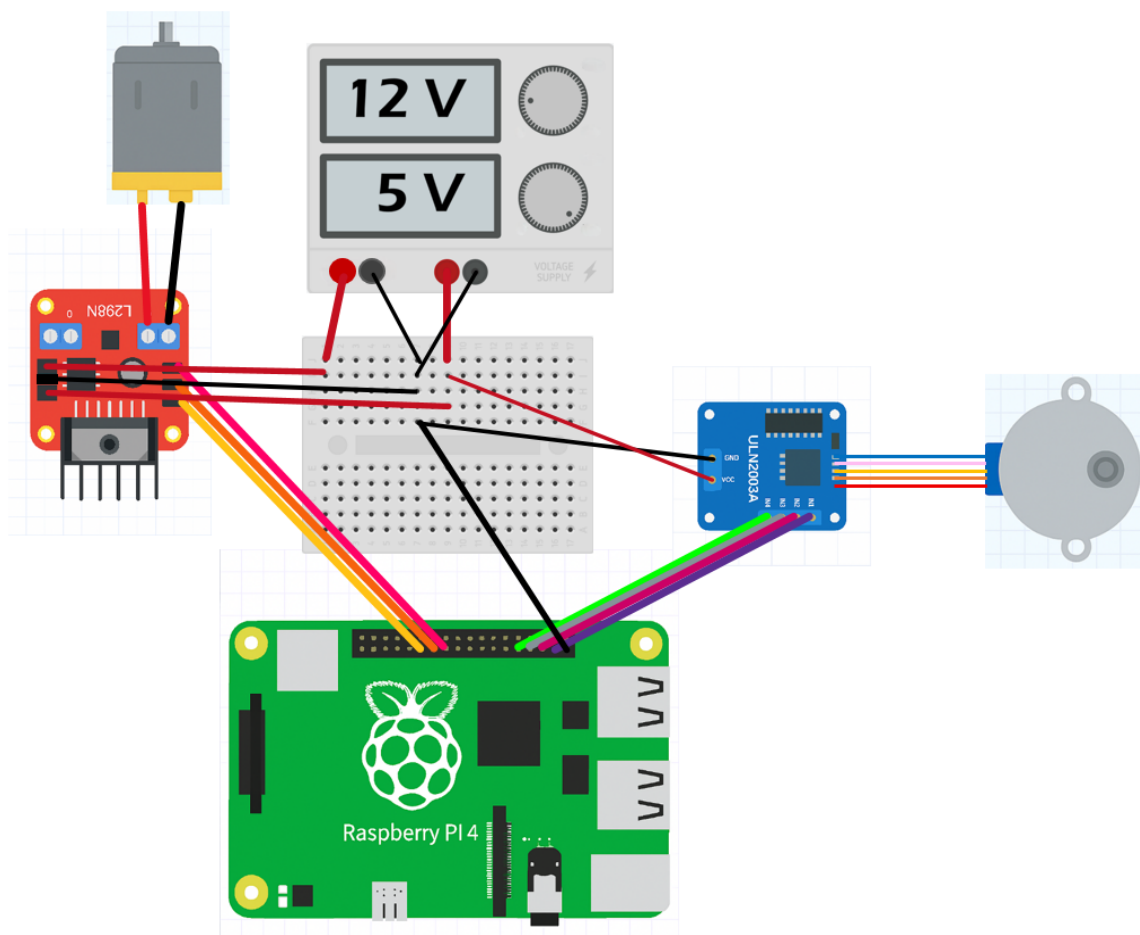


Figura 59: Diagrama del diseño electrónico del sistema

En la figura 59, se muestra el diseño electrónico del sistema, donde se incluye una fuente de alimentación dual de 12 y 5 V, un motor DC con driver L298N y un motor a pasos con driver ULN2003, ambos drivers conectados a una raspberry pi 4.

El pinout del circuito electrónico se mostrará posteriormente mientras se expone el programa de control de de cada prueba de los actuadores.

ALGORITMO DE PRUEBA DE MOTOR DC

Para la prueba de control del motor DC se utilizó un algoritmo básico de control para motores DC mediante un L298N en lenguaje Python, lo cual está explicado en el algoritmo [1](#).

Algorithm 1: Prueba de control de Motor DC con Raspberry Pi y L298N

Input: Pines IN1=17, IN2=27, ENA=22

Output: Motor controlado en avance, retroceso y alto

Inicializar librerías GPIO y tiempo;

Definir pines: IN1 $\leftarrow$ 17, IN2 $\leftarrow$ 27, ENA $\leftarrow$ 22;

Configurar pines como salida;

Inicializar PWM en ENA con frecuencia de 1 kHz;

Iniciar PWM con ciclo de trabajo 0;

Función Avance(speed):;

IN1 $\leftarrow$ HIGH; IN2 $\leftarrow$ LOW;

PWM $\leftarrow$ speed;

Función Retroceso(speed):;

IN1 $\leftarrow$ LOW; IN2 $\leftarrow$ HIGH;

PWM $\leftarrow$ speed;

Función Stop():;

IN1 $\leftarrow$ LOW; IN2 $\leftarrow$ LOW;

PWM $\leftarrow$ 0;

Ejecutar Avance(100) durante 10 s;

Ejecutar Retroceso(100) durante 10 s;

Ejecutar Stop();

Finalizar PWM y limpiar configuración GPIO;

ALGORITMO DE PRUEBA PARA MOTOR A PASOS

Posteriormente se probó un algoritmo básico para el control de motor a pasos con ULN2003 en Python, como se muestra en el algoritmo 2.

Algorithm 2: Prueba de control de motor 28BYJ-48 con ULN2003 en Raspberry Pi

Input: Pines IN1=5, IN2=6, IN3=13, IN4=19

Output: Motor paso a paso controlado en sentido horario y antihorario

Inicializar librerías GPIO y tiempo;

Definir lista de pines: `pins = [IN1, IN2, IN3, IN4];`

Configurar pines como salida y ponerlos en bajo;

Definir secuencia de pasos `seq` para motor 28BYJ-48;

Función `step_motor(steps, delay, clockwise);`

 Seleccionar secuencia: `seq` si horario, `reversed(seq)` si antihorario;

 Para cada paso en `steps`::

 Para cada patrón en la secuencia::

 Asignar valores a los pines según el patrón;

 Esperar `delay`;

Ejecutar `step_motor(100, 0.008, horario)` → una vuelta;

Ejecutar `step_motor(100, 0.008, antihorario)` → una vuelta;

Finalizar y limpiar configuración GPIO;

Una vez realizadas las pruebas para cada actuador, se procedió con el desarrollo del sistema de corrección de los insertos mediante procesamiento de imágenes.

SOFTWARE DE CONTROL

En esta sección se presenta el desarrollo del sistema de corrección de insertos basado en procesamiento de imágenes, cuyo objetivo es asegurar que cada inserto sea colocado con precisión en su posición correspondiente. Este algoritmo permite identificar desviaciones y corregir el posicionamiento mediante ajustes mecánicos. Como se explicó al inicio del capítulo, el prototipo integra un mecanismo de desplazamiento lineal, accionado por un motor DC, y un

mecanismo rotacional, controlado por un motor a pasos. La combinación de ambos genera un patrón de movimiento en espiral que facilita la ubicación precisa de los insertos dentro de la torre de invernadero. En la siguiente figura se ilustra gráficamente el funcionamiento general de este mecanismo.

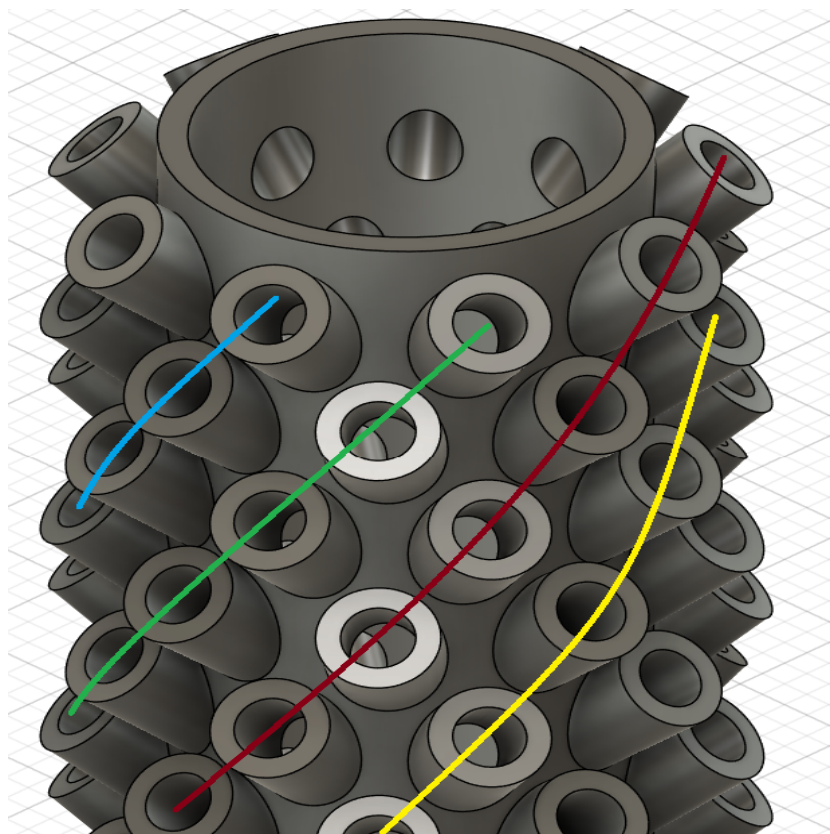


Figura 60: Representación gráfica de la proyección de movimiento de la torre de invernadero

Como se muestra en la figura 60, las líneas de colores muestran la trayectoria que tendrá la torre de invernadero en forma de espiral. El algoritmo de corrección de posicionamiento consistirá primero en la detección de los insertos en una zona particular, posteriormente, el algoritmo calculará el error de desfase de cada inserto para luego accionar los actuadores para la corrección de sus posiciones.

Existen diferentes métodos de procesamiento de imágenes que se pudieron haber empleado para el objetivo de este trabajo (detección de insertos), los cuales se validaron las ventajas y desventajas de cada uno para emplear la elección más óptima para el objetivo principal.

En siguiente, tabla, se mostrarán los principales métodos de procesamiento de imágenes, sus descripciones, ventajas y desventajas.

Métodos	Descripción	Ventajas	Desventajas
Detección de objetos (basada en aprendizaje automático o redes neuronales).	Utiliza modelos entrenados (YOLO, SSD, Faster R-CNN, etc) para reconocer y localizar objetos mediante características aprendidas.	Alta precisión en entornos complejos; adaptable a múltiples tipos de objetos.	Requiere una gran cantidad de datos de entrenamiento y alto consumo computacional, además de sensibilidad a variaciones de iluminación y ángulo.
Detección de formas (basadas en contornos o transformadas en hough).	Utiliza extracción de bordes y geometrías específicas.	Útil para objetos de formas bien definidas y procesamiento más ligero.	Sensible al ruido visual y sombras, limitada ante variaciones de color o textura.
Detección por color (basada en umbrales de color en espacios RGB o HSV)	Identifica regiones de interés (ROI) mediante segmentación de rangos de color previamente definidos.	Bajo costo computacional, fácil de implementar, alta velocidad de procesamiento, buena precisión en entornos controlados.	Sensible a variaciones extremas de iluminación o sombras, requiere calibración de los rangos de color.

Tabla 3: Comparación de técnicas de detección de objetos en imágenes. [2].

Como se observa en la tabla 4, existen 3 principales métodos/técnicas de procesamiento de imágenes. Al contemplar la complejidad del proyecto y los recursos disponibles, la opción más práctica e idónea para aplicar en este trabajo fue la técnica de detección de colores. Por esta

razón, cada espiral (conformada por los insertos) fue teñida de distintos colores para facilitar el desarrollo del sistema de procesamiento de imágenes. Como se muestra en la siguiente figura.

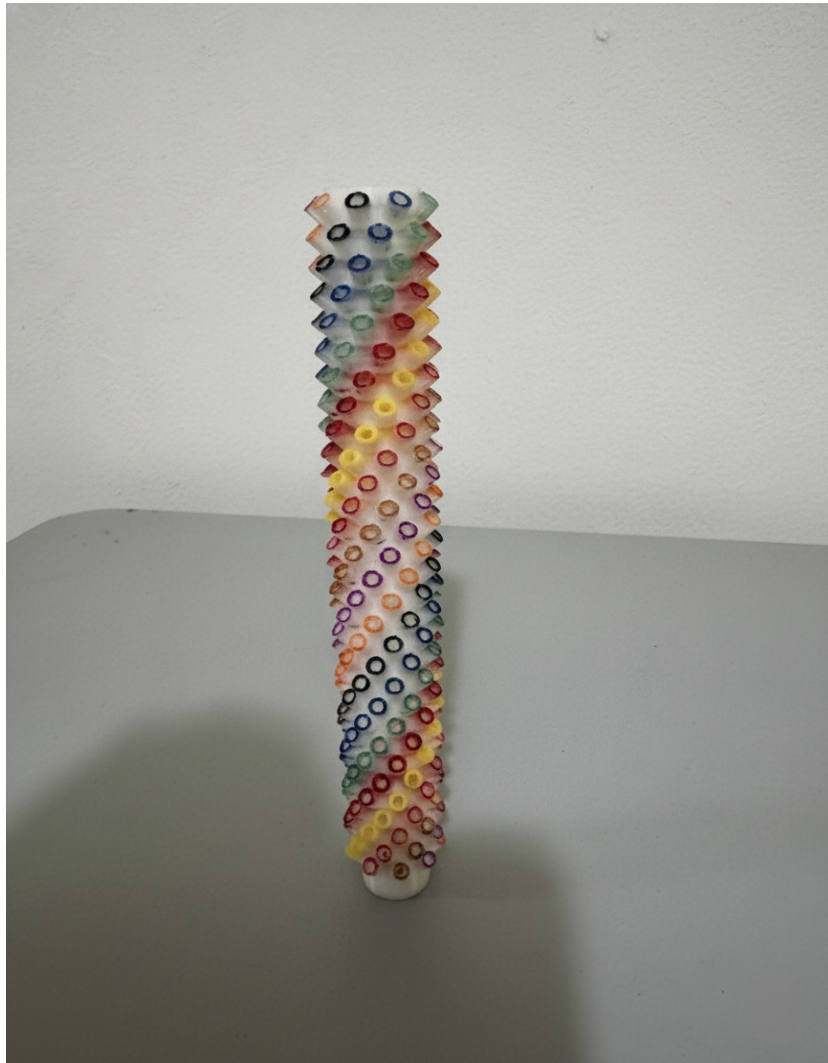


Figura 61: Torre de invernadero con espirales de insertos diferenciados por colores.

Una vez comprendido el objetivo del mecanismo del prototipo, se procedió con el desarrollo del sistema de procesamiento de imágenes. El sistema fue desarrollado en el entorno de programación Visual Studio Code, empleando el lenguaje de programación Python debido a su versatilidad y amplia compatibilidad con bibliotecas de visión por computadora.

Se utilizó una cámara USB de tipo plug and play para el sistema de detección.



Figura 62: Cámara USB utilizada en el prototipo.

La cámara, mostrada en la figura 62, tiene un enfoque automático de 2 MP, una calidad de imagen FHD con una resolución de captura de video de 1080 p, adicionalmente cuenta con un sensor fotográfico CMOS.

El acoplamiento de la cámara con el prototipo consistió en el diseño de una base para la antes mencionada. El diseño de la base radicó en 2 soportes, que se expondrán a continuación.

En primera estancia, debido a la arquitectura de la cámara, se diseñó una base que albergará el cuerpo de la misma, debido a su incapacidad de firmeza.

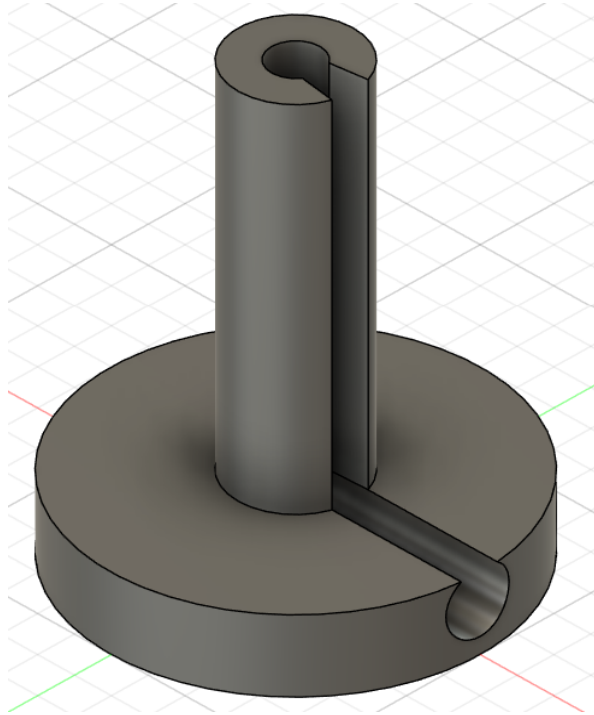


Figura 63: Base para cuerpo de cámara USB.

Como se muestra en la figura 64, se tomó de referencia el diseño de una lámpara de escritorio con ajuste de resorte, por lo que la base le ayudará a la cámara permanecer estático y firme.

Por otra parte, se diseñó un soporte para la cámara, ya que esta deberá permanecer estática durante todo el mecanismo para evitar movimientos involuntarios o ruido visual.

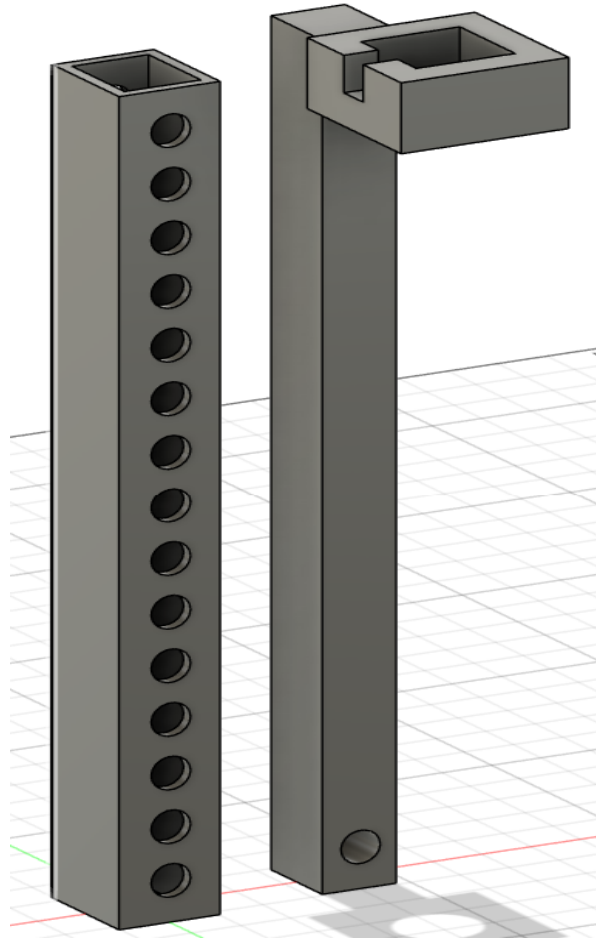


Figura 64: Base para cámara USB.

Para el diseño del soporte de la cámara de la figura 64, se tomó de referencia la estructura de una máquina de levantamiento de un gimnasio, en donde se puede ajustar la altura del lente de la cámara mediante un seguro. En la figura 65 se podrá observar el acoplamiento completo de la cámara con el prototipo.

De igual manera, se colocó una silla de tamaño a escala para referenciar el lugar en el que permanecerá el operador durante el tiempo del proceso.

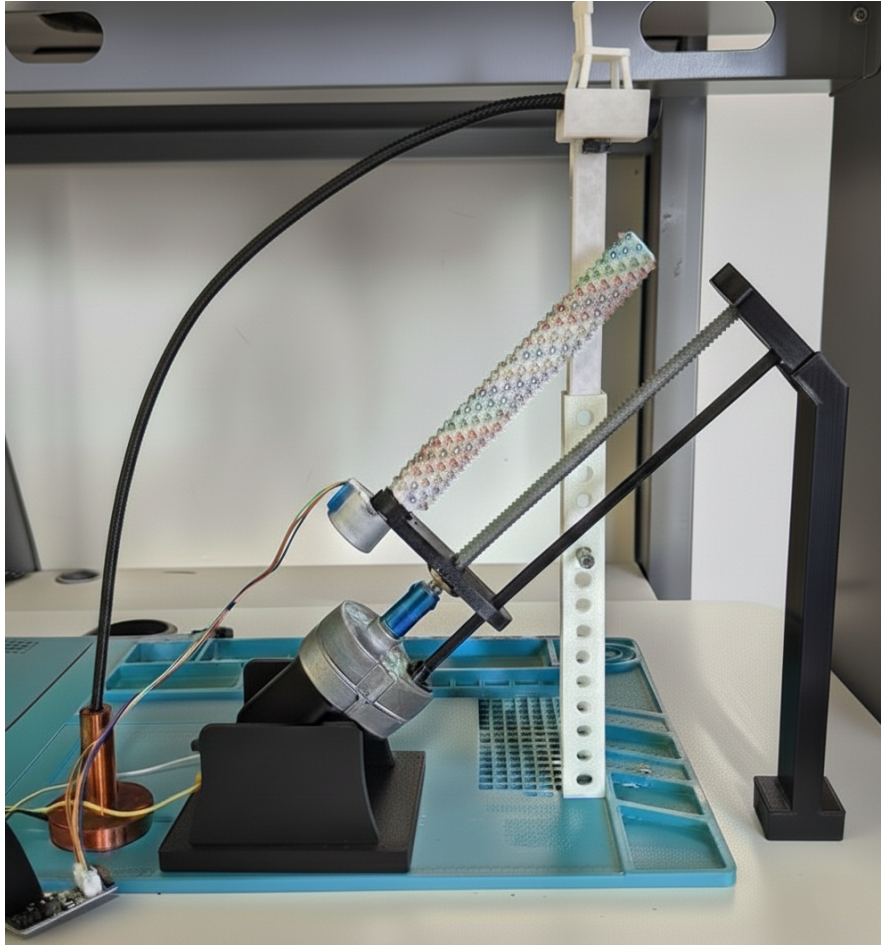


Figura 65: Prototipo con cámara USB.

Una vez acondicionada la cámara USB, se procedió con el desarrollo del algoritmo del sistema de corrección de posición de los insertos.

La metodología seguida para el desarrollo del algoritmo se muestra de la siguiente manera:

1. Creación de Frame.

En primera estancia, se hizo una prueba preliminar sobre el funcionamiento de la cámara creando un frame. Un frame o fotograma se refiere a una imagen concreta dentro de una sucesión de imágenes en movimiento [21], el frame permite el área que se tendrá para la posición inicial del prototipo, en la figura 66 se muestra un frame creado para ajustar la posición y altura de la cámara.

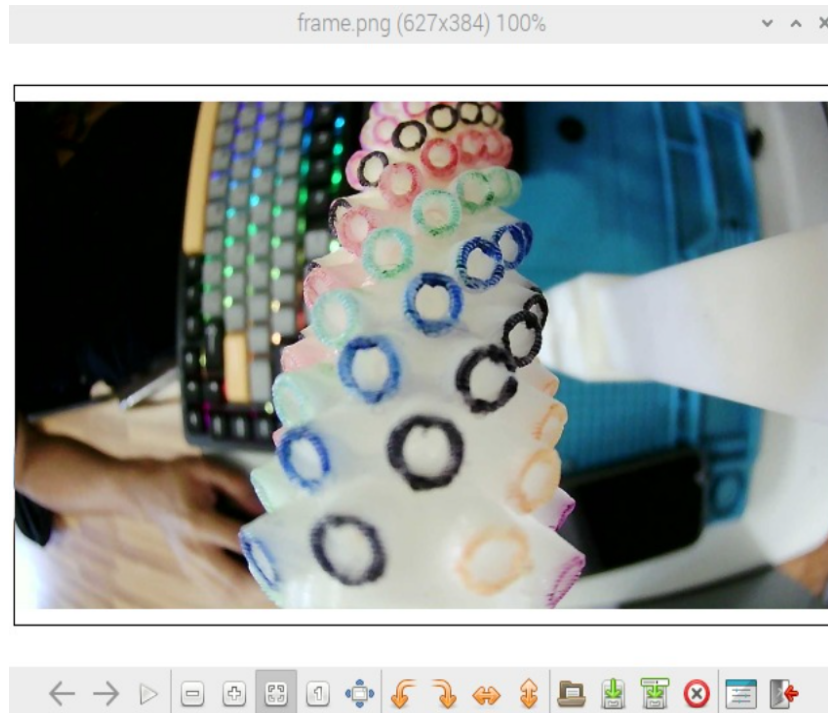


Figura 66: Frame mostrando torre de invernadero.

2. Programación de ROI dentro del frame.

Una región de interés (ROI), de acuerdo con [22], es una porción de una imagen que se desea filtrar o manipular. Se puede representar una ROI como una imagen de máscara binaria. En la imagen de máscara, los píxeles que pertenecen a la ROI se establecen en 1 y los píxeles fuera de ella se establecen en 0. Se programó una ROI, en donde se detectarán los insertos. Este proceso facilita el procesamiento, ya que al detectar varios insertos en un mismo frame, afectaría drásticamente el sistema y existiría mucha inestabilidad. Filtrando el frame en un ROI, se detectan los insertos únicamente en una sola región del frame, aislandolos de los demás y discriminando cualquier elemento fuera del ROI.

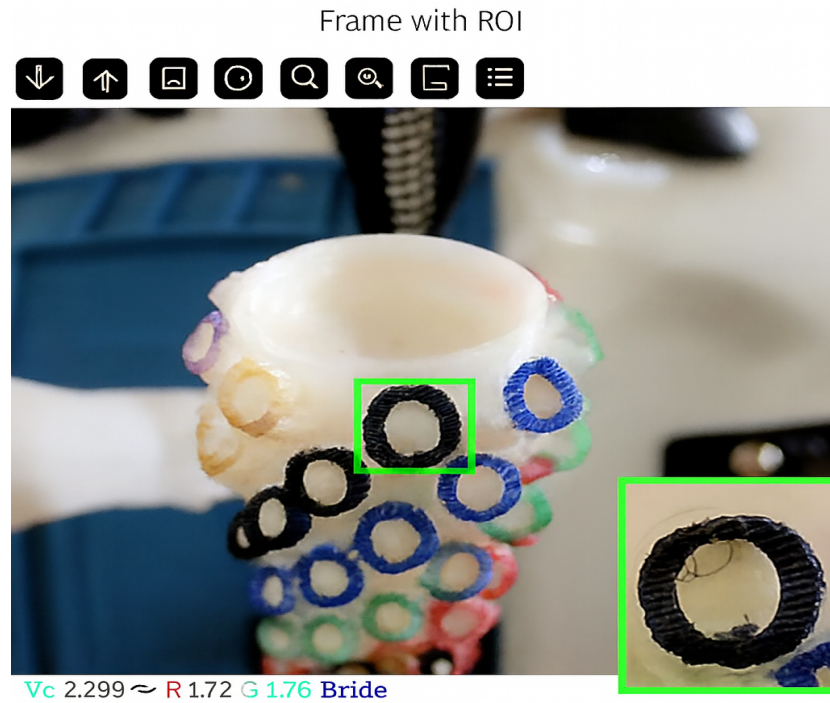


Figura 67: Frame con ROI programado.

Como se muestra en la figura 67, dentro del frame se programó un ROI en color verde. El tamaño y color de este puede modificarse en el código fuente del algoritmo. Además, se programó una ventana adicional para observar lo que proyecta la cámara dentro del ROI.

Como ya se mencionó, se programó un segundo frame (ROI en color verde), pero este no fue utilizado de manera práctica a lo largo del proyecto, su programación se realizó con fines visuales, para proyectar lo que detecta el ROI y así delimitar el tamaño del mismo para su practicidad.

3. Detección de insertos por colores

Para la detección de insertos por colores se utilizó la herramienta OpenCV, como se mencionó en los antecedentes. OpenCV es una biblioteca de visión artificial de código abierto, que presenta un enfoque fundamental para la detección de movimiento y colores (el presente caso), que abarca la adquisición de fotogramas, reducción de ruido, seguimiento de objetos, umbral de colores, etc.



Figura 68: Inserto detectado en ROI.

Como se muestra en la figura 68, se programó la detección de los insertos por colores, en este caso el ROI detecta los insertos en color negro.

4. Cálculo de error de desfaseamiento

Una vez programada la detección de los insertos, se procedió a calcular el error de desfaseamiento con respecto al ROI en unidades de píxeles (px), dependiendo del sentido en el que esté desfasado cada inserto, es el tipo de error y con ello el tipo de actuador que corregirá el inserto, en la figura 69 se muestra la lógica visual del sistema.

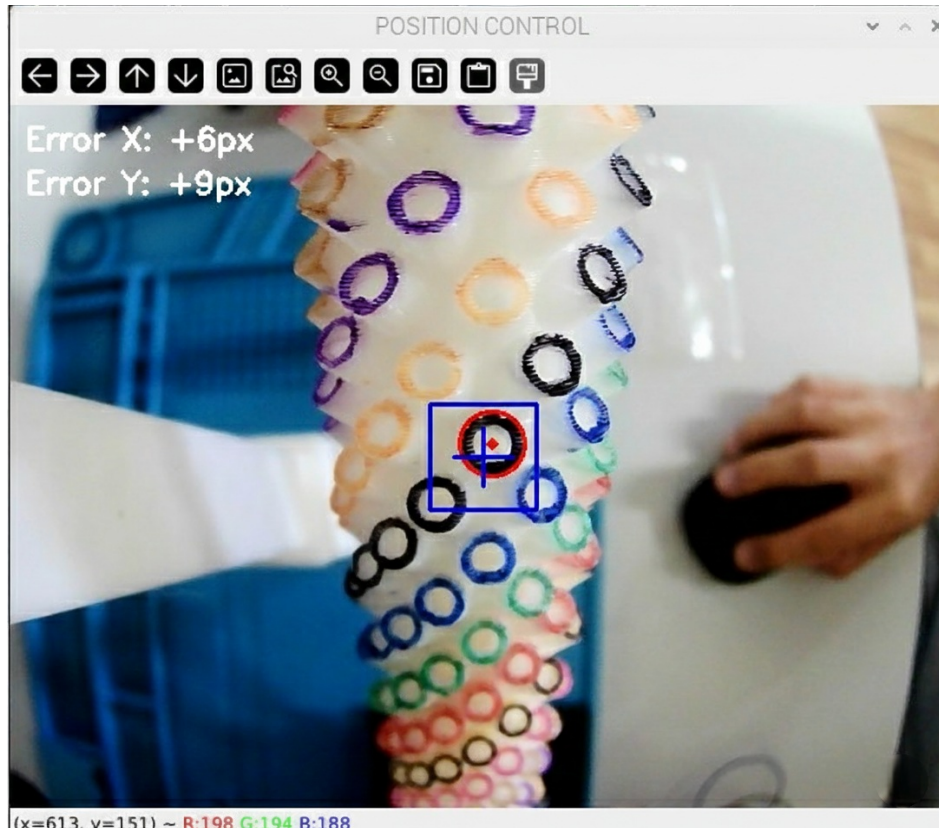


Figura 69: Cálculo de error de posicionamiento de ROI.

La lógica visual del programa consiste, tomando de referencia la figura 69, en el trazo de un plano cartesiano dentro del ROI, mostrando así el centro exacto del mismo. De acuerdo a los cuadrantes del plano es el sentido de desfaseamiento del inserto, si se encuentra a la izquierda en el eje X, el error es negativo y si se encuentra a la derecha, el error es positivo. Por otra parte, si el inserto se encuentra por arriba en el eje Y, el error es positivo, de lo contrario, el error es negativo.

De acuerdo con la figura anterior, el error de desfaseamiento del inserto detectado es de 6px en el eje X y 9 px en el eje Y, los cuales se proyectan en el frame.

5. Corrección de desfaseamiento

Una vez el error es calculado, como se mostró en la figura 69, el programa corrige, con el motor a pasos para el eje X y con el motor dc para el eje Y, colocando el inserto en el centro

del ROI. Para agilizar el proceso del sistema, se programó una tolerancia del 15 %, es decir, una tolerancia en la cual el inserto se le permite estar desfasado a cierto rango de error, si la sumatoria del desfaseamiento en ambos ejes (errores) se encuentra por debajo de la tolerancia, el sistema lo considera como centrado, además se le programó un tiempo de 5 segundos, que simula el tiempo que le tomaría a una persona colocar el sustrato en el inserto, como se muestra en la figura 70. Esto con el fin de “centrar” los insertos lo más rápido posible y así agilizar el proceso de colocación de sustrato.

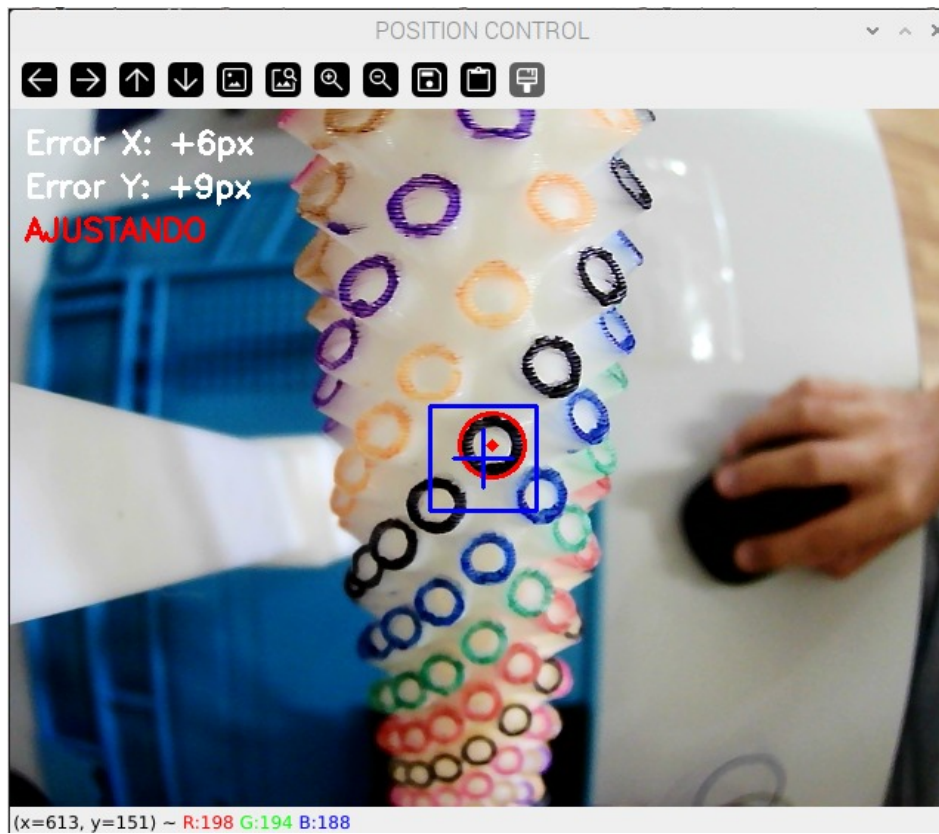


Figura 70: Ajuste de desfaseamiento de inserto negro.

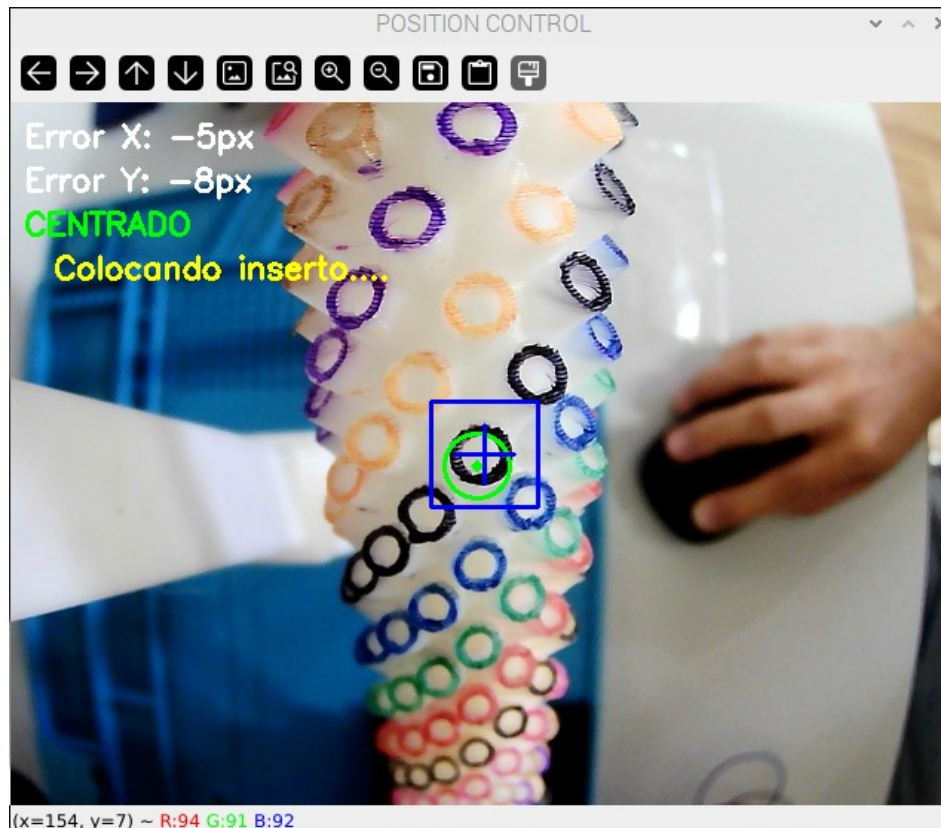


Figura 71: Inserto negro centrado.

Como se muestra en las figuras 70 y 71, de acuerdo con el tipo de error es el actuador que se acciona para corregir el error y centrar el inserto. En el código fuente se programó el patrón y orden de colores de los insertos a detectar, los cuales son: negro, azul, verde, rojo, negro, fucsia, café, morado y naranja.

Con respecto a la cantidad total de los insertos (270) cada espiral está conformada por 30 insertos. El algoritmo se encarga de corregir cada uno de los 30 insertos de un solo color hasta llegar al inserto no. 30, cuando este esté corregido, la torre de invernadero vuelve a su posición inicial y reinicia el proceso con el siguiente color junto con el conteo de insertos detectados y así sucesivamente con todos los colores. A continuación se mostrará el proceso de la corrección de los insertos de los colores restantes de la torre de invernadero.

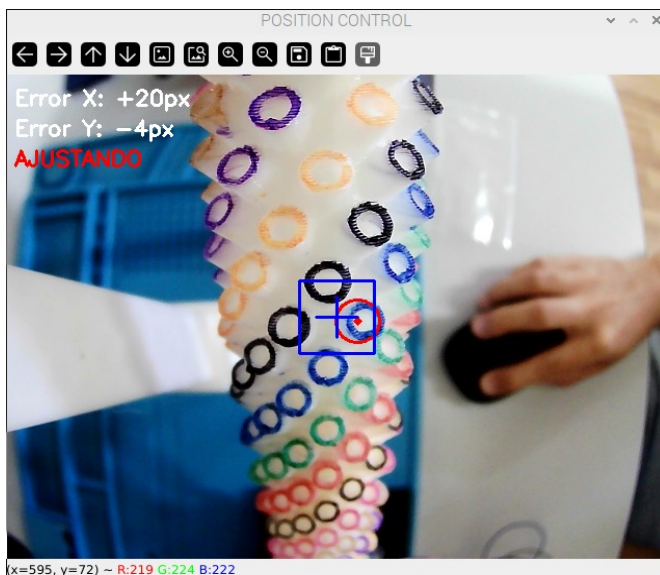


Figura 72: Ajuste de desfasamiento de inserto azul.

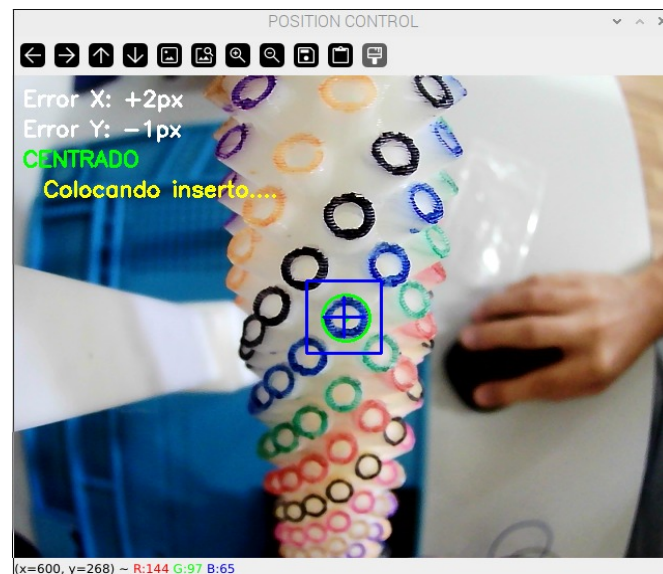


Figura 73: Inserto azul centrado.

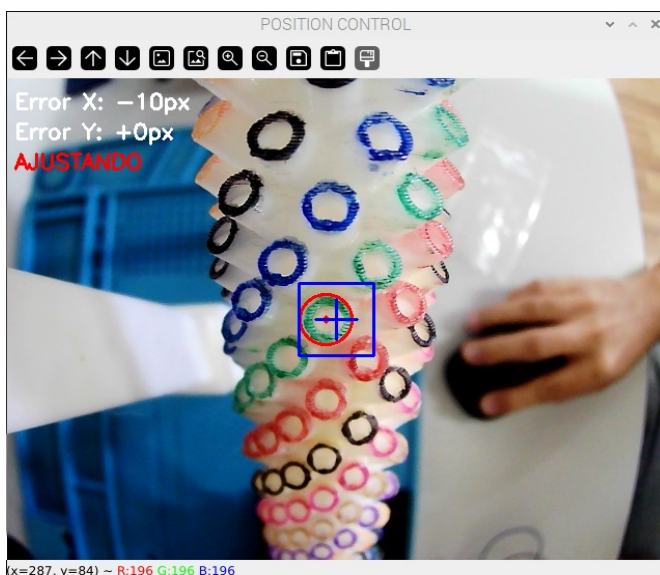


Figura 74: Ajuste de desfasamiento de inserto verde.

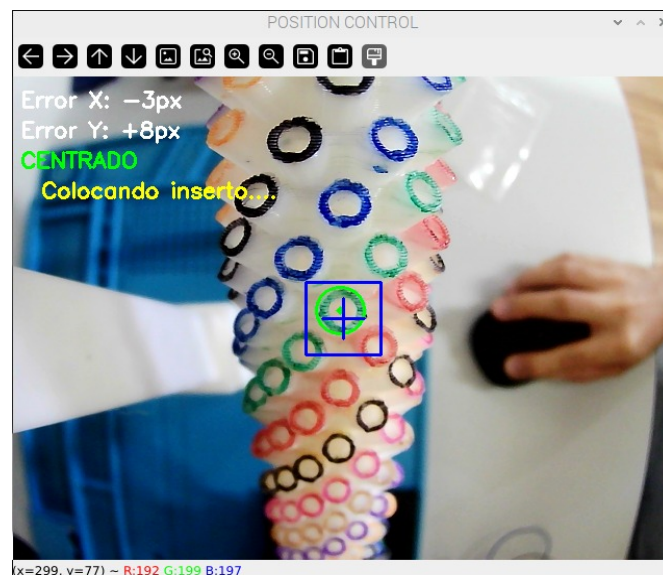


Figura 75: Inserto verde centrado.

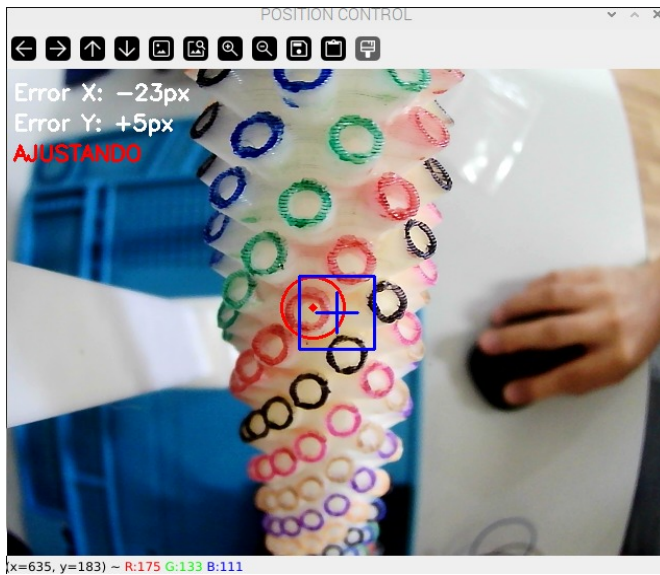


Figura 76: Ajuste de desfasamiento de inserto rojo.

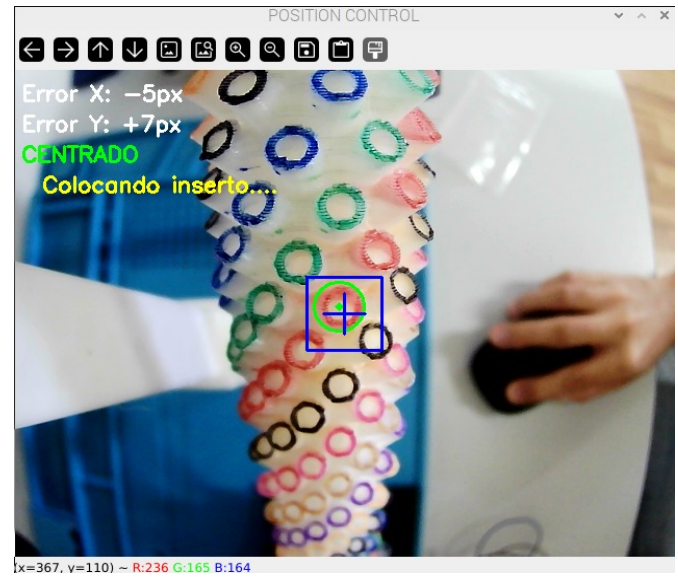


Figura 77: Inserto rojo centrado.

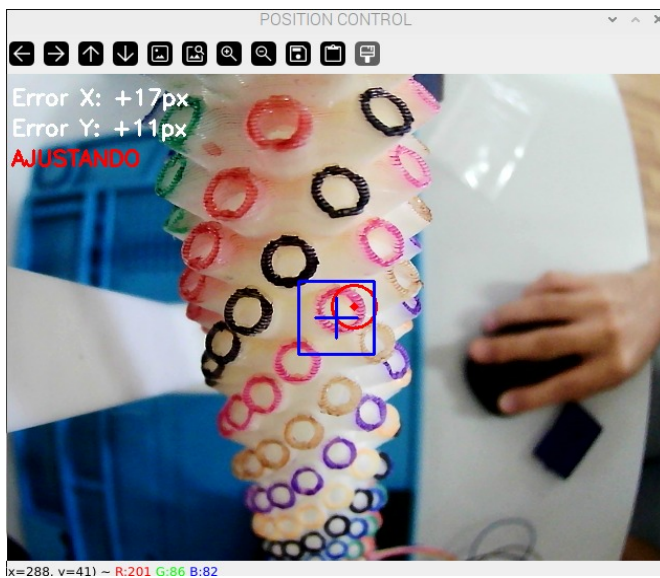


Figura 78: Ajuste de desfasamiento de inserto fucsia.

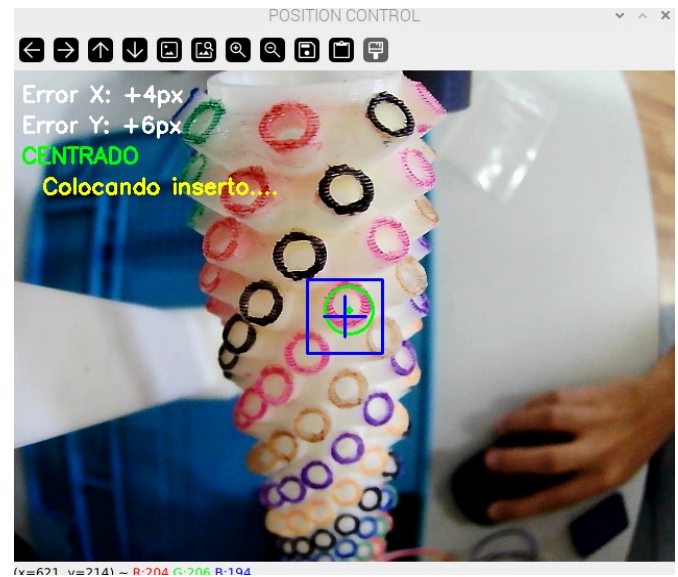


Figura 79: Inserto fucsia centrado.

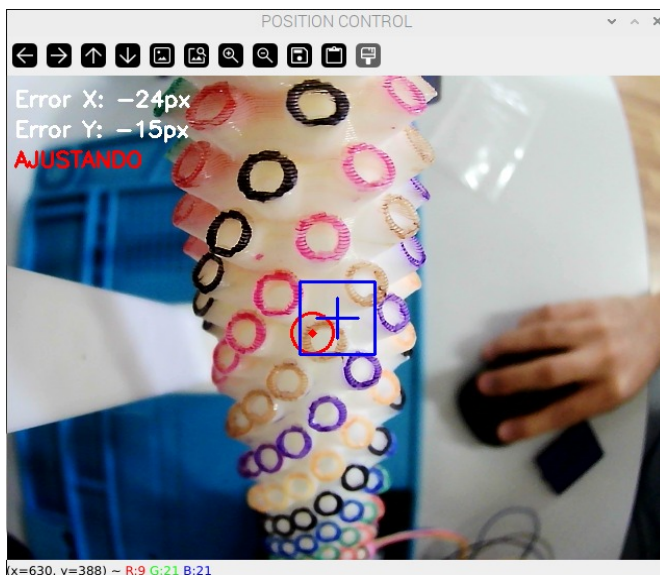


Figura 80: Ajuste de desfasamiento de inserto café.

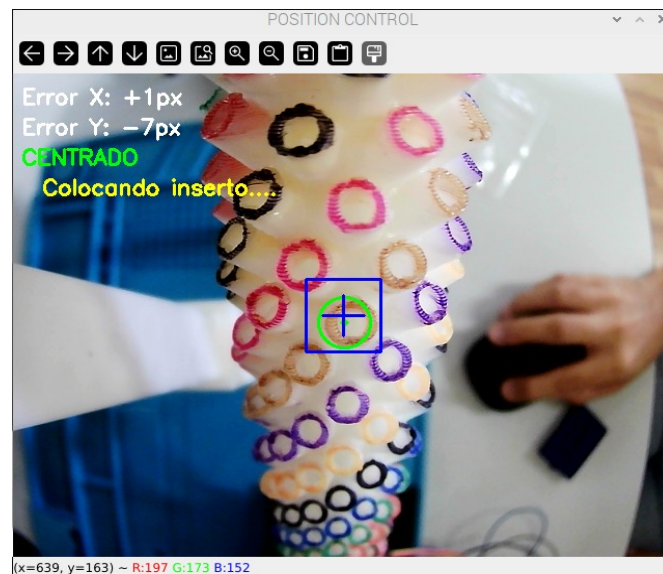


Figura 81: Inserto café centrado.

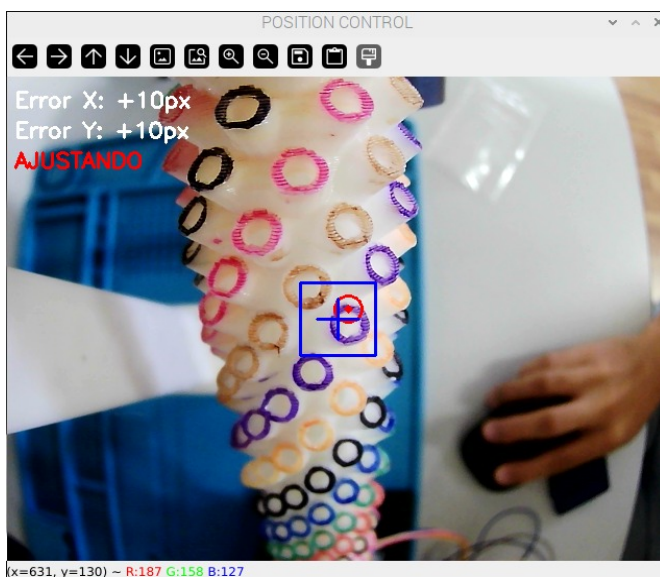


Figura 82: Ajuste de desfasamiento de inserto morado.



Figura 83: Inserto morado centrado.

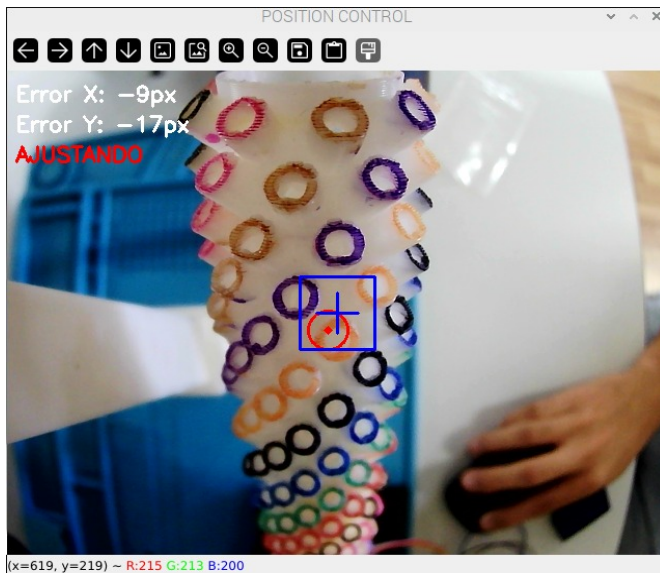


Figura 84: Ajuste de desfaseamiento de inserto naranja.

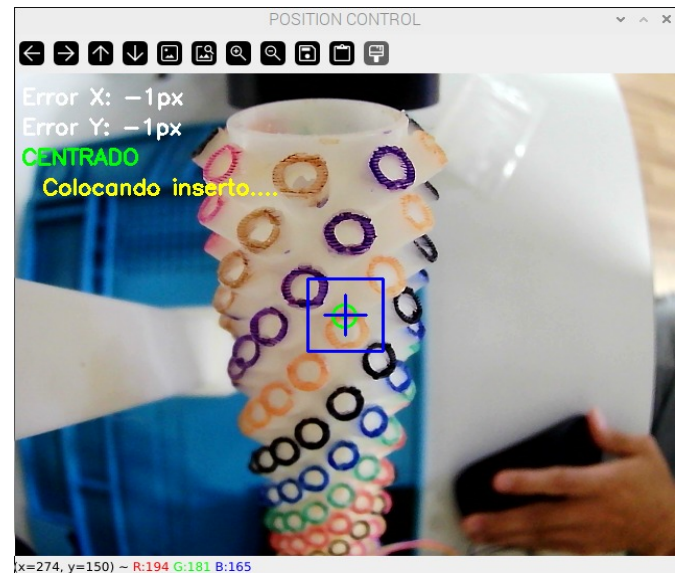


Figura 85: Inserto naranja centrado.

Una vez el algoritmo detecte la corrección de todos los insertos, el programa finaliza su ejecución.

A continuación, se presentará el algoritmo principal del sistema de corrección de posicionamiento de insertos de la torre de invernadero.

ALGORITMO PRINCIPAL DE SOFTWARE DE CONTROL

Algorithm 3: Sistema de procesamiento de imágenes para sistema de corrección de posicionamiento de insertos de torre de invernadero

Data: Secuencia de pasos del motor, rangos HSV de colores, tolerancias de centrado

Result: Detección del inserto y cálculo del error de posición

Inicialización:

Configurar pines GPIO para motores y botón;

Configurar cámara y cargar lista de colores HSV: (Negro, azul, verde, rojo, negro, fucsia, cafe, morado, naranja);

Establecer tolerancias de centrado y parámetros de movimiento;

while *el sistema está en ejecución* **do**

 Leer estado del botón de activación;

if *el sistema está desactivado* **then**

 Detener motores y mostrar pantalla vacía;

continuar;

end

 Capturar frame y rotarlo;

 Definir ROI cuadrado centrado;

 Convertir ROI a HSV y aplicar máscara del color actual;

 Obtener contorno más grande detectado;

if *se detecta un contorno válido* **then**

 Calcular centro del objeto;

 Calcular error en X y en Y;

 Determinar si el objeto está dentro de la tolerancia de centrado;

end

else

 Mostrar estado SIN DETECCION;

end

 Dibujar ROI, referencias y estado en pantalla;

end

if *el inserto está centrado* then

Detener motores;

Incrementar frames centrado;

if *frames centrado* $\geq$ *umbral* then

Confirmar inserto centrado;

Iniciar conteo de permanencia;

end

if *tiempo de inserción cumplido* then

Avanzar mecanismo para depositar sustrato;

Mover motor a pasos a siguiente inserto;

Reiniciar contadores;

if *se completan 30 insertos* then

Ejecutar retroceso largo en motor DC;

Mover eje X para alinear siguiente fila;

Cambiar al siguiente color HSV;

end

end

end

else

Reiniciar contadores de centrado;

Calcular pasos dinámicos según el error;

if *error en X* > *tolerancia* then

Mover motor a pasos hacia la izquierda o derecha;

end

else if *error en Y* > *tolerancia* then

Mover motor DC en avance o retroceso;

end

end

Actualizar visualización y mostrar información del estado;

Adicional a lo que se ha mencionado anteriormente, se añadió al prototipo un circuito de interrupción de programa, encargado de interrumpir/pausar el sistema en el momento que se requiera, una vez que se oprima de nuevo, el sistema reanudará su proceso en justo donde se interrumpió.

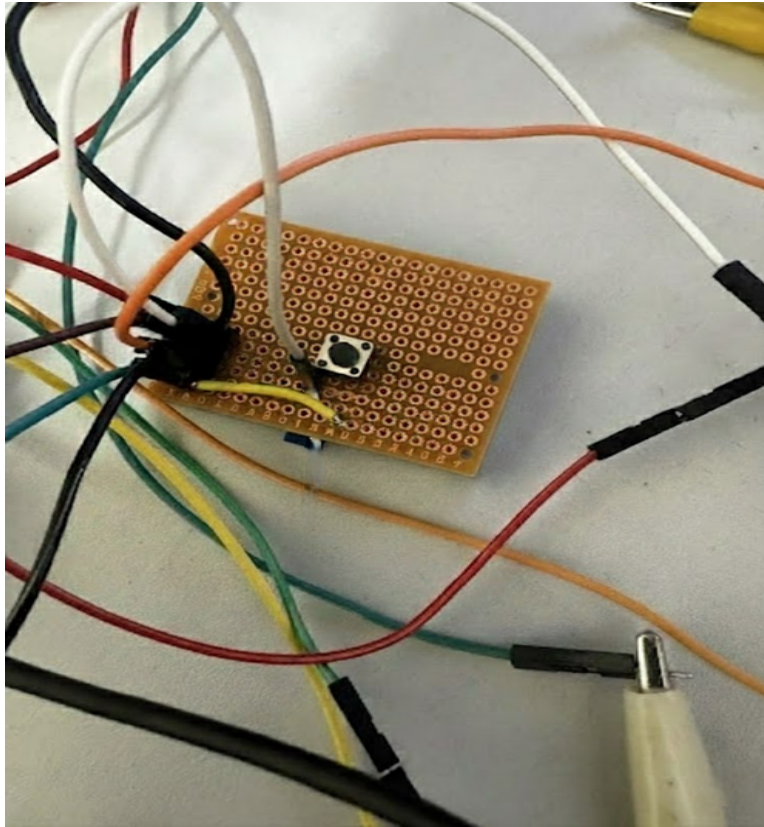


Figura 86: Placa fenólica con circuito de interrupción con push button.

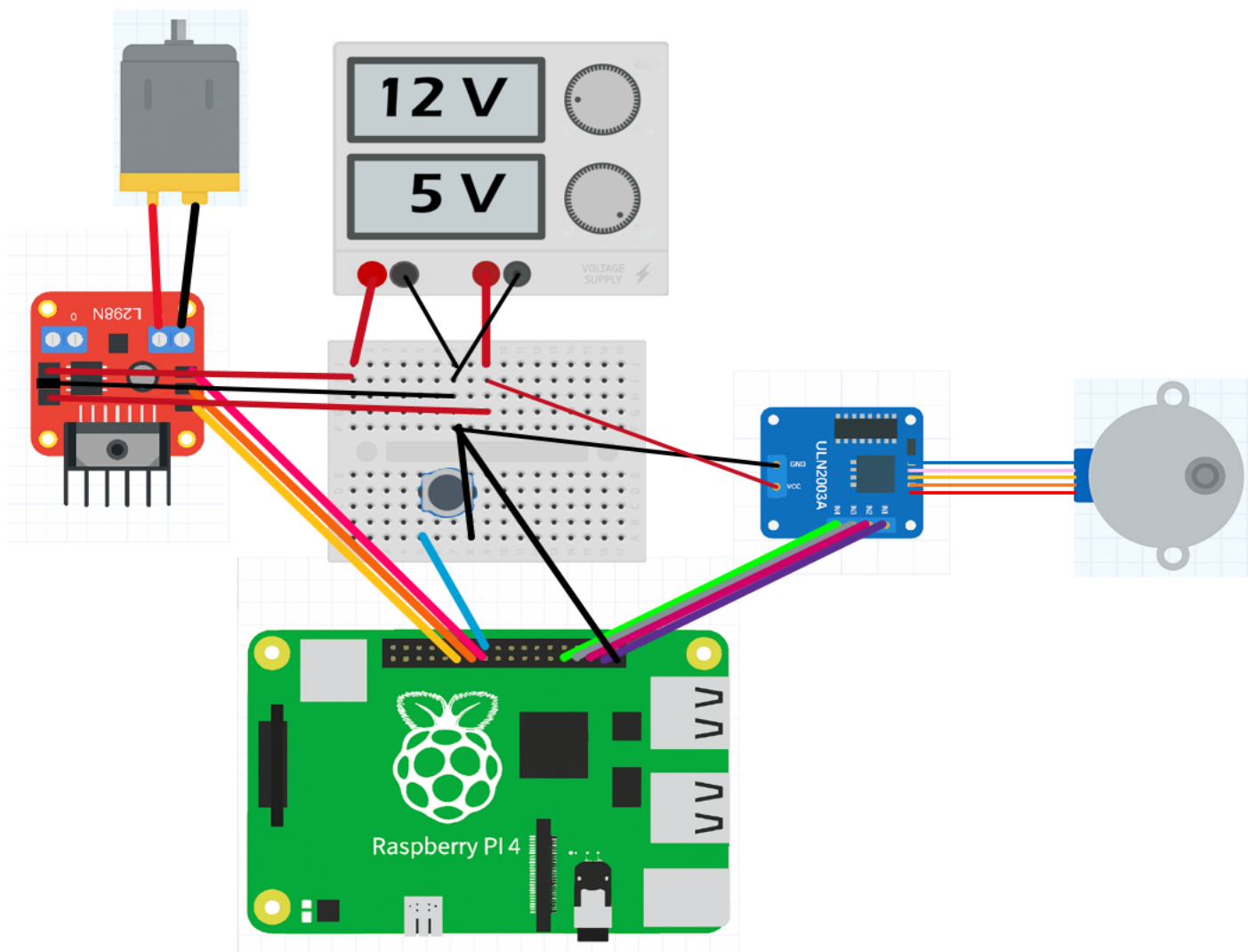


Figura 87: Circuito electrónico de sistema completo.

CAPÍTULO 5: RESULTADOS

En este capítulo se mostrarán los resultados obtenidos de la experimentación del proyecto.

Se realizaron 15 pruebas experimentales bajo condiciones similares de iluminación y ambientales para asegurar su eficiencia. En cada prueba se midió el tiempo y la precisión del posicionamiento de los insertos mediante el sistema de control basado en procesamiento de imágenes. Además se realizó una evaluación comparativa con el método de colocación manual, donde el operador realiza todo el proceso de pie, el cual se hicieron 2 pruebas solamente, debido al tiempo tardío que se toma en realizar la operación, como se muestra en la siguiente tabla.

Prueba	Tiempo de operación con sistema de corrección	Tiempo de operación manual
1	41:35 minutos.	1:28:24 horas.
2	44:48 minutos.	1:38:40 horas.
3	43:23 minutos.	-
4	45:12 minutos.	-
5	45:50 minutos.	-
6	43:32 minutos.	-
7	42:55 minutos.	-
8	44:08 minutos.	-
9	43:42 minutos.	-
10	41:47 minutos.	-
11	42:22 minutos.	-
12	44:39 minutos.	-
13	45:07 minutos.	-
14	42:46 minutos.	-
15	44:16 minutos.	-

Tabla 4: Comparación de tiempo de operación con sistema de detección con tiempo de operación manual.

La comparación del sistema de posicionamiento con el método manual mostró que el sistema automatizado obtuvo un tiempo de operación promedio de 44 minutos con 16 segundos, en comparación al tiempo promedio de operación manual de 1 hora con 33 minutos y 32 segundos, lo que significa que el sistema de posicionamiento redujo el tiempo total de operación en un 52 % aproximadamente. El rendimiento del sistema de posicionamiento se mantuvo constante entre las pruebas, con solo pequeñas variaciones atribuidas a los cambios de iluminación. Si bien no se realizó un análisis ergonómico cuantitativo, la evaluación observacional indicó una clara reducción de los movimientos repetitivos y el esfuerzo físico en comparación con la operación manual.

Aunque las mejoras ergonómicas no se midieron cuantitativamente y/o cualitativamente, la reducción del tiempo del sistema y la detección automática de los insertos podrían demostrar una mejora ergonómica, dado que, según [23], la frecuencia y la duración de las tareas repetitivas permiten encontrar mejoras ergonómicas en procesos automáticos, y [24] destacan cómo la reducción de posturas incómodas y movimientos repetitivos es un indicador de mejora ergonómica en los procesos productivos.

Para evaluar la precisión de posicionamiento del sistema, se recopilaron las mediciones de 30 pruebas para cuantificar la desviación entre la posición objetivo y la posición final alcanzada por el mecanismo.

En la Ecuación 1 podemos observar la ecuación para el cálculo del error posicional en pixeles (px).

Ecuación 1: Ecuación de cálculo de error de posicionamiento (px)

$$E_p = \sqrt{(X_{esp} - X_{obt})^2 + (Y_{esp} - Y_{obt})^2}$$

En la [Ecuación 2](#) se observa la ecuación para el cálculo del error posicional en milímetros (mm), considerando que 1 px = 0.25 mm.

Ecuación 2: Ecuación de cálculo de error de posicionamiento real (mm)

$$E_r = E_p * 0.25mm$$

Tabla 5: Resultados de detección y error en 30 pruebas

Prueba	X detectado (px)	Y detectado (px)	Error px	Error (mm) (escala 0.25mm)
1	2	-3	3.606	0.902
2	-1	4	4.123	1.031
3	5	2	5.385	1.346
4	-4	-3	5.000	1.250
5	3	1	3.162	0.791
6	-2	6	6.325	1.581
7	4	-5	6.403	1.601
8	1	3	3.162	0.791
9	-3	2	3.606	0.902
10	6	-4	7.211	1.803
11	-5	-1	5.099	1.275
12	2	5	5.385	1.346
13	-1	-4	4.123	1.031
14	3	-6	6.708	1.677
15	-2	3	3.606	0.902
16	4	2	4.472	1.118

Prueba	X detectado (px)	Y detectado (px)	Error px	Error (mm) (escala 0.25mm)
17	-6	1	6.083	1.521
18	5	-3	5.831	1.458
19	-4	4	5.657	1.414
20	1	-2	2.236	0.559
21	3	4	5.000	1.250
22	-5	2	5.385	1.346
23	2	-6	6.325	1.581
24	-3	-5	5.831	1.458
25	6	3	6.708	1.677
26	-1	1	1.414	0.354
27	4	-2	4.472	1.118
28	-2	-3	3.606	0.902
29	5	1	5.099	1.275
30	-4	6	7.211	1.803
Promedio:				1.205 mm

Una vez obtenidos estos valores y el promedio de error posicional de nuestro sistema, se procedió con el cálculo de intervalo de confianza para validar la eficiencia del proyecto.

A continuación se muestran las ecuaciones utilizadas:

Ecuación 3: Ecuación de cálculo de varianza (mm)

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Ecuación 4: Ecuación de cálculo de Desviación estándar (mm)

$$s = \sqrt{s^2}$$

Ecuación 5: Ecuación de cálculo de error estándar (mm)

$$SE = \frac{s}{\sqrt{n}}$$

Ecuación 6: Ecuación de cálculo de intervalo de confianza del 95 % (mm)

$$IC_{95\%} = \bar{x} \pm t_{\alpha/2, n-1} \cdot SE$$

Ecuación 7: Ecuación de cálculo de eficiencia basado en error

$$\text{Eficiencia}_{\text{error}} = 1 - \frac{\bar{x}_{\text{error}}}{T}$$

Ecuación 8: Ecuación de cálculo de eficiencia basado en tiempo

$$\text{Eficiencia}_{\text{tiempo}} = 1 - \frac{T_{\text{automático}}}{T_{\text{manual}}}$$

Ecuación 9: Ecuación de cálculo de eficiencia global

$$\text{Eficiencia}_{\text{global}} = \frac{1}{2} \text{Eficiencia}_{\text{error}} + \frac{1}{2} \text{Eficiencia}_{\text{tiempo}}$$

Los resultados mostraron un error promedio de 1.2 mm con un intervalo de confianza del 95 % entre 1.06 y 1.35 mm. Esto indica que el valor de eficiencia con respecto al error es de 75,9 % hacia una tolerancia máxima de 5 mm de error posicional, mientras que el valor de eficiencia con respecto al tiempo de operación es de 51,9 %. Estos hallazgos demuestran que el prototipo funciona de forma consistente y que el módulo de procesamiento de imágenes proporciona una retroalimentación posicional fiable al integrarse con el sistema de control.

CONCLUSIONES

Los resultados obtenidos confirman la viabilidad de automatizar la corrección posicional de insertos en una torre de invernadero mediante el desarrollo de un prototipo de posicionador vertical equipado con un sistema de procesamiento de imágenes. Esta integración permitió un control más preciso en el posicionamiento del mecanismo, lo que se tradujo en una colocación más dinámica y eficiente del sustrato con semilla, reduciendo la necesidad de intervención manual y mejorando la ergonomía del operador.

El uso combinado de componentes impresos en 3D y materiales metálicos resultó adecuado para lograr un sistema ligero y adaptable, aunque se identificaron ciertas restricciones funcionales que abren la posibilidad de mejoras futuras. A pesar de estas limitaciones, el prototipo constituye una base sólida para la optimización y aplicación en entornos reales de invernadero, donde podrían implementarse estructuras de mayor rigidez para incrementar la estabilidad general del dispositivo.

En relación con los objetivos específicos, se logró:

- Automatizar el movimiento de avance y retroceso de la torre mediante un tornillo sin fin, garantizando precisión en la operación.
- Implementar el giro con inclinación controlada de la torre, lo que favoreció la colocación ergonómica del sustrato de caña.
- Acelerar el proceso de obtención de plántula, contribuyendo a una germinación más ágil y eficiente.

Finalmente, se reconoce que el sistema puede evolucionar hacia versiones más robustas mediante la incorporación de algoritmos avanzados y técnicas de Inteligencia Artificial, lo que potenciaría su desempeño y ampliaría su aplicabilidad en la automatización de procesos agrícolas.

RECOMENDACIONES

A pesar de las limitaciones identificadas, el sistema constituye una base sólida y prometedora para su optimización e implementación futura. Las líneas de desarrollo posteriores se orientan hacia:

- **Escalabilidad:** El diseño del prototipo en una escala funcional para procesos operativos reales y funcionales.
- **Mejora Estructural:** El diseño y la implementación de una estructura con mayor rigidez, posiblemente mediante el uso de perfiles de aluminio o compuestos, para incrementar la estabilidad general y la precisión durante su operación.
- **Avance en el Control Inteligente:** La implementación de algoritmos de control más robustos y la incorporación de técnicas de Inteligencia Artificial (IA) y Machine Learning. Esto permitiría un análisis predictivo, la identificación automática de plántulas no viables, y la optimización en tiempo real de los parámetros de siembra basados en datos históricos y ambientales.
- **Adaptabilidad:** El prototipo sienta las bases para el diseño de sistemas escalables que puedan adaptarse a diferentes configuraciones de invernadero y variedades de cultivo, aumentando así su versatilidad y potencial de comercialización.
- **Integración con IoT:** Una futura integración con tecnologías del Internet de las Cosas (IoT) facilitaría el monitoreo remoto, la recolección continua de datos y la gestión centralizada de múltiples torres de cultivo, allanando el camino hacia invernaderos completamente autónomos.
- **Automatización:** Implementar un proceso automático para colocación automática de sustrato en insertos de torres de invernadero para reducir la intervención humana.

En conclusión, este trabajo no solo demuestra la funcionalidad del prototipo, sino que también establece un camino claro para su evolución, posicionándolo como un paso significativo hacia la agricultura de precisión y la automatización total en entornos de cultivo controlado.

Bibliografía

- [1] ATO.com. (2026) Servo motor vs. stepper motor in performance & control. Accedido: 05-Feb-2026. [Online]. Available: <https://www.ato.com/difference-between-servo-motor-and-stepper-motor>
- [2] J. L. Martín Pérez, “Comparación de técnicas de detección de objetos en imágenes,” 2023, trabajo de Fin de Máster en Ingeniería Informática, Facultad de Informática UCM, Departamento de Ingeniería del Software e Inteligencia Artificial. [Online]. Available: <https://hdl.handle.net/20.500.14352/88540>
- [3] M. G. G. Mendoza, “Impacto geoespacial de especies invasoras en la agricultura mexicana: la próxima amenaza a la seguridad alimentaria,” *Revista Inclusiones*, vol. 9, no. Especial, pp. 127–151, Jan. 2022.
- [4] M. F. de la Vega, “Economy of the mexican agriculture sector in times of covid-19,” *Revista Mexicana de Fitopatología*, vol. 39, no. 4, pp. 425–432, 2021. [Online]. Available: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-33092021000400011
- [5] A. Kluczkowski, P. Hadley, C. Yap, U. Ehgartner, B. Doherty, and K. Denby, “Urban vertical farming: innovation for food security and social impact?” *Philosophical Transactions of the Royal Society B*, vol. 380, no. 1935, p. 20240154, 2025. [Online]. Available: <https://royalsocietypublishing.org/rstb/article/380/1935/20240154/235082/Urban-vertical-farming-innovation-for-food>
- [6] J. Matthews, “The role of ergonomics in the efficient utilisation of agricultural labour,” Agricultural Adjustment Unit, University of Newcastle upon Tyne, Tech. Rep. Technical Paper 11, 1970, accessed: Nov. 21, 2025. [Online]. Available: <https://ageconsearch.umn.edu/record/272888/files/newcastle030.pdf>
- [7] G. N. Tiwari, *Greenhouse Technology for Controlled Environment*. Alpha Science International Ltd., 2003. [Online]. Available: <https://books.google.com.mx/books?id=K5LVrc6k4WIC>

- [8] D. Despommier, “The vertical farm: Food production in urban environments,” *Gastronomica: The Journal of Food and Culture*, vol. 7, no. 3, pp. 80–87, 2007. [Online]. Available: <https://www.jstor.org/stable/10.1525/gfc.2007.7.3.80>
- [9] G. A. Kranzler and M. P. Rigney, “Machine vision grading for tree seedlings,” in *Proceedings of the 11th International Congress on Agricultural Engineering*, Sep. 1989, pp. 1883–1888.
- [10] P. Joshi, D. M. Escriva, and V. Godoy, *OpenCV By Example*. Packt Publishing Ltd, 2015. [Online]. Available: <https://books.google.com.mx/books?id=WwYcDAAAQBAJ>
- [11] D. T. Joy, G. Kaur, A. Chugh, and S. B. Bajaj, “Computer vision to color detection,” *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, vol. 9, no. 3, pp. 53–59, May 2021, disponible también en SSRN: <https://ssrn.com/abstract=3879981>. [Online]. Available: <https://www.ijircst.org/viewfulltext.php?article=IRP1153>
- [12] J. David, M. Vélez, L. Felipe, and C. Segura, “Diseño de un sistema automático de cultivo hidropónico para forraje verde forraje verde,” 2018. [Online]. Available: https://ciencia.lasalle.edu.co/ing_automatizacionF.
- [13] D. R. Quintero, “Desarrollo de un sistema de cultivo hidropónico vertical automatizado y remoto,” 2023. [Online]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/60478/TFG-I-2564.pdf?sequence=1>
- [14] A. Martínez-Sibaja, “Diseño de un invernadero vertical e integración de herramientas para su instrumentación electrónica,” vol. 10, 3 2018. [Online]. Available: <https://revista-agroproductividad.org/index.php/agroproductividad/article/view/87>
- [15] A. F. E. Iván, J. S. Daniel, and V. C. D. Tonatiuh, “Diseño y elaboración de prototipo de una máquina sembradora semiautomática para charolas de invernadero,” Instituto Politécnico Nacional, Tech. Rep., 5 2013. [Online]. Available: <https://tesis.ipn.mx/bitstream/handle/123456789/12174/dise%C3%B1oyelaboracion.pdf?sequence=1&isAllowed=y>

- [16] J. J. M. García, “Diseño de un semillero automatizado,” 2020, grado en Electrónica Industrial y Automática. Tutor: Nicolás Laguarda Miró.
- [17] Álvaro Domingo Gordo Ruiz, “Desarrollo e implementación de un invernadero automatizado con cultivo hidropónico y aplicación móvil para el seguimiento de datos,” Universidad de Sevilla, Tech. Rep., 2017. [Online]. Available: <https://idus.us.es/server/api/core/bitstreams/bc47e38e-417f-471>
- [18] A. varios, “Técnicas de procesamiento digital de imágenes para detección de plagas y enfermedades en cultivos: una revisión,” *Ingeniería y Competitividad*, 2022. [Online]. Available: https://revistaingenieria.univalle.edu.co/index.php/ingenieria_y_competitividad/article/view/10973
- [19] C. D. Blair, “Apparatus for high density, automated cultivation,” U.K. Patent 1816403.8, Apr. 16 2020.
- [20] E. B. R. Rodríguez, “Sistema de procesamiento de imágenes para análisis multiespectral e hiperespectral en agricultura de precisión,” Patent WO2 017 105 177A1, Jun., 2017, patente internacional, prioridad 2015.
- [21] H. Frank, *Frame by Frame: A Materialist Aesthetics of Animated Cartoons*. Oakland: University of California Press, 2019. [Online]. Available: <https://library.oapen.org/handle/20.500.12657/25004>
- [22] C. M. Yeum, J. Choi, and S. J. Dyke, “Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure,” *Structural Health Monitoring*, vol. 17, no. 6, pp. 1363–1382, 2018. [Online]. Available: <https://journals.sagepub.com/doi/10.1177/1475921718765419>
- [23] CENEA, “¿cómo evaluar riesgos por movimientos repetitivos en un puesto de trabajo?” 2023, consultado en noviembre de 2025. [Online]. Available: <https://www.cenea.eu/como-evaluar-riesgos-por-movimientos-repetitivos-en-un-puesto-de-trabajo>
- [24] J. Sánchez Ortega, H. P. Chérrez Sandoval, M. Barriga Gavilanes, and G. K. Urquizo Flores, “Rediseño de puestos de trabajo orientado a criterios ergonómicos

generales: Una revisión bibliográfica,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, no. 5, pp. 160–177, 2024. [Online]. Available: https://doi.org/10.37811/cl_rcm.v8i5.13287

PRODUCTOS ACADÉMICOS

Como producto académico, adicional al presente trabajo, se publicó un artículo científico en la revista Agro Productividad enfocada a la agricultura y agroindustria sobre un enfoque centrado hacia el programa de sistema de corrección de posición de insertos.

AGRO PRODUCTIVIDAD



Image processing and recognition for controlling a greenhouse tower mechanism focused on ergonomics and acceleration in substrate placement for sugarcane germination

García-Peñuela, B. F.¹; Águila-Rodríguez, G.¹; Cuellar-Cortés, E. A.; Herrera-Aguilar, I.^{1*}

¹ Tecnológico Nacional de México - Instituto Tecnológico de Orizaba, Av. Oriente 9 No 852, Col. Emiliano Zapata, Orizaba, Veracruz, México. C.P. 94320.

* Correspondence: ignacio.ha@orizaba.tecnm.mx

ABSTRACT

Objective: To make the process of placing substrate in greenhouse towers for sugarcane germination faster and easier, through an automated prototype that uses image processing to reduce work time and physical effort.

Design/methodology/approach: A small-scale prototype was built to test how it works and to see if it could be used in real conditions. The system used a camera to recognize and locate points inside the work area, allowing smoother and more accurate movement of the mechanism.

Results: The tests showed that the system detected positions correctly and responded well to control commands, even with small changes in the environment. It also helped reduce working time and the operator's physical effort, improving both efficiency and comfort.

Limitations/implications: During testing, changes in lighting and small vibrations affected the accuracy of the system at times. These issues were mostly caused by the materials used in the prototype and the light conditions around it. Still, the results show that the system can be improved to work with greater precision and stability.

Findings/conclusions: The prototype proved to be a good option to make substrate placement in greenhouse towers faster and more efficient, while reducing physical strain for the operator.

Keywords: Image processing; ergonomics; automation; sugarcane; vertical greenhouses.

Citation: García-Peñuela, B. F., Águila-Rodríguez, G., Cuellar-Cortés, E. A., & Herrera-Aguilar, I. (2025). Image processing and recognition for controlling a greenhouse tower mechanism focused on ergonomics and acceleration in substrate placement for sugarcane germination. *Agro Productividad*. <https://doi.org/10.32854/7hr52k71>

Academic Editor: Jorge Cadena Iñiguez
Associate Editor: Dra. Lucero del Mar Ruiz Posadas
Guest Editor: Daniel Alejandro Cadena Zamudio

Received: July 23, 2025.

Accepted: November 1, 2025.

Published on-line: December 11, 2025.

Agro Productividad, 18(11). November. 2025. pp: 157-170.

This work is licensed under a Creative Commons Attribution-Non-Commercial 4.0 International license.



INTRODUCTION

The Sugarcane (*Saccharum officinarum*) is one of the main crops grown in tropical and subtropical regions across Latin America, especially in Mexico, which ranks as the sixth-largest producer worldwide according to Senties *et al.*, (2018). Besides being an essential source of sugar, sugarcane is also used to produce biofuels like ethanol, animal feed, paper and biodegradable plastics. It plays an important socioeconomic role in several rural areas and stands out for its high biomass production and ability to capture CO₂.



ANEXOS

En este apartado se anexaron los códigos implementados en este proyecto.

```
1 #-----Prueba de control de Motor DC-----
2 # Se declaran las variables
3 import RPi.GPIO as GPIO
4 import time
5
6 # Pines de L298N con raspberry
7 IN1 = 17
8 IN2 = 27
9 ENA = 22 # PWM
10
11 GPIO.setmode(GPIO.BCM)
12 GPIO.setup(IN1, GPIO.OUT)
13 GPIO.setup(IN2, GPIO.OUT)
14 GPIO.setup(ENA, GPIO.OUT)
15
16 pwm = GPIO.PWM(ENA, 1000) # 1kHz
17 pwm.start(0)
18
19 # Funcion de avance
20 def forward(speed=100):
21     GPIO.output(IN1, GPIO.HIGH)
22     GPIO.output(IN2, GPIO.LOW)
23     pwm.ChangeDutyCycle(speed)
24
25 # Funcion de retroceso
26 def backward(speed=100):
27     GPIO.output(IN1, GPIO.LOW)
28     GPIO.output(IN2, GPIO.HIGH)
29     pwm.ChangeDutyCycle(speed)
```

```

30
31 #Funcion de alto
32 def stop():
33     GPIO.output(IN1, GPIO.LOW)
34     GPIO.output(IN2, GPIO.LOW)
35     pwm.ChangeDutyCycle(0)
36
37 try:
38     print("Motor_adelante...")
39     forward(100)
40     time.sleep(10)
41
42     print("Motor_atras...")
43     backward(100)
44     time.sleep(10)
45
46     print("Motor_detenido")
47     stop()
48
49 finally:
50     pwm.stop()
51     GPIO.cleanup()

```

Código 1: Código de control de motores DC con L298N en python

```

1 # PRUEBA PARA MOTOR A PASOS CON RASPBERRY PI
2 # Se declaran las variables
3 import RPi.GPIO as GPIO
4 import time
5
6 # Pines ULN2003 (ajusta a tu conexion)
7 IN1 = 5
8 IN2 = 6

```

```

9  IN3 = 13
10 IN4 = 19
11
12 pins = [IN1, IN2, IN3, IN4]
13
14 GPIO.setmode(GPIO.BCM)
15 for pin in pins:
16     GPIO.setup(pin, GPIO.OUT)
17     GPIO.output(pin, 0)
18
19 # Secuencia de pasos para 28BYJ-48
20 seq = [
21     [1,0,0,1],
22     [1,0,0,0],
23     [1,1,0,0],
24     [0,1,0,0],
25     [0,1,1,0],
26     [0,0,1,0],
27     [0,0,1,1],
28     [0,0,0,1]
29 ]
30
31 def step_motor(steps, delay=0.008, clockwise=True):
32     """
33     Mueve el motor paso a paso
34     """
35     sequence = seq if clockwise else list(reversed(seq))
36
37     for _ in range(abs(steps)):
38         for step in sequence:
39             for pin, val in zip(pins, step):
40                 GPIO.output(pin, val)

```

```

41         time.sleep(delay)
42
43     try:
44         print("Motor_␣horario...")
45         step_motor(100,0.008, clockwise=True)    # ~ 1 vuelta
46
47         print("Motor_␣antihorario...")
48         step_motor(100,0.008, clockwise=False) # ~ 1 vuelta
49
50
51     finally:
52         GPIO.cleanup() # Se limpian los pines (reinicio)

```

Código 2: Código de control de motor a pasos con ULN2003 en python

```

1  import cv2
2  import numpy as np
3  import RPi.GPIO as GPIO
4  import time
5
6  # ----- CONFIGURACION MOTOR A PASOS
7  # -----
8
9  GPIO.setmode(GPIO.BCM)
10
11
12 # Pines motor eje X (motor a pasos)
13 STEPPER_X_PINS = [5, 6, 13, 19]
14
15
16 for pin in STEPPER_X_PINS:
17     GPIO.setup(pin, GPIO.OUT)
18     GPIO.output(pin, 0)
19
20
21 # Secuencia paso a paso
22 SEQ = [

```

```

18     [1,0,0,1],
19     [1,0,0,0],
20     [1,1,0,0],
21     [0,1,0,0],
22     [0,1,1,0],
23     [0,0,1,0],
24     [0,0,1,1],
25     [0,0,0,1]
26 ]
27
28 def step_motor(steps, delay=0.015, clockwise=True):
29     sequence = SEQ if clockwise else list(reversed(SEQ))
30     for _ in range(abs(steps)):
31         for step in sequence:
32             for pin, val in zip(STEPPER_X_PINS, step):
33                 GPIO.output(pin, val)
34                 time.sleep(delay)
35     for pin in STEPPER_X_PINS:
36         GPIO.output(pin, 0)
37
38 # -----CONFIGURACION MOTOR DC
39 # -----
39 IN1 = 17
40 IN2 = 27
41 ENA = 22 # PWM
42
43 GPIO.setup(IN1, GPIO.OUT)
44 GPIO.setup(IN2, GPIO.OUT)
45 GPIO.setup(ENA, GPIO.OUT)
46
47 pwm = GPIO.PWM(ENA, 1000)
48 pwm.start(0)

```

```

49
50 def avance(speed=100):
51     GPIO.output(IN1, GPIO.HIGH)
52     GPIO.output(IN2, GPIO.LOW)
53     pwm.ChangeDutyCycle(speed)
54
55 def retroceso(speed=100):
56     GPIO.output(IN1, GPIO.LOW)
57     GPIO.output(IN2, GPIO.HIGH)
58     pwm.ChangeDutyCycle(speed)
59
60 def stop():
61     GPIO.output(IN1, GPIO.LOW)
62     GPIO.output(IN2, GPIO.LOW)
63     pwm.ChangeDutyCycle(0)
64
65 # ----- PUSH BUTTON (TOGGLE) -----
66 BUTTON_PIN = 23
67 GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
68
69 sistema_activo = False
70 ultimo_estado_boton = 1
71 tiempo_ultimo_cambio = time.time()
72
73 def leer_boton():
74     global sistema_activo, ultimo_estado_boton,
75         tiempo_ultimo_cambio
76     estado = GPIO.input(BUTTON_PIN)
77
78     # Previene rebotes (50 ms)
79     if time.time() - tiempo_ultimo_cambio < 0.05:
80         return

```

```

80
81     # Detectar flanco de bajada
82     if estado == 0 and ultimo_estado_boton == 1:
83         sistema_activo = not sistema_activo
84         if sistema_activo:
85             print("\nSistema_ACTIVADO")
86         else:
87             print("\nSistema_PAUSADO-Motores_detenidos")
88             stop()
89
90         tiempo_ultimo_cambio = time.time()
91
92         ultimo_estado_boton = estado
93
94
95     # ----- PARAMETROS -----
96     TOLERANCIA_X = 8
97     TOLERANCIA_Y = 8
98     PASOS_BASE = 2
99     DELAY_STEP = 0.01
100     speed = 100
101     insercion = 3
102
103     cap = cv2.VideoCapture(0)
104
105     print("Sistema_esperando_activacion_del_boton...")
106
107     # ----- COLORES DE ESPIRALES -----
108     colores = [
109         {"nombre": "negro", "lower": np.array([0,0,0]), "upper": np.
            array([180,255,50])},
110         {"nombre": "azul", "lower": np.array([90,80,50]), "upper": np

```



```

        .array([130,255,255])),
111 {"nombre": "verde", "lower": np.array([35,60,60]), "upper":
        np.array([85,255,255])},
112 {"nombre": "rojo", "lower": np.array([0,100,100]), "upper":
        np.array([10,255,255])},
113 {"nombre": "negro", "lower": np.array([0,0,0]), "upper": np.
        array([180,255,50])},
114 {"nombre": "fucsia", "lower": np.array([135,80,80]), "upper":
        np.array([170,255,255])},
115 {"nombre": "cafe", "lower": np.array([5,100,20]), "upper": np
        .array([20,200,200])},
116 {"nombre": "morado", "lower": np.array([135,80,80]), "upper":
        np.array([170,255,255])},
117 {"nombre": "naranja", "lower": np.array([10,150,100]), "upper
        ": np.array([25,255,255])},
118 ]
119 indice_color = 0
120
121 try:
122     ultimo_centrado = 0
123     centrado_anterior = False
124     frames_centrado = 0
125     frames_requeridos = 5
126     Insertos_centrados = 0
127     mecanismo_realizado = False
128
129     while True:
130
131         leer_boton()      # REVISAMOS EL BOTON EN CADA FRAME
132
133         if not sistema_activo:
134             stop()

```

```

135         cv2.imshow("POSITION_CONTROL", np.zeros((200,300,3),
136             dtype=np.uint8))
137         if cv2.waitKey(1) & 0xFF == 27:
138             break
139         continue # NO EJECUTAR LOGICA
140
141     ret, frame = cap.read()
142     if not ret:
143         print("Error al capturar frame")
144         break
145
146     frame = cv2.rotate(frame, cv2.ROTATE_180)
147     height, width = frame.shape[:2]
148
149     # ----- ROI cuadrado centrado -----
150
151     side = int(min(width, height) * 0.15)
152     cx_f, cy_f = width // 2, height // 2
153     x1 = max(0, cx_f - side // 2)
154     y1 = max(0, cy_f - side // 2)
155     x2 = x1 + side
156     y2 = y1 + side
157     roi = frame[y1:y2, x1:x2]
158
159     cx_roi = roi.shape[1] // 2
160     cy_roi = roi.shape[0] // 2
161
162     hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
163     color_actual = colores[indice_color]
164     mask = cv2.inRange(hsv, color_actual["lower"],
165         color_actual["upper"])
166     mask = cv2.medianBlur(mask, 7)

```

```

165     mask = cv2.erode(mask, None, iterations=1)
166     mask = cv2.dilate(mask, None, iterations=2)
167
168     contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
169                                     cv2.CHAIN_APPROX_SIMPLE)
170
171     if contours:
172         # Contorno mas grande
173         c = max(contours, key=cv2.contourArea)
174         ((cx, cy), radius) = cv2.minEnclosingCircle(c)
175         M = cv2.moments(c)
176
177         if M["m00"] > 0 and radius > 5:
178             cx = int(M["m10"] / M["m00"])
179             cy = int(M["m01"] / M["m00"])
180
181             # Calculo del error de desfase
182             error_x = cx - cx_roi
183             error_y = -cy + cy_roi
184             centrado_x = abs(error_x) <= TOLERANCIA_X
185             centrado_y = abs(error_y) <= TOLERANCIA_Y
186             centrado_actual = centrado_x and centrado_y
187
188             if centrado_actual:
189                 frames_centrado += 1
190                 color = (0, 255, 0) # Verde centrado
191                 estado = "CENTRADO"
192                 stop()
193                 #print(f" Inserto centrado (error {error_x:+d
194                     }px en x, error {error_y:+d}px y - Motores
195                     detenidos")

```

```

194         # Si lleva varios frames centrado, cuenta
           como inserto terminado
195     if frames_centrado >= frames_requeridos and
       not centrado_anterior:
196         print(f"Inserto_{Insertos_centrados+1}
           _centrado_correctamente.")
197         centrado_anterior = True
198         ultimo_centrado = time.time()
199         mecanismo_realizado = False
200
201     # Mientras esta centrado, mostramos tiempo de
       permanencia en consola
202     if centrado_anterior:
203         tiempo_centrado= time.time() -
           ultimo_centrado
204         print(f"Tiempo_centrado:{tiempo_centrado
           :.1f}_s", end="\r")
205
206     # Mostrar texto en frame
207     cv2.putText(frame, "placing_subtract....
           ", (30, 120),
208                 cv2.FONT_HERSHEY_SIMPLEX,
           0.7, (0,255,255), 2)
209
210     # Si ya paso el tiempo de insercion,
       pasar al siguiente inserto
211     if tiempo_centrado >= insercion and not
       mecanismo_realizado:
212         Insertos_centrados += 1
213         print(f"_Pasando_al_siguiente_inserto
           _({Insertos_centrados}/30)")
214         avance(80)

```

```

215         time.sleep(2)
216         step_motor(23, DELAY_STEP, clockwise=
                True)
217         stop()
218
219         # Reinicio de estados
220         mecanismo_realizado = True
221         centrado_anterior = False
222         frames_centrado = 0
223
224         # --- CAMBIO DE COLOR AUTOMATICO CADA
                30 INSERTOS ---
225         if Insertos_centrados == 30:
226             retroceso(100)
227             time.sleep(83)
228             step_motor(140, DELAY_STEP,
                    clockwise=True)
229             indice_color = (indice_color + 1)
                    % len(colores)
230             print(f"===CAMBIOA COLOR:{
                    colores[indice_color]['nombre
                    '].upper()}==")
231             Insertos_centrados = 0
232
233         else:
234             frames_centrado = 0
235             centrado_anterior = False
236             ulitmo_centrado = 0
237             color = (0, 0, 255) # Rojo fuera de
                    tolerancia
238             estado = "AJUSTANDO"
239             #vel = min(30 + int(abs(error_y) / 2), 80)

```

```

240
241     # Magnitud del error (distancia al centro)
242     magnitud_errorx = abs(error_x)
243     # magnitud_error_y = abs(error_y)
244
245     # Escalado: entre 100\% y 20\% segun que tan
        cercano este
246     factorx = np.clip(magnitud_errorx / 100, 0.2,
        1.0)
247     #factory = np.clip(magnitud_error_y / 100,
        0.2, 1.0)
248
249     #Ajuste dinamico
250     pasos_dinamicos = max(1, int(PASOS_BASE *
        factorx))
251     #velocidad_dinamica = int(vel * factory)
252
253     if error_x > TOLERANCIA_X:
254         print(f"Error_X: {error_x:+4d}px /
        Inserto_DERECHA - Motor_IZQUIERDA {
        pasos_dinamicos} pasos")
255         step_motor(pasos_dinamicos, DELAY_STEP,
        clockwise=False)
256     elif error_x < -TOLERANCIA_X:
257         print(f"Error_X: {error_x:+4d}px /
        Inserto_IZQUIERDA - Motor_DERECHA {
        pasos_dinamicos} pasos")
258         step_motor(pasos_dinamicos, DELAY_STEP,
        clockwise=True)
259     elif error_y > TOLERANCIA_Y:
260         print(f"Error_Y: {error_y:+4d}px /
        Inserto_ARRIBA - RETROCESO")

```

```

261         retroceso(50)
262     elif error_y < -TOLERANCIA_Y:
263         print(f"Error_Y: {error_y:+4d}px /  

264             Inserto ABAJO - AVANCE")
265         avance(50)
266
267     # Coordinadas globales
268     cx_global = x1 + cx
269     cy_global = y1 + cy
270
271     # Dibujo del inserto
272     cv2.circle(frame, (cx_global, cy_global), int(
273         radius), color, 2) # Contorno
274     cv2.circle(frame, (cx_global, cy_global), 2,
275         color, 3) # Centro
276
277     # Texto
278     cv2.putText(frame, f"Error_X: {error_x:+d}px",
279         (10, 30),
280         cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,
281             255, 255), 2)
282     cv2.putText(frame, f"Error_Y: {error_y:+d}px",
283         (10, 60),
284         cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,
285             255, 255), 2)
286     cv2.putText(frame, estado, (10, 90),
287         cv2.FONT_HERSHEY_SIMPLEX, 0.7, color,
288         2)
289
290 else:
291     cv2.putText(frame, "SIN DETECCION", (10, 90),

```

```

285         cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0,
286             255), 2)
287
288     # ----- Dibujar ROI y cruz central -----
289     cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)
290     center_x_global = x1 + cx_roi
291     center_y_global = y1 + cy_roi
292     cv2.line(frame, (center_x_global - 20, center_y_global),
293         (center_x_global + 20, center_y_global), (255,
294             0, 0), 2)
295
296     cv2.line(frame, (center_x_global, center_y_global - 20),
297         (center_x_global, center_y_global + 20), (255,
298             0, 0), 2)
299
300     # Mostrar frame
301     cv2.imshow("POSITION CONTROL", frame)
302
303     if cv2.waitKey(1) & 0xFF == 27:
304         break
305
306 except KeyboardInterrupt:
307     print("\nPrograma detenido manualmente")
308
309 finally:
310     print("Limpiando GPIO...")
311     for pin in STEPPER_X_PINS:
312         GPIO.output(pin, 0)
313
314     GPIO.cleanup()
315     cap.release()
316     cv2.destroyAllWindows()
317     print("Sistema apagado")

```

Código 3: Sistema de procesamiento de imágenes para sistema de corrección de posicionamiento de insertos de torre de invernadero