



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA INGENIERÍA INDUSTRIAL

**“CONSTRUCCIÓN DE UN PROTOTIPO ROBÓTICO VINCULADO
CON VISIÓN ARTIFICIAL PARA LA ASISTENCIA EN LA
ELIMINACIÓN DE MALEZA EN CULTIVOS DE FRESA”**

Trabajo de Titulación

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

INGENIERO INDUSTRIAL

AUTOR:

ALEXANDER JAVIER VILLACIS PALACIOS

Riobamba – Ecuador

2022



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA INGENIERÍA INDUSTRIAL

**“CONSTRUCCIÓN DE UN PROTOTIPO ROBÓTICO VINCULADO
CON VISIÓN ARTIFICIAL PARA LA ASISTENCIA EN LA
ELIMINACIÓN DE MALEZA EN CULTIVOS DE FRESA”**

Trabajo de Titulación

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

INGENIERO INDUSTRIAL

AUTOR: ALEXANDER JAVIER VILLACIS PALACIOS

DIRECTOR: Ing. EDUARDO FRANCISCO GARCIA CABEZAS.

Riobamba – Ecuador

2022

©2022, Alexander Javier Villacis Palacios

Se autoriza la reproducción total o parcial con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, ALEXANDER JAVIER VILLACIS PALACIOS, que el presente trabajo de titulación es de mi autoría y los resultados de este son auténticos. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 25 de febrero de 2022



Alexander Javier Villacis Palacios

C.I. 0605404417-7

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA INGENIERÍA INDUSTRIAL

El Tribunal del Trabajo de Titulación certifica que: El Trabajo de Titulación; Tipo: Propuesta Tecnológica, “**CONSTRUCCIÓN DE UN PROTOTIPO ROBÓTICO VINCULADO CON VISIÓN ARTIFICIAL PARA LA ASISTENCIA EN LA ELIMINACIÓN DE MALEZA EN CULTIVOS DE FRESA**”, realizado por el señor: **ALEXANDER JAVIER VILLACIS PALACIOS**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, el mismo cumple con los requerimientos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

FIRMA

FECHA

Ing. Eugenia Mercedes Naranjo Vargas



Firmado electrónicamente por:
**EUGENIA MERCEDES
NARANJO VARGAS**

2022-02-25

PRESIDENTE DEL TRIBUNAL

Ing. Eduardo Francisco García Cabezas



Firmado electrónicamente por:
**EDUARDO
FRANCISCO GARCIA
CABEZAS**

2022-02-25

DIRECTO DEL TRABAJO DE

TITULACIÓN

Ing. Ángel Geovanny Guamán Lozano



Firmado electrónicamente por:
**ANGEL GEOVANNY
GUAMAN LOZANO**

2022-02-25

MIEMBRO DEL TRIBUNAL DE

TITULACIÓN

DEDICATORIA

Dedico esta tesis a mi madre Jenny Palacios con todo mi corazón y amor, puesto que, gracias a su sacrificio y esfuerzo acompañado de sus enseñanzas, valores, apoyo, motivación y su gran amor he podido llegar a cumplir cada uno de mis sueños. Su bendición durante cada día de mi vida me cuida, protege y siempre me guía por el camino del bien. Eres la mejor madre del mundo.

A quien considero mi padre, quien, desde su bondad, aprecio y buenos consejos ha sabido guiarme con sabiduría para avanzar en cada uno de los peldaños que conforman mi vida por lo que le guardo un gran respeto y cariño.

A mi amada Jessi, por comprenderme, apoyarme y acompañarme en todo momento de manera incondicional; tu sonrisa, calidez y ocurrencias llenan de felicidad mi vida. Es un privilegio poder contar con tu compañía y amor.

A mi hermano Eduardo con mucho cariño, por ser un ejemplo en valores y trabajo duro; por estar pendiente de mí y brindarme tu cariño y apoyo en todo momento.

Alexander.

AGRADECIMIENTO

En especial a mis hermanos David, Steven y tíos Estela, Klever, Wilo, Mario, por siempre alentarme y apoyarme en cada paso que he dado en mi vida.

Al Ing. Eduardo García, director del presente trabajo de titulación, por compartir sus conocimientos, su apoyo, paciencia y disposición para que el presente trabajo llegue a buen término.

Al Abg. Javier Villacis por su apoyo para el desarrollo del presente trabajo de titulación.

Estoy muy agradecido con mis familiares y amigos que me brindaron su apoyo en todo el camino que he recorrido para llegar a esta meta.

Por último, pero no menos importante quiero agradecer a la Escuela Superior Politécnica de Chimborazo y los docentes de la Carrera de Ingeniería Industrial, que me brindaron sus conocimientos para formarme profesionalmente.

Alexander

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xvi
ÍNDICE DE ANEXOS	xvii
SUMMARY	xix
INTRODUCCIÓN	1
Antecedentes.....	1
Planteamiento del problema.....	2
Justificación.....	2
Alcance.....	3
Objetivos	3
<i>Objetivo General</i>	3
<i>Objetivos Específicos</i>	3
1. MARCO TEÓRICO	5
1.1. Fresa	5
1.2. Malezas.....	5
1.2.1. <i>Maleza presente en cultivos de fresa</i>	5
1.3. Deshierbe.....	5
1.4. Tipos de deshierbe.....	6
1.4.1. <i>Deshierbe manual</i>	6
1.4.2. <i>Deshierbe mecánico</i>	6
1.4.3. <i>Deshierbe químico</i>	7
1.4.4. <i>Desbrozadora</i>	8
1.5. Robots utilizados en el control de malezas.....	8
1.6. Robots móviles.....	9
1.6.1. <i>Tipos de locomoción en robots móviles</i>	9
1.7. Robots móviles con ruedas	10

1.7.1.	<i>Tracción diferencial</i>	10
1.7.2.	<i>Triciclo clásico</i>	10
1.7.3.	<i>Ackerman</i>	11
1.8.	Control de robots móviles	11
1.9.	Sistemas de Alimentación	11
1.9.1.	<i>Corriente Continua</i>	11
1.9.2.	<i>Corriente Alterna</i>	12
1.10.	Elementos de robots móviles	12
1.10.1.	<i>Actuadores</i>	12
1.10.2.	<i>Motores DC</i>	12
1.10.3.	<i>Motores paso a paso</i>	13
1.10.4.	<i>Microcontroladores</i>	14
1.10.5.	<i>Ruedas</i>	14
1.11.	Arduino	15
1.12.	Visión artificial	16
1.12.1.	<i>Etapas de un sistema de visión artificial</i>	16
1.12.2.	<i>Adquisición de imágenes</i>	17
1.12.3.	<i>Preproceso</i>	17
1.12.4.	<i>Segmentación</i>	18
1.12.5.	<i>Reconocimiento</i>	18
1.13.	Python	19
1.13.1.	<i>Open CV</i>	19
1.14.	SOLIDWORKS	19
1.15.	Protocolo SPI	19
1.16.	Protocolo UDP	20
1.17.	Modelo HSV	20
2.	METODOLOGÍA	22
2.1.	Caracterización del cultivo experimental	22
2.1.1.	<i>Localización</i>	22
2.1.2.	<i>Características</i>	23
2.2.	Selección de la mejor opción de robot móvil	24
2.2.1.	<i>Criterios de valorización</i>	25

2.3.	Selección del tipo de rueda	26
2.4.	Modelación CAD	26
2.4.1.	<i>Modelación de la estructura base</i>	29
2.4.2.	<i>Ensamble Triciclo Clásico</i>	30
2.4.2.1.	<i>Análisis Estático de la estructura.....</i>	30
2.5.	Selección de hardware	31
2.5.1.	<i>Selección del motor del prototipo.....</i>	31
2.5.1.1.	<i>Criterios de valorización.....</i>	32
2.5.1.2.	<i>Modulo Driver IBT-2</i>	33
2.5.2.	<i>Motor paso a paso</i>	34
2.5.2.1.	<i>Driver para motor paso a paso.....</i>	34
2.5.3.	<i>Selección de procesador</i>	35
2.5.4.	<i>Selección de microcontrolador</i>	37
2.5.5.	<i>Shield Ethernet para Arduino.....</i>	38
2.5.6.	<i>Modulo relé de 2 canales</i>	38
2.5.7.	<i>Regulador de voltaje DC/DC Step Down LM2596.....</i>	39
2.5.8.	<i>Baterías.....</i>	39
2.5.9.	<i>Router inalámbrico</i>	40
2.5.10.	<i>Selección de Cámara.....</i>	40
2.5.11.	<i>Herramienta</i>	42
2.6.	Placa de control AUTOPRO	43
2.6.1.	<i>Diagrama electrónico placa AUTOPRO</i>	43
2.7.	Selección de Software.....	45
2.8.	Esquema de funcionamiento del prototipo	45
2.9.	Construcción y ensamble de prototipo robótico.....	46
2.9.1.	<i>Construcción de la estructura base</i>	46
2.9.2.	<i>Ensamble del sistema de tracción.....</i>	47
2.9.3.	<i>Ensamble de la herramienta de corte</i>	48
2.9.4.	<i>Pintura.....</i>	48
2.9.5.	<i>Ensamble y conexiones de hardware.....</i>	49
2.9.6.	<i>Elaboración y colocación de tapas protectoras</i>	50
2.10.	Sistema de visión artificial.....	51
2.10.1.	<i>Parámetros de programación del sistema.....</i>	51
2.10.2.	<i>Adquisición de imágenes.....</i>	51
2.10.3.	<i>Principio de funcionamiento del sistema de visión artificial.....</i>	52

2.11.	Modo Manual y Automático	58
2.12.	Diagrama de flujo del prototipo robótico	59
2.12.1.	<i>Diagrama de flujo del modo automático del prototipo</i>	61
2.12.1.1.	<i>Diagrama de flujo del procesamiento del color negro</i>	62
2.12.1.2.	<i>Diagrama de flujo del procesamiento del color verde.....</i>	64
2.12.1.3.	<i>Diagrama de flujo del control de la dirección.....</i>	66
2.13.	Programación en Python	68
2.14.	Programación de Arduino	69
2.15.	Pruebas de funcionalidad	70
2.15.1.	<i>Funcionamiento del prototipo</i>	74
2.15.1.1.	<i>Paso por el surco</i>	74
2.15.1.2.	<i>Detección y eliminación de la maleza.....</i>	77
2.15.1.3.	<i>Tiempo de procesamiento de imágenes.....</i>	80
2.15.1.4.	<i>Generación de trayectorias.....</i>	82
2.15.2.	<i>Consumo de energía.....</i>	83
3.	GESTIÓN DEL PROYECTO.....	85
3.1.	Costos Directos	85
3.2.	Costos Indirectos	86
3.3.	Costoso Totales	87
	CONCLUSIONES.....	88
	RECOMENDACIONES.....	89

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-3:	Opciones para la selección del robot móvil con ruedas.....	24
Tabla 2-3:	Evaluación de criterios de valorización.....	25
Tabla 3-3:	Elementos del prototipo robótico	28
Tabla 4-3:	Dimensiones de la estructura base.....	29
Tabla 5-3:	Opciones para la selección de motor DC.....	31
Tabla 6-3:	Evaluación de criterios de valorización.....	32
Tabla 7-3:	Especificaciones motor DC RS775	33
Tabla 8-3:	Características Driver IBT-2	33
Tabla 9-3:	Especificación del controlador DM556Y	35
Tabla 10-3:	Comparativa de características de ordenadores.....	36
Tabla 11-3:	Especificaciones técnicas de Arduino Uno	37
Tabla 12-3:	Especificaciones técnicas Shield Ethernet W5100.....	38
Tabla 13-3:	Especificaciones técnicas modulo relé de 2 canales	38
Tabla 14-3:	Especificaciones técnicas de Step Down LM2596.....	39
Tabla 15-3:	Comparativa de cámara Ip y analógica.....	41
Tabla 16-3:	Características de la cámara seleccionada.....	42
Tabla 17-3:	Especificaciones técnicas de la desbrozadora DES-440T	43
Tabla 18-3:	Conexión de pines de Arduino Uno	49
Tabla 19-3:	Prueba de funcionamiento	75
Tabla 20-3:	Comparativa de maleza contabilizada, detectada y eliminada	78
Tabla 21-3:	Consumo energético del prototipo.....	83
Tabla 1-4:	Costos de elementos mecánicos y electrónicos	85
Tabla 2-4:	Costo de mano de obra	86
Tabla 3-4:	Costo directo total.....	86

Tabla 4-4:	Cálculo de costos indirectos	86
Tabla 5-4:	Cálculo de costos totales	87

ÍNDICE DE FIGURAS

Figura 1-2:	Deshierbe manual.....	6
Figura 2-2:	Herramientas para desherbar: a) de canasta, b) de dedos, c) de torsión, d) Eco-Weeder	7
Figura 3-2:	Aplicación de herbicidas.....	7
Figura 4-2:	Desbrozadora DES-440T.....	8
Figura 5-2:	Plataforma agrícola robótica AgBot II.....	8
Figura 6-2:	Tipos de locomoción en robots móviles.....	9
Figura 7-2:	Tracción diferencial.....	10
Figura 8-2:	Triciclo clásico.....	10
Figura 9-2:	Ackerman.....	11
Figura 10-2:	Partes de un motor DC.....	12
Figura 11-2:	Motor paso a paso Nema 23.....	13
Figura 12-2:	Driver DM542Y.....	13
Figura 13-2:	Microcontrolador PIC.....	14
Figura 14-2:	Tipos de ruedas para robots móviles.....	15
Figura 15-2:	Arduino Uno Rev3.....	16
Figura 16-2:	Diagrama de bloques de las fases de la visión artificial.....	17
Figura 1-3:	Localización del cultivo experimental.....	22
Figura 2-3:	Cultivo experimental de fresa.....	23
Figura 3-3:	Rueda carretilla.....	26
Figura 4-3:	Diseño CAD del prototipo ensamblado.....	27
Figura 5-3:	Modelación del prototipo detallado.....	28
Figura 6-3:	Estructura base.....	28
Figura 7-3:	Ensamble triciclo clásico.....	30

Figura 8-3:	Motor Nema 23	34
Figura 9-3:	Driver DM556Y	34
Figura 10-3:	Ordenador Toshiba Satellite S55-B	37
Figura 11-3:	Batería 12N7BL	39
Figura 12-3:	Router inalámbrico TL-WR840N	40
Figura 13-3:	Diagrama electrónico Placa de control AUTOPRO, parte 1.....	44
Figura 14-3:	Diagrama de electrónico Placa de control AUTOPRO, parte 2.....	44
Figura 15-3:	Esquema de funcionamiento del prototipo robótico	45
Figura 16-3:	Construcción de la estructura base.....	46
Figura 17-3:	Estructura base con sistema de dirección y tren trasero.....	47
Figura 18-3:	Base del motor Nema 23 y acople	47
Figura 19-3:	Acople de la herramienta de corte.....	48
Figura 20-3:	Pintura del prototipo	48
Figura 21-3:	Colocación de la Placa de Control AUTOPRO en el prototipo	49
Figura 22-3:	Cortes láser de acrílico de las tapas del prototipo	50
Figura 23-3:	Colocación de tapa lateral derecha.....	50
Figura 24-3:	Imagen captada por la cámara Ip.	51
Figura 25-3:	Delimitación de imagen.	52
Figura 26-3:	Paneles de control HSV de color negro y verde.	53
Figura 27-3:	Imagen resultante de la extracción de HSV	54
Figura 28-3:	Imagen aplicada filtro gaussiano.....	54
Figura 29-3:	Selección de contornos	55
Figura 30-3:	Resultado de la búsqueda de contornos	55
Figura 31-3:	Detección de color verde.....	56
Figura 32-3:	Cálculo de la trayectoria del prototipo	56
Figura 33-3:	Deslizadores del panel color negro	57

Figura 34-3:	Deslizadores panel color verde	57
Figura 35-3:	Diagrama principal, parte 1.....	59
Figura 36-3:	Diagrama principal, parte 2.....	60
Figura 37-3:	Diagrama de flujo del modo automático.....	61
Figura 38-3:	Diagrama del procesamiento del color negro, parte 1.....	62
Figura 39-3:	Diagrama del procesamiento del color negro, parte 2.....	63
Figura 40-3:	Diagrama del procesamiento del color verde, parte 1.....	64
Figura 41-3:	Diagrama del procesamiento del color verde, parte 2.....	65
Figura 42-3:	Diagrama de flujo del control de la dirección, parte 1	66
Figura 43-3:	Diagrama de flujo del control de la dirección, parte 2.....	67
Figura 44-3:	Código de programación en Python.....	68
Figura 45-3:	Programación Arduino UNO.	69
Figura 46-3:	Carga de las baterías	70
Figura 47-3:	Inspección del código.....	71
Figura 48-3:	Interfaz gráfica de paneles de control blanco y verde.....	71
Figura 49-3:	Interfaz gráfica del modo de accionamiento	72
Figura 50-3:	Imagen en tiempo real con la detección de trayectoria y maleza.....	72
Figura 51-3:	Prototipo robótico en funcionamiento en el cultivo experimental	73
Figura 52-3:	Prototipo robótico en funcionamiento.....	73
Figura 53-3:	Interfaz gráfica del modo manual	74
Figura 54-3:	Desplazamientos no completados.	76
Figura 55-3:	Colisión del prototipo con la hilera.....	76
Figura 56-3:	Maleza por detectar.....	77
Figura 57-3:	Maleza detectada.....	77
Figura 58-3:	Funcionamiento de la herramienta de corte	78
Figura 59-3:	Maleza no eliminada	80

Figura 60-3: Generación de trayectorias..... 82

ÍNDICE DE GRÁFICOS

Gráfico 1-3:	Análisis de deformación.....	31
Gráfico 2-3:	Curva del par motor nema 23.....	35
Gráfico 3-3:	Porcentaje de eficacia de paso por surco.....	75
Gráfico 4-3:	Eficacia de detección de maleza.....	79
Gráfico 5-3:	Eficacia de eliminación de maleza.....	79
Gráfico 6-3:	Tiempo de procesamiento de imágenes.....	81
Gráfico 7-3:	Porcentaje de tiempos de procesado de imágenes.....	81

ÍNDICE DE ANEXOS

ANEXO A: PLANOS DEL PROTOTIPO ROBOTICO PARA LA ASISTENCIA EN LA ELIMINACION EN LA MALEZA Y SUS ELEMENTOS

ANEXO B: ESQUEMA GENERAL DE CONEXIONES

ANEXO C: DIAGRAMAS DE FLUJO MODO MANUAL Y ARDUINO

ANEXO D: CODIGO PYTHON DEL PROTOTIPO ROBÓTICO

ANEXO E: CÓDIGO DE PROGRAMACIÓN EN ARDUINO

RESUMEN

El presente proyecto describe la construcción de un prototipo robótico para la asistencia en la eliminación de maleza en cultivos de fresa basado en el uso de técnicas de visión artificial con el fin de aportar con el sector agrícola de la provincia de Chimborazo. Se realizó el análisis del caso de estudio, en base a esto, se modeló el prototipo robótico con la utilización del software SolidWorks; el resultado fue un robot móvil de ruedas con un diseño de triciclo clásico que tiene la capacidad de moverse a través del cultivo de fresa con solvencia. El análisis estático realizado sobre la estructura base con el material utilizado arrojó que al aplicar una fuerza de 500 N existe una deformación de 0,000193 mm, esta deformación es imperceptible y corrobora que el modelado es correcto y soporta las cargas a las que fue sometido. El sistema de control y procesamiento de imágenes lo realizó un ordenador que se encargó de la comunicación con el prototipo utilizando los protocolos SPI y UDP. El código de programación del sistema visión artificial para el reconocimiento de trayectorias de recorrido e identificación de malezas se lo realizó usando el lenguaje de programación Python y Arduino IDE. Los resultados obtenidos a través de las pruebas de funcionalidad indican que el prototipo robótico posee una eficacia de detección de maleza de 94,13% y de eliminación de 90,65%, estos datos son un indicativo de que el prototipo cumple con las funciones para las cuales fue construido. La velocidad de procesamiento de imágenes fue en promedio de 0,039 s, este tiempo comprueba que existe un procesamiento veloz que garantizó un correcto funcionamiento del sistema de visión artificial. Se recomienda utilizar elementos de protección personal mientras el prototipo se encuentre realizando funciones de corte.

Palabras clave: <ASISTENTE ROBÓTICO AGRICULTOR> <VISIÓN ARTIFICIAL>
<ELIMINACIÓN DE MALEZA> <CULTIVO DE FRESA> <PYTHON (SOFTWARE)>.



Firmado electrónicamente por:
**HOLGER GERMAN
RAMOS UVIDIA**

0574-DBRA-UPT-2022

2022-04-04

SUMMARY

This project describes the construction of a robotic prototype to assist in the elimination of weeds in strawberry crops based on the use of artificial vision techniques in order to contribute to the agricultural sector in Chimborazo province. The analysis of the case study was performed, based on this, the robotic prototype was modeled with the use of SolidWorks software; the result was a mobile wheeled robot with a classic tricycle design that has the ability to mobilize it through the strawberry crop with solvency. The static analysis performed on the base structure with the material used, resulted that when applying a force of 500 N, there is a deformation of 0.000193 mm, this deformation is imperceptible and corroborates that the modeling is correct and supports the loads to which it was subjected. The control and image processing system was carried out by a computer that was responsible for communication with the prototype using SPI and UDP (User Datagram Protocol) protocols. The programming code of the machine vision system for path-path recognition and weed identification was done using the Python and Arduino IDE (Integrated Development Environment) programming language. The results obtained from the functionality tests indicate that the robotic prototype has a weed detection efficiency of 94.13% and a removal efficiency of 90.65%. This data is an indication that the prototype fulfils the functions for which it was built. The speed of image processing was on average 0.039 s, this time proves that there is a rapid processing that guaranteed the correct functioning of the machine vision system. It is recommended to use personal protective equipment while the prototype is performing cutting functions.

Keywords: < ROBOTIC ASSISTANT AGRICULTURAL > <ARTIFICIAL VISION>
<WEED REMOVAL> <STRAWBERRY CROP> <PYTHON (SOFTWARE)>

ANGELA
CECIBEL
MORENO
NOVILLO

Firmado digitalmente
por ANGELA CECIBEL
MORENO NOVILLO
Fecha: 2022.04.14
20:53:51 -05'00'

Introducción

Antecedentes

Según el (FOASAT, 2019) la producción mundial de fresa en el 2019 se estima en un total de 8885028 toneladas con un rendimiento de 224142 hg/ha, ocupando de esta manera un área de cosecha total de 396401 hectáreas que es una extensión de territorio considerable comparándolas con el cultivo de otras frutas.

En América del Sur la producción total de fresa en el 2019 fue de 312766 toneladas representando de esta forma el 3,5 % de la producción total mundial de la fresa, esto evidencia que nuestra región no es un referente en la producción de esta fruta ya que, comparándola con un solo país de América del Norte como lo es México se aprecia una notable diferencia, este país aporta con el 9,7 % de la producción mundial.

En el Ecuador según los datos estadísticos obtenidos del (FOASAT, 2019) existe una disminución en la producción de fresa, de 1900 toneladas producidas en el 2009 paso a 787 en el 2019, esto se origina por la disminución progresiva del área cosechada de fresas en el país, lo que nos indica que hay una tendencia pronunciada por parte de los agricultores a elegir otro tipo de cultivo a producir. Si bien la producción ha bajado en gran cantidad, el rendimiento de hectogramos por hectárea es mayor en comparación al rendimiento de los años pasados. Esto es un claro indicador de la evolución paulatina gracias a la implementación de nuevas técnicas de agricultura que permite al productor aprovechar de mejor manera sus cultivos.

La relación existente entre tecnología y agricultura a nivel nacional históricamente ha sido casi nula, pero en los últimos años a través de proyectos de investigación se han generado diversos sistemas robóticos utilizando herramientas CAD, según (Castro, 2016, pp. 10-11) el uso de estas herramientas brindan la oportunidad de simular diferentes condiciones sobre el diseño, esto permite tener una idea basada en la ciencia de cuál será el comportamiento del elemento dicho lo que permite tomar decisiones sobre cada uno de los sistemas simulados y así validar o desestimar cada uno de estos.

En la actualidad muchos de los procesos industriales y productivos han sido relegados debido a la irrupción de sistemas inteligentes, estos sistemas tienen la capacidad de realizar actividades agrícolas como lo son la siembra, cosecha, eliminación de maleza y preparación de la tierra. Estos avances se

ven impulsados por la necesidad de encontrar medios alternativos a los comunes para producir una gran cantidad de alimentos al menor costo y con la mínima cantidad de mano de obra. Por las razones expuestas se ha incentivado a la agricultura para que tome parte de la automatización de sus procesos y para obtener cultivos que brinden un mayor rendimiento.(Velasco, 2017)

Planteamiento del problema

En el Ecuador existe un suelo muy fértil en el cual se puede generar una producción agrícola rica y variada, esto a su vez genera que en los cultivos crezcan a parte de la planta deseada otro tipo de vegetación que según sea el caso se la puede catalogar como maleza.

En este sentido, el problema que aqueje a los productores de la de la fresa es la existencia de varias clases de malezas que crecen en los surcos existentes entre las hileras de los cultivos de esta fruta, estas malezas tienen una incidencia negativa en el rendimiento del cultivo, pudiendo ser grande el daño si no son controladas permanentemente. La necesidad del control permanente de la maleza genera que el productor agrícola requiera de mano de obra que se encuentre de manera constante en el cultivo, detectando y eliminando la maleza. En nuestro país esta actividad se lo realiza generalmente de forma manual y en ocasiones con la utilización de químicos con herbicidas; por lo tanto, es de gran importancia dar solución a este problema y como parte de la solución se propone el sistema que consta de un prototipo robótico que servirá como asistente para la eliminación de la maleza, con el fin de garantizar que los nutrientes del suelo sean destinados exclusivamente a la fresa y a su vez disminuir la mano obra requerida en cultivo de esta fruta.

Justificación

En el Plan Nacional para el Buen Vivir recalca la importancia de fomentar el desarrollo en diferentes competencias, que garanticen un empleo digno y el emprendimiento. Se pretende fomentar esto a través del crecimiento económico inclusivo que vincule la educación superior tecnológica con las diversas aptitudes productivas existentes en el país.(SENPLADES, 2017, p. 67)

Con el presente proyecto se pretende fortalecer el enlace actual entre la agricultura y la educación superior, y que de esta forma se inserten recursos tecnológicos como lo es la robótica en el proceso de cultivo facilitando la labor agrícola.

Dentro del contexto del cultivo de fresa, el control y eliminación de la maleza existente en los surcos es favorable para que la planta pueda dotarse de los nutrientes necesarios durante su crecimiento y desarrollo como a su vez, mantenerse saludable y pueda brindar frutos de calidad de manera constante. Al realizar un deshierbe en el que se evite la utilización de herbicidas se obtienen frutos orgánicos que son de mejor recepción en el mercado frutícola lo que le da una mayor apertura al agricultor de comerciar su producto. También al contar con un sistema que elimine la maleza permite que el agricultor pueda ocupar el tiempo que destinaba a esta actividad en el mantenimiento y cuidado de la fresa.

Es importante recordar que este sistema robótico no pretende reemplazar al agricultor que realiza esta actividad sino más bien, será desarrollado con la finalidad de realizar una mejor distribución del recurso humano para optimizar otros procesos dentro del cultivo de esta fruta, este proyecto también proporcionará un gran paso en la evolución de la agricultura de la provincia de Chimborazo al incluir en ella tecnología con la contamos en la actualidad.

Alcance

El presente trabajo de titulación está enfocado en desarrollar un sistema de detección y eliminación de malezas mediante un prototipo robótico vinculado con visión artificial. El objetivo gira en torno a innovar la manera en la que es eliminada la maleza teniendo la intención de obtener como resultado un sistema que aporte con nueva tecnología a los agrícolas de la provincia, innovación, investigación, y que sobre todo busque mejorar el cultivo. El proyecto abarca tanto la modelación CAD como su construcción física.

Objetivos

Objetivo General

Construir un prototipo robótico vinculado con visión artificial para la asistencia en la eliminación de maleza en cultivos de fresa.

Objetivos Específicos

- Definir el caso de estudio para la caracterización del cultivo de fresa para la generación de diferencias con la maleza.
- Modelar un sistema robótico para la definición del sistema de movilización y eliminación de la maleza más idóneos.

- Determinar el hardware necesario para cubrir los requerimientos de la implementación del prototipo robótico.
- Definir el software para la codificación del algoritmo de visión artificial para la identificación de trayectorias y la maleza dentro del cultivo.
- Efectuar pruebas de funcionalidad en un cultivo experimental.

1. MARCO TEÓRICO

1.1. Fresa

La fresa es una fruta que pertenece a la familia de las rosáceas y a su vez al género *Fragaria*. En el sector agrícola se ha experimentado con la fresa, esta experimentación guiada a la obtención de técnicas de cultivo menos costosas y en mejorar a la planta con el fin de conseguir fresas resistentes a enfermedades y con la capacidad de brindar un rendimiento elevado junto con una calidad gustativa elevada. (Bianchi, 2018, pp. 1-2)

1.2. Malezas

En la agricultura se consideran malezas a ciertas especies vegetales no deseadas por el hombre que afectan la calidad y producción de cultivos, ya que estas plantas tienen una importante capacidad competitiva por espacio, nutrientes, luz y agua con otras existentes en el cultivo. Esto hace que el control de malezas en la agricultura sea fundamental para obtener un rendimiento óptimo en los cultivos. (Morales, 2009, pp. 102-106)

1.2.1. Maleza presente en cultivos de fresa

Las malezas perennes son predominantes en los cultivos de fresa, en los cuales por lo general podemos encontrar malezas como: Correhuela (*Convolvulus arvensis* L), Romacilla (*Rumex acetosella*), Chufa (*Cyperus esculentus* L), Diente de León (*Taraxacum officinalis*), Chépica (*Paspalum dilatatum* Poin) y Correhuela (*Convolvulus arvensis* L). Estas malezas perjudican al cultivo, pudiendo quitar nutrientes los cuales son necesarios para un correcto desarrollo de la fresa. (Sepúlveda et al., 2015, pp. 93-94)

1.3. Deshierbe

Se define el deshierbe como el acto de retirar especies vegetales no deseadas de los cultivos también conocidas como malezas (Rivera, 2015, pp. 20-21). Esta labor generalmente es realizada manualmente, en la actualidad existen varios tipos de deshierbe que facilitan este tipo de trabajo.

1.4. Tipos de deshierbe

Existen varios tipos de deshierbe utilizados para la eliminación de las malezas los cuales son comunes en la mayoría de los cultivos y son utilizados según las características de los terrenos y sembríos.

1.4.1. *Deshierbe manual*

En el deshierbe manual se realiza de forma que la parcela del cultivo quede libre de la presencia de vegetación ajena a la que se encuentra sembrada. Se considera de forma manual ya que se lo realiza utilizando azadones o a través de la extracción directa por medio de las manos. La eficiencia de este tipo de deshierbe se obtendrá de las horas utilizadas por hectárea trabajada. (Moreira et al., 2012, pp. 311-314)



Figura 1-2: Deshierbe manual

Fuente: (Cueva & Groten, 2010, p.102)

1.4.2. *Deshierbe mecánico*

Es un tipo de deshierbe en el cual es utilizado una labranza habitual en el cual se ocuparán herramientas como azadones rotatorios y de precisión. Su eficiencia será calculada en función a las horas empleadas por hectárea. (Moreira et al., 2012, pp. 311-314)

Estas herramientas de deshierbe utilizan técnicas específicas las cuales son arrancar, enterrar y cortar (Ahmad et al., 2014, pp. 56-136), las técnicas mencionadas a lo largo de la historia y en la actualidad son los más utilizados, siendo su uso más frecuente que el uso de químicos herbicidas. Las herramientas mecánicas de deshierbe son utilizadas en su mayoría fuera de la línea de cultivo ya que no poseen la característica de selección, por lo que no pueden diferenciar entre la maleza y el cultivo. Los desherbadores mecánicos gozan de cierta variedad entre los cuales podemos mencionar los siguientes: desherbadores de canasta, desherbadores de dedos, desherbadores de torsión de muelles helicoidales

flexibles, desbrozadores, escardador de torsión, ECO-Weeder, entre otros. Algunas de estas herramientas las podemos observar en la siguiente figura:

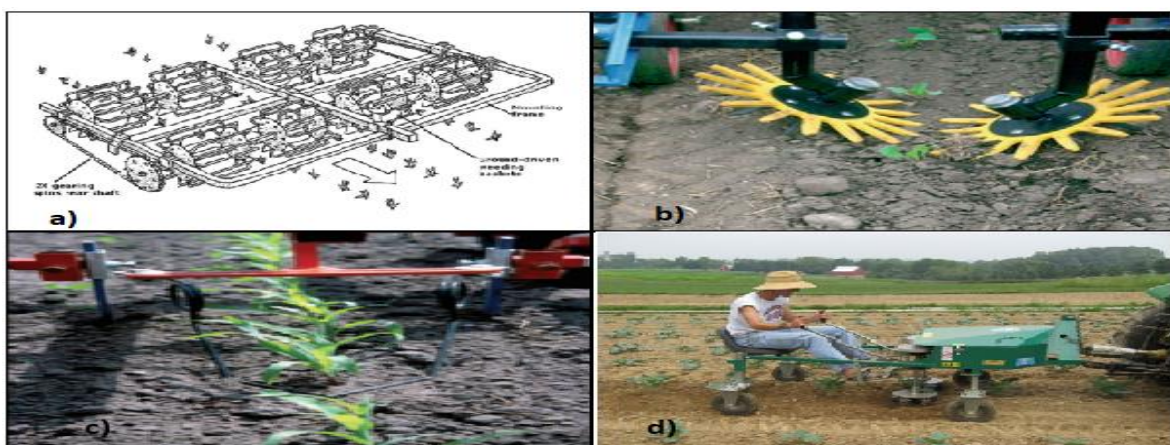


Figura 2-2: Herramientas para desherbar: a) de canasta, b) de dedos, c) de torsión, d) Eco-Weeder
Fuente: (Ahmad et al., 2014, pp. 56-136)

1.4.3. Deshierbe químico

El deshierbe químico es aquel que utiliza herbicidas para la eliminación de la maleza, para esto se utilizan equipos como lo son aspersores de mochila, aspersores, aguilonos entre otros. En este tipo de control de maleza el volumen de aplicación va a diferir según el cultivo. (Moreira et al., 2012, pp. 311-314)

Los herbicidas utilizados en el deshierbe químicos pueden ser aplicados de forma directa en el suelo, estos generalmente perjudican la germinación de malezas, aunque tienden a tener un efecto prolongado por lo que se les denomina también herbicidas residuales; también se los aplica directamente en el follado de esta forma solo afectan el área de maleza que ha tenido contacto con el herbicida, por ello se los conoce también como herbicidas de contacto. (Caseley & Parker, 1996, pp. 147-162)



Figura 3-2: Aplicación de herbicidas
Fuente: (Inostroza et al., 2011)

1.4.4. Desbrozadora

La desbrozadora es una máquina herramienta que cumple con la función de cortar maleza al ras del suelo. Un hilo de nailon es el cual realiza el corte de la maleza, también se usan cuchillas diseñadas para la eliminación de este tipo de vegetación, estas están presentadas en forma de disco. (Oña, 2019, pp. 1-2)

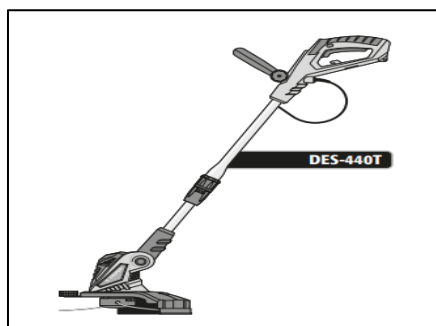


Figura 4-2: Desbrozadora DES-440T

Fuente: (TRUPER, 2018)

1.5. Robots utilizados en el control de malezas

En la actualidad existen varios tipos de robots que cumplen la función de controlar la proliferación de la maleza en los cultivos, entre estos podemos mencionar al AgBot II el cual es desarrollado por la Universidad Tecnológica de Queensland, este robot utiliza un sistema de visión artificial que detecta y clasifica en tipo real la maleza existente en el cultivo, a su vez tiene a su disposición herramientas químicas o mecánicas intercambiables que están diseñadas en base a características tanto de la maleza como del suelo, por lo que el robot tomara decisiones en base a lo mencionado y utilizara estas herramientas. (Hall et al., 2018, PP.107-120)



Figura 5-2: Plataforma agrícola robótica AgBot II

Fuente: (Hall et al., 2018, pp. 107-120)

1.6. Robots móviles

Se denomina robots móviles a las plataformas mecánicas a las que se les concede sistemas de locomoción el cual posee la capacidad de navegar o trasladarse de un lugar a otro sin estar sujetos a un punto físicamente; estos robots móviles para transportar sus cargas poseen cierto nivel de autonomía. Las aplicaciones que poseen los robots móviles en la actualidad son muy variadas, y van desde tareas básicas hasta actividades que son riesgosas para el hombre, un ejemplo de esto es su utilización en la agricultura o el transporte de elementos o cargas peligrosas.(Bambino, 2008)

1.6.1. Tipos de locomoción en robots móviles

La clasificación de los robots móviles va de acuerdo con el tipo de locomoción que utiliza. Para estos tipos de locomoción se tiene por lo general tres medios de movimiento los cuales son por patas, por ruedas y por orugas. De los tipos de locomoción mencionados, el que a través del tiempo ha tenido más desarrollo e investigación son los robots móviles con ruedas, este mayor desarrollo se debe a las diversas ventajas que ofrecen las ruedas sobre las ofrecidas por las patas y orugas.(Barrientos et al., 2007, pp. 12-17)

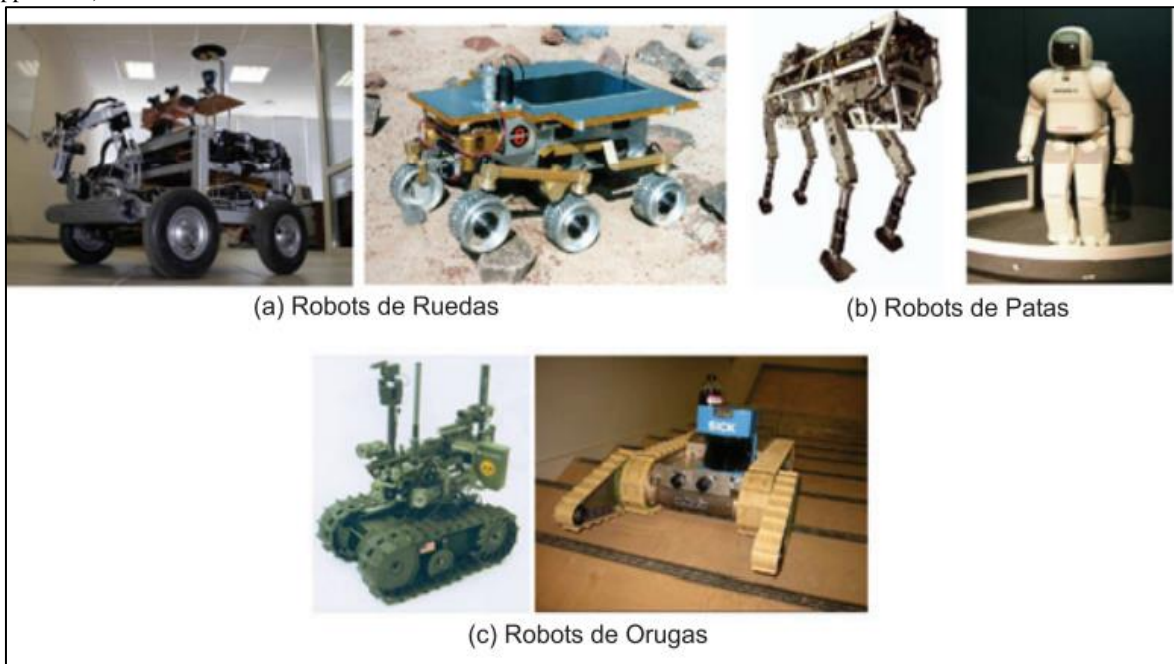


Figura 6-2: Tipos de locomoción en robots móviles

Fuente: (Barrientos et al., 2007, pp. 12-17)

1.7. Robots móviles con ruedas

Se considera a los robots móviles con ruedas como una solución simple y eficiente para obtener una movilidad adecuada en terrenos firmes y libre de obstáculos, con este tipo de robots se puede lograr una velocidad relativamente alta. Para estos robots móviles existen varios tipos de configuraciones de locomoción que se pueden llevar a cabo mediante sus ruedas.(Jiménez, 2012, pp.8-23)

1.7.1. Tracción diferencial

Los robots con tracción diferencial poseen dos ruedas las cuales tienen la capacidad de girar de forma independiente la una de la otra. El robot representara una trayectoria u otra en función a la velocidad independiente de sus ruedas.(Jiménez, 2012, pp 8-23)

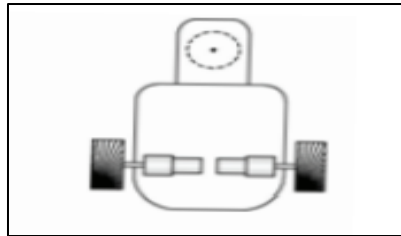


Figura 7-2: Tracción diferencial

Fuente: (Barrientos et al., 2007)

1.7.2. Triciclo clásico

Los robots móviles de ruedas tipo triciclo tienen como característica poseer dos ruedas traseras que serán las encargadas de la propulsión y tiene una rueda delantera la cual será orientable a la dirección deseada o requerida. (Jiménez, 2012, pp. 8-23)

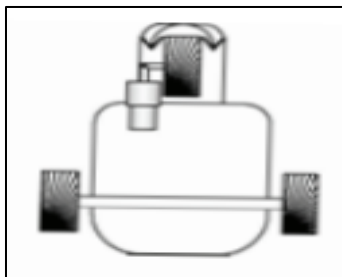


Figura 8-2: Triciclo clásico

Fuente: (Barrientos et al., 2007)

1.7.3. Ackerman

Este tipo de robot móvil es lo más común y lo podemos encontrar en automóviles y en muchos de los vehículos autónomos diseñados para exteriores. Esta configuración se caracteriza por la diferencia de ángulo existente entre la rueda interior y la exterior, la rueda interior representara un ángulo mayor que el exterior. (Jiménez, 2012, pp. 8-23)

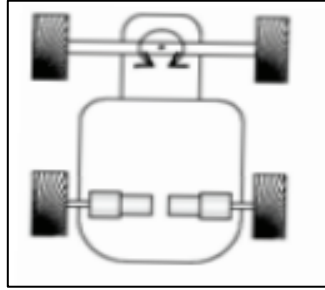


Figura 9-2: Ackerman
Fuente:(Barrientos et al., 2007)

1.8. Control de robots móviles

El control de los robots móviles está basado en dos tipos de controles, que son el control cinemático y dinámico. El control cinemático es el encargado de calcular la orientación y la posición en cada instante del robot, para de esta manera, el robot pueda realizar las acciones necesarias para que el este se movilice hasta el punto designado. El control dinámico es el encargado de controlar la velocidad angular de los motores propulsores. (Jiménez, 2012, pp. 8-23)

1.9. Sistemas de Alimentación

Dado que los robots contienen diversos componentes eléctricos y electrónicos deben de poseer una energía constante, la cual les permita funcionar de manera continua para que estos robots cumplan con los objetivos para los cuales fueron diseñados. Se pueden obtener dos tipos de energía eléctrica los cuales se explican a continuación:

1.9.1. Corriente Continua

Se considera como corriente continua a aquella corriente que se mantiene siempre constante. Es decir que el flujo de electrones se mantiene siempre en una misma dirección (Morales, 2005, pp. 77-80). En

este caso para los robots móviles la energía eléctrica de corriente continua se almacena en baterías o es generada a través de paneles solares para su posterior utilización.

1.9.2. Corriente Alterna

Si existe una variación periódica en la dirección del flujo de electrones se considera la corriente eléctrica como una corriente eléctrica alterna (Morales, 2005, pp. 77-80). Puede ser obtenida a través de generadores de corriente alterna mejor conocidas como alternado o a su vez se pueden obtener de la que proveen las empresas de energía.

1.10. Elementos de robots móviles

1.10.1. Actuadores

Los actuadores son aquellos dispositivos que poseen la capacidad de generar un cambio de velocidad, posición o estado sobre un dispositivo o elemento mecánico a través de una fuerza producida por la transformación de la energía.(Corona et al., 2014, pp. 280-281)

1.10.2. Motores DC

Los motores DC o motores de corriente continua son maquinas que se encargan de convertir energía eléctrica en energía mecánica, esto genera un par motor que hace que el eje que posee gire (Corona et al., 2014, pp. 280-281). Su principio de funcionamiento está basado en la fuerza que produce un conductor debido a que existe la presencia de un campo magnético sobre una intensidad de corriente eléctrica.

Estos motores están compuestos por los elementos que se muestran en la siguiente figura:

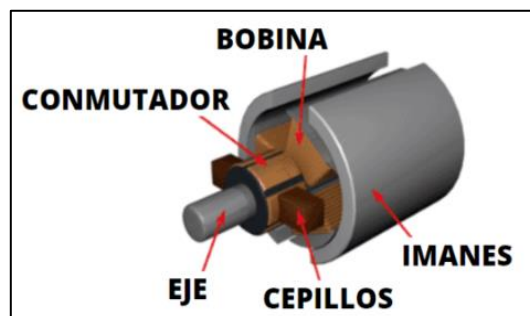


Figura 10-2: Partes de un motor DC

Fuente: (Crespo, 2017)

1.10.3. Motores paso a paso

También denominados motores de movimiento indexado son motores que son diseñados para que puedan girar un ángulo determinado en base a señales eléctricas recibidas a través de sus terminales de control. Estos motores reciben en forma de pulsos ordenes de lazo abierto con la finalidad de hacer rotar su eje en un ángulo exacto definido. (Conti, 2013, pp. 1-2)

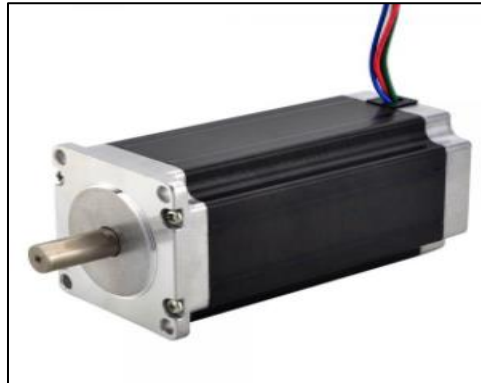


Figura 11-2: Motor paso a paso Nema

Fuente: <https://bit.ly/3wdQDDB>

Estos motores por lo general necesitan drivers los cuales son circuitos que permiten controlarlos a través de corriente continua de forma simple. Estos drivers permiten el control de intensidad de corriente y voltajes que van a ser entregador al motor para de esta manera controlar la velocidad o ángulo de giro del eje.



Figura 12-2: Driver DM542Y

Fuente: (STEPPERONLINE, 2019)

1.10.4. Microcontroladores

Se defino a los microcontroladores como circuitos integrados de alta escala, los cuales poseen en su totalidad componentes de un procesador digital secuencial síncrono programable. Los microcontroladores tienen la ventaja de ser pequeños lo que permite que puedan estar empotrados en los dispositivos que gobiernan.(Mandado et al., 2007, pp. 11-12)



Figura 13-2: Microcontrolador PIC

Fuente: (Inzunza, 2016, p.16)

Por lo general los microcontroladores contienen los siguientes elementos:

- Unidad Central de Proceso
- Líneas de Entrada/Salida
- Módulos de control de periféricos
- Generador de impulsos de reloj
- Memoria RAM
- Memoria ROM/PROM/EPROM.(Barra & Barra, 2015, pp 19-21.)

Los microcontroladores ofrecen ciertas de ventajas lo que explica que sean tan comunes y se utilicen en diversos sistemas, entre estas ventajas tenemos que poseen un espacio optimizado lo que hace que ocupe el menor espacio a un costo aceptable, es un dispositivo de bajo consumo cuando no está efectuando acciones esto hace que sea muy común la utilización de microcontroladores con fuentes de alimentación como baterías.(Valdés & Pallás, 2007, pp. 14-21)

1.10.5. Ruedas

Para los robots móviles según la configuración cinemática utilizada, se tienen para la locomoción cuatro tipos de ruedas.(Barrientos et al., 2007, pp. 12-17)

- Ruedas omnidireccionales: También conocida como ruedas suecas, en este tipo de rueda se incluye rodillos alrededor de su periferia esto le brinda un grado de libertad por lo que adquiere movilidad en el eje perpendicular en el sentido de rodadura que poseen las ruedas convencionales. (Martínez & Sisto, 2009, pp.25-26)
- Rueda orientable descentrada: Denominada como rueda de castor, es una rueda que su rotación esta alrededor de un eje vertical, este eje no pasa por el centro de la rueda. Se utiliza generalmente para proveer de una mayor estabilidad a la estructura mecánica del robot móvil siendo una rueda de dirección.
- Rueda convencional: Son las ruedas más utilizadas en la fabricación de robots móviles y automóviles, estas ruedas consiguen girar en sentido de la orientación que se le asigno. Las ruedas convencionales pueden ser fijas u orientables. (Serrano, 2010, pp. 45-46)
- Rueda de bolas: Este tipo de ruedas suelen ser utilizadas cuando se necesita que el movimiento de giro sea sencillo con la sola inversión del sentido en el que giran las ruedas motrices. (Barrientos, 2008, pp. 4-7)



Figura 14-2: Tipos de ruedas para robots móviles

Fuente: (Silva et al., 2007, pp. 1-14)

1.11. Arduino

Arduino es considerada una plataforma electrónica que posee un código abierto la cual está basada en hardware y software sencillos de usar. Las placas fabricadas por Arduino están en la capacidad de leer entradas y convertirlas en una salida accionando algún tipo de actuador. La plataforma Arduino en la actualidad es una pieza fundamental en miles de proyectos los cuales abarcan desde objetos sencillos hasta ser parte de complejos instrumentos tecnológicos y científicos. (ARDUINO, 2021)

Arduino ofrece ciertas ventajas para sus usuarios especialmente para aficionados, profesores y estudiantes sobre lo que pueden ofrecer otras plataformas de microcontroladores como:

- Económicas: Las placas comercializadas por Arduino son de fácil adquisición ya que son relativamente más económicas en comparación al valor en el que se pueden comprar otras plataformas similares en el mercado.
- Multiplataformas: El Arduino Software se puede correr en Windows, Macintosh OS X y Linux.
- Entorno de programación simple: El Arduino Software es de fácil comprensión y utilización para principiantes, pero a su vez ofrece una buena flexibilidad para que usuarios con mayores conocimientos y experiencia lo aprovechen. (ARDUINO, 2021)



Figura 15-2: Arduino Uno Rev3
Fuente:(ARDUINO, 2021)

1.12. Visión artificial

La visión artificial tiene como objetivo extraer información utilizando un procesador o computador para obtener datos del mundo físico en base a imágenes. La representación de una realidad ofrece datos sobre colores, formas, brillo, etcétera; esta información es sobre la cual un sistema de visión artificial actuará. La representación ya mencionada por lo general está en forma de escenas tridimensionales, imágenes en movimiento o imágenes estáticas.(Vélez et al., 2003, pp. 18-21)

1.12.1. Etapas de un sistema de visión artificial

La visión artificial posee cuatro fases principales. En estas etapas trata de replicar la visión del ser humano, esta captura la luz por medio de los ojos, la información captada circula a través del nervio óptico hasta llegar al cerebro que es donde será procesada (Vélez et al., 2003, pp.18-21).

- La primera fase es la captura, la cual consiste en la obtención de imágenes digitales con la utilización de algún tipo de sensor.

- El preprocesamiento es la segunda fase, en esta se realiza un tratamiento digital sobre las imágenes con la finalidad de facilitar las siguientes fases. Se utilizan transformaciones geométricas, filtros y a su vez se eliminan partes no deseadas de la imagen o se realzan partes importantes de la misma.
- A la tercera fase se la denomina segmentación, esta fase consiste en aislar objetos de mayor interés de una imagen o escena para comprenderla.
- La cuarta fase es el reconocimiento o clasificación, el objetivo de esta fase es distinguir los objetos segmentados, con la ayuda de la caracterización que se instaura previamente con el fin de diferenciarlos.

A continuación, se muestra un diagrama de bloques en el cual se puede apreciar que las 4 fases no necesariamente deben realizar este proceso de manera secuencial.

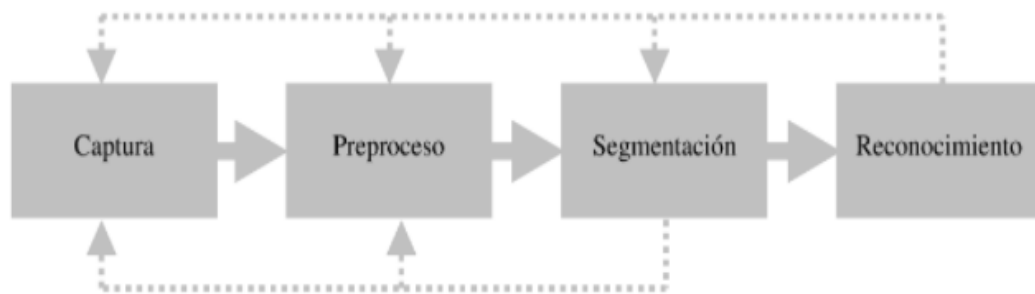


Figura 16-2: Diagrama de bloques de las fases de la visión artificial

Fuente: (Vélez et al., 2003, pp. 18-21)

1.12.2. Adquisición de imágenes

Según (González et al., 2006, pp. 19-20) la etapa de captura se trata de obtener que la imagen conseguida sea de la mayor calidad posible para que de esta forma puedan funcionar las etapas siguientes. La captura de manera correcta es un proceso del cual depende mucho que el reconocimiento tiene éxito o no.

1.12.3. Preproceso

El preprocesado de imagen es aquel que se encarga de mediar los efectos que se manifiestan en la adquisición de imágenes, los que pueden aparecer como pérdida de definición, ruido, entre otras. De

manera que el preprocesado trata de mejorar o repara la imagen de las imperfecciones o errores producidos por el hardware. (Vélez et al., 2003)

En el preproceso existen para su utilización varios filtros los cuales son muy utilizados, en el caso de este trabajo de titulación nos enfocaremos en los siguientes:

- Filtro de Mediana: Este filtro consiste en conseguir los valores de cada píxel existente en la ventana para proceder a ordenarlos y tomar la mediana, en otras palabras, escoger el valor una igual cantidad de valores inferiores y superiores en el área de la ventana.
- Filtro detector de bordes: Cuando se quieren detectar bordes de elementos de un tamaño considerable las máscaras de convolución son muy utilizadas. Estas mascarar en el centro poseen un valor positivo y este se curva a un negativo el cual termina en cero. (González et al., 2006)

1.12.4. Segmentación

Se considera como segmentación a la acción de dividir la imagen digital capturada en zonas individualizadas en las que se diferencien los objetos y a su vez se identifique en que posición se encuentran respecto al fondo. Esta fase es fundamental para el reconocimiento de formas, pudiéndolo facilitar o dificultar según sea el caso.

1.12.5. Reconocimiento

En esta fase se basa en obtener las características que definan a cada objeto. Según (González et al., 2006) estas características son forma, textura, color, orientación, etcétera. De los mencionados se deben elegir aquellos que puedan ser discriminantes, suficientes o independientes entre sí.

Los discriminantes son aquellos que difieren de manera clara de los elementos de una misma clase o familia; ser suficientes, es decir, que puede caracterizar de forma suficiente si un elemento pertenece a una familia predeterminada; por último, ser independiente entre sí, es decir, que los elementos se definan como si no estuviesen relacionados entre sí de tal manera que si varia alguno los demás no se ven afectados.

1.13. Python

Python es un lenguaje de programación que es muy interactivo y didáctico, ya sea por su sencillez como por su sintaxis. Una de las muchas ventajas de usar Python es que los programas son muchos más compactos, es decir son más cortos, además la escritura en Python es más simple y su lectura es mucho más que si se estuviera utilizando otros lenguajes y otra entre todos sus ventajas es que el entorno de ejecución ayuda a detectar todos los errores de la programación y así nos da gran cantidad de información para detectarlos y corregirlos. (Rodríguez, 2016, pp. 10-11)

1.13.1. Open CV

Es una herramienta que nos ayuda en todo el camino de la visión artificial. Desde la detección de movimiento, además de la identificación de los objetos y todo esto en vida real En definitiva es una multiplataforma que está disponible para todas las plataformas de sistemas operativos. E incluye un soporte para lenguajes de programación entre ellos Python, C++, Java. (Navacerrada, 2018, pp. 35-36)

1.14. SOLIDWORKS

El software CAD denominado Solidworks es una plataforma en la cual se puede realizar automatizaciones de diseños mecánicos. A través de los procedimientos debidos se crean modelos 3D de elementos a partir de un croquis. El resultado son piezas con las cuales podemos realizar análisis, ensamblajes o suben samblajes en 3D. Es una herramienta tecnológica utilizada a nivel mundial por la mayoría de los diseñadores, debido a su diseño y a los complementos. (Dassault Systemes SolidWorks Corporation, 2015, pp. 9-10)

1.15. Protocolo SPI

El *Serial Peripheral Interface* es un bus tres líneas, por el cual se transmite paquetes de información de un tamaño de 8 bits. Está presente en un sin números de dispositivos en la actualidad. Aquellos dispositivos que se encuentran conectados al bus se les considera como maestros y esclavos. De esta

forma se tiene que dentro de este protocolo el maestro es quien inicia el envío o transferencia de información sobre el bus y este genera señales de control y reloj. (López, 2015, p. 1)

1.16. Protocolo UDP

El protocolo UDP se limita a recoger el mensaje y enviar el paquete por la red sin necesidad de establecer una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Para garantizar la llegada, el protocolo exige a la máquina de destino del paquete que envíe un mensaje. Si dicho mensaje no llega pasado un tiempo establecido, la máquina de destino envía el mensaje de nuevo. Esto puede originar la duplicación y/o desordenación de los datagramas a su destino. (García, 2012, 186-187)

1.17. Modelo HSV

Los modelos HSV está concebido para ser sencillo y fácilmente interpretable. Este modelo de color utiliza denominaciones más comunes para sus partes que lo componen. Así tenemos las siguientes características.

- Intensidad de color: Se define a la capacidad de ser más claro u oscuro.
- Saturación: Es la diferenciación existente entre el color y un gris que posea la misma intensidad. Esto quiere decir que mientras más diferente sea más saturado estará.
- Matiz de un color: su ángulo dentro de la rueda cromática.
- El espacio HSV se representa como un cilindro.

La conversión del modelo RGB a HSV se explica en el siguiente modelo matemático:

Sea $max = \max\{R, G, B\}$ y $min = \min\{R, G, B\}$

$$H \begin{cases} \frac{(G - B) * 60}{max - min}, si R = max \\ \frac{(B - R) * 60}{(max - min) + 120}, si G = max \\ \frac{(R - G) * 60}{(max - min) + 240}, si B = max \end{cases} \quad (1)$$

$$S = \frac{(max - min)}{max} \quad (2)$$

$$V = max \quad (3)$$

Donde:

H: Matiz

S: Saturación.

V: Valor de intensidad.

Es muy importante tener en cuenta que si R=G=B, el H no está definido. Por lo que este disminuirá según disminuye la saturación. (García, 2016, pp. 31-35).

Esta conversión se puede realizar por medio de códigos de programación por ejemplo los conversores utilizados en Python.

2. METODOLOGÍA

En este capítulo se describen la metodología y procedimientos que se realizaron en la construcción del prototipo robótico para la eliminación de maleza vinculado con visión artificial. Este prototipo robótico ha sido realizado en base a fundamentos lógicos y metódicos en base a fundamentos científicos expuestos en el Marco Teórico. Se especificará los elementos componentes del prototipo, esquemas electrónicos y eléctricos, selección de hardware y software, diseño estructural y programación.

2.1. Caracterización del cultivo experimental

2.1.1. Localización

Para la caracterización de los cultivos de fresa y de la maleza existente en ellos, se consideró un cultivo experimental de fresa de proporcionado por la Sra. Nora Orozco. Este cultivo experimental se encuentra ubicado en el Barrio Chingazo Alto, Cantón Guano – Chimborazo -Ecuador.

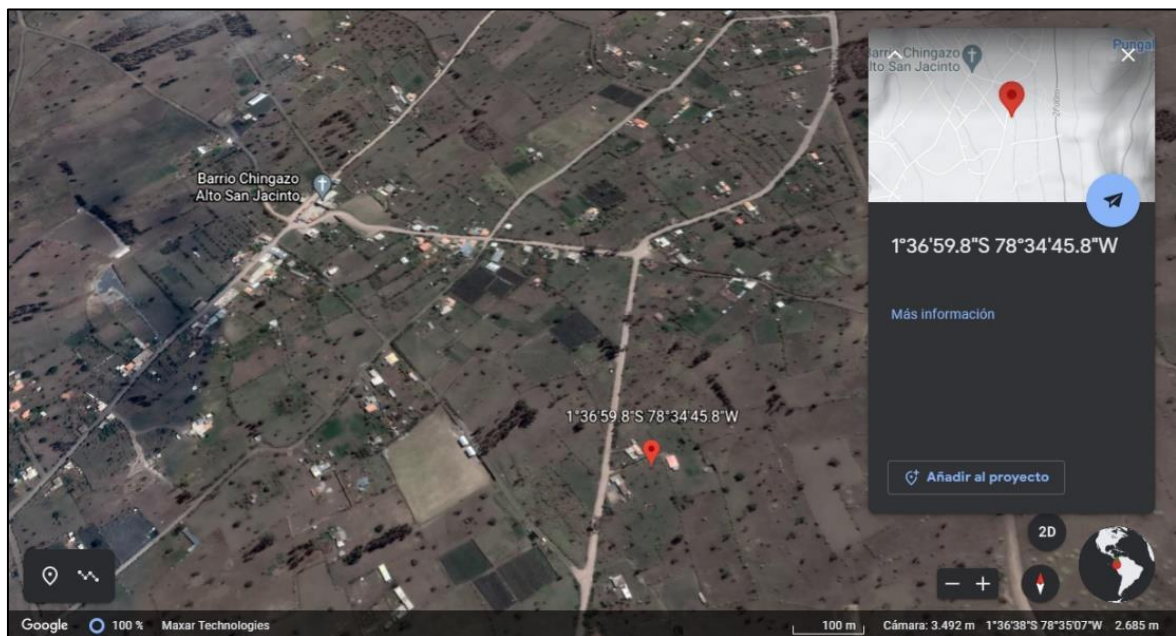


Figura 1-3: Localización del cultivo experimental

Fuente: (Google, s.f.)

2.1.2. Características

Las siguientes características encontradas en el cultivo experimental fueron de suma importancia para establecer los primeros parámetros que debía cumplir el prototipo robótico:

- El terreno tiene un grado de inclinación leve.
- El cultivo experimental de fresas está formado por 13 hileras y 12 surcos los cuales no son uniformes y presentan variaciones en sus medidas.
- El cultivo posee un sistema de riego por goteo el cual está cubierto de principio a fin por lonas plásticas de color negro en el cual sobre sale en la parte superior las plantas de fresa.
- Las hileras tienen una altura irregular tomada desde los surcos de entre 22 y 33 centímetros.
- El ancho existente en los surcos varía entre 27 y 34 centímetros.
- La fresa y la maleza están separadas por la diferencia de altura entre la hilera y el surco.
- La maleza existente en los surcos es abundante y se encuentra presente en la mayoría del cultivo.
- El largo medio de los surcos es de 35 m.

En la Figura 2-3 se puede observar el cultivo experimental.



Figura 2-3: Cultivo experimental de fresa

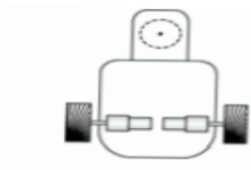

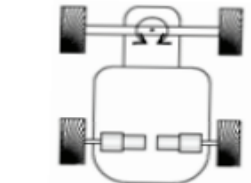
Realizado por: Villacis, A. 2021

2.2. Selección de la mejor opción de robot móvil

Como se indica en el capítulo anterior existen distintos tipos de robots móviles los cuales pueden ser de mayor o menor utilidad dependiendo el campo de aplicación.

En este caso al tener que movilizarse por un terreno irregular con poco espacio se ha considerado como la mejor opción a los robots móviles de ruedas. Existen distintos tipos de robots móviles de ruedas que se pueden emplear dentro de estos cultivos. En la Tabla 1-3 se exponen las opciones para ser evaluadas y por siguiente seleccionar la mejor opción.

Tabla 1-3: Opciones para la selección del robot móvil con ruedas

Opción	Tipo de robot móvil con ruedas	Imagen
1	Tracción diferencial	
2	Triciclo clásico	
3	Ackerman	

Fuente: (Barrientos et al., 2007)

Realizado por: Villacis, A. 2021

2.2.1. Criterios de valorización

- **Facilidad de modelación y construcción:** Se debe considerar el nivel de dificultad que representa modelar estos diseños con todos sus componentes en el espacio reducido existente en los surcos y conseguir un desempeño optimo.
- **Costo de fabricación:** Esta basado en el costo de adquisición de los elementos necesarios para la construcción
- **Funcionalidad:** Se considera la capacidad con la que pueda realizar giros en espacios reducidos
- **Consumo de energía bajo:** Los robots móviles para su funcionamiento deben convertir energía eléctrica en energía mecánica por lo que existe consumo de energía eléctrica, por ello se debe procurar que el robot realice sus funciones con el menor consumo de energía posible.

Utilizando la matriz de ponderación se elige el tipo de robot móvil de ruedas que obtenga la calificación ponderada más alta basado en los criterios de valorización ya expuestos. Se detallan los resultados en la siguiente tabla.

Tabla 2-3: Evaluación de criterios de valorización

Criterios de valorización	Peso asignado	1		2		3	
		Calificación	Ponderación	Calificación	Ponderación	Calificación	Ponderación
Facilidad de modelación y construcción	0,30	3	0,90	4	1,20	3	0,90
Costo de fabricación	0,15	2	0,30	3	0,45	2	0,30
Funcionalidad	0,35	3	1,05	5	1,75	2	0,70
Consumo de energía bajo	0,20	3	0,60	3	0,60	3	0,60
TOTAL	1,00		2,85		4,00		2,50

Realizado por: Villacis, A. 2021

Para la evaluación de la matriz de ponderación se calificará de 1 a 5 puntos; siendo 1 la calificación más baja y 5 la más alta. A su vez se consideran a los valores altos como opciones factibles y que presentan menos dificultades y a los valores bajos, como opciones menos factibles y que presentan mayores dificultades.

El resultado obtenido en la Tabla 2-3 indica que la calificación más alta fue de 4 ,00; la misma que pertenece a la opción 2 correspondiente al robot móvil de ruedas de tipo triciclo clásico. Este tipo de robot satisface las necesidades presentadas.

2.3. Selección del tipo de rueda

La correcta elección del tipo de rueda es fundamental ya que será el encargado de proveer movilidad a la estructura del prototipo y a su vez dependiendo de la rueda seleccionada contará con un pequeño nivel de suspensión. Al tener que movilizarse en un terreno muy irregular la tracción que pueda proveer la rueda ayudara a que su movilización se efectuó con mayor facilidad.

En este caso se selecciona una rueda convencional, específicamente las denominadas ruedas de carretilla, estas poseen una cámara de aire la cual las vuelve livianas en comparación a otros tipos de ruedas de carretilla; además tanto el material con el que están fabricadas como el labrado que posee han sido pensados para ser utilizados en la agricultura y sobre todo en terrenos irregulares.



Figura 3-3: Rueda carretilla
Realizado por: Villacis, A. 2021

2.4. Modelación CAD

La modelación del prototipo robótico que se presenta en este trabajo de titulación se basa en los diseños de los robots móviles con ruedas específicamente con el diseño del denominado triciclo clásico, a su vez está realizado para que pueda efectuar sus funciones con eficiencia en el medio en el que se va a desenvolver. Este diseño está compuesto por dos partes fundamentales que son la estructura de la plataforma móvil, herramienta de corte y el sistema de tracción, cada uno de estos fueron pensados para que sus materiales sean de fácil adquisición, económicos y también en el caso del sistema de tracción que sean de un sencillo mantenimiento y que los elementos que lo conforman tengan a su disposición repuestos si así lo requieren.

Se realizó el modelado en el software SolidWorks, tanto de la parte estructural como del sistema de tracción, estos nos permiten tener un diseño bidimensional tridimensional de la estructura a la cual podremos validar con simulaciones en el software mencionado; las dimensiones del prototipo robótico están ideadas de tal forma que pueda movilizarse en el limitado espacio en el que realizara sus actividades.

A continuación, podemos observar el modelado CAD del ensamble del prototipo.

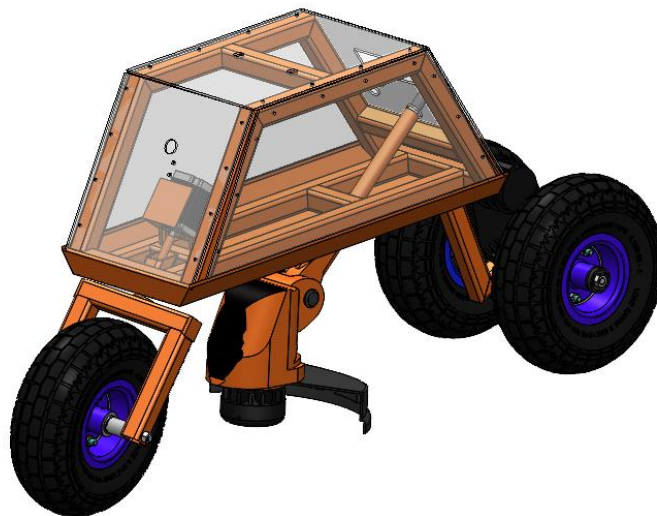


Figura 4-3: Diseño CAD del prototipo ensamblado

Realizado por: Villacis, A. 2021

En la Figura 20-3 se puede apreciar una vista explosionada del diseño del prototipo; en la Tabla 33 se detallará cada una de las partes que lo constituyen. Para el armado del prototipo se debe seguir la numeración establecida en orden descendiente, de esta forma se puede ensamblar correctamente el prototipo.

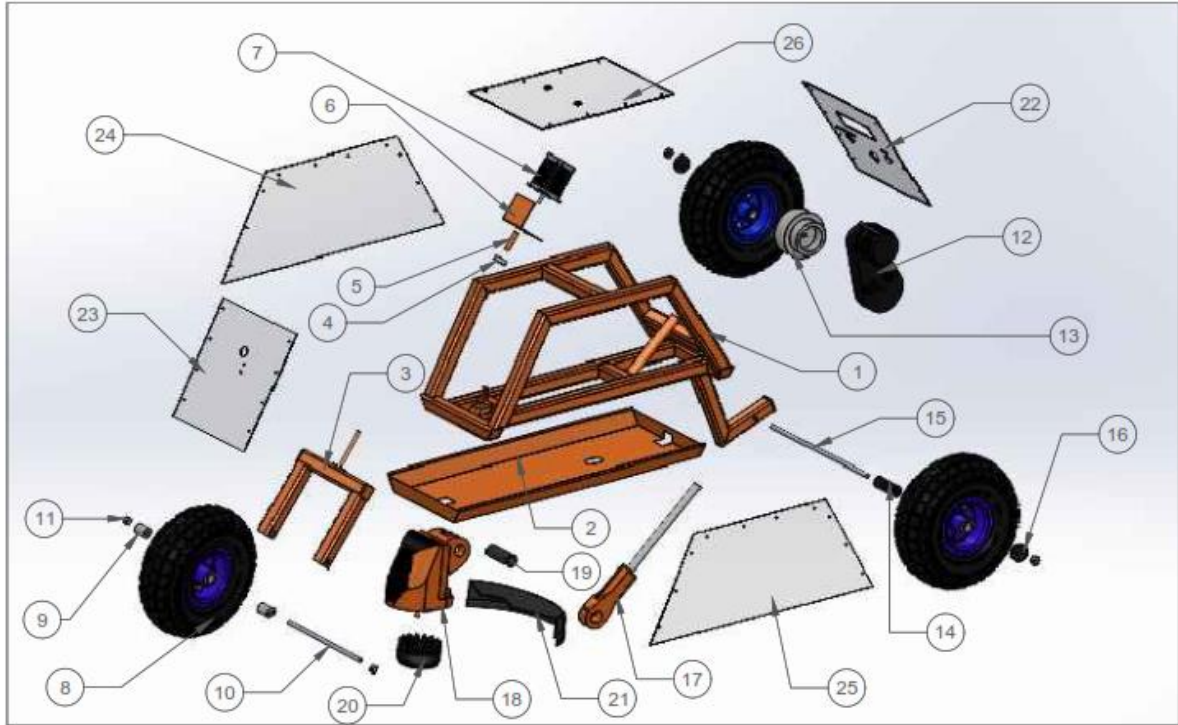


Figura 5-3: Modelación del prototipo detallado.

Realizado por: Villacis, A. 2021

Tabla 3-3: Elementos del prototipo robótico

ELEMENTOS DEL PROROTIPO ROBÓTICO			
Elemento	Cantidad	Nombre	Material
1	1	Estructura base	ASTM A36
2	1	Tapa inferior	ASTM A653
3	1	Soporte rueda delantera	ASTM A36
4	1	Rodamiento dirección	ASTM A295
5	1	Acople del motor nema 23	ANSI 1020-1050
6	1	Soporte motor/cámara	ASTM A36
7	1	Motor paso a paso Nema 23	Varios
8	3	Rueda de carretilla	Nylon y caucho natural
9	2	Porta eje delantero	ASTM A36
10	1	Eje rueda delantera	ANSI 1020-1050
11	4	Tuerca hexagonal	ASTM A194
12	1	Conjunto motor-caja reductora	Varios
13	1	Acople caja reductora-rueda	Nylon
14	1	Porta eje trasero	PVC
15	1	Eje ruedas traseras	ANSI 120-1050
16	6	Acople rueda	PVC

17	1	Barra de soporte de herramienta	ASTM A36
18	1	Cabezal de la desbrozadora	Varios
19	1	Botón de ajuste de cabezal	PVC
20	1	Guarda	PVC
21	1	Cabezal de corte	Varios
22	1	Tapa trasera	Acrílico
23	1	Tapa delantera	Acrílico
24	1	Tapa lateral derecha	Acrílico
25	1	Tapa lateral izquierda	Acrílico
26	1	Tapa superior	Acrílico

Realizado por: Villacis, A. 2021

En el Anexo A se pueden encontrar los planos de la estructura base en la unidad de medida mm del sistema internacional de medida.

2.4.1. Modelación de la estructura base

A continuación, se puede observar la base del diseño de la estructura, en esta base estarán acoplados los diferentes elementos que componen el prototipo.

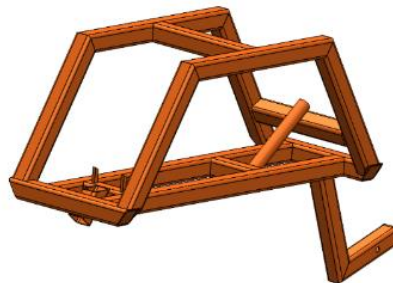


Figura 6-3: Estructura base

Realizado por: Villacis, A. 2021

En la Tabla 4-3 se indica las dimensiones de la estructura base, en ella cuenta tanto ancho, largo y altura.

Tabla 4-3: Dimensiones de la estructura base

Dimensiones del prototipo robótico	
Ancho	250,00 mm
Largo	762,30 mm
Altura	437,56 mm

Realizado por: Villacis, A. 2021

2.4.2. *Ensamble Triciclo Clásico*

El tipo de robot móvil seleccionado es un triciclo clásico debido, para lo cual se modeló un eje que se acopla con la estructura base. A su vez se diseñó una tapa inferior que protegerá a los componentes internos del prototipo mientras esté en funcionamiento. En la Figura 7-3 se aprecia el acople de los elementos mencionados.



Figura 7-3: Ensamble triciclo clásico
Realizado por: Villacis, A. 2021

2.4.2.1. *Análisis Estático de la estructura*

La estructura del triciclo clásico será el encargado de soportar ciertas cargas estáticas que serán provocadas por el peso de los componentes y a su vez del mismo material que lo compone. Para lo cual se realiza con la ayuda del software Solidworks un análisis estático. Dentro del software se definen las cargas externas que va a soportar, a de más de, sujeciones que permitirán realizar la simulación.

El material elegido para la estructura es el acero ASTM A36 el cual es muy utilizado en el país, es de bajo costo y fácil adquisición. Este acero presenta un límite elástico de 250 MPa. El valor obtenido de la simulación utilizando el análisis de tensión de Von Mises es de 78,53 MPa el cual es menor al límite elástico del acero utilizado, lo que nos indica que la estructura soportara las cargas con las que se hizo el análisis.

También se obtuvieron los resultados del análisis de deformación, estos resultados nos indica la existencia de deformaciones en la estructura a causa de las cargas externas aplicadas y a su vez

comprobamos la rigidez que tiene la misma. El parámetro de deformación máxima que nos arroja la simulación es de 0,000193 mm en los laterales de la estructura, esta deformación es imperceptible para el ojo humano, por consiguiente, la estructura es considerada rígida.

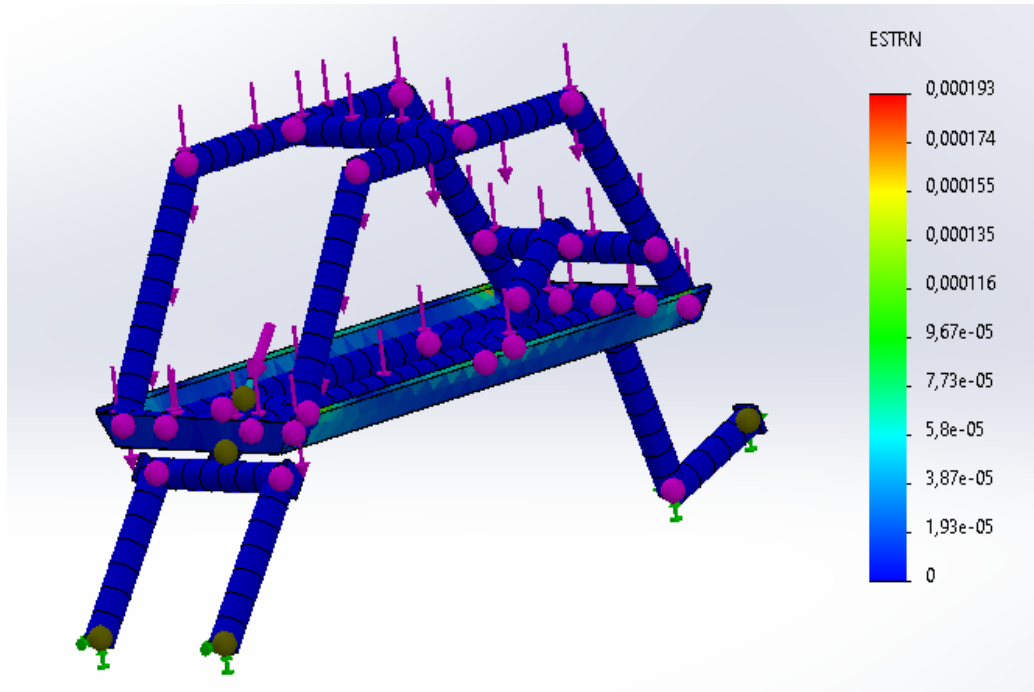


Gráfico 1-3: Análisis de deformación.
Realizado por: Villacis, A. 2021

2.5. Selección de hardware

2.5.1. Selección del motor del prototipo

Para proveer de la energía mecánica necesaria para que el prototipo robótico pueda movilizarse a través del cultivo experimental se requiere de un motor DC que este en la capacidad de realizar esta función. Para esto se ha considerado dos tipos de motores DC los cuales serán evaluados y comparados entre sí y por consiguiente elegir la mejor opción.

Tabla 5-3: Opciones para la selección de motor DC

Opción	Tipo de motor DC
1	Motor con escobillas
2	Motor Brushless

Realizado por: Villacis, A. 2021

2.5.1.1. Criterios de valorización

- **Costo:** Este criterio se basa en el costo de cada tipo de motor existente en el mercado.
- **Tamaño:** Se considera de manera favorable que el motor sea pequeño y pueda utilizarse en lugares reducidos.
- **Potencia:** Esta basado en la relación entre la potencia de salida del motor y tamaño.
- **Eficiencia:** Se debe considerar con mejor calificación el tipo de motor que genere menor desperdicio de potencia.

Utilizando la matriz de ponderación se elige el tipo de motor DC que obtenga la calificación ponderada más alta basado en los criterios de valorización ya expuestos. Se detallan los resultados en la siguiente tabla.

Tabla 6-3: Evaluación de criterios de valorización

Criterios de valorización	Peso asignado	1		2	
		Calificación	Ponderación	Calificación	Ponderación
Costo	0,50	5,00	2,50	3,00	1,50
Tamaño	0,10	5,00	0,50	4,00	0,40
Potencia	0,25	3,00	0,75	4,00	1,00
Eficiencia	0,15	3,00	0,45	5,00	0,75
TOTAL	1,00		4,20		3,65

Realizado por: Villacis, A. 2021


Para la evaluación de la matriz de ponderación se calificará de 1 a 5 puntos; siendo 1 la calificación más baja y 5 la más alta. A su vez se consideran a los valores altos como opciones factibles y que presentan menos dificultades y a los valores bajos, como opciones menos factibles y que presentan mayores dificultades.

El resultado obtenido en la Tabla 6-3 indica que la calificación más alta fue de 4,20; la misma que pertenece a la opción 1 correspondiente al motor con escobillas. Este tipo de motor DC satisface las necesidades presentadas.

El motor DC con escobillas que se selecciono es un motor DC RS775 de 12v, debido a su bajo consumo de energía, su bajo costo y sobre todo la disponibilidad de este tipo de motor en el mercado. El control de este motor es muy sencillo y esto también facilita su utilización en el prototipo. Este motor será el encargado de proveer la tracción al robot y que este se pueda movilizar en el cultivo.

A continuación, en la Tabla 6-3 se indican las características del motor DC seleccionado.

Tabla 7-3: Especificaciones motor DC RS775

MOTOR DC RS775		
Alimentación	DC	
Voltaje	12	
Configuración	Síncrono	
Potencia	111,68 W	
Velocidad	18800	
Eficiencia	75,7 %	


Fuente: (Longkangtech, S.f.)

Realizado por: Villacis, A. 2021

2.5.1.2. Modulo Driver IBT-2

Este módulo es el adecuado para el control del motor RS775 ya que está diseñado para motores DC de alta potencia. Además, posee una interconexión considerada fácil con Arduino lo que lo hace un módulo de control ideal ya que la placa Arduino Uno será la que lo controle. Este control consta de 2 conectores de alimentación y 5 de trabajo haciendo de muy fácil utilización. Por las razones expuestas este Driver es el elegido para cumplir con la función de ser quien controle al motor de tracción. En la Tabla 6-3 se puede apreciar tanto el aspecto del puente H IBT-2 como sus principales características.

Tabla 8-3: Características Driver IBT-2

Características IBT-2		
Rango de voltaje de funcionamiento	5,5v - 27v	
Corriente máxima	43A	
Nivel de entrada	3,3v -5,5v	
Frecuencia máxima de operación	25kHz	

Fuente: (Rainer, 2013)

Realizado por: Villacis, A. 2021

2.5.2. Motor paso a paso

El motor paso a paso será utilizado para proveer movilidad a la dirección del prototipo por lo que debe considerar que el motor seleccionado posea un alto torque a bajas velocidad, que sea de una consistencia robusta y de alta confiabilidad. De esta manera pueda operar adecuadamente en el entorno en el que será aplicado.

En lo que a motores paso a paso se refiere, la mejor alternativa en el mercado hoy en día son los motores provistos por Steperonline debido a su costo y su fiabilidad. En este caso el motor seleccionado es el motor paso a paso Nema 23 Bipolar 3Nm. Este motor posee la habilidad de al ser utilizado a baja velocidad, puede conducirla la misma sin perder un solo paso, lo que lo vuelve un motor preciso y el más adecuado para la dirección del prototipo robótico.



Figura 8-3: Motor Nema 23

Realizado por: Villacis, A. 2021

2.5.2.1. Driver para motor paso a paso

Stepperonline provee para sus motores paso a paso el mejor controlador según el modelo de motor elegido. El controlador que se utiliza para este motor Nema 23 es el Driver DM556Y. A continuación, se muestra en la Figura 9-3 el aspecto del controlador.

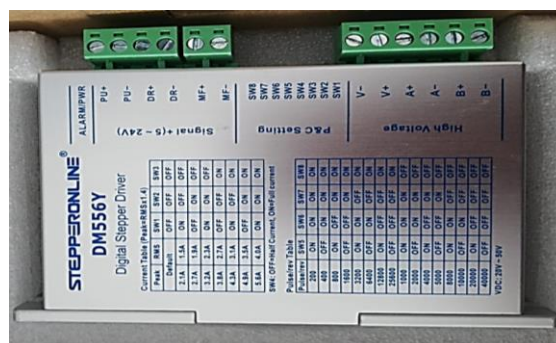


Figura 9-3: Driver DM556Y

Realizado por: Villacis, A. 2021

Este controlador tiene las especificaciones mostradas en la siguiente tabla.

Tabla 9-3: Especificación del controlador DM556Y

ESPECIFICACIÓN DEL CONTROLADOR	
Voltaje de entrada	DC 20v -50v
Corriente de señal lógica	10 mA
Ancho de pulso	2,5 μ S
Temperatura de funcionamiento	50 °C máximo.
Vibración	10-50 Hz

Realizado por: Villacis, A. 2021

En la siguiente figura se puede observar el comportamiento de la curva par de motor, esto nos brindara la información necesaria para configurar el controlador según lo que se requiera.

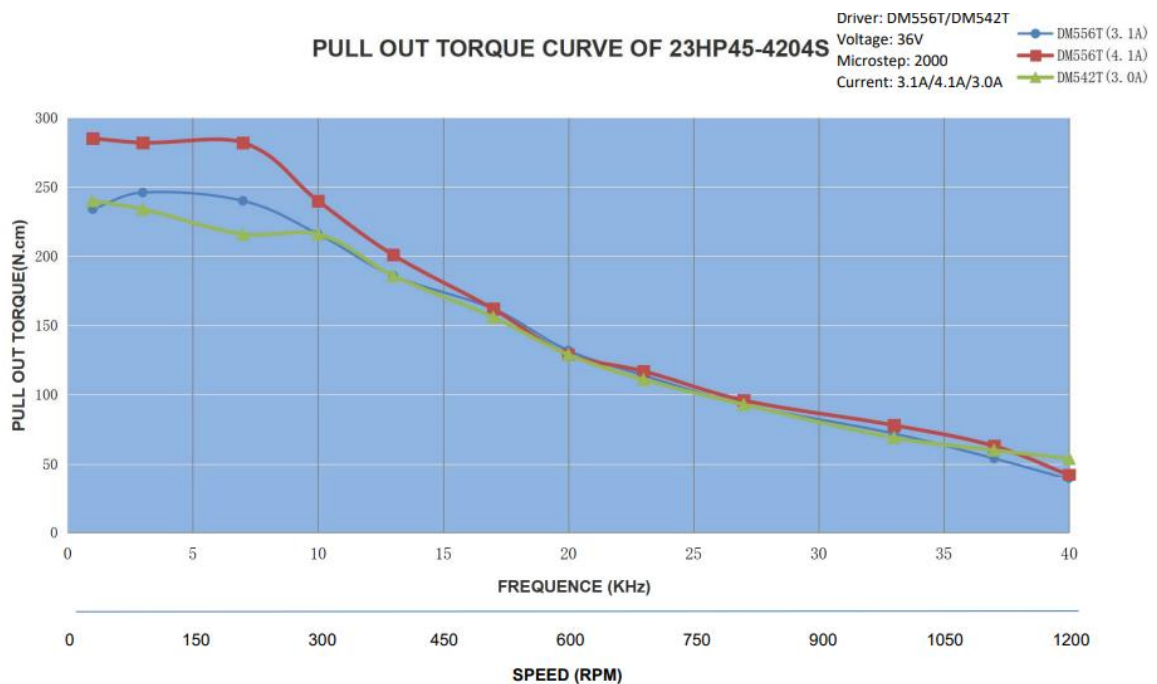


Gráfico 2-3: Curva del par motor nema 23

Fuente: <https://bit.ly/3wdQDDB>

2.5.3. Selección de procesador

El ordenador portátil va a ser el componente fundamental para el funcionamiento del prototipo, ya que será el encargado del procesar, detectar y enviar datos a la plataforma móvil a través de una red wifi. La comunicación del ordenador con el router que posee el prototipo permitirá que el ordenador se encuentre inmóvil mientras el prototipo robótico se movilice a través del cultivo

experimental de fresa. Se cuentan con dos ordenadores portátiles a los cuales se les hará una comparación de sus características para constatar cuál de estos poseen la solución más aceptable, como se puede evidenciar en la siguiente tabla:

Tabla 10-3: Comparativa de características de ordenadores

CARACTERÍSTICA	HP PAVILION 14 NOTEBOOK	TOSHIBA SATELITE S55- B
Procesador	Intel(R) Core (TM) i-3230M	Intel(R) Core (TM) i7-4510u
Velocidad del procesador	2,6 GHz	2,6 GHz
Memoria RAM	8192 MB	12288 MB
Puertos USB	2 USB 3.0 SuperSpeed 1 universal Serial Bus (USB) 2.0	2 USB 3.0 SuperSpeed 1 universal Serial Bus (USB) 2.0
Chip Gráfico	Intel(R) HD Graphics Family	AMD Radeon R7 M260
Puertos de video	1 HDMI 1 VGA (15 pines)	1 HDMI
Tarjeta de red	LAN Ethernet 10/100BASE-T (conector RJ-45)	Integrated 10/100 Ethernet LAN – (Puerto RJ – 45)
Conectividad inalámbrica	WLAN 802.11b/g/n 1x1 Bluetooth	WLAN 802.11b/g/n 1x1
Pantalla	LED de alta definición BrightView de 14 pulgadas en diagonal (1366 x 768)	15,6 pulgadas LED Wide Screen 1366 x 768 P
Dimensiones	2,71 cm (alt. mín.) / 3,14 cm (alt. máx.) x 34,6 cm (largo) x 23,1 cm (ancho).	(An. x Pr. x Al. frontal / Al. posterior): 380 mm x 260 mm x 23.5mm.
Compatibilidad con cámaras Ip	Si	Si

Realizado por: Villacis, A. 2021

Una vez analizado y comparado las características de los ordenadores disponibles para la utilización en el prototipo robótico, el ordenador más apropiado conforme a los requerimientos es el TOSHIBA Satellite S55-B ya que posee un procesador más potente que el otro ordenador además una memoria RAM más amplia lo que permitirá que la velocidad de procesamiento de datos que se requiere sea

alta, lo que no se conseguiría del todo con el ordenador HP Pavilion 14 Notebook. A demás tiene compatibilidad con cámaras Ip, puertos externos USB de alta velocidad y que es compatible con la mayoría de los entornos de programación hace que esta opción sea la mejor. En la figura 10-3 se puede visualizar el aspecto del ordenador TOSHIBA Satellite S55-B.



Figura 10-3: Ordenador Toshiba Satellite S55-B
Fuente: (TOSHIBA America Information Systemns Inc, 2014)

2.5.4. Selección de microcontrolador

El microcontrolador seleccionado fue el Arduino Uno debido a sus características sencillas y de fácil control, bajo costo, pequeño, multifuncional y que tiene características acoplables en forma de módulos. A su vez posee un protocolo de comunicación SPI el cual es un sistema rápido lo cual es fundamental dado que se desea un procesamiento de imágenes con la mayor velocidad posible para el funcionamiento de la visión artificial.

Tabla 11-3: Especificaciones técnicas de Arduino Uno

ESPECIFICACIONES TÉCNICAS DE ARDUINO UNO	
Microcontrolador	ATMEGA328P
Voltaje de ingreso	7-12V
Voltaje de funcionamiento	5V
Pines digitales de entrada/salida	14 (6 pines para salidas PWM)
Pines PWM de entrada/salida	6
Pines de entrada analógica	6
Memoria Flash	32 KB (ATMEGA328P)
SRAM	2KB (ATMEGA328P)
EEPROM	1KB (ATMEGA328P)
Velocidad de reloj	16 MHz

Fuente: (ARDUINO, 2021)
Realizado por: Villacis, A. 2021

2.5.5. Shield Ethernet para Arduino

El shield ethernet W5100 permite que la placa Arduino que hemos seleccionado se conecta a una red. Esta shield se comunicará con el Arduino a través del protocolo ISP el cual asegura una comunicación veloz.

Tabla 12-3: Especificaciones técnicas Shield Ethernet W5100

ESPECIFICACIONES TECNICAS DE SHIELD ETHERNET W5100	
Chip Ethernet	Wiznet W5100
Velocidad Ethernet	10-100 Mb/s
Conector	RJ45
Protocolo de comunicación	SPI
Compatibilidad	Arduino Uno, Leonardo, Mega
Lector	MicroSD



Fuente: (ARDUINO, 2010)

Realizado: Villacis, A. 2021

2.5.6. Modulo relé de 2 canales

En la Figura 29-3 se muestra un módulo relé de dos canales CH que puede va a ser controlado por Arduino, este módulo requiere que el microcontrolador entregue 5V 20mA para cada pin.

Tabla 13-3: Especificaciones técnicas modulo relé de 2 canales

ESPECIFICACIONES TÉCNICAS	
Señal de control	TTL (5V)
Voltaje de funcionamiento	5V DC
Modulo Relé	SRD-05VDC-SL-C
Capacidad máxima	10A /250VCA -30VDC
Indicadores	LED de accionamiento



Realizado por: Villacis, A. 2021

2.5.7. Regulador de voltaje DC/DC Step Down LM2596

Este regulador ha sido seleccionado para cumplir con la entrega de un voltaje DC de salida inferior al voltaje de entrada. Este regulador es tipo conmutado reductor y se considera que tiene alta eficacia en la conversión de voltaje. Como punto positivo se tiene que es capaz de tolerar cargas de hasta 3A.

Tabla 14-3: Especificaciones técnicas de Step Down LM2596

ESPECIFICACIONES TÉCNICAS	
Voltaje de entrada	De 4,5 a 40V DC
Voltaje de salida	De 1,23 a 37V DC
Convertidor DC/DC	LM2596
Corriente de salida	Máximo 3A
Potencia de salida	25 W
Eficiencia de conversión	92%



Realizado por: Villacis, A. 2021

2.5.8. Baterías

Se ha seleccionado como fuente de alimentación baterías de gel debido a su bajo costo y tamaño, ya que otros tipos son muy costosos y poseen un tamaño y peso que no va de acuerdo con el diseño del prototipo. Se han elegido dos baterías de 12V con dos terminales un negativo y positivo.



Figura 11-3: Batería 12N7BL
Fuente: (GRUPO 86, 2021)

2.5.9. Router inalámbrico

Se ha seleccionado el router inalámbrico TL-WR840N debido a que posee una velocidad inalámbrica 300 Mbps lo cual es ideal para la transmisión de la información que necesita el prototipo. Este router posee una fácil configuración y uso. Se puede establecer seguridad a la red que proporcione de esta forma no se tienen interferencias o conexiones no deseadas.

A continuación, podemos observar el router inalámbrico.



Figura 12-3: Router inalámbrico TL-WR840N

Realizado por: Villacis, A. 2021

2.5.10. Selección de Cámara

La cámara será la encargada de adquirir las imágenes que serán procesadas por el ordenador portátil, para esto hay que considerar que el prototipo robótico se movilizara por el cultivo y el centro de procesamiento se encontrara fijo en un punto previamente definido dentro del cultivo experimental. La cámara deberá poder enviar imagen a través de una red inalámbrica que en este caso será wifi, también se debe enfatizar que la resolución debe ser la más alta posible.

En la siguiente tabla se establece una comparativa entre las cámaras que se encuentran comúnmente en el mercado.

Tabla 15-3: Comparativa de cámara Ip y analógica

COMPARATIVA PARA SELECCIÓN DE CAMARA		
Característica	Cámara IP	Cámara analógica
Resolución	Brindas resoluciones de 1,3 a 5 megapíxeles o (2560x1920)	Resolución estándar 720x480 píxeles (PAL), o lo que es 0,4 megapíxeles.
Calidad de video	Se puede ampliar o estrechar el campo de visión y cuenta con una excelente capacidad de acercamiento.	Calidad de acercamiento, generalmente granulada y degradada. No varía el campo de visión.
Inteligencia y gestión	Posee inteligencia de red y gestión remota.	No posee inteligencia de red, de difícil control remoto.
Instalación	Puede recibir alimentación, audio, video y señales de control a través de un solo cable.	Requiere cables individuales de alimentación, audio, video.
Wireless	Posee conexiones Wireless	No posee conexiones Wireless
Medios de transmisión	Transmite a través de cable de par trenzado, cable coaxial y conexiones inalámbricas. Manteniendo resolución de imagen.	Transmiten por cable coaxial, si usan cable de par trenzado esto con menor resolución
Costo	Costo un poco elevado.	Costos bajos.

Realizado por: Villacis, A. 2021

La cámara elegida después de hacer una comparativa entre las cámaras disponibles en el mercado, es la cámara HIKVISION DS-2CD2421G0-I(D)(W). Esta cámara es utilizada generalmente en sistemas de vigilancia y seguridad, en este caso debido a las características que posee, será adaptada para utilizarla en un sistema de visión artificial. Las características se pueden apreciar en la tabla que se muestra a continuación.

Tabla 16-3: Características de la cámara seleccionada.

CÁMARA	
Sensor de imagen	1/2,7 pulgadas Progressive Scan CMOS
Iluminación mínima	2 mm: Color: 0,01 Lux, B/N: 0 Lux con IR
Día y Noche	Filtro de corte IR
Video	
Máxima Resolución	1920 x 1080
Main Stream	50 Hz: 25 fps (1920 x 1080, 1280 x 720) 60 Hz: 30 fps (1920 x 1080, 1280 x 720)
Wi-Fi	
Protocolo Wi-Fi	-IW: 802.11b: CCK, QPSK, BPSK 802.11g/n: OFDM
Rango de frecuencia	-IW: 20/40 MHz
Radio de transferencia	-IW: 11b/g: 54 Mbps 11n: Hasta 150 Mbps
Rango Wireless	IW: Hasta 50 m

Realizado por: Villacis, A. 2021

2.5.11. Herramienta

Debido a la abundante cantidad de maleza existente en los surcos se seleccionó como la herramienta más adecuada a una desbrozadora eléctrica, debido a la alta velocidad de corte y que el radio de corte se puede modificar de tal forma que cubra el ancho del surco, esto sin causar daño a la lona plástica negra que cubre el sistema de riego por goteo que existe en las hileras del cultivo experimental de fresa.

Se elige una desbrozadora Truper modelo DES-440T con una de 450 W a causa de que es un modelo potente, económico y posee repuestos en la mayoría de sus partes lo cual facilita un eventual mantenimiento.

Esta herramienta ha sido modificada y adaptada al prototipo robótico de manera que mantiene su eficiencia de corte.

La desbrozadora posee las siguientes especificaciones técnicas.

Tabla 17-3: Especificaciones técnicas de la desbrozadora DES-440T

ESPECIFICACIONES TECNICAS	
Voltaje de funcionamiento	127 V CA
Frecuencia	60 Hz
Corriente	3,54A
Potencia	450 W
Velocidad	10300 r/min
Capacidad de corte	330 mm



Realizado por: Villacis, A. 2021

2.6. Placa de control AUTOPRO

Las conexiones entre los diferentes elementos electrónicos que son necesarios para el funcionamiento del prototipo robótico se realizaron por medio del diseño de una placa electrónica denominada Placa de control AUTOPRO. La placa diseñada está constituida por los siguientes elementos.

- Dos reguladores de voltaje DC/DC Step Down LM2596
- Placa Arduino Uno
- Modulo relé de 2 canales

2.6.1. Diagrama electrónico placa AUTOPRO

Estos elementos mencionados se encuentran conectados de tal manera que satisfagan las necesidades del prototipo robótico y a su vez el control de sus actuadores y la dotación de alimentación energética específica para cada elemento. En la Figura 13-3 y 14-3 se puede visualizar el diagrama electrónico de la Placa de control AUTOPRO.

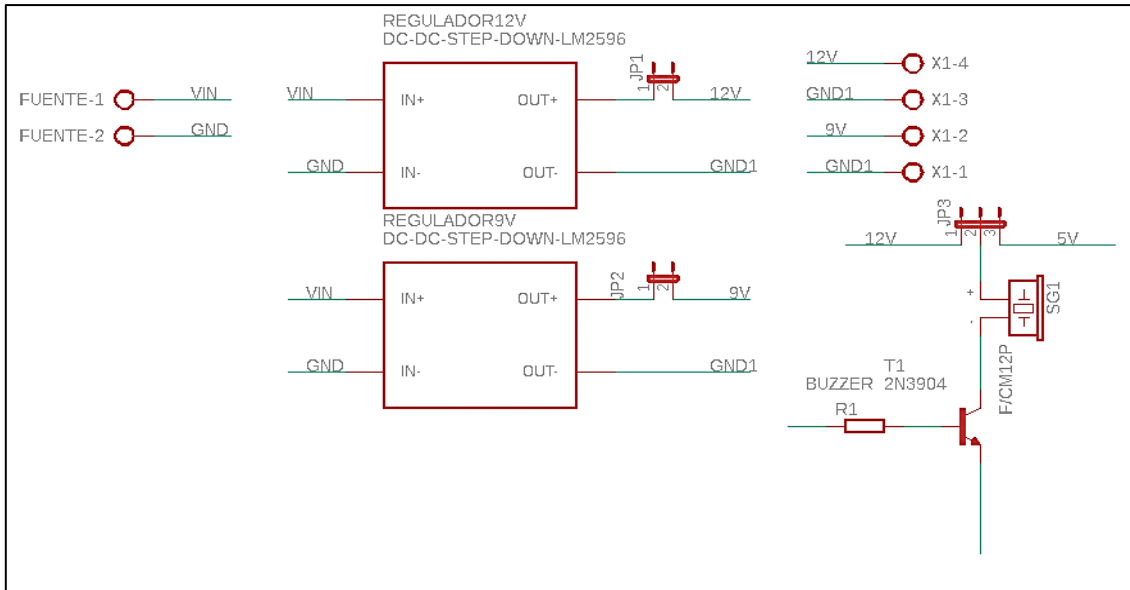


Figura 13-3: Diagrama electrónico Placa de control AUTOPRO, parte 1.
Realizado por: Villacis, A. 2021

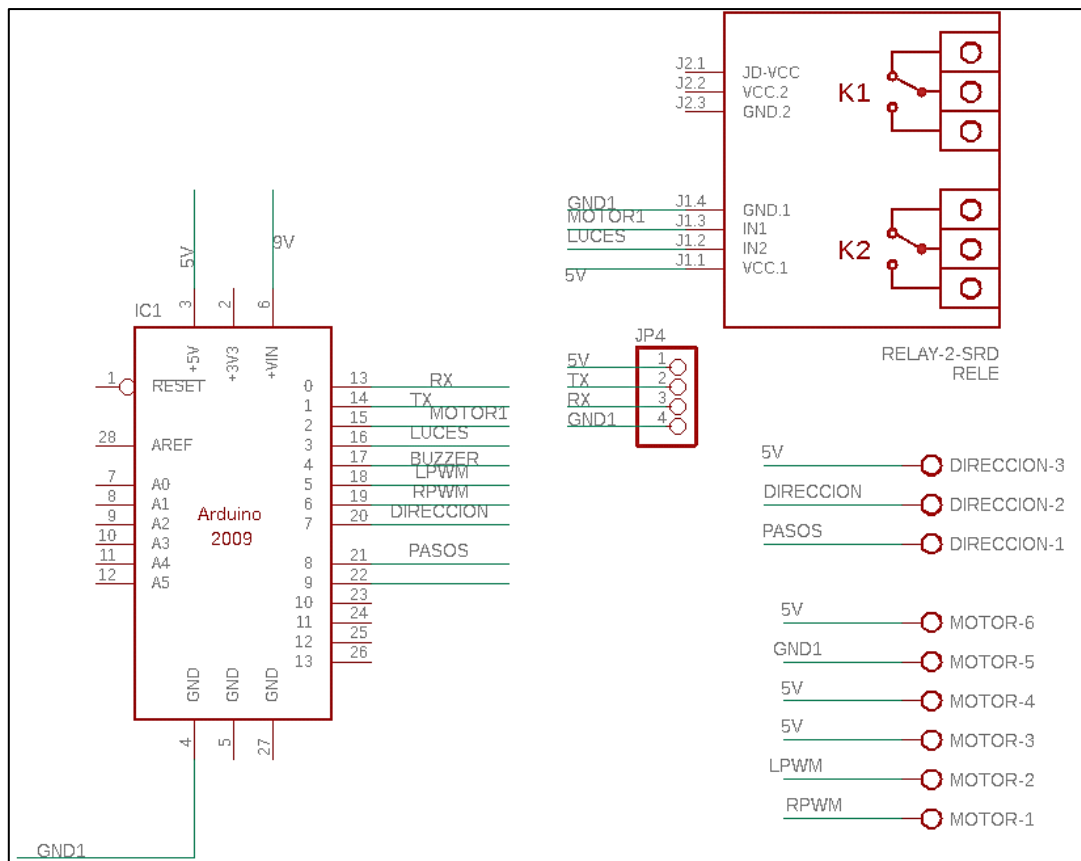


Figura 14-3: Diagrama de electrónico Placa de control AUTOPRO, parte 2.
Realizado por: Villacis, A. 2021

2.7. Selección de Software

A fin de que todos los elementos que componen el prototipo realicen una función en específico, es primordial que se cuente con el software necesario para programar estas funciones. Al contar con diferentes tipos de componentes se necesita contar con varias librerías en cada uno de los softwares de programación.

Se va a utilizar el lenguaje de programación Python para la creación del sistema de visión artificial, tanto para el control de la trayectoria como de la detección de la maleza. Para ello se utiliza la interfaz del software PyCharm Community Edition juntamente con el software libre OpenCV y librerías propias del lenguaje de programación Python.

Se cuenta con una placa Arduino Uno, la cual será programada a través de la IDE que posee este microcontrolador para su desarrollo y programación. Así como Python esta IDE de Arduino posee una gran cantidad de librerías que serán de utilidad.

2.8. Esquema de funcionamiento del prototipo

Una vez seleccionado tanto el hardware y software del prototipo robótico planteado en este trabajo de titulación, se concibe el esquema de funcionamiento autónomo del prototipo el cual será muestra en la Figura 15-3.

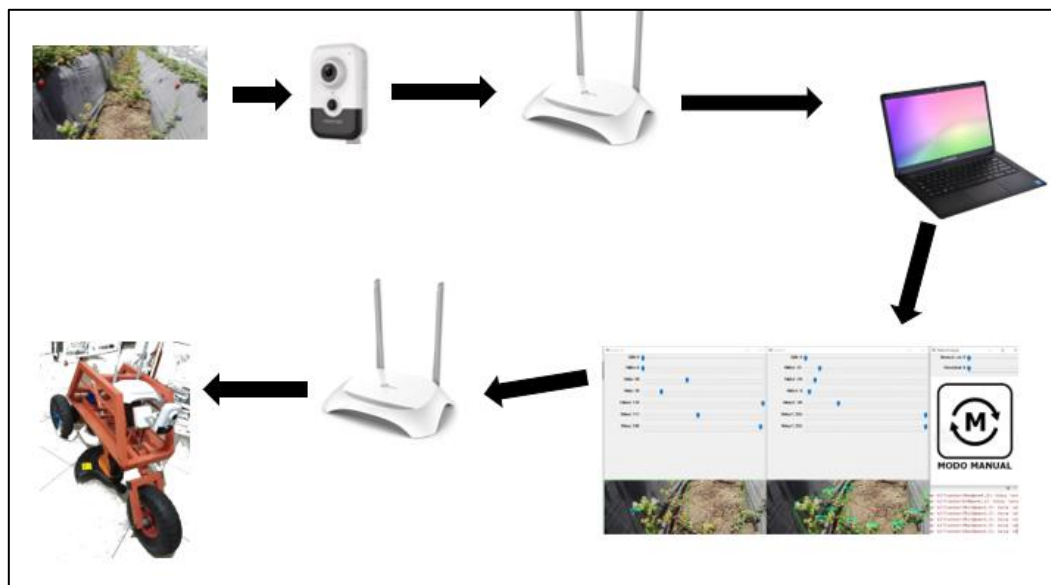


Figura 15-3: Esquema de funcionamiento del prototipo robótico
Realizado por: Villacis, A. 2021

En la Figura 15-3 podemos observar varias fases las cuales serán explicadas a continuación:

- Toma las imágenes del entorno de trabajo mediante la cámara Ip Hikvision.
- Mediante conexión ethernet se envía la imagen capturada al computador.
- Las imágenes llegan al ordenador para ser procesadas.
- Las imágenes procesadas y las ordenes son enviadas por protocolo UDP a la placa de control por medio del router.
- Las ordenes receptadas son decodificadas y aplicadas tanto a la velocidad de traslación como ángulo de giro.
- Si se detectara por medio de la visión artificial maleza se procede a activar la herramienta de corte.

2.9. Construcción y ensamble de prototipo robótico

2.9.1. Construcción de la estructura base

La construcción de la estructura base se realizó a partir de un tubo cuadrado de 1 in de acero ASTM A36. El tubo fue cortado según las medidas de los planos obtenidos en el modelado CAD. Posterior a esto fueron soldados con una soldadura por arco.



Figura 16-3: Construcción de la estructura base

Realizado por: Villacis, A.2021

2.9.2. Ensamble del sistema de tracción

Para esta parte de la construcción se realizó la elaboración del eje de la dirección, así como la estructura base del motor nema 23. A continuación se aprecia el eje de la dirección con la rueda unido a la estructura base, a su vez se ensambla el sistema de tracción del prototipo.



Figura 17-3: Estructura base con sistema de dirección y tren trasero.
Realizado por: Villacis, A. 2021

Se realizó la construcción de la base del motor nema 23 el cual cumple también como base de la cámara Ip. Se soldó con la estructura base del prototipo, también colocó el acople de unión del eje y el motor. Esto se aprecia en la siguiente imagen.

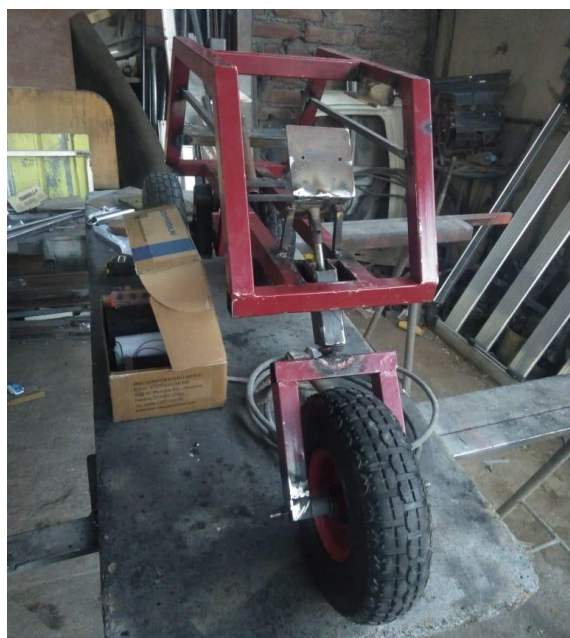


Figura 18-3: Base del motor Nema 23 y acople
Realizado por: Villacis, A. 2021

2.9.3. Ensamble de la herramienta de corte

La herramienta de corte desbrozadora Truper DES-440T, en el cual se recorte el tubo telescópico y su perilla de ajuste. Esto con el fin de que pueda acoplarse de buena manera en el prototipo. A demás su sistema de alimentación fue cambiando a una posición dentro del robot móvil. Se puede apreciar el acople de la herramienta de corte a continuación.



Figura 19-3: Acople de la herramienta de corte
Realizado por: Villacis, A. 2021

2.9.4. Pintura

El ambiente en el que se va a desenvolver el prototipo robótico es complejo dado que existe la presencia de humedad y agentes externos que pueden corroer la estructura metálica que este tiene. La elección del color va de acuerdo con el color usado usualmente por las herramientas agrícola ya que debe ser un color que se distinga del cultivo y permita ubicarlo en todo momento. Es así como en la Figura 20-3 se puede visualizar el prototipo pintado.



Figura 20-3: Pintura del prototipo
Realizado por: Villacis, A. 2021

2.9.5. Ensamble y conexiones de hardware

Una vez lista la estructura se procedió a realizar todas las conexiones del hardware dentro de la plataforma móvil del prototipo robótico para lo cual se tiene un esquema general de conexiones en el que se guio para la correcta instalación de cada uno de los componentes. El diagrama se podrá observar en el Anexo B.



Figura 21-3: Colocación de la Placa de Control AUTOPRO en el prototipo
Realizado por: Villacis, A. 2021

Los pines de conexión de Arduino Uno se pueden verificar en la siguiente tabla.

Tabla 18-3: Conexión de pines de Arduino Uno

CONEXIÓN DE PINES DE ARDUINO UNO		
Nº	Pin	Conexión
1	0	Comunicación serial PC
2	1	Comunicación serial PC
3	2	Pin conectado modulo relé de desbrozadora
4	3	Pin conectado modulo relé luces
5	4	Pin conectado buzzer
6	5	Pin conectado modulo Lpwm control motor
7	6	Pin conectado modulo Rpwm control motor
8	7	Pin conectado dirección modulo motor paso a paso
9	8	Pin conectado pasos modulo motor paso a paso
10	9	Pin modulo ethernet
11	10	Pin modulo ethernet
12	11	Pin modulo ethernet
13	12	Pin modulo ethernet
14	13	Pin modulo ethernet
15	14 o A0	Sin conectar
16	15 o A1	Sin conectar
17	16 o A2	Sin conectar
18	17 o A3	Sin conectar
19	18 o A4	Sin conectar
20	19 o A5	Sin conectar

Realizado: Villacis, A. 2021

2.9.6. Elaboración y colocación de tapas protectoras

Se modeló las tapas que aislarán el interior del prototipo de agentes externos que puedan ocasionar fallos o daños en los circuitos electrónicos y en los elementos que se encuentran allí. Se decidió la utilización de acrílico debido a ser un material resistente y liviano además de proporcionar una vista en todo momento del interior del prototipo.

El acrílico es cortado mediante la utilización de láser lo que asegura la precisión en el corte. En la Figura 22-3 se puede ver el proceso de corte del acrílico.

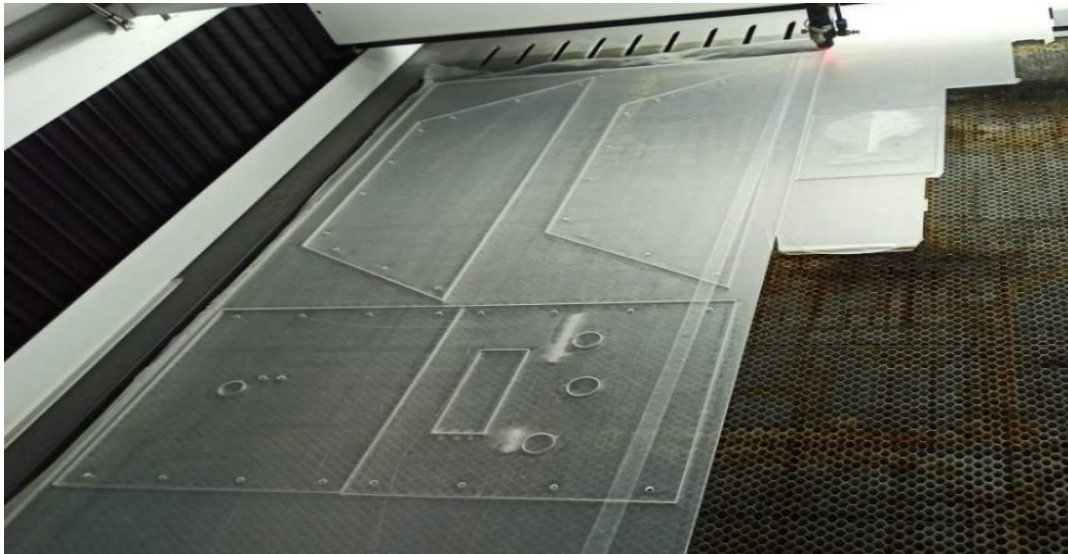


Figura 22-3: Cortes láser de acrílico de las tapas del prototipo

Realizado por: Villacis, A. 2021

Se aseguran las tapas a la estructura lo que se visualiza en la siguiente figura.



Figura 23-3: Colocación de tapa lateral derecha

Realizado por: Villacis, A. 2021

2.10. Sistema de visión artificial

Para la creación del sistema de visión artificial que satisfaga el objetivo propuesto en el presente proyecto, se establecerán varios parámetros y características que debe cumplir este sistema.

2.10.1. *Parámetros de programación del sistema.*

El software de visión artificial debe cumplir con los siguientes parámetros a considerar:

- Procesamiento y análisis de imágenes provistas por la cámara Ip.
- Utilización sencilla e intuitiva del lenguaje de programación en este caso del lenguaje Python.
- Especificar librerías de funciones utilizadas para el tratamiento de imágenes.
- El funcionamiento debe ser en tiempo real.

Para el inicio de la programación sistema de visión artificial instalamos Python a su última versión y también el software libre Open CV2, además también debemos tener el IDE de Arduino el cual nos permitirá realizar la programación de la placa la cual deberá estar comunicada todo el tiempo con el sistema diseñado.

2.10.2. *Adquisición de imágenes*

La encargada de esta función será la cámara Ip, la cual estará ubicada a una altura de 50 cm sobre el nivel del suelo del cultivo, lo que le permite tener un campo de visión adecuado del surco.

Al estar en un entorno no controlado la adquisición de la imagen variará dependiendo diversos factores como los es la iluminación, la hora del día, el clima y el movimiento que tendrá el prototipo al no estar en un suelo regular. Por lo que se debe idear un sistema que solviente esta variación para que el prototipo realice sus funciones de manera continua.



Figura 24-3: Imagen captada por la cámara Ip.

Realizado por: Villacis, A. 2021

2.10.3. Principio de funcionamiento del sistema de visión artificial

Como se visualiza en la Figura 25-3, el prototipo capta la imagen del terreno, esta imagen es amplia y en ella se puede observar gran parte del surco, así como la lona de plástico negro que cubre las hileras. Se observa también la maleza existente. Se aprecia de forma clara que, debido a la altura de las hileras, la planta de fresa no está en contacto con la maleza; lo que facilita la caracterización de la maleza; esto también produce que se facilite la detección de la maleza y a de la misma forma la labor de la herramienta, que podrá cortar y eliminar la plantas cada vez que la visión detecte el color verde. A su vez para el control de la trayectoria se tomó como base las lonas de color negro existentes a los lados del surco para que a través de la programación y la visión el prototipo se mantenga en el centro del surco mientras se moviliza.

De la imagen captada por la cámara se toma una parte determinada y se la sustrae. Esto lo podemos observar en la siguiente figura.



Figura 25-3: Delimitación de imagen.

Realizado por: Villacis, A. 2021

La imagen sustraída se va a analizar en dos partes distintas. Por programación se abrirán dos paneles uno para la detección del color negro y otro panel para el color verde. Estos dos paneles están compuestos exactamente por los mismos componentes los que son, 6 deslizadores que controlan el modelo de color HSV, un activador de imagen para visualizar la imagen a color o a su vez la imagen con la máscara creada por la selección de color de los deslizadores.

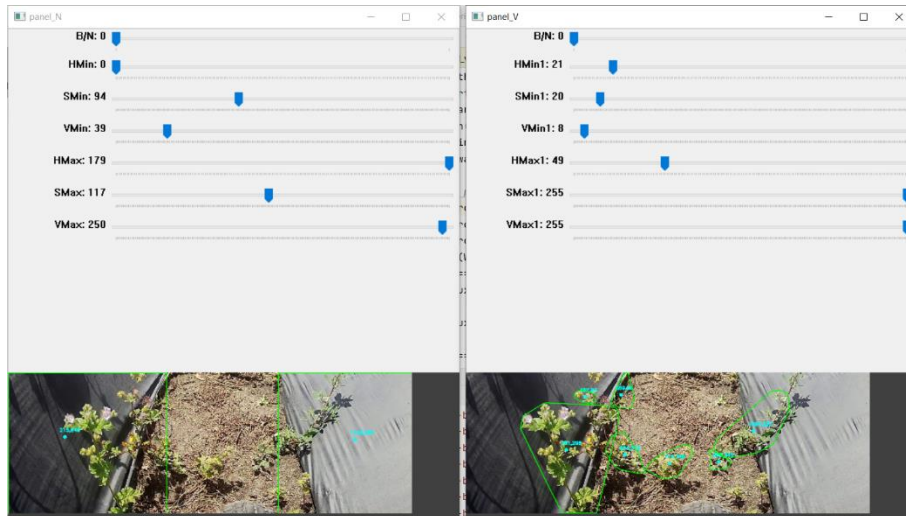


Figura 26-3: Paneles de control HSV de color negro y verde.
Realizado por: Villacis, A. 2021

Como ya se mencionó en el marco teórico, el modelo HSV es una transformación no lineal del modelo RGB en coordenadas cilíndricas, de manera que, cada color viene definido por las siguientes dimensiones:

- Tinte o matiz: El ángulo que representa el tinte por lo general está definido entre 0 y 360°, pero debido a que utilizamos el lenguaje de programación Python este valor cambia, en esta plataforma el matiz se encuentra entre los 0 y 180°.
- Saturación: En esta dimensión el nivel de saturación del color estará dado entre 0 y 1, entonces se establece que 0 representará que la imagen esta sin saturación alguna (blanco) y 1 sería el matiz en toda su intensidad. Es común manejar estos valores en datos percentiles es decir de 0% a 100%. En Python estos valores están representados entre 0 y 255.
- Brillo: Para el brillo tenemos los siguientes datos que son entre 0 y 1. Al igual que sucedía con la saturación puede darse en percentiles entre 0% y 100%. De esta forma el 50% indicara el nivel medio o normal del brillo del color. De igual manera en Python estos valores fluctúan ente 0 y 255.

Como se puede apreciar en los paneles ya mencionados, se tienen los rangos de HSV mínimo y máximo. Por lo que si se extraen se obtiene una mascara en blanco y negro. En este aspecto debido a que la resolución de la cámara es de muy buena calidad, no es necesario aplicar un filtro antes del análisis de estas imágenes, sino más bien, se lo realizara después del análisis para limpiar de cualquier ruido a la imagen.

La resultante de la extracción mencionada es la figura mostrada a continuación.



Figura 27-3: Imagen resultante de la extracción de HSV

Realizado por: Villacis, A. 2021

A la imagen resultante se le aplicara un filtro, como ya es conocido existen un sin número de filtros, desde filtros sencillos como lo es un filtro mediano hasta filtros de considerable complejidad. En este caso se aplicará un filtra gaussiano. Al aplicar este filtro obtenemos lo siguiente.

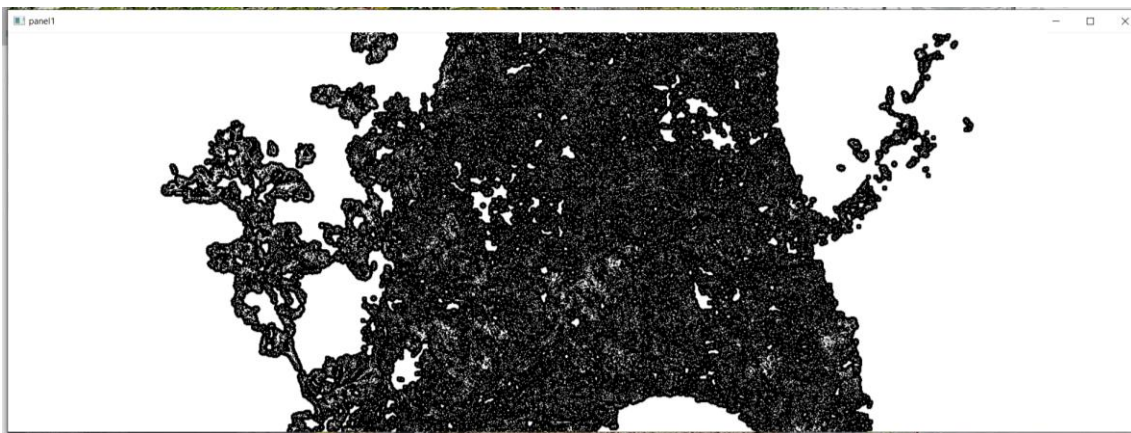


Figura 28-3: Imagen aplicada filtro gaussiano

Realizado por: Villacis, A. 2021

A la imagen de la Figura 28-3 se le aplica una búsqueda de contornos el cual nos dará un vector con la información de todos los contornos de dicha imagen por ello solo delimita a los dos contornos de mayor dimensión.

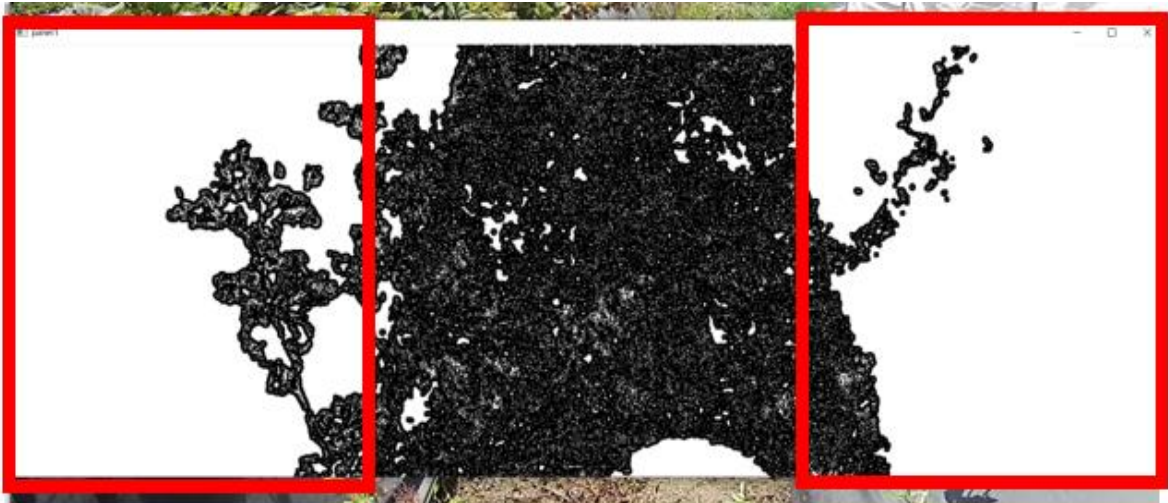


Figura 29-3: Selección de contornos

Realizado por: Villacis, A. 2021

Se puede apreciar en la imagen los contornos de mayor dimensión, estos tienen un tamaño aproximado de 70000 píxeles. Entonces el resultado obtenido aplicando lo ya mencionado se muestra a continuación.



Figura 30-3: Resultado de la búsqueda de contornos

Realizado por: Villacis, A. 2021

Como se observa en la imagen, el sistema detecta la lona de plástico debido a la búsqueda de contornos, esto se usará para el control de la trayectoria del prototipo robótico.

Para la detección de maleza se usará una búsqueda de color en este caso el color verde. Aplicando esta búsqueda obtenemos lo siguiente.

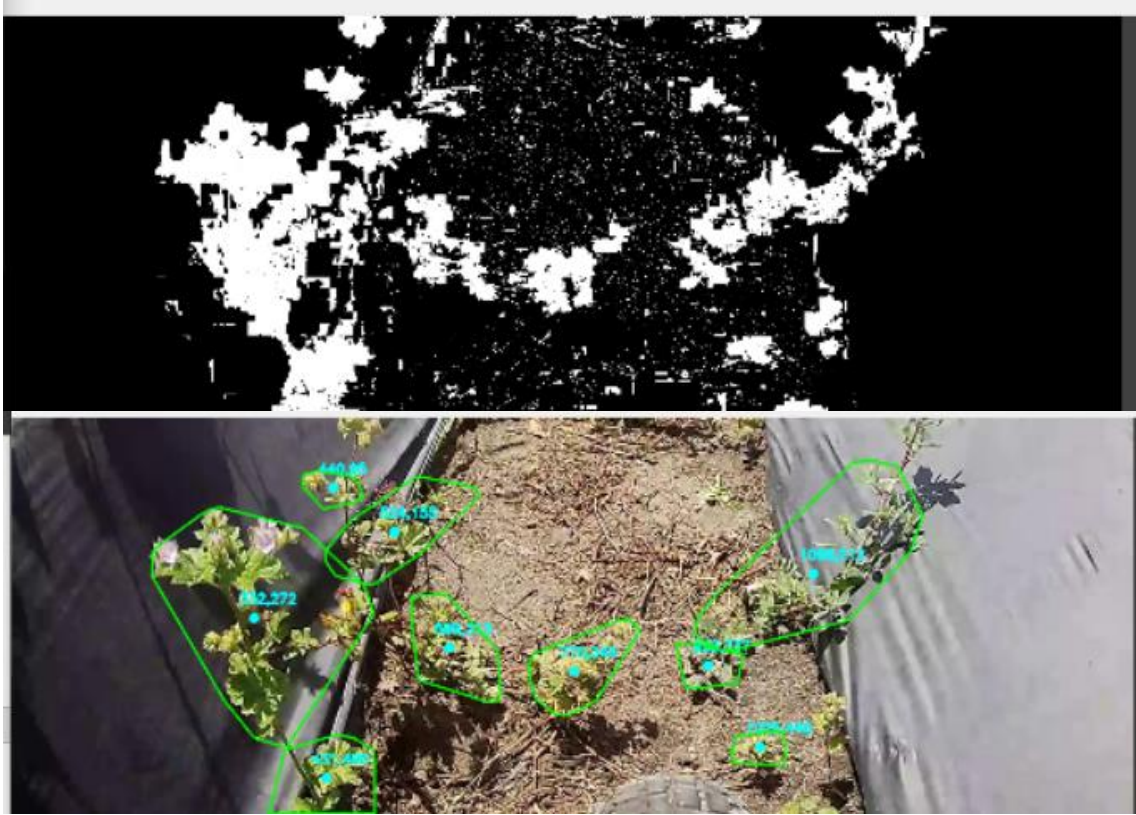


Figura 31-3: Detección de color verde.
Realizado por: Villacis, A. 2021

Una vez realizados estos procesos el sistema procede a calcular el camino encontrado entre las lonas plásticas negras, dibujando así un rectángulo como se aprecia en la siguiente figura.



Figura 32-3: Cálculo de la trayectoria del prototipo
Realizado por: Villacis, A. 2021

Además, se dibuja el centro de la pantalla en color rojo y un punto de color celeste. Este punto es un indicador de la posición de la dirección del prototipo el cual se moverá de tal manera que el punto quede entre el rectángulo de color rojo guiando de esta manera al robot por la mitad del camino.

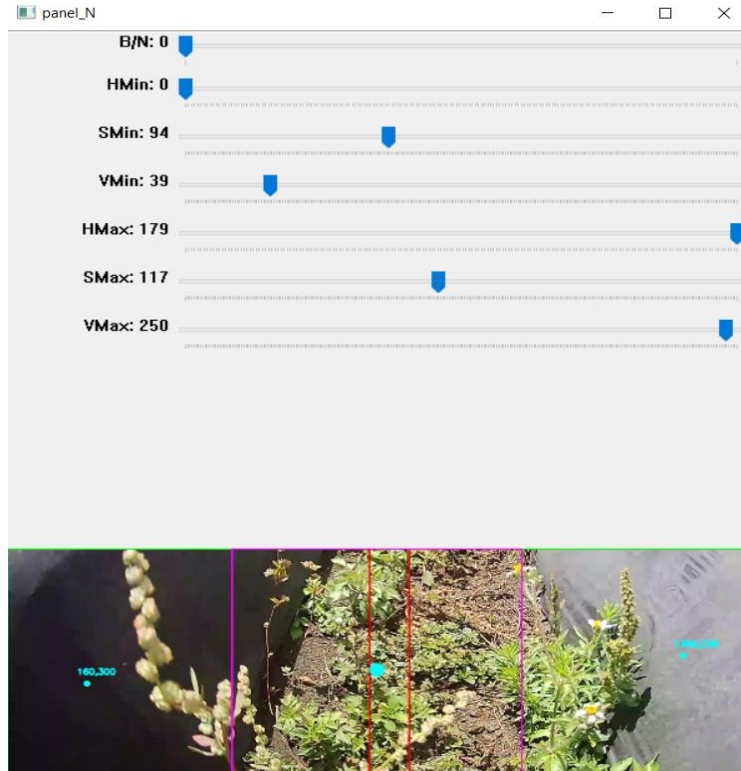


Figura 33-3: Deslizadores del panel color negro
 Realizado por: Villacis, A. 2021

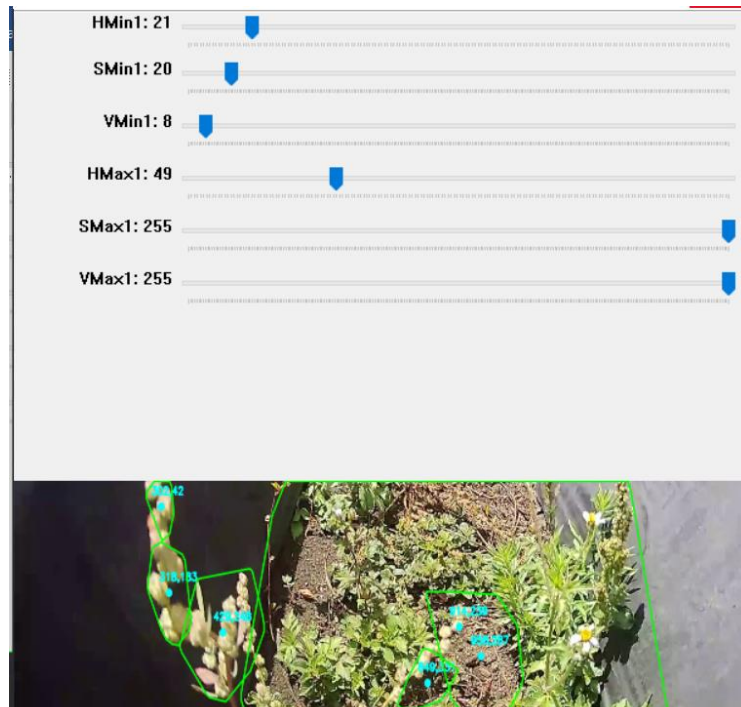


Figura 34-3: Deslizadores panel color verde
 Realizador por: Villacis, A. 2021

La decisión de diseñar paneles con deslizadores que puedan ser modificados mientras esté en funcionamiento el prototipo, es debido a que al encontrarse en un ambiente no controlado, la variación de la iluminación a distintas horas del día u ocasionada por el clima ocasionarían que pueda existir un fallo en la detección de la trayectoria como de la maleza si los valores de HSV fuesen fijos. Por ende, estos valores pueden ser modificados y calibrados con el fin de que siempre se tenga la mejor calidad de imagen posible para el procesamiento y detección, mitigando de cierta forma la incertidumbre existente en este ambiente no controlado.

2.11. Modo Manual y Automático

El prototipo robótico consta de dos modos de operación los cuales serán explicados a continuación:

- **Modo Manual:** En este modo el prototipo robótico es controlado totalmente por el operador a través del teclado del ordenador. En este modo el operario debe definir tanto la velocidad como la trayectoria del prototipo y cuando encender la herramienta. La finalidad de este modo manual es que el prototipo pueda ser usado fuera de los surcos ya que existen malezas que afectan al cultivo de fresa que están a los alrededores del terreno.
- **Modo Automático:** En este modo el operario solo debe realizar la calibración inicial del rango HSV y la velocidad del prototipo, la programación vinculada con la visión artificial será la encargada de definir la trayectoria, detectar la maleza y eliminarla.

2.12. Diagrama de flujo del prototipo robótico

Los diagramas presentados a continuación describen los pasos que realizan los diferentes en el prototipo robótico para su funcionamiento.

El diagrama de flujo presentado a continuación es el que gobierna todo el prototipo robótico, de este se desprenden los diagramas de flujo de subprocesos que serán evidenciados más adelante.

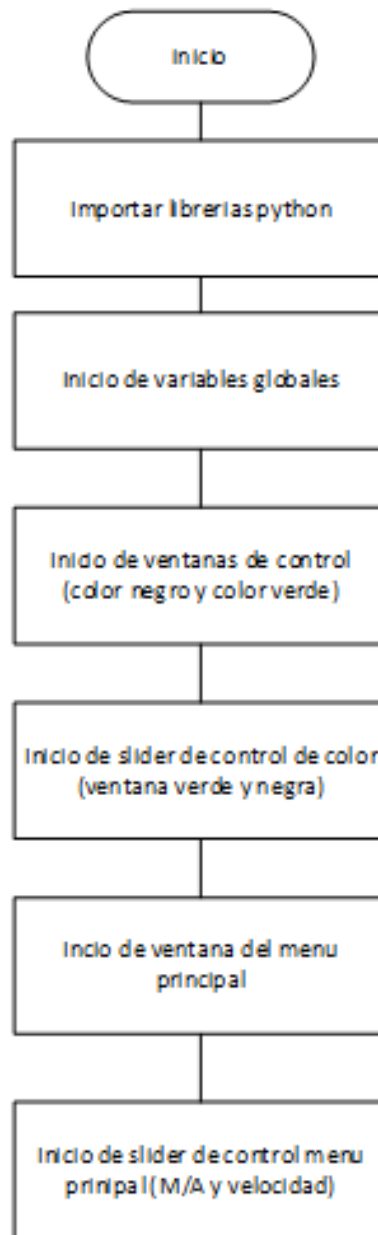


Figura 35-3: Diagrama principal, parte 1

Realizado por: Villacis, A. 2021

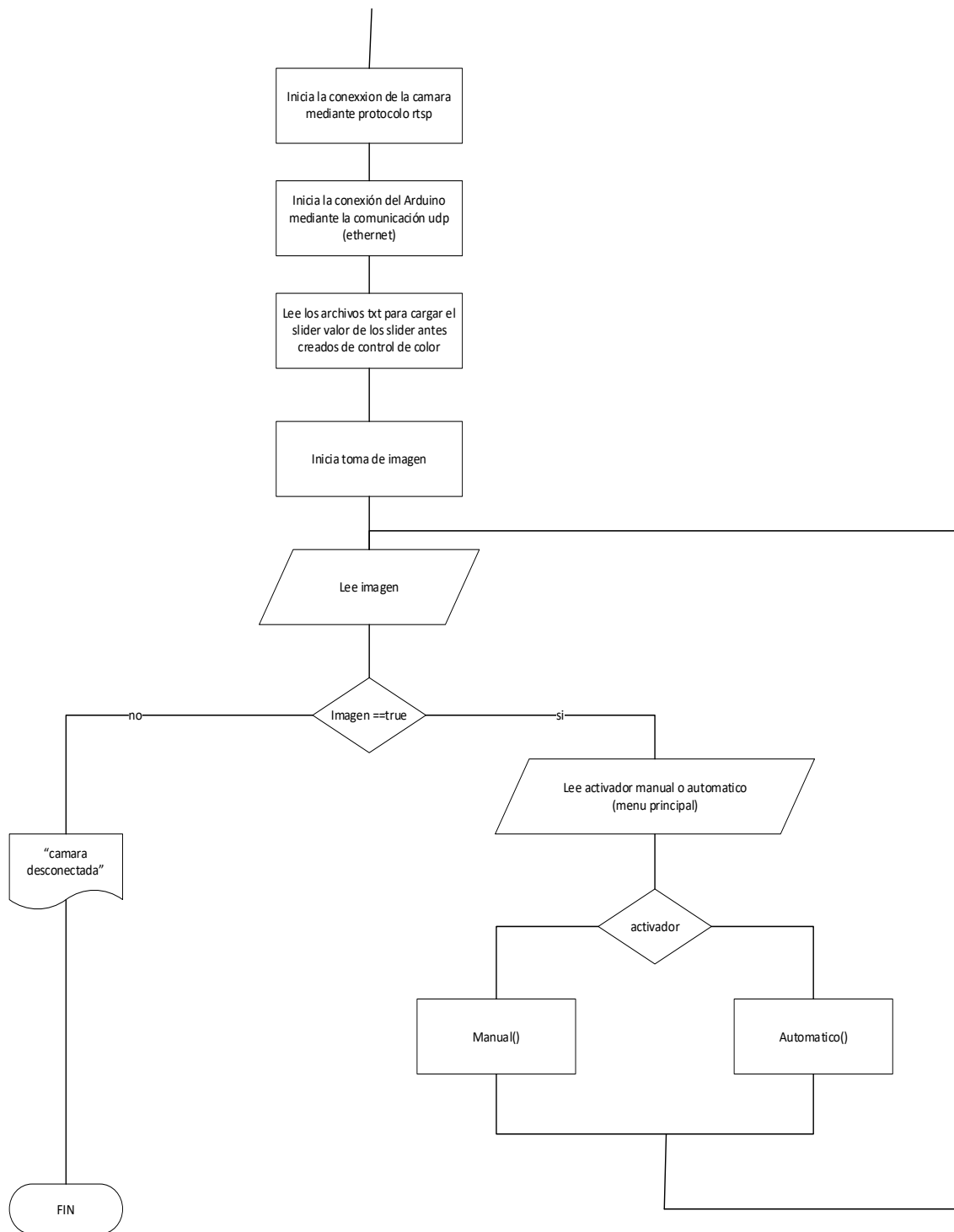


Figura 36-3: Diagrama principal, parte 2
 Realizado por: Villacis, A. 2021

En el Anexo C encontraremos los diagramas de flujo del modo manual y Arduino.

2.12.1. Diagrama de flujo del modo automático del prototipo

A continuación, se puede observar el diagrama los pasos que realiza el modo automático para dar funcionamiento al prototipo robótico.

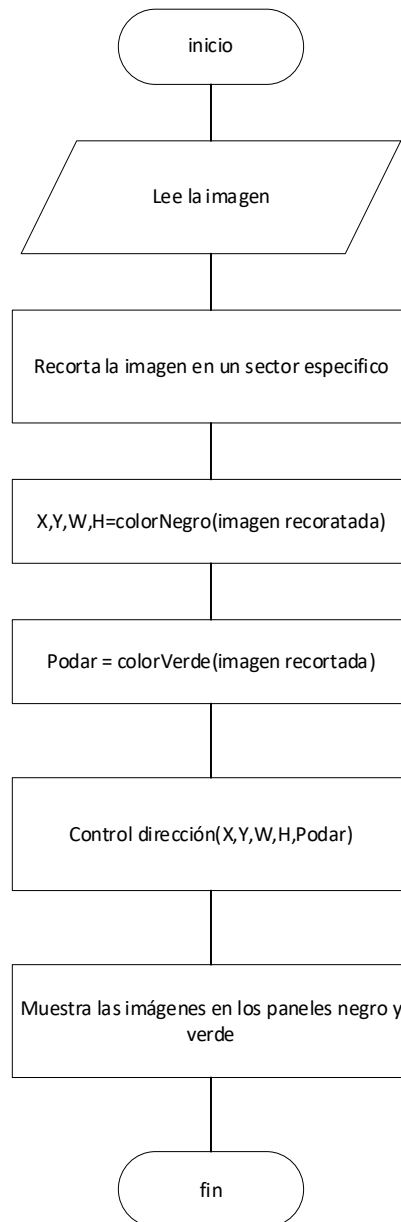


Figura 37-3: Diagrama de flujo del modo automático.
Realizado por: Villacis, A. 2021

De este modo automático se desprenden tres subprocesos los cuales son el procesamiento del color negro, color verde y el control de dirección.

2.12.1.1. Diagrama de flujo del procesamiento del color negro

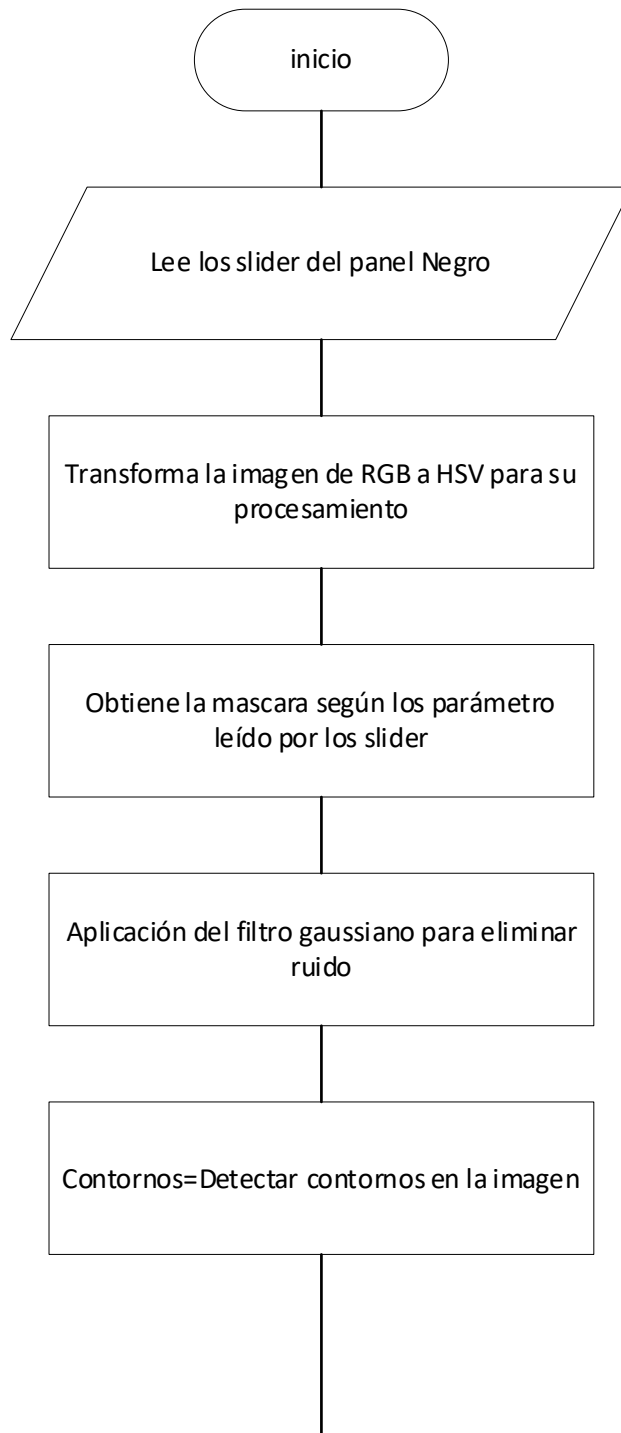


Figura 38-3: Diagrama del procesamiento del color negro, parte 1
Realizado por: Villacis, A. 2021

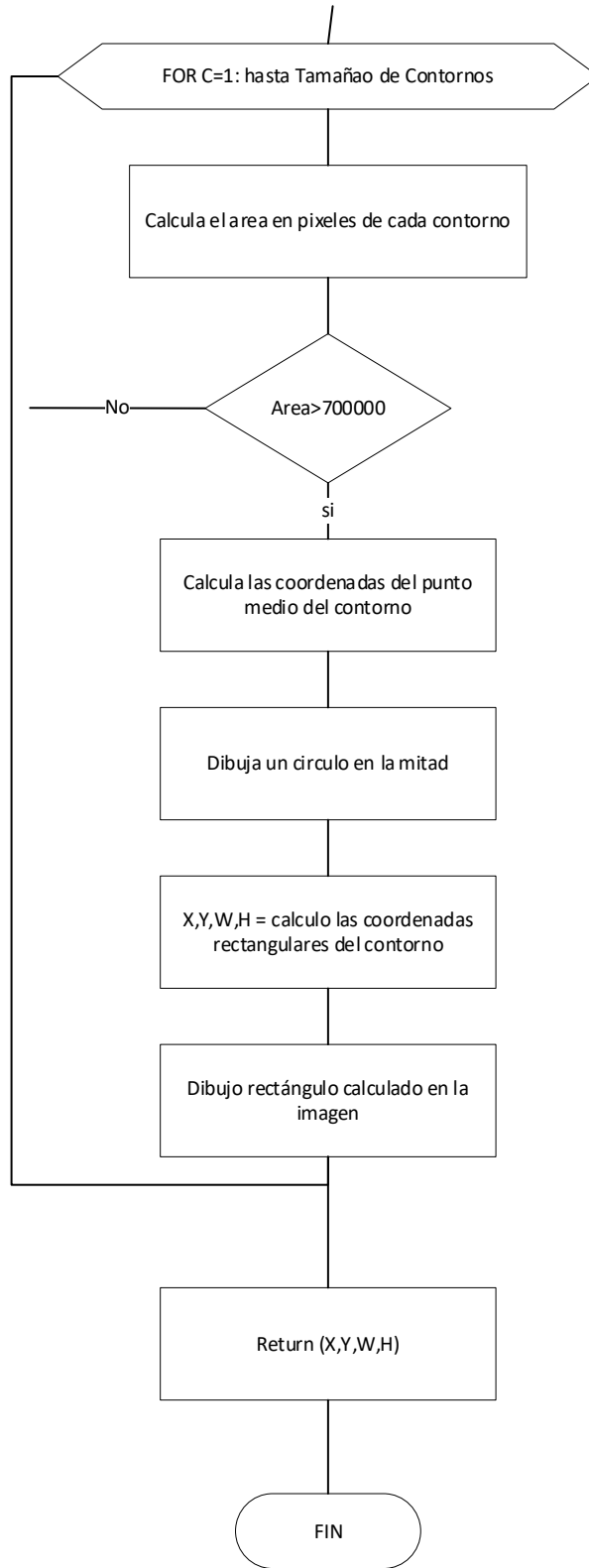


Figura 39-3: Diagrama del procesamiento del color negro, parte 2
 Realizado por: Villacis, A. 2021

2.12.1.2. Diagrama de flujo del procesamiento del color verde

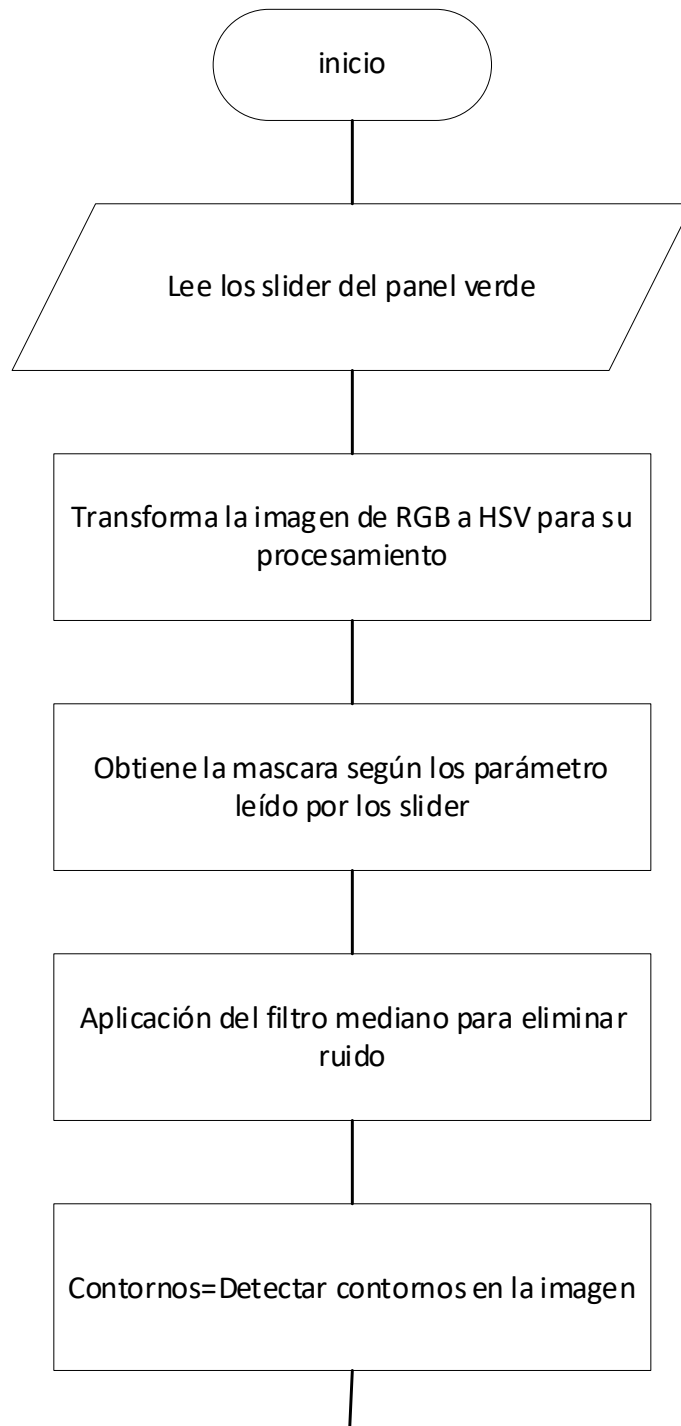


Figura 40-3: Diagrama del procesamiento del color verde, parte 1
Realizado por: Villacis, A. 2021

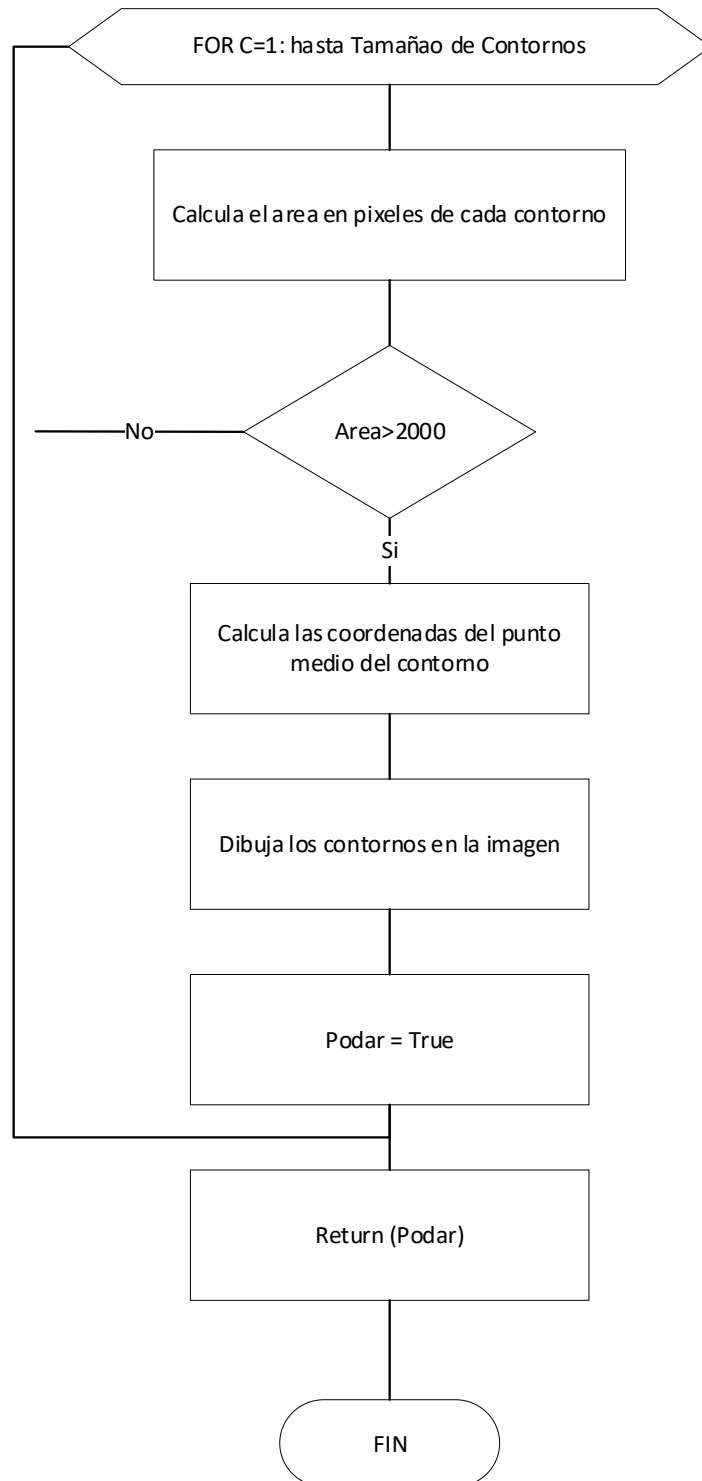


Figura 41-3: Diagrama del procesamiento del color verde, parte 2
Realizado por: Villacis, A. 2021

2.12.1.3. Diagrama de flujo del control de la dirección

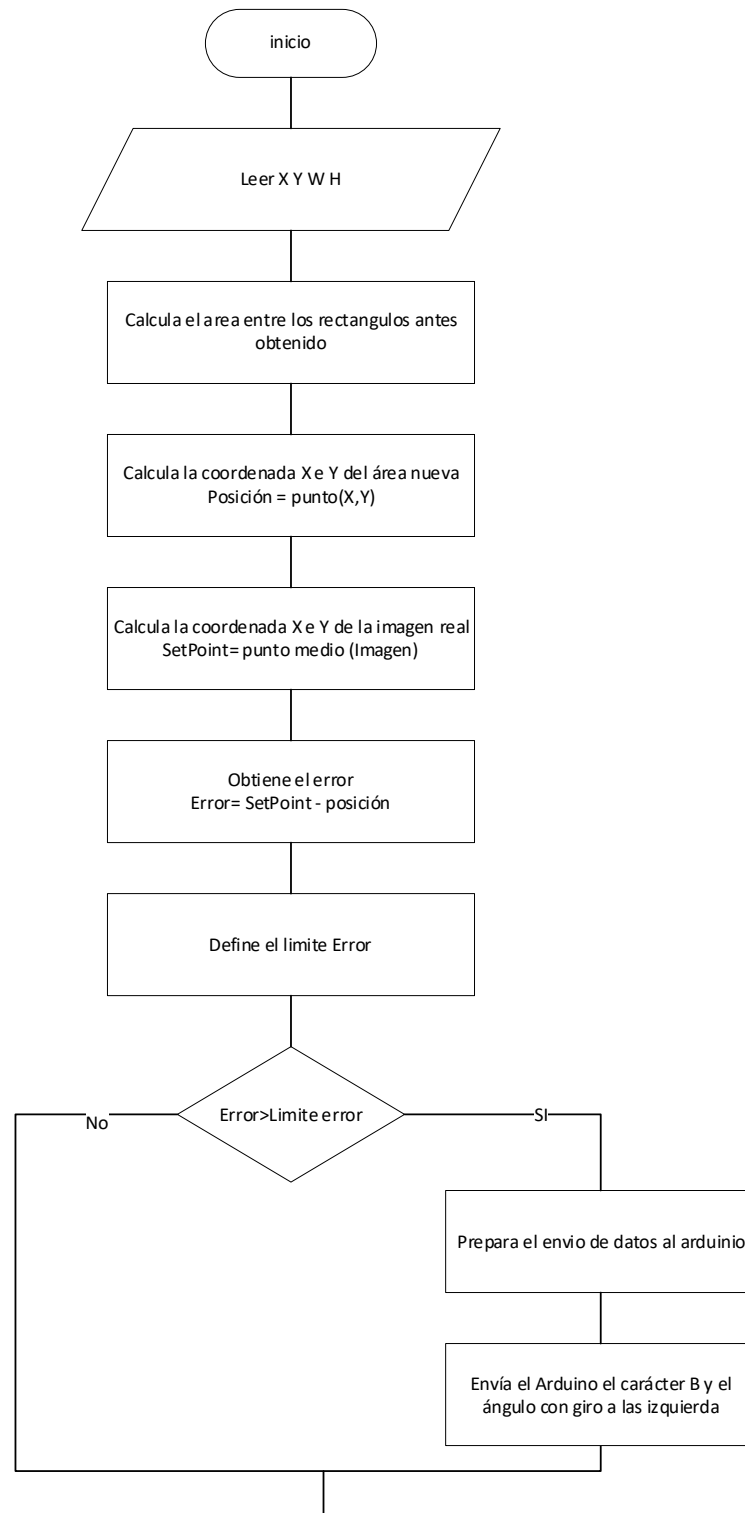


Figura 42-3: Diagrama de flujo del control de la dirección, parte 1
Realizado por: Villacis, A. 2021

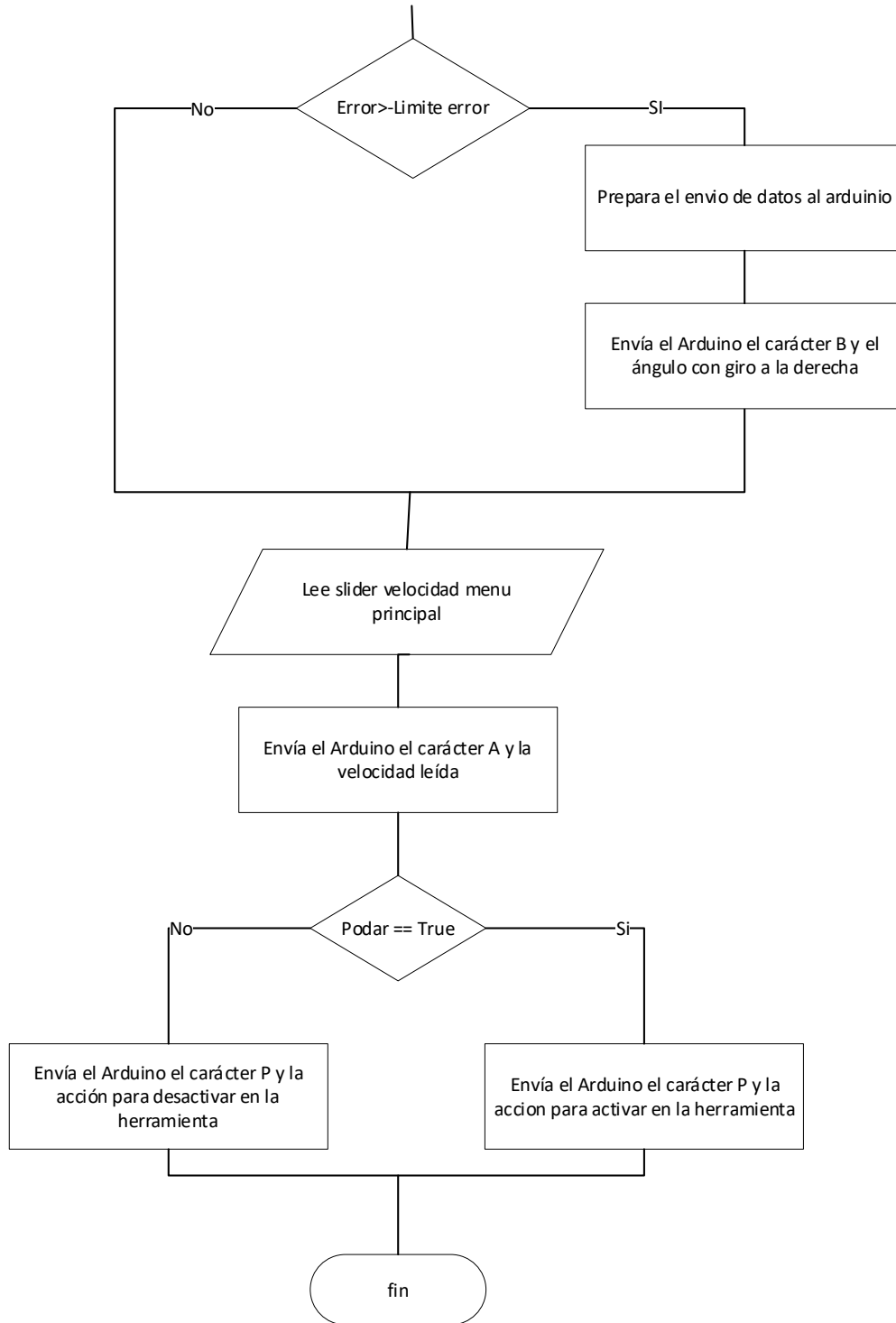


Figura 43-3: Diagrama de flujo del control de la dirección, parte 2
 Realizado por: Villacis, A. 2021

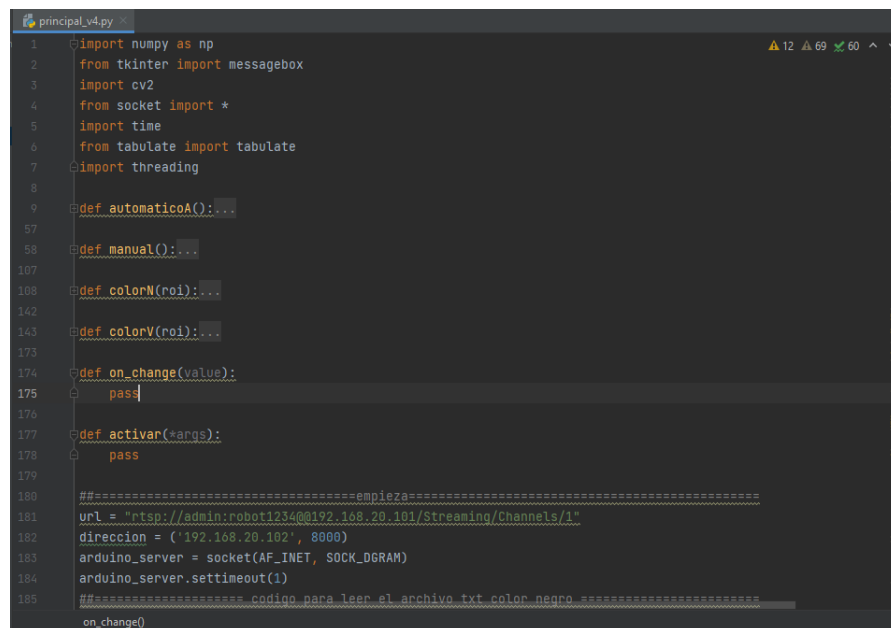
2.13. Programación en Python

La programación que se efectuó con el lenguaje Python se realizó a través de la IDE de PyCharm el cual es un entorno muy favorable y potente para desarrollar códigos de programación como el realizado para este trabajo de titulación. Para la utilización de Python se añadieron librerías que eran necesarias para satisfacer las necesidades del código.

Fueron utilizadas las siguientes librerías:

- **Numpy:** Provee soporte para la creación de vectores y matrices multidimensionales
- **Tkinter:** Es una librería que proporciona una interfaz gráfica al usuario de fácil programación.
- **Cv2:** con esta librería se realiza la visión artificial puede detectar movimientos, reconstruir objetos 3D, reconocer objetos, etcétera.
- **Socket:** librería utilizada para la comunicación ethernet.
- **Tabulate:** Imprimir datos de manera ordenada.
- **Threading:** Permite ejecutar varios hilos de programación al mismo tiempo, conocida como programación en paralelo.

En la Figura 44-3 se puede ver una parte del código de programación realizado en Python para el prototipo robótico, el código completo se encontrará en el Anexo D.



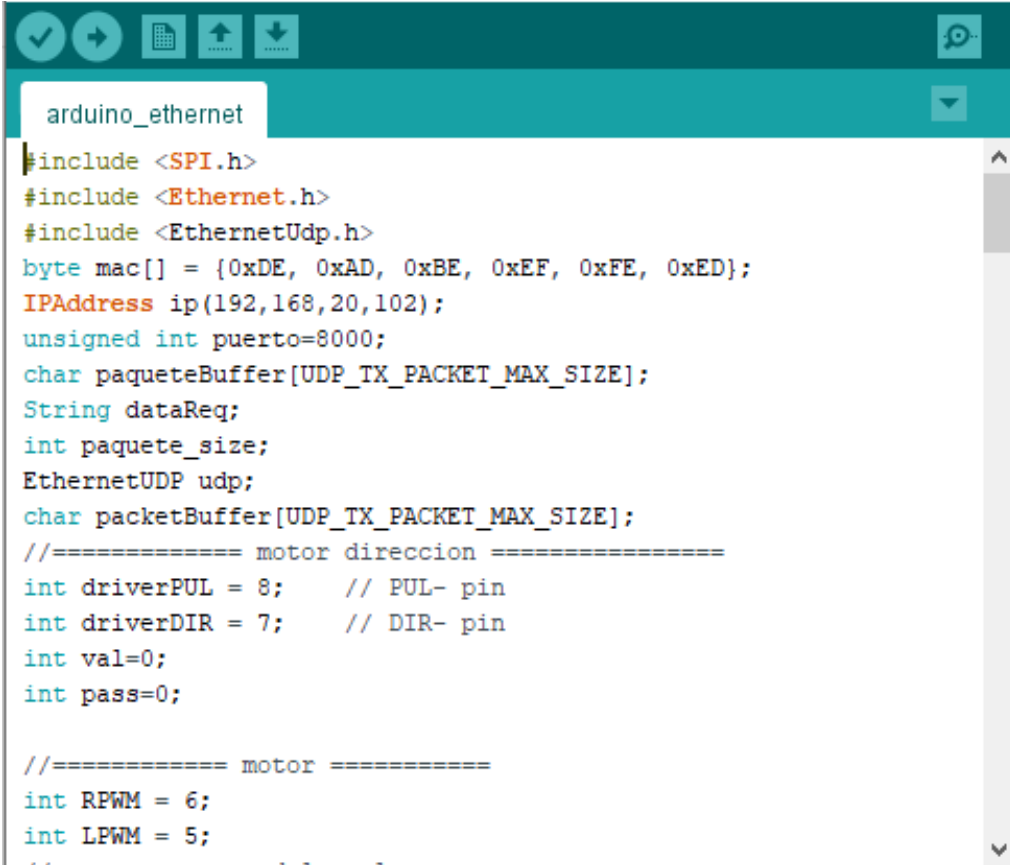
```
principal_v4.py
1 import numpy as np
2 from tkinter import messagebox
3 import cv2
4 from socket import *
5 import time
6 from tabulate import tabulate
7 import threading
8
9 def automaticoA():...
57
58 def manual():...
107
108 def colorN(roi):...
142
143 def colorV(roi):...
173
174 def on_change(value):
175     pass
176
177 def activar(*args):
178     pass
179
180 ##### empieza #####
181 url = "rtsp://admin:robot1234@192.168.20.101/Streaming/Channels/1"
182 direccion = ("192.168.20.102", 8000)
183 arduino_server = socket(AF_INET, SOCK_DGRAM)
184 arduino_server.settimeout(1)
185 ##### codigo para leer el archivo txt color negro #####
on_change()
```

Figura 44-3: Código de programación en Python
Realizado por: Villacis, A. 2021

2.14. Programación de Arduino

El entorno de programación del IDE de Arduino es uno de los más sencillos que se pueden encontrar y esa facilidad permite realizar un sin número de proyectos en él. En este caso la placa Arduino ha sido el microcontrolador elegido para realizar muchas funciones fundamentales del prototipo robótico por lo que una correcta programación resulta fundamental.

El software de programación Arduino posee muchas librerías a su haber, en esta ocasión se usaron sobre todo librerías de comunicación. Estas librerías se observarán en la siguiente figura y parte del código de programación. El código completo se lo podrá observar completo en el Anexo E.



```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192,168,20,102);
unsigned int puerto=8000;
char paqueteBuffer[UDP_TX_PACKET_MAX_SIZE];
String dataReq;
int paquete_size;
EthernetUDP udp;
char packetBuffer[UDP_TX_PACKET_MAX_SIZE];
//===== motor direccion =====
int driverPUL = 8;    // PUL- pin
int driverDIR = 7;   // DIR- pin
int val=0;
int pass=0;

//===== motor =====
int RPWM = 6;
int LPWM = 5;
```

Figura 45-3: Programación Arduino UNO.
Realizado por: Villacis, A. 2021

2.15. Pruebas de funcionalidad

Las pruebas de funcionalidad de prototipo robótico se basaron en la capacidad del prototipo de moverse a través del surco guiado por la visión artificial y a su vez que cumpla con la detección y eliminación de la maleza. El prototipo estará en el cultivo experimental para la realización de las pruebas de funcionalidad

1. *Encendido del prototipo:* Se debe comprobar que las baterías se encuentren completamente cargadas para el encendido del prototipo.



Figura 46-3: Carga de las baterías
Realizado por: Villacis, A. 2021

Una vez realizado la carga se acciona el switch de encendido que energiza todos los componentes internos y se comprueba si la notificación luminosa provista por las tiras led se efectúa ya que este es el indicador de que el prototipo se inició correctamente.

Después de unos segundos iniciado el prototipo, se comprueba a través del ordenador que se genere la red wifi privada denominada “ROBOT” la cual servirá para la comunicación entre el prototipo y el ordenador. Esta red será generada por el router que contiene el prototipo.

A continuación, se procede a verificar que la conexión a la red “ROBOT” sea exitosa.

2. *Inspección del código:* En la IDE de PyCharm se verifica que el código a ejecutar no tenga errores de programación o conflictos que generen un fallo.

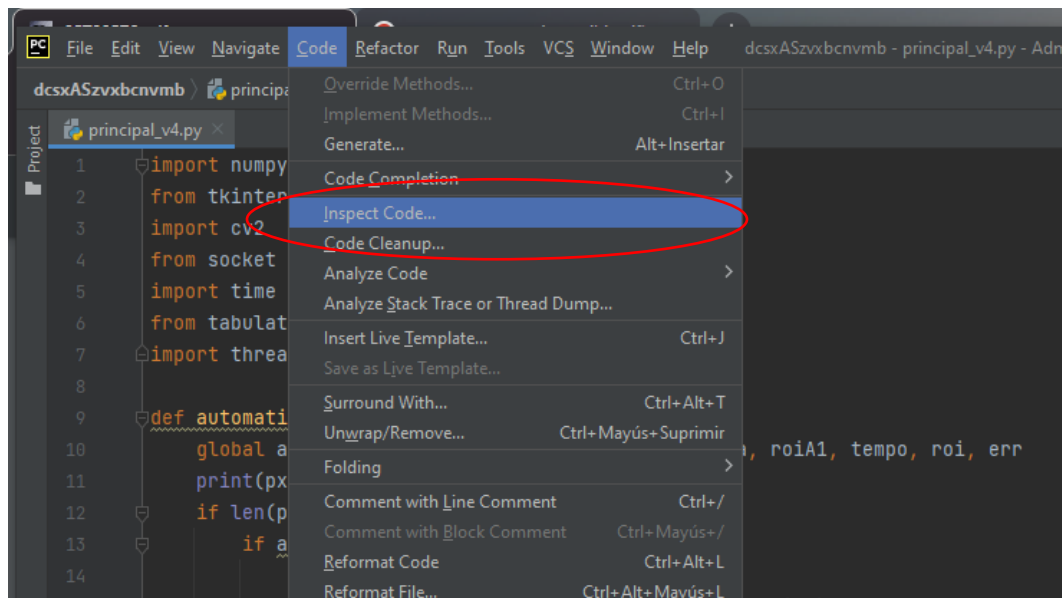


Figura 47-3: Inspección del código
Realizado por: Villacis, A. 2021

3. *Ejecución del código en Python:* Inspeccionado el código procedemos a ejecutarlo, realizado esto se mostrará la interfaz gráfica de la visión artificial, en el cual podremos calibrar los parámetros HSV y el modo de accionamiento.

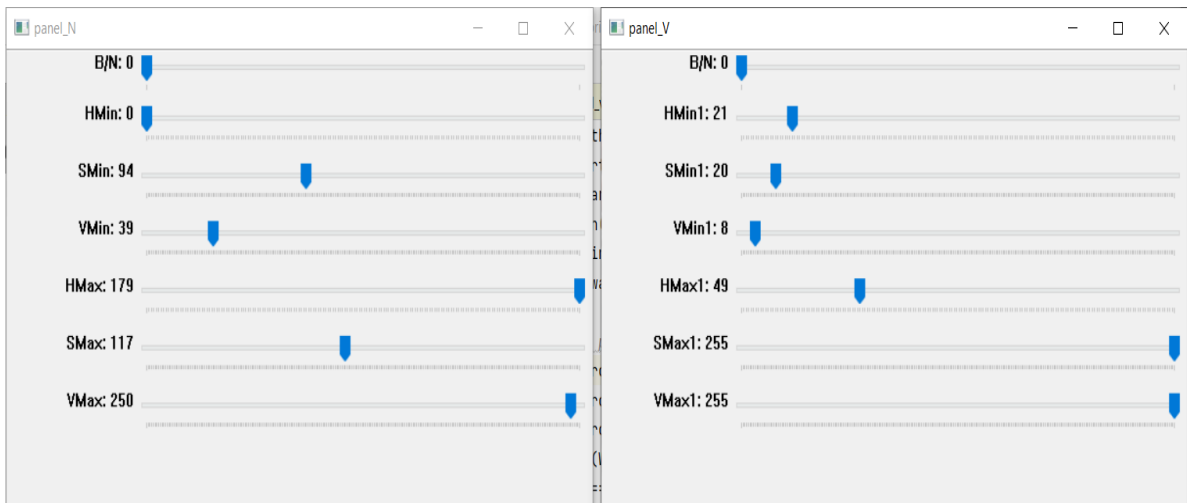


Figura 48-3: Interfaz gráfica de paneles de control blanco y verde
Realizado por: Villacis, A. 2021



Figura 49-3: Interfaz gráfica del modo de accionamiento
Realizado por: Villacis, A. 2021

4. *Ejecución modo automático:* Para ello se ajusta la velocidad del prototipo y se procede a iniciar a través del deslizador el modo automático. Se verifica si en la interfaz graficas de los paneles se visualice imagen en tiempo real de la detección tanto de la trayectoria como la maleza.

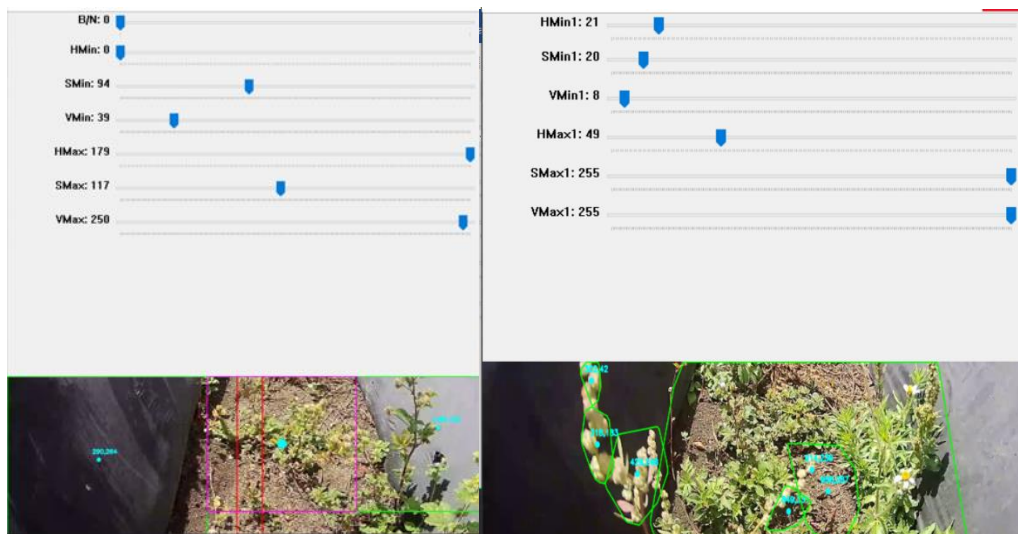


Figura 50-3: Imagen en tiempo real con la detección de trayectoria y maleza
Realizado por: Villacis, A. 2021

5. *Verificación del funcionamiento en campo del prototipo:* El procesamiento de imágenes detecta la maleza y controla la dirección del prototipo, se debe verificar visualmente que el prototipo no tenga inconvenientes en la ejecución de la trayectoria y que la herramienta se accione en los lugares en donde exista presencia de maleza. En la figura siguiente se puede visualizar al prototipo en el cultivo experimental de fresa en funcionamiento.



Figura 51-3: Prototipo robótico en funcionamiento en el cultivo experimental
Realizado por: Villacis, A. 2021



Figura 52-3: Prototipo robótico en funcionamiento
Realizado por: Villacis, A. 2021

6. *Verificación de funcionamiento de modo manual:* En este modo de funcionamiento se puede muestra en la siguiente interfaz gráfica la cual contine deslizadores que van a permitir controlar la velocidad del prototipo. Se debe comprobar que control de la dirección que se realiza por medio del teclado del ordenador con las letras w, a, s, d. De igual forma la activación y desactivación de la herramienta se la realiza con la letra p. Comprobar que al presionar las teclas realice las acciones definidas.



Figura 53-3: Interfaz gráfica del modo manual
Realizado por: Villacis, A. 2021

2.15.1. *Funcionamiento del prototipo*

2.15.1.1. *Paso por el surco*

Se realizaron pruebas con el prototipo en los diversos surcos existentes en el cultivo experimental, en el que se verifico la movilidad en la correcta trayectoria del surco y el corte de la maleza. Para la prueba se establecen 11 surcos de 12 existentes, debido a que en uno de ellos existe un árbol el cual obstruye todo el surco.

Para contemplar un paso exitoso por el surco el prototipo debió cruzar de inicio a fin en modo de funcionamiento automático, cortando la maleza existente y se deberá detener solo cuando llegue al final del surco.

Se realizaron un total de tres pasos por cada uno de los surcos. En la siguiente tabla se evidencian los resultados de funcionamiento del prototipo.

Tabla 19-3: Prueba de funcionamiento

PRUEBA DE FUNCIONAMIENTO DEL PROTOTIPO ROBOTICO					
Numero de surco	PASO POR EL SURCO			Cantidad completada	Cantidad incompletos
	1	2	3		
1	Completado	Completo	Completo	3	0
2	Completado	Completo	Incompleto	2	1
3	Completado	Incompleto	Completo	2	1
4	Completado	Completo	Completo	3	0
5	Completado	Incompleto	Completo	2	1
6	Completado	Completado	Completado	3	0
7	Completado	Completado	Completado	3	0
8	Completado	Completado	Completado	3	0
9	Completado	Incompleto	Incompleto	1	2
10	Completado	Completado	Completado	2	1
11	Completado	Completado	Completado	3	0
TOTAL				27	6

Realizado por: Villacis, A. 2021



Gráfico 3-3: Porcentaje de eficacia de paso por surco

Realizado por: Villacis, A. 2021

Según los datos obtenidos existe un porcentaje de pasos completados a través de los surcos del 81,82%, el porcentaje obtenido es alentador. Se analizó cada uno de los casos en los que el prototipo no pudo completar la ruta y se llegó a las causas que provocaron el fallo, las cuales son:

- Fallo propiciado por la reducción abrupta del ancho del surco impidiendo la circulación del prototipo, esta razón causó el 50% de los fallos ya que en el surco que se encuentra esta reducción el prototipo no pudo completar la ruta ni una sola vez.
- Dos de los fallos fueron producidos debido a la pérdida de tracción del prototipo. En ciertas partes se encuentran desprendimientos de la lona de plástico negra en su parte inferior, haciendo que la rueda trasera entre en contacto con la lona perdiendo tracción.
- El fallo restante se debió a la presencia de maleza alta y densa, esto causó el enredo de la herramienta con la maleza provocando el fallo.

A continuación, se mostrarán imágenes tomadas de los fallos.



Figura 54-3: Desplazamientos no completados.

Realizador por: Villacis, A. 2021



Figura 55-3: Colisión del prototipo con la hilera

Realizado por: Villacis, A. 2021

2.15.1.2. Detección y eliminación de la maleza.

Se realizó una inspección visual de los surcos en los que se contabilizaron la cantidad de maleza existente en cada uno de ellos para de esta manera realizar una comparativa con lo detectado por el sistema de visión artificial y la maleza eliminada con la herramienta. En las siguientes imágenes podemos observar a la maleza existente en el surco, como son detectadas con el sistema de visión artificial y como se aprecian en la interfaz de la detección del color verde.



Figura 56-3: Maleza por detectar.
Realizado por: Villacis, A. 2021

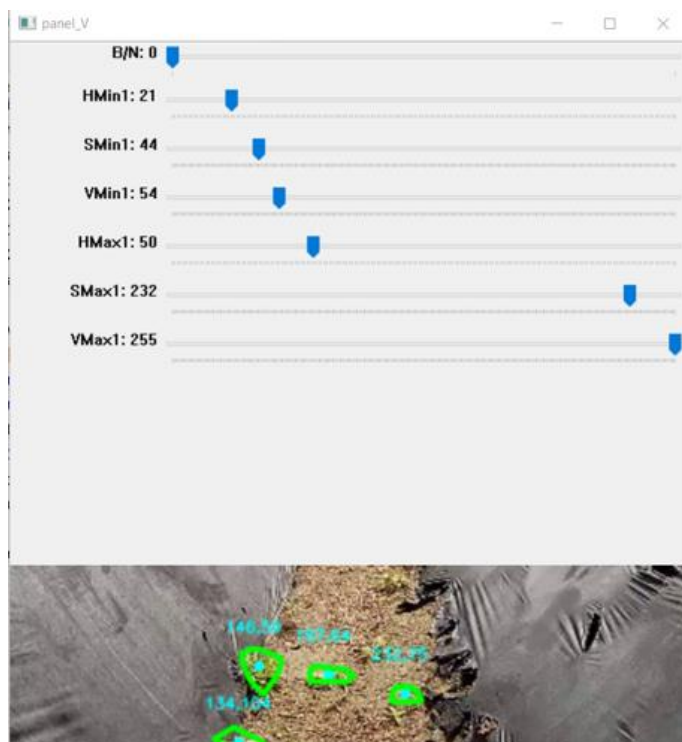


Figura 57-3: Maleza detectada
Realizado por: Villacis, A. 2021

El funcionamiento de la herramienta de corte se comportó de la forma esperada, realizando un corte limpio y eliminando la maleza desde su base. La calibración de la altura y la inclinación de la herramienta es fundamental para la obtener un buen corte ya que si se inclina demasiado o está muy baja produce que particular de tierra se levanten del piso interfiriendo con la cámara y ocasionando que existan errores en la detección de maleza y trayectorias.



Figura 58-3: Funcionamiento de la herramienta de corte
Realizado por: Villacis, A. 2021

Entonces al contabilizar la maleza detectada y eliminada tenemos la siguiente tabla.

Tabla 20-3: Comparativa de maleza contabilizada, detectada y eliminada

Surco	Maleza contabilizada	Maleza detectada	Maleza eliminada
S1	39	36	31
S2	65	65	58
S3	61	59	56
S4	30	30	26
S5	32	29	28
S6	35	34	30
S7	42	39	33
S8	58	53	47
S9	46	41	40
S10	37	35	31
S11	32	28	27
TOTAL	477	449	407

Realizado por: Villacis, A. 2021

La eficacia tanto de la detección como de la eliminación de la maleza se pueden apreciar en los gráficos, este porcentaje de eficacia nos dará un claro indicador de que el prototipo está cumpliendo o no con la tarea para la que ha sido desarrollado.

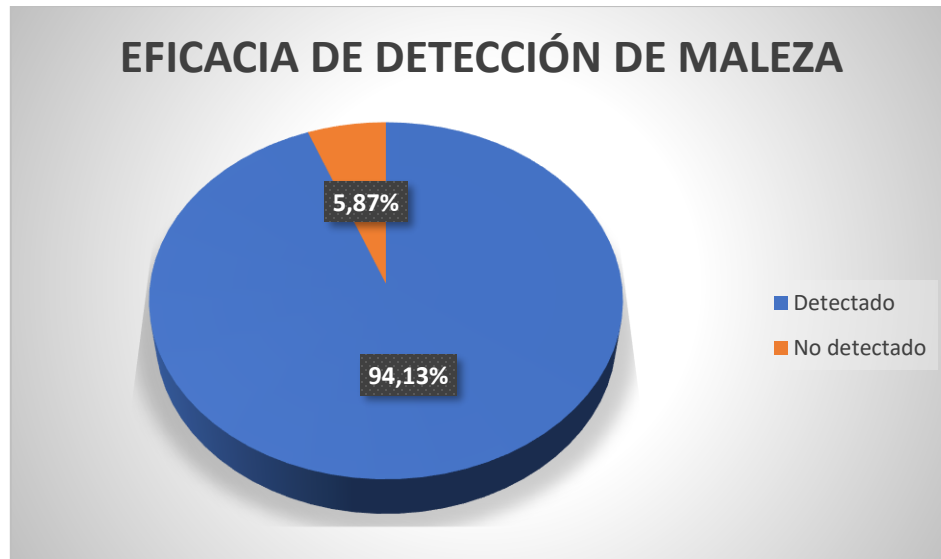


Gráfico 4-3: Eficacia de detección de maleza
 Realizado por: Villacis, A. 2021

La eficacia de la detección de la maleza es muy alta, este porcentaje del 94,13% indica que el sistema desarrollado funciona de acuerdo a lo esperado, a su vez se visualizó que el sistema de visión en la que se utiliza el color verde como indicador de la existencia de maleza omite a cierta maleza que es aún pequeña, ya que se estableció un mínimo de píxeles el cual es necesario para tener una detección correcta ya que al disminuir hasta cierto punto el número de píxeles existen fallos en la detección y el sistema marca a ciertos elementos existentes en el surco que tienden al color verde como maleza cuando no lo son. Por lo que el porcentaje de maleza no detectada es aceptable ya que al paso del tiempo debido a su desarrollo y crecimiento serán detectadas en una próxima ocasión.



Gráfico 5-3: Eficacia de eliminación de maleza
 Realizado por: Villacis, A. 2021

La eliminación de maleza depende de la detección de esta a través del sistema de visión artificial creado, es decir que si la eficacia en la detección es deficiente la eliminación también lo será. Según lo comprobado la eficacia obtenida es del 90,65% lo que significa que de 449 plantas detectadas 407 fueron eliminadas con éxito. Se reviso los surcos y se evidencio que el porcentaje de maleza que no fue eliminada se debió a que se encontraban junto a la lona plástica negra por lo que no fueron alcanzadas por la herramienta, ya que, con el fin de precautelar la integridad física de la lona el largo del hilo de nylon con el que se realiza el corte, esta calibrado para que en ningún momento pueda llegar a tener contacto con la lona plástica. En la siguiente figura podemos observar uno de los casos mencionados.



Figura 59-3: Maleza no eliminada
Realizado por: Villacis, A. 2021

2.15.1.3. Tiempo de procesamiento de imágenes.

Se analizo un total de 10939 imágenes y cuánto tiempo tardo la computadora en procesar en tiempo real cada una de estas imágenes mientras el prototipo robótico realizaba sus funciones dentro del cultivo experimental. Según los datos obtenidos se tienen tiempos de procesamientos buenos, lo que asegura que el prototipo podrá reaccionar de manera veloz al detectar maleza y trazar la trayectoria para en un siguiente paso eliminar lo que detecte. A continuación, tenemos un gráfico en el cual podemos contemplar el comportamiento en tiempo real del procesamiento de imágenes.

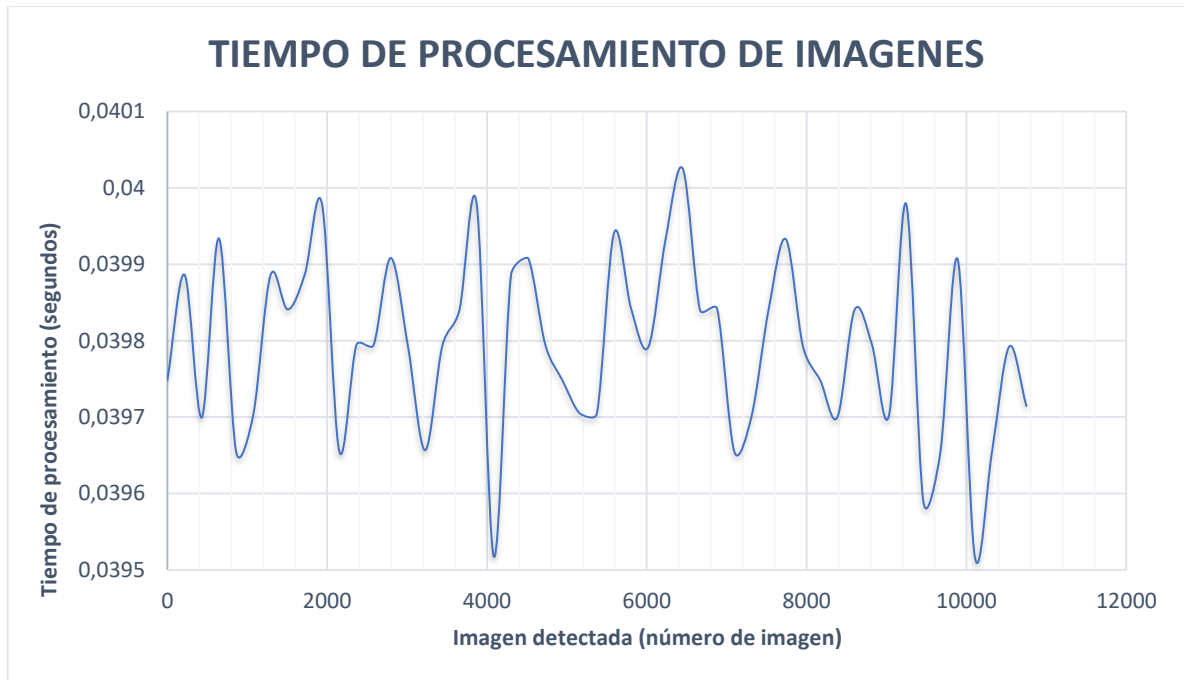


Gráfico 6-3: Tiempo de procesamiento de imágenes
 Realizado por: Villacis, A. 2021

Se puede apreciar que el tiempo que tarda en procesar las imágenes es medianamente estable esto se debe a que en su mayoría el tiempo de procesado por imagen varía entre 0,0395 s a 0,0401 s lo que garantiza que no se generen colas de imágenes para procesar y así se evita un retraso que afecte al funcionamiento del prototipo.

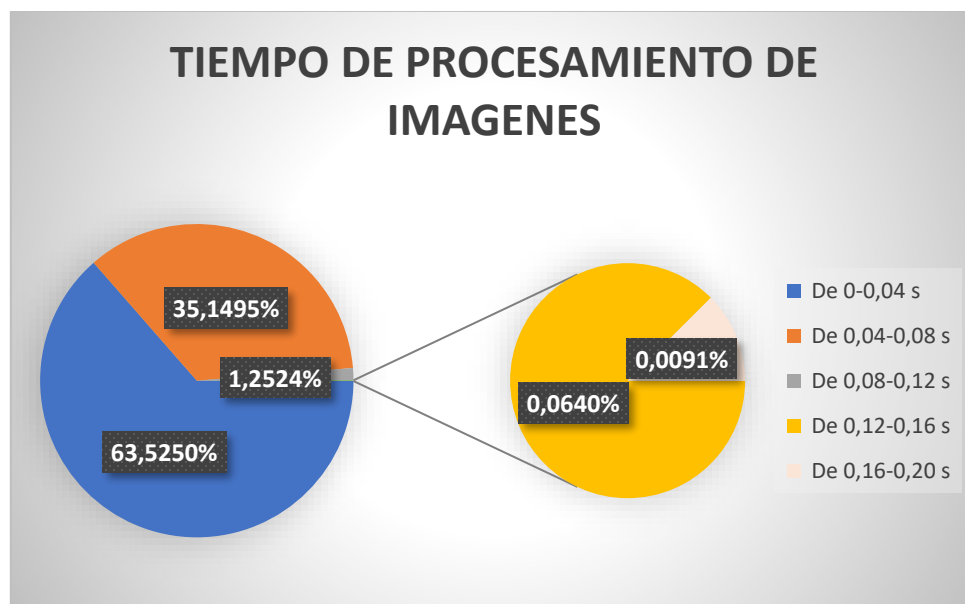


Gráfico 7-3: Porcentaje de tiempos de procesado de imágenes
 Realizado por: Villacis, A. 2021

En el Gráfico 7-3 podemos observar el porcentaje de las imágenes según el tiempo que tomo su procesamiento en intervalos definidos, del total de las 10939 imágenes procesadas se apreció que el porcentaje que supera los 80 milisegundos es sumamente bajo lo que comprueba que el procesamiento es veloz.

2.15.1.4. Generación de trayectorias.

Se estableció un centro en el cual se definió un límite de error permisible el cual si es superado tanto por el nivel superior como inferior la dirección será corregida con el fin de encaminar al prototipo al centro. En la siguiente imagen podemos observar cómo se genera la trayectoria que sigue el prototipo robótico.

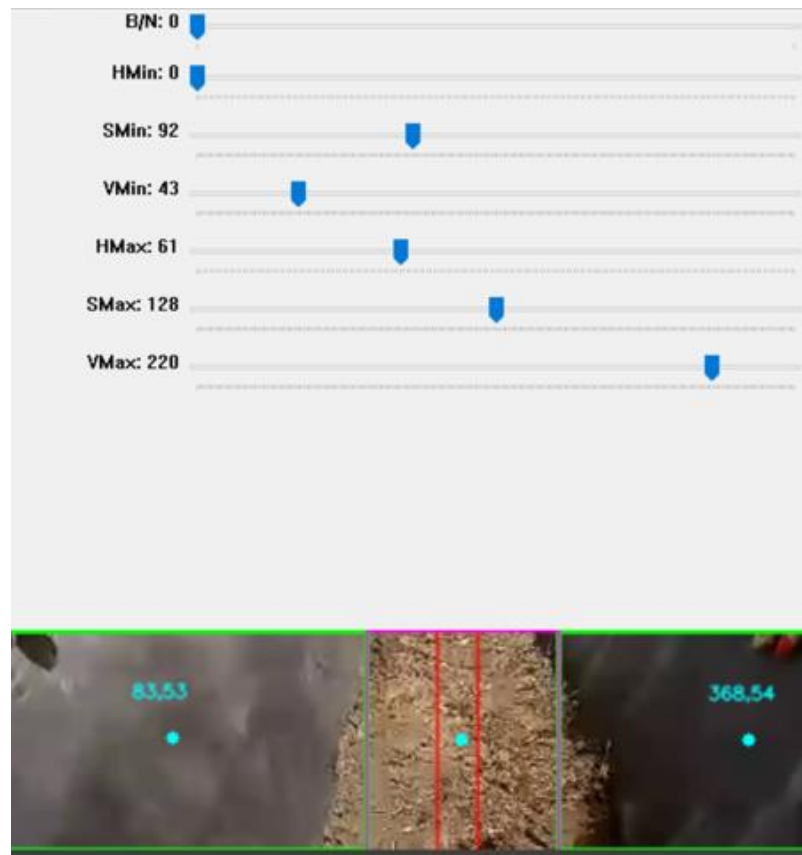
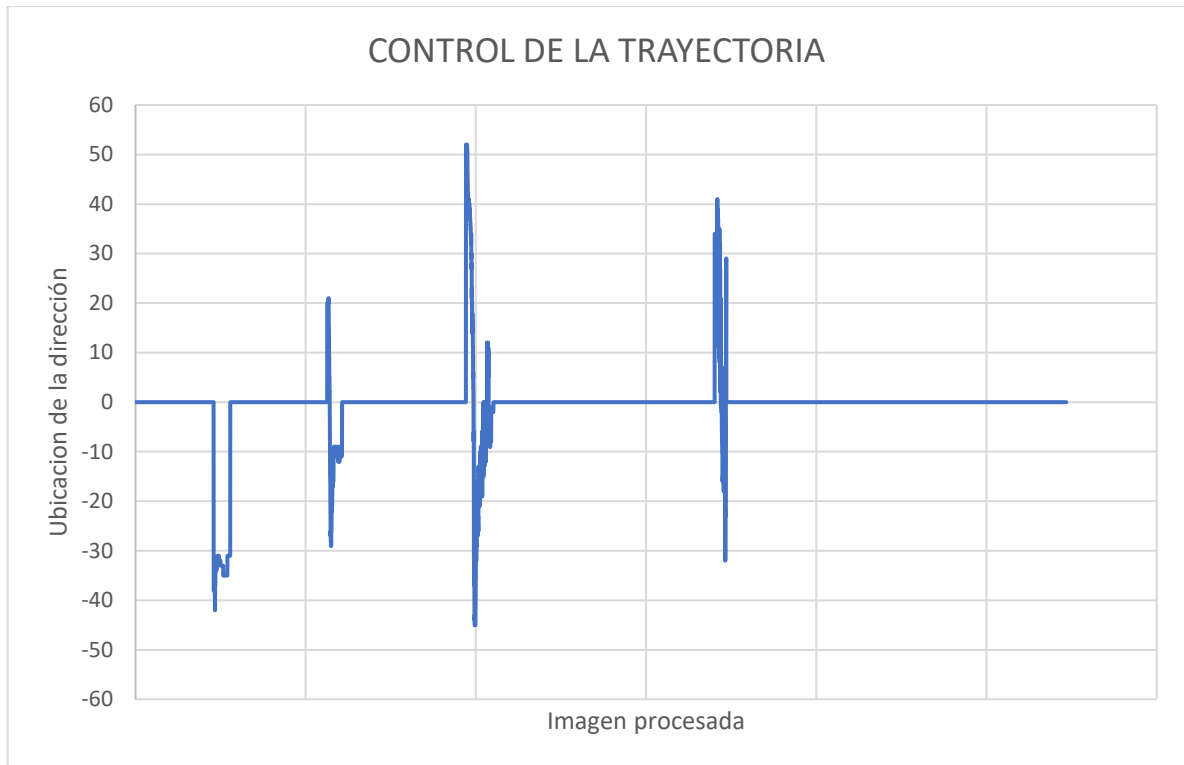


Figura 60-3: Generación de trayectorias

Realizado por: Villacis, A. 2021

La corrección de la trayectoria del prototipo a través del surco se pudo observar en el siguiente gráfico.



Realizado por: Villacis, A. 2021

Se puede apreciar las correcciones que realiza el prototipo para que la dirección este en el centro del surco y pueda movilizarse sin dificultades. En este aspecto el comportamiento del robot móvil fue el indicado y no se encontraron inconvenientes en el trazado de la trayectoria a seguir.

2.15.2. Consumo de energía

Para el cálculo de consumo de corriente del prototipo robótico se consideraron los dispositivos electrónicos mostrados en la Tabla 18-3, durante el proceso de eliminación de la maleza.

Tabla 21-3: Consumo energético del prototipo

CONSUMO DE ENERGÍA DEL PROTOTIPO			
Elemento	Corriente (mA)	Voltaje (V)	Potencia (W)
Placa de control AUTOPRO	500	5	2,5
Camara Ip	500	12	6
Router	250	9	2,25
Driver motor tracción	3500	12	42
Driver moto Nema 23	2700	24	64,8
Luces LED	1000	12	12
TOTAL	8450		129,55

Fuente: Villacis, A. 2021

La fuente de alimentación que provee de energía al prototipo en este caso dos baterías de gel de 12V brinda una corriente de 160000 mA/h, se pudo obtener que el prototipo tiene una autonomía de funcionamiento de 113,4 minutos, se llegó a este resultado utilizando la formula siguiente.

$$Tiempo = \frac{Corriente\ de\ la\ Bateria}{Corriente\ total}$$

$$Tiempo = \frac{16000\ mA/h}{8450\ mA}$$

$$Tiempo = 1,89\ h$$

$$Tiempo = 113,4\ min$$

3. GESTIÓN DEL PROYECTO

Se considera a la gestión como uno de los puntos más importantes en la ejecución de un proyecto para determinar su viabilidad. Para ello se realiza un análisis de costos directos e indirectos en los cuales se detallarán todos los valores económicos necesarios para la implementación del prototipo robótico.

3.1. Costos Directos

Los costos directos son aquellos costos que influyen de manera inmediata en la construcción del prototipo robótico. En la siguiente tabla se detallan cada uno de los costos directos.

Tabla 1-4: Costos de elementos mecánicos y electrónicos

Descripción	Cantidad	Costo unitario (USD)	Costo total (USD)
Tubo estructural cuadrado de 1 in	2	9,00	18,00
Rueda de carretilla neumática	3	8,50	25,50
Plancha de acero negro 210x695x2 mm	1	35,00	35,00
Rodamiento	1	7,50	7,50
Eje ANSI 1020-1050 de ¼ in	1	8,00	8,00
Tuercas, arandelas, tornillos.	60	0,16	10,00
Acople de nylon para caja reductora	1	35,00	35,00
Tubo redonde de acero negro de ½ in	1	8,00	8,00
Acople de acero negro para eje	1	16,00	16,00
Plancha de acrílico de 4 mm 1800x1200	1	40,00	40,00
Cables eléctricos	10	0,40	4,00
Conjunto motor – caja reductora	1	45,00	45,00
Cámara Ip. HIKVISION DS-2CD2421G0-I	1	120,00	120,00
Desbrozadora Truper DES-440T	1	80,00	80,00
Batería 12n7bl de 12V	2	26,00	52,00
Modulo Driver IBT2 43A	1	18,00	18,00
Shield Ethernet W5100	1	14,00	14,00
Modulo relé de dos canales	1	5,00	5,00
Modulo regulador Step Down LM2596	2	2,00	4,00
Router Tp-link TI-wr840n	1	21,00	21,00
SUBTOTAL 1			566,00

Realizado por: Villacis, A. 2021

En la siguiente tabla se detalla el costo de mano de obra utilizada en la construcción de la estructura metálica y en el corte del acrílico.

Tabla 2-4: Costo de mano de obra

Descripción	Cantidad	Horas/hombre	USD/hora	Costo total (USD)
Técnico soldador	1	24	4,50	108,00
Operario de cortadora laser	1	2	5,00	10,00
SUBTOTAL 2				118,00

Realizado por: Villacis, A. 2021

En la tabla 20-4 se podrá observar el cálculo total de los costos directos.

Tabla 3-4: Costo directo total

Descripción	Costo total (USD)
Subtotal 1	566,00
Subtotal 2	118,00
TOTAL	684,00

Realizado por: Villacis, A. 2021

3.2. Costos Indirectos

Para los costos indirectos se consideran aquellos costos producidos por actividades fuera del proceso de fabricación.

Para este trabajo de titulación se designó un valor estimado para conceptos de gasto de transporte, referentes a la necesidad constante de movilizarse al cultivo experimental el cual se encuentra a una distancia de 18,7km, además incluir la utilización del vehículo para realizar una búsqueda de los diferentes componentes necesarios para el prototipo. Estas actividades producen un gasto de 60 dólares.

En la siguiente tabla se especificará el cálculo de costos indirectos.

Tabla 4-4: Cálculo de costos indirectos

Descripción	Costo (USD)
Transporte	60,00
Imprevistos	50,00
TOTAL	110,00

Realizado por: Villacis, A. 2021

3.3. Costoso Totales

Los costes totales realizados a lo largo del desarrollo de este trabajo de titulación se pueden observar en la siguiente tabla.

Tabla 5-4: Cálculo de costos totales

Descripción	Costo (USD)
Costos directos	684,00
Costos indirectos	110,00
TOTAL	794,00

Realizado por: Villacis, A. 2021

La fuente de financiamiento para cubrir los gastos que conlleva realizar este proyecto es autónoma.

El Grupo de Investigación de la Facultad de Mecánica – ESPOCH “Tecnologías de la información, comunicación y procesos industriales AUTOPRO” facilito para la realización de este proyecto los siguientes elementos.

- Motor paso a paso Nema 23 y controlador DM556Y.
- Dos Placas Arduino Uno.

Conclusiones

- Se construyó un prototipo robótico vinculado con visión artificial para la asistencia en la eliminación de maleza en cultivos de fresa, este prototipo basa su funcionamiento en los robots móviles de ruedas; la trayectoria y el accionamiento de la herramienta que elimina la maleza es controlado a través de un sistema de visión artificial el cual es procesado por un ordenador portátil.
- Se modeló un sistema robótico en base a la caracterización del cultivo experimental, su estructura fue validada con una simulación estática en SolidWorks que indica que el robot es apto para el trabajo encomendado ya que su simulación con el material seleccionado (Acero ASTM A36) no presentó deformaciones ni desplazamientos,
- Se definió a la desbrozadora eléctrica DES-440T como la mejor opción para la eliminación de maleza en cultivos de fresa debido a sus 450W de potencia, su velocidad y capacidad de corte; esta herramienta es eficaz con malezas de bajo tamaño, sin embargo, se evidenció una falencia al tratar de eliminar maleza de alta envergadura.
- Se seleccionó el lenguaje de programación Python para la realización del algoritmo del sistema de visión artificial complementado con las librerías de OpenCV y la IDE de Arduino.
- El sistema de visión para la detección de la maleza brinda un adecuado procesamiento de imagen para el prototipo, por medio de este se logra observar una buena segmentación gracias a la aplicación de los filtros en la imagen y búsqueda de color verde.
- Se realizaron pruebas de funcionalidad en el cultivo experimental de fresa, el sistema de visión artificial para el control de la trayectoria del prototipo demostró ser eficiente, este género la mayoría de las ocasiones la trayectoria esperada y guió de manera precisa el sistema de dirección del prototipo, controlando al motor nema 23 el cual es responsable de la precisión observada.
- Las pruebas de funcionalidad indicaron que la configuración del robot móvil con ruedas, triciclo clásico, presenta pequeñas dificultades e inestabilidad en zonas en donde el cultivo es muy irregular y pierde tracción al entrar en contacto con la lona plástica presente en las hileras del cultivo, estas dificultades se atribuyen netamente a la irregularidad del cultivo experimental mas no al diseño del prototipo robótico.
- La eliminación de la maleza es la esperada según lo evidenciado es las pruebas de funcionalidad, sin embargo, se detectó que es necesario que el prototipo robótico realice la trayectoria en el mismo surco en más de una ocasión si se desea eliminar la mayor cantidad posible de maleza.

Recomendaciones

- Se recomienda la utilización del prototipo robótico en cultivos planos que no presenten grandes irregularidades, esto ayudara a que el prototipo tenga un mejor rendimiento.
- Continuar aportando con nuevas ideas en el campo de la investigación agrícola bajo la premisa de llevar tecnología a cultivos y motivarles para que exista un mayor interés en la utilización de prototipos y sistemas como el realizado en este proyecto.
- Si se desea aplicar el prototipo en zonas muy irregulares se debe realizar nuevas investigaciones con el fin de proporcionarle un sistema más amplio que pueda ser utilizado en la mayoría de los entornos.
- Se recomienda investigar a fondo los sistemas de visión artificial ya que existen un sin número de aplicaciones que aún no se han diseñado y existe un campo de trabajo muy amplio en esa área.
- Investigar sobre las librerías necesarias para la programación de un sistema de visión artificial ya que al contar con ese conocimiento se puede simplificar de manera significativa el desarrollo del código.
- Se recomienda la utilización de la programación en hilos, esto ayuda a que el procesamiento de imágenes sea más veloz.
- Investigar, analizar y corroborar la efectividad de diversos métodos que sean de uso para la eliminación de malezas en diversos tipos de cultivos.

BIBLIOGRAFÍA

AHMAD, T; & STWAARD, B. *Automation: The future of Weed Control in Cropping Systems.* North Plate – USA: Springer, 2014, pp. 56-136.

ARDUINO. *Arduino Ethernet Shield.* [En línea] [Consulta: 16 julio 2021]. Disponible en: https://www.mouser.com/catalog/specsheets/a000056_datasheet.pdf

ARDUINO. *¿Qué es Arduino?.* [En línea] [Consulta: 17 julio 2021]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>

BAMBINO, I. *Una Introducción a los robots móviles.* [En línea] [Consulta: 05 agosto 2021]. Disponible en: https://sistemamid.com/panel/uploads/biblioteca/2017-01-01_09-42-20138787.pdf

BARRA, O; & BARRA, F. *Microcontroladores PIC con programación PBP.* Madrid – España: Editorial RA – MA, 2015. pp. 19-21.

BARRIENTOS, V. Análisis, diseño, construcción y control en tiempo real de un robot móvil tipo shakey en el seguimiento de trayectoria [En línea] (Trabajo de titulación) (Maestría) Instituto Politécnico Nacional, México D.F., México. 2008. pp. 4-7. [Consulta: 09 junio 2021]. Disponible en: <https://tesis.ipn.mx/handle/123456789/26191>

BARRIENTOS, V; et al. “Robots Móviles: Evolución y Estado del Arte”. *Revista Polibits* [En línea], 2007, (México) 1(35), pp. 12-16. [Consulta: 06 junio 2021]. Disponible en: http://www.polibits.gelbukh.com/2007_35/Robots%20Moviles_%20Evolucion%20y%20Estado%20del%20Arte.pdf

BIANCHI, P. *Guía completa del cultivo de las fresas*. Barcelona – España: Editorial De Vecchi, S.A., 2018, pp. 1-2.

CASELEY, J; & PARKER, C. *Manejo de Malezas para Países en Desarrollo*. Roma – Italia: Organización de las Naciones Unidas para la Agricultura y la Alimentación, 1996, pp. 147-162.

CASTRO, C. Diseño, simulación. estudio de factibilidad de un robot sembrador de maíz por surcos. [En línea] (Trabajo de titulación). Universidad Tecnológica Equinoccial, Quito, Ecuador. 2016. pp. 10-11. [Consulta: 08 junio 2021]. Disponible en: http://repositorio.ute.edu.ec/xmlui/bitstream/handle/123456789/14628/67402_1.pdf?sequence=1&isAllowed=y

CONTI, F. *Motores paso a paso*. Buenos Aires – Argentina: Librería y Editorial Alsina, 2013, pp. 1-2.

CORONA, L; et al. *Sensores y actuadores. Aplicaciones con Arduino*. México D.F. – México: Grupo Editorial Patria, 2014, pp. 280-281.

CRESPO, J. *Motores Arduino* [Blog]. Logroño – España: 2017 [Consulta: 23 junio de 2021]. Disponible en: <https://aprendiendoarduino.wordpress.com/tag/motores/>

CUEVA, K; & GROTEN, U. *Saberes y Prácticas Andinas. Una muestra de los sistemas de conocimiento biocultural local*. [En línea]. Quito – Ecuador: ECOCIENCIA, 2010. [Consulta: 11 junio de 2021]. Disponible en: <https://biblio.flacsoandes.edu.ec/libros/122194-opac>

DASSAULT SYSTEMES SOLIDWORKS CORPORATION. *Introducción a SolidWorks*. [En línea]. Massachusetts – USA: DASSAULT SYSTEMES, 2015. [Consulta: 21 junio de 2021]. Disponible en: https://my.solidworks.com/solidworks/guide/SOLIDWORKS_Introduction_ES.pdf

GARCIA, A. *Protocolos de Interconexión de Redes*. [En línea]. Cantabria – España: Universidad de Cantabria, 2012. [Consulta: 17 junio de 2021]. Disponible en: <https://ocw.unican.es/pluginfile.php/1357/course/section/1682/Tema%203.pdf>

GARCIA, G. *Espacios de color y el dominio frecuencial*. Murcia – España: Departamento de informática y sistemas de la Universidad de Murcia, 2016, pp. 31-35.

GONZALES, A; et al. *Técnicas y algoritmos básicos de visión artificial*. Logroño – España: Universidad de la Rioja, 2006, pp. 18-19.

HALL, D; et al. “A Rapidly Deployable Classification System using Visual Data for the Application of precision Weed Management.” *Computers and Electronics in Agriculture*. [En línea], 2018, (Australia) 148(2), pp. 107-120. [Consulta: 17 junio de 2021]. ISSN 0168-1699. Disponible en: <https://eprints.qut.edu.au/117663/8/117663.pdf>

INOSTROZA, J; et al. *Manual del campo. Uso de equipos pulverizadores*. [En línea]. Temuco–Chile: Institutos de Investigaciones Agropecuarias, 2011. [Consulta: 20 junio de 2021]. Disponible en: <https://biblioteca.inia.cl/bitstream/handle/20.500.14001/7432/NR38166.pdf?sequence=1>

INZUNZA, E. *Microcontroladores*. [En línea]. Ensenada – México: Universidad Autónoma de Baja California, 2016. [Consulta: 22 junio 2021]. Disponible en: <http://ing.ens.uabc.mx/docencia/apuntes/electronica/microcontroladores%5B11686%5D.pdf>

JIMÉNEZ, J. Simulación de vehículos eléctricos ligeros. [En línea] (Trabajo de titulación). Universidad Politécnica de Valencia, Valencia, España. 2012. pp. 8-23. [Consulta: 12 junio 2021]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/18173/Memoria.pdf?sequence=1>

LÓPEZ, E. *Protocolo SPI (Serial Peripheral Interface)* [En línea]. México: Ingeniería en Microcontroladores, 2015. [Consulta: 17 junio 2021]. Disponible en: <https://docplayer.es/7393962-Spi-teoria-y-aplicaciones-ingenieria-en-microcontroladores-protocolo-spi-serial-peripheral-interface-protocolo.html>

MANDADO, E; et al. *Microcontroladores PIC. Sistema integrado para el autoaprendizaje.* Barcelona – España: MARCOMBO, 2007. pp. 11-12.

MARTÍNEZ, S; & SISTO, R. Control y Comportamiento de Robots Omnidireccionales. [En línea] (Trabajo de titulación). Universidad de la República, Montevideo, Uruguay. 2009. pp. 25-26. [Consulta: 12 junio 2021]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/18173/Memoria.pdf?sequence=1>

MORALES, C; et al. *Aspectos relevantes en la producción de frambuesa.* [En línea]. Villa Alegre – Chile: Instituto de Investigaciones Agropecuarias INIA Raihuen. [Consulta: 5 junio 2021]. Disponible en: http://www.latranqueraweb.com.ar/web/archivos/menu/Aspectos_de_su_produccion.pdf

MORALES, G. *Análisis de circuitos eléctricos en DC: Nueva metodología de enseñanza.* México D.F – México: LIMUSA, 2005. pp. 77-80.

MOREIRA, F; et al. *Manual de biología de suelos tropicales. Muestreo y caracterización de la biodiversidad bajo suelo.* [En línea]. México: Instituto Nacional de Ecología, 2012. [Consulta: 26 junio 2021]. Disponible en: <http://189.240.101.244:8080/xmlui/handle/publicaciones/217?show=full>

NAVACERRADA, J. Sistema de detección de matrículas con Open CV. ANPR (Automation Number Plate Recognition). [En línea] (Trabajo de titulación). Universidad Politécnica de Madrid, Madrid, España. 2017. pp. 35-36. [Consulta: 19 junio 2021]. Disponible en: https://oa.upm.es/51869/1/TFG_JORGE_NAVACERRADA.pdf

OÑA, A. *Procedimiento Uso adecuado de motoguadaña.* [En línea]. Ecuador: SEK, 2019. [Consulta: 14 junio 2021]. Disponible en: <https://repositorio.uisek.edu.ec/bitstream/123456789/3486/8/PROCEDIMIENTO%20USO%20AD ECUADO%20DE%20%20MOTOGUADA%C3%91A.pdf>

RIVERA, M. *Labores Culturales. Guía para mantener un huerto orgánico y saludable.* [En línea]. La Paz – Bolivia: Alternativas CC, 2015. [Consulta: 24 junio 2021]. Disponible en: https://alternativascc.org/wp-content/uploads/2018/05/labores-culturales_web-1.pdf

RODRÍGUEZ, L. *Python Programación.* [En línea]. Guayaquil – Ecuador: Escuela Superior Politécnica del Litoral, 2016. [Consulta: 17 junio 2021]. Disponible en: http://www.fcnm.espol.edu.ec/sites/fcnm.espol.edu.ec/files/PYTHON_PROGRAMACION_V2_3.pdf

SENPLADES. *Plan Nacional para el Buen Vivir 2017 - 2021.* [En línea]. Quito – Ecuador: Secretaria Nacional de Planificación y Desarrollo, 2017. [Consulta: 07 junio 2021]. Disponible en: <https://www.gobiernoelectronico.gob.ec/wp-content/uploads/downloads/2017/09/Plan-Nacional-para-el-Buen-Vivir-2017-2021.pdf>

SEPÚLVEDA, P; et al. *Cultivo de frutilla, en una realidad sin bromuro de metilo en Chile.* [En línea]. Santiago de Chile – Chile: Ministerio del Medio Ambiente y ONUDI, 2015. [Consulta: 15 junio 2021]. Disponible en: https://www.unido.org/sites/default/files/2016-11/straw_0.pdf

SERRANO, J. *Diseño y construcción de un robot móvil con dos motores independientes y una rueda libre.* [En línea] (Trabajo de titulación). Instituto Politécnico Nacional, México D.F, México. 2017. pp. 45-46. [Consulta: 18 junio 2021]. Disponible en: <https://tesis.ipn.mx/jspui/bitstream/123456789/9309/1/160.pdf>

SILVA, R; et al. “Una panorámica de los robots móviles.” *Revista Electronica de Estudios Telematicos*. [En línea], 2007, (Venezuela) 6(3), pp. 1-14. [Consulta: 17 junio de 2021]. ISSN 1856-4194. Disponible en: <https://www.redalyc.org/pdf/784/78460301.pdf>

STEPPERONLINE. *User Manual for Full Digital Stepper Driver*. [En línea]. Nankín – China: STEPPERONLINE, 2019. [Consulta: 20 junio 2021]. Disponible en: <https://www.omc-stepperonline.com/download/Y-series-open-loop-stepper-driver.pdf>

TRUPER. *Instructivo de Debrozadora/orilladora electrica*. [En línea] [Consulta: 20 junio 2021]. Disponible en: <https://www.truper.com/admin/descargables/manual/10975.pdf>

VALDÉS, F; & PALLAS, R. *Microcontroladores: Fundamentos y aplicaciones con PIC*. Barcelona – España : MARCOMBO S.A, 2007. pp. 14-21.

VELASCO, J. *El avance de la automatización en la agricultura - Redagrícola, Tendencias tecnológicas en la industria frutícola*. [Blog]. Chile: 2017 [Consulta: 13 junio de 2021]. Disponible en: <http://www.redagricola.com/cl/el-avance-de-la-automatizacion-en-la-agricultura/>

VÉLEZ, J; et al. *Visión por computador*. [En línea]. Madrid – España: Editorial DYKINSON, 2003. [Consulta: 19 junio 2021]. Disponible en: <https://elibro.net/es/ereader/espoch/104894>

ANEXOS

ANEXO A: PLANOS DEL PROTOTIPO ROBOTICO PARA LA ASISTENCIA EN LA ELIMINACION EN LA MALEZA Y SUS ELEMENTOS

ANEXO B: ESQUEMA GENERAL DE CONEXIONES

ANEXO C: DIAGRAMAS DE FLUJO MODO MANUAL Y ARDUINO

ANEXO D: CODIGO PYTHON DEL PROTOTIPO ROBÓTICO

```
import numpy as np
from tkinter import messagebox
import cv2
from socket import *
import time
from tabulate import tabulate
import threading

def automaticoA():
    global actA, actH, actM, px1, pyl, pA1, pA2, ima, roiA1, tempo, roi, err
    print(px1)
    if len(px1) > 1:
        if actA == False:
            data = 'A' + str(velocidad) + 'Z'
            arduino_server.sendto(data.encode(), direccion)
            actA = True
        actM = False
        error = 40
        ima = cv2.imread('auto.jpg')
        wi = roi.shape[1]
        he = roi.shape[0]
        setpoint = int(wi / 2)
        p1 = (int(wi / 2) - error, 0)
        p2 = (int(wi / 2) + error, he)
        cv2.rectangle(roiA1, p1, p2, (0, 0, 255), 2, cv2.LINE_AA)
        p_1 = (pA1[1], pyl[1])
        p_2 = (px1[0], pA2[0])
        pointx1 = pA1[1] + (abs(int((px1[0] - pA1[1]) / 2)))
        pointy1 = abs(int((pyl[1] - pA2[0]) / 2))
        point = (pointx1, pointy1)
        cv2.circle(roiA1, point, 15, (255, 255, 0), -1)
        cv2.rectangle(roiA1, p_1, p_2, (255, 0, 255), 2, cv2.LINE_AA)
        err = setpoint - pointx1

    print(err, setpoint, pointx1)
    if err > error:
        data = 'B' + str(1) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
    elif err < -error:
        data = 'B' + str(-1) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
    else:
        pass
    if ph == 1 and actH == 0:
        actH = 1
        tempo = time.time()
        data = 'P' + str(0) + 'Z'
        arduino_server.sendto(data.encode(), direccion)

    if actH == 1:
        elapse = time.time()
```



```

        lapso = abs(tempo - elapse)
        if lapso > 5:
            actH = 0
            data = 'P' + str(1) + 'Z'
            arduino_server.sendto(data.encode(), direccion)

def manual():
    global h, h1, actA, actH, actM, ima
    data = ''
    if actM == False:
        data = 'S' + str(0) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
        data = 'P' + str(1) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
        actM = True
    actA = False
    actH = 0
    ima = cv2.imread('manual.jpg')
    if k == ord('w'):
        data = 'A' + str(velocidad) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
    if k == ord('a'):
        data = 'B' + str(1) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
    if k == ord('d'):
        data = 'B' + str(-1) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
    if k == ord('e'):
        data = 'S' + str(0) + 'Z'
        arduino_server.sendto(data.encode(), direccion)

    if k == ord('s'):
        data = 'R' + str(velocidad) + 'Z'
        arduino_server.sendto(data.encode(), direccion)

    if k == ord('p'):
        if h == 0:
            h = 1
        else:
            h = 0
        data = 'P' + str(h) + 'Z'
        arduino_server.sendto(data.encode(), direccion)

    if k == ord('l'):
        if h1 == 0:
            h1 = 1
        else:
            h1 = 0
        data = 'L' + str(h1) + 'Z'
        arduino_server.sendto(data.encode(), direccion)

    if k == ord('r'):
        data = 'U' + str(0) + 'Z'
        arduino_server.sendto(data.encode(), direccion)
    print(data)

```

```

def colorN(roi):
    global roiA1, roiV1, th, px1, py1, pA1, pA2
    global hMin, hMax, sMin, sMax, vMin, vMax
    roiA1 = roi.copy()
    hMin = cv2.getTrackbarPos('HMin', 'panel_N')
    sMin = cv2.getTrackbarPos('SMin', 'panel_N')
    vMin = cv2.getTrackbarPos('VMin', 'panel_N')
    hMax = cv2.getTrackbarPos('HMax', 'panel_N')
    sMax = cv2.getTrackbarPos('SMax', 'panel_N')
    vMax = cv2.getTrackbarPos('VMax', 'panel_N')
    lower = np.array([hMin, sMin, vMin])
    upper = np.array([hMax, sMax, vMax])
    roiA = cv2.cvtColor(roiA1, cv2.COLOR_BGR2HSV)
    th = cv2.inRange(roiA, lower, upper)

    th = cv2.adaptiveThreshold(th, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 11, 2)
    cnts, _ = cv2.findContours(th, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
    for c in cnts:
        area = cv2.contourArea(c)

        if area > 70000:
            M = cv2.moments(c)
            if (M["m00"] == 0): M["m00"] = 1
            x = int(M["m10"] / M["m00"])
            y = int(M["m01"] / M["m00"])
            cv2.circle(roiA1, (x, y), 7, (255, 255, 0), -1)
            cv2.putText(roiA1, '{}{}'.format(x, y), (x - 20, y - 20),
cv2.FONT_ITALIC, 0.6, (255, 255, 0), 2,
                        cv2.LINE_AA)
            x, y, w, hc = cv2.boundingRect(c)
            px1.append(x)
            py1.append(y)
            pA1.append(w)
            pA2.append(hc)
            roiA1 = cv2.rectangle(roiA1, (x, y), (x + w, y + hc), (0,
255, 0), 2)

def colorV(roi):
    global roiV1, ph, thV
    global hMin1, hMax1, sMin1, sMax1, vMin1, vMax1
    hMin1 = cv2.getTrackbarPos('HMin1', 'panel_V')
    sMin1 = cv2.getTrackbarPos('SMin1', 'panel_V')
    vMin1 = cv2.getTrackbarPos('VMin1', 'panel_V')
    hMax1 = cv2.getTrackbarPos('HMax1', 'panel_V')
    sMax1 = cv2.getTrackbarPos('SMax1', 'panel_V')
    vMax1 = cv2.getTrackbarPos('VMax1', 'panel_V')
    lower1 = np.array([hMin1, sMin1, vMin1])
    upper1 = np.array([hMax1, sMax1, vMax1])
    roiV1 = roi.copy()
    roiB = cv2.cvtColor(roiV1, cv2.COLOR_BGR2HSV)
    thV = cv2.inRange(roiB, lower1, upper1)
    thV = cv2.medianBlur(thV, 3)
    cnts1, _ = cv2.findContours(thV, cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)

```

```

for c in cnts1:
    area = cv2.contourArea(c)
    if area > 2000:
        M = cv2.moments(c)
        if (M["m00"] == 0): M["m00"] = 1
        x = int(M["m10"] / M["m00"])
        y = int(M["m01"] / M["m00"])
        nuevoContorno1 = cv2.convexHull(c)
        ph = 1
        cv2.circle(roiV1, (x, y), 7, (255, 255, 0), -1)
        cv2.putText(roiV1, '{}{}'.format(x, y), (x - 20, y - 20),
cv2.FONT_ITALIC, 0.6, (255, 255, 0), 2,
                    cv2.LINE_AA)
        cv2.drawContours(image=roiV1, contours=[nuevoContorno1],
contourIdx=-1, color=(0, 255, 0), thickness=2,
                    lineType=cv2.LINE_AA)

```

```

def on_change(value):
    pass

```

```

def activar(*args):
    pass

```

```

##=====empieza=====
=====
url = "rtsp://admin:robot1234@192.168.20.101/Streaming/Channels/1"
direccion = ('192.168.20.102', 8000)
arduino_server = socket(AF_INET, SOCK_DGRAM)
arduino_server.settimeout(1)
##===== codigo para leer el archivo txt color negro
=====
file = open('colorNegro.txt', 'r')
data = file.read()
print(data)
file.close()
valueHSV = []
posA = 0
for pos, char in enumerate(data):
    if (char == ','):
        datos = data[posA:pos]
        posA = pos + 1
        valueHSV.append(int(datos))
print(valueHSV)
##===== codigo para leer el archivo txt color negro
=====
file = open('colorVerde.txt', 'r')
data = file.read()
print(data)
file.close()
valueHSV_V = []
posA = 0
for pos, char in enumerate(data):
    if (char == ','):
        datos = data[posA:pos]
        posA = pos + 1

```

```

        valueHSV_V.append(int(datos))
print(valueHSV_V)
##=====
=====

cap = cv2.VideoCapture(url)
#cap = cv2.VideoCapture("video.avi")
ret, frame = cap.read()

## codigo panel de control principal
control = False
messagebox.showinfo("Aviso", "sistema en Manual")
salida = cv2.VideoWriter('videoSalida.avi',
cv2.VideoWriter_fourcc(*'XVID'), 20.0, (1920, 1080))
cv2.namedWindow('panel_N')
cv2.createTrackbar('B/N', 'panel_N', 0, 1, on_change)
cv2.createTrackbar('HMin', 'panel_N', 0, 179, on_change)
cv2.createTrackbar('SMin', 'panel_N', 0, 255, on_change)
cv2.createTrackbar('VMin', 'panel_N', 0, 255, on_change)
cv2.createTrackbar('HMax', 'panel_N', 0, 179, on_change)
cv2.createTrackbar('SMax', 'panel_N', 0, 255, on_change)
cv2.createTrackbar('VMax', 'panel_N', 0, 255, on_change)
cv2.setTrackbarPos('HMin', 'panel_N', valueHSV[0])
cv2.setTrackbarPos('SMin', 'panel_N', valueHSV[1])
cv2.setTrackbarPos('VMin', 'panel_N', valueHSV[2])
cv2.setTrackbarPos('HMax', 'panel_N', valueHSV[3])
cv2.setTrackbarPos('SMax', 'panel_N', valueHSV[4])
cv2.setTrackbarPos('VMax', 'panel_N', valueHSV[5])

cv2.namedWindow('panel_V')
cv2.createTrackbar('B/N', 'panel_V', 0, 1, on_change)
cv2.createTrackbar('HMin1', 'panel_V', 0, 179, on_change)
cv2.createTrackbar('SMin1', 'panel_V', 0, 255, on_change)
cv2.createTrackbar('VMin1', 'panel_V', 0, 255, on_change)
cv2.createTrackbar('HMax1', 'panel_V', 0, 179, on_change)
cv2.createTrackbar('SMax1', 'panel_V', 0, 255, on_change)
cv2.createTrackbar('VMax1', 'panel_V', 0, 255, on_change)
cv2.setTrackbarPos('HMin1', 'panel_V', valueHSV_V[0])
cv2.setTrackbarPos('SMin1', 'panel_V', valueHSV_V[1])
cv2.setTrackbarPos('VMin1', 'panel_V', valueHSV_V[2])
cv2.setTrackbarPos('HMax1', 'panel_V', valueHSV_V[3])
cv2.setTrackbarPos('SMax1', 'panel_V', valueHSV_V[4])
cv2.setTrackbarPos('VMax1', 'panel_V', valueHSV_V[5])
cv2.namedWindow('Menu Principal')
switch = 'Manual : Automatico'
cv2.createTrackbar(switch, 'Menu Principal', 0, 1, on_change)
cv2.createTrackbar('Velocidad', 'Menu Principal', 0, 255, on_change)
cv2.createTrackbar('Tiempo de ejecucion', 'Menu Principal', 0, 1,
on_change)
## activadores
global h, h1, actA, actH, actM
actA = False
actM = False
actH = 0
tempo = 0
elapse = 0

```

```

act1 = 0
h = 1
hl = 1

while ret:
    try:
        global roiA1, roiV1, th, thV, px1, py1, pA1, pA2, ph, ima,
roi, err
        global hMin1, hMax1, sMin1, sMax1, vMin1, vMax1
        global hMin, hMax, sMin, sMax, vMin, vMax
        start = time.time()
        err = 0
        ret, frame = cap.read()
        # frame = cv2.imread('1.jpg')
        if not (ret):
            st = time.time()
            cap = cv2.VideoCapture(url)
            print("error en la conexion restableciendo : ", time.time() -
st)

            ret, frame = cap.read()
            continue
        frame2 = frame.copy()
        # salida.write(frame2)
        act1 = cv2.getTrackbarPos('B/N', 'panel_N')
        act2 = cv2.getTrackbarPos('B/N', 'panel_V')
        auto = cv2.getTrackbarPos(switch, 'Menu Principal')
        velocidad = cv2.getTrackbarPos('Velocidad', 'Menu Principal')
        actPri= cv2.getTrackbarPos('Tiempo de ejecucion', 'Menu
Principal')
        cv2.imshow('Imagen', frame2)
        # prepara la imagen para recorte solo d ela seccion escogida
        width = frame.shape[1]
        height = frame.shape[0]
        dim = (width, height)
        val1 = int(height * 0.5)
        val2 = int(height * 1.0)
        val3 = int(width * 0.0)
        val4 = int(width * 0.80)
        roi = frame[val1:val2, val3:val4]
        px1 = []
        py1 = []
        pA1 = []
        pA2 = []
        ## codigo para reconocimiento color negro y direccion
        k = cv2.waitKey(1)
        hilo = threading.Thread(target=colorN, args=(roi,))
        hilo1 = threading.Thread(target=colorV, args=(roi,))
        hilo.start()
        hilo1.start()
        hilo.join()
        hilo1.join()
        ## campo para la muestra de las imagenes

        k = cv2.waitKey(1)
        if auto != 1:
            manual()
        else:

```

```

        automaticoA()

# concat_vertical = cv2.vconcat([th, roiAux])
scale_percent = 35
W = int(roi.shape[1] * scale_percent / 100)
H = int(roi.shape[0] * scale_percent / 100)
dsize = (W, H)
if act1 == 1:
    roiAux = cv2.resize(th, dsize)
else:
    roiAux = cv2.resize(roiA1, dsize)

if act2 == 1:
    roiVux = cv2.resize(thV, dsize)
else:
    roiVux = cv2.resize(roiV1, dsize)
cv2.imshow('panel_N', roiAux)
cv2.imshow('panel_V', roiVux)
cv2.imshow('Menu Principal', ima)
cv2.resizeWindow('panel_N', 600, 200)
cv2.resizeWindow('panel_V', 600, 200)

end = time.time()
tiempo = end - start
if actPri:

print('=====')
print(tabulate({"Informacion": ["Tiempo (s)", "Error"],
"Medida": [tiempo, err]}, headers="keys"))

## campos para salir
if k == ord('g'):
    ## guarda los limites de cada reconocimiento de color
    dataN = str(hMin) + ',' + str(sMin) + ',' + str(vMin) + ',' +
str(hMax) + ',' + str(sMax) + ',' + str(
        vMax) + ','
    file = open('colorNegro.txt', 'w')
    file.write(dataN)
    file.close()
    dataN = str(hMin1) + ',' + str(sMin1) + ',' + str(vMin1) +
',' + str(hMax1) + ',' + str(sMax1) + ',' + str(
        vMax1) + ','
    file = open('colorVerde.txt', 'w')
    file.write(dataN)
    file.close()
    messagebox.showinfo("Aviso", "Se guardo los valores")

if k == ord('q'):
    break

except:
    print('error')
salida.release()
cv2.destroyAllWindows()

```

ANEXO E: CÓDIGO DE PROGRAMACIÓN EN ARDUINO

```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192,168,20,102);
unsigned int puerto=8000;
char paqueteBuffer[UDP_TX_PACKET_MAX_SIZE];
String dataReq;
int paquete_size;
EthernetUDP udp;
char packetBuffer[UDP_TX_PACKET_MAX_SIZE];
//===== motor direccion =====
int driverPUL = 8; // PUL- pin
int driverDIR = 7; // DIR- pin
int val=0;
int pass=0;

//===== motor =====
int RPWM = 6;
int LPWM = 5;
//===== modulo rele =====
int luces = 3;
int podadora= 2;
//===== Buzzer =====
int buzz = 4;
int valor =1;
int valor2 =1;
int angM=10;
```

```
void setup()
{
  pinMode(driverPUL,OUTPUT);
  pinMode(driverDIR,OUTPUT);
  pinMode(RPWM,OUTPUT);
  pinMode(LPWM,OUTPUT);
  pinMode(luces,OUTPUT);
  pinMode(podadora,OUTPUT);
  pinMode(buzz,OUTPUT);
  digitalWrite(luces,HIGH);
  digitalWrite(podadora,HIGH);
  Serial.begin(9600);
  for (int i = 0; i <= 20; i++)
  {
    digitalWrite(luces,!digitalRead(luces));
    delay(1000);
  }
  Ethernet.begin(mac,ip);
  udp.begin(puerto);
  analogWrite(RPWM,0);
  digitalWrite(podadora,1);
}

void loop()
{
  if(udp.parsePacket(>0)
  {
    udp.read(packetBuffer,UDP_TX_PACKET_MAX_SIZE);
    String dat(packetBuffer);
    Serial.println(dat);
  }
}
```



```
char a =dat[0];  
int b= dat.indexOf('Z');  
int data= (dat.substring(1, b)).toInt();  
Serial.println(data);
```

```
switch(a)  
{  
  case 'A':  
    analogWrite(LPWM,data);  
    analogWrite(RPWM,0);  
    break;  
  
  case 'R':  
    analogWrite(LPWM,0);  
    analogWrite(RPWM,data);  
    break;  
  
  case 'S':  
    analogWrite(LPWM,0);  
    analogWrite(RPWM,0);  
    break;  
  
  case 'B':  
    if(data>0)  
    {  
      val=val+1;  
      if(val>angM)  
      {  
        val=angM;  
      }  
    }  
}
```

```
    }  
    else  
    {  
        val=val-1;  
        if(val<-angM)  
        {  
            val=-angM;  
        }  
    }  
    Serial.println(val);  
    break;  
  
    case 'P':  
        valor=data;  
        break;  
  
    case 'L':  
        valor2=data;  
        break;  
  
    case 'U':  
        for (int i = 0; i <= 20; i++)  
        {  
            digitalWrite(buzz,!digitalRead(buzz));  
            digitalWrite(luces,!digitalRead(luces));  
            delay(100);  
        }  
        digitalWrite(buzz,LOW);  
        digitalWrite(luces,HIGH);  
        asm volatile ( "jmp 0");
```

```
        break;
    }
}
direccionV();
digitalWrite(podadora,valor);
digitalWrite(luces,valor2);
}
```

```
void direccionV()
{
    int error=val-pass;
    if(error>0)
    {
        digitalWrite(driverDIR,HIGH);
        mover();
        pass++;
    }
    if(error<0)
    {
        digitalWrite(driverDIR,LOW);
        mover();
        pass--;
    }
}
```

```
void mover()
{
    for (int i = 1; i <= 55; i++)
    {
        digitalWrite(driverPUL,HIGH);
```

```
    delayMicroseconds(100);  
    digitalWrite(driverPUL,LOW);  
    delayMicroseconds(100);  
  }  
}
```