



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**ROBOT MÓVIL AUTÓNOMO PARA LA
SIEMBRA DE SEMILLAS EN EL CAMPO**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Electrónica y Automatización

AUTORES: David Fernando Quiroga Guerra

Boris Leonardo Rubio Amaya

TUTOR: Carmen Johanna Celi Sánchez

Quito – Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, David Fernando Quiroga Guerra con documento de identificación N° 1720883832 y Boris Leonardo Rubio Amaya con documento de identificación N° 1720243268; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 9 de marzo del año 2023

Atentamente,



David Fernando Quiroga Guerra

1720883832



Boris Leonardo Rubio Amaya

1720243268

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, David Fernando Quiroga Guerra con documento de identificación N° 1720883832 y Boris Leonardo Rubio Amaya con documento de identificación N° 1720243268, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Robot móvil autónomo para la siembra de semillas en el campo”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 9 de marzo del año 2023

Atentamente,



David Fernando Quiroga Guerra

1720883832



Boris Leonardo Rubio Amaya

1720243268

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Carmen Johanna Celi Sánchez con documento de identificación N° 1717437808, docente de la Universidad Politécnica Salesiana Sede Quito, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **ROBOT MÓVIL AUTÓNOMO PARA LA SIEMBRA DE SEMILLAS EN EL CAMPO**, realizado por, David Fernando Quiroga Guerra con documento de identificación N° 1720883832 y Boris Leonardo Rubio Amaya con documento de identificación N° 1720243268 obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 9 de marzo del año 2023

Atentamente,



Ing. Carmen Johanna Celi Sánchez. MSc

1717437808

DEDICATORIA

A mi madre, por siempre haber confiado en mí y haberme apoyado en todo momento sin importar lo difícil que fuese la situación, a mi abuelo, sin su motivación y cariño ningún logro tendría la misma sensación y sin dejar de lado a quien estuvo a lado en todo momento con su incondicional compañía en todas las noches largas, mi Rex. Sin estos invaluable seres, nada sería posible, esto y más siempre será por ustedes.

Quiroga Guerra David Fernando.

A mi familia Galo, Chary, David y Jhastin ya que han sido mi principal inspiración y sin su apoyo incondicional no sería posible lograr esta meta, a mis abuelitos que con sus principios y valores me hicieron ser una persona de bien, a mi mascota Luna por su compañía en las noches cuando me quedaba estudiando.

Rubio Amaya Boris Leonardo

AGRADECIMIENTO

La vida es un camino largo, la cual está llena de retos y gratas experiencias, que nos forman y nos enseñan a apreciarla y querer tenerla para siempre.

Agradezco a esta bella institución y a sus docentes, quienes me han acogido durante este tramo de mi vida, han puesto su dedicación y conocimientos de la mejor manera para que el día de hoy pueda llamarme un profesional más, a mi tutora Mgtr. Carmen Johanna Celi Sánchez, por la paciencia y dedicación que nos puso en este proyecto, sabiendo guiarnos paso a paso hasta alcanzar el objetivo por el que tanto soñamos.

Un agradecimiento especial a mi madre, Susana; por sus enseñanzas y por compartirme el valor de la educación. Y sin dejar de lado a mis amigos, quienes con sus palabras de aliento jamás me han dejado solo y siempre confiaron en mí.

Quiroga Guerra David Fernando

Agradezco en primer lugar a Dios por haberme permitido llegar a esta etapa final de la carrera, a mis padres Galo y Chary por ser mi ejemplo para seguir, por todos sus sacrificios, consejos los cuales no me han permitido desistir en todo este camino y los valores enseñados. A mis hermanos David y Jhastin por ser mis amigos incondicionales, por todos los momentos que me apoyaron y todas las experiencias que vivimos juntos. Un agradecimiento especial a mi tío Xavier por su ayuda incondicional, por la ayuda intelectual y manual para lograr inventos extraordinarios. Agradezco a mis amigos de la carrera Ivonne, Cristian, David, Diego, Germán por todos los momentos compartidos el apoyo tanto académico como en lo personal, a mi profesora Ing. Johana Celi por la guía, la ayuda prestada y los consejos que me han brindado en esta etapa.

Rubio Amaya Boris Leonardo

ÍNDICE DE CONTENIDOS

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.....	2
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA.....	i
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE DE CONTENIDOS	v
ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE TABLAS	x
RESUMEN.....	xi
ABSTRACT.....	xii
INTRODUCCIÓN	xiii
CAPÍTULO 1	1
ANTECEDENTES.....	1
1.1 Planteamiento del problema.....	1
1.2 Justificación del proyecto.....	1
1.3 Objetivos	2
1.3.1 Objetivo General	2
1.3.2 Objetivos específicos	2
CAPÍTULO 2.....	4
MARCO TEÓRICO.....	4
2.1 Introducción	4
2.2 Estado del arte	4
2.3 Robot Móvil	5
2.4 Sistema de locomoción	6

2.4.1 Ruedas/ cintas de deslizamiento.....	6
2.5 Visión artificial.....	7
2.5.1 Captura: Proceso donde se obtiene una imagen a través de un dispositivo de cámara digital.	7
2.5.2 Preprocesamiento: Proceso de filtración el cual sirve para la reducción de ruido, mejoramiento del contraste y la nitidez de la imagen.....	7
2.5.3 Segmentación: Proceso donde se divide una imagen en objetos o regiones (grupo de píxeles), que sean el interés de estudio.....	7
2.5.4 Descripción: Proceso donde se obtienen características relevantes convenientes para distinguir un objeto de otro.	7
2.5.5 Reconocimiento: Proceso que clasifica en categorías los objetos presentes en la imagen utilizada en el proceso de descripción.	7
2.5.6 Interpretación: Proceso que da sentido o un significado a las categorías de objetos para entender la escena.	7
2.6 Inteligencia artificial (IA)	8
2.7 Redes neuronales.....	8
2.8 Arquitectura de una red neuronal.....	8
2.9 Capas de una red neuronal	9
2.9.1 Capa de entrada.....	9
2.9.2 Capa de salida	9
2.9.3 Capa oculta.....	10
2.10 Red neuronal convolucional.....	10
2.11 Entrenamiento de una red neuronal.....	11
2.12 Entrenamiento profundo.....	12
2.12.1 Entrenamiento	12
2.12.2 Inferencia.....	12
2.13 Pasos para realizar un aprendizaje profundo.....	12
2.13.1 Identificación de la función de aprendizaje	12

2.13.2 Selección de la plataforma	13
2.13.3 Preparación de datos	13
2.13.4 Entrenamiento y validación	14
2.13.5 Implementación de la red neuronal	14
CAPÍTULO 3	15
DISEÑO E IMPLEMENTACIÓN DE ROBOT SEMBRADOR.....	15
3.1 Selección del controlador	15
3.2 Diseño electrónico.....	18
3.3 Esquema de alimentación.....	22
3.4 Diseño estructural.....	23
3.5 Cinemática del robot inteligente	31
3.6 Red neuronal	34
3.7 Entrenamiento de la red neuronal	36
3.8 Algoritmo inteligente.	38
3.9 Programación de los motores.	40
3.10 Resumen de costos	47
CAPÍTULO 4	49
PRUEBAS – RESULTADOS.....	49
4.1 Pruebas de reconocimiento de surco.....	49
4.2 Pruebas con diferente número de muestras	51
4.3 Pruebas de funcionamiento del robot en campo	54
CONCLUSIONES	56
RECOMENDACIONES	58
REFERENCIAS.....	60
ANEXOS	62

ÍNDICE DE FIGURAS

Figura 2.1 Diagrama de capas	11
Figura 3.1 Tarjeta OpenMV Cam H7	16
Figura 3.2 Tarjeta Raspberry Pi Pico	16
Figura 3.3 Tarjeta Arduino Mega 2560	17
Figura 3.4 Módulo controlador de motores L298N.	20
Figura 3.5 Regulador LM2596.....	20
Figura 3.6 Ruedas Dagu Wild Thumper Wheel.....	21
Figura 3.7 Batería First Power 12V / 2.8 A	22
Figura 3.8 Esquema de alimentación del robot	22
Figura 3.9 Chasis del robot con la base.....	23
Figura 3.10 Pieza fija Cara A.....	24
Figura 3.11 Pieza móvil cara A.....	25
Figura 3.12 Pieza móvil cara B	25
Figura 3.13 Dispensador de Semillas.....	26
Figura 3.14 Piezas diseñadas en 3D.....	27
Figura 3.15 Piezas impresas	28
Figura 3.16 Ruedas y motores de desplazamiento	29
Figura 3.17 Elementos electrónicos	30
Figura 3.18 Prototipo Final	30
Figura 3.26 Modelo cinemático	31
Figura 3.19 Diagrama de bloques de la arquitectura de la red neuronal.....	34
Figura 3.20 Ejemplos de imágenes	35
Figura 3.21 Capa oculta y sus neuronas.....	35
Figura 3.22 Robot posicionado correctamente.	36
Figura 3.23 Interfaz de Edge Impulse	37
Figura 3.24 Archivos generados por Edge Impulse	37
Figura 3.25 Resultados del entrenamiento	38
Figura 3.27 Algoritmo Inteligente comentado primera parte.....	39
Figura 3.28 Algoritmo inteligente comentado segunda parte	39
Figura 3.29 Interfaz de la aplicación Bluetooth RC Car	40
Figura 3.30 Programación de los motores para trabajo autónomo.....	41
Figura 3.31 Programación del robot para trabajo manual parte 1	42

Figura 3.32 Programación del robot para trabajo manual parte 2.....	43
Figura 3.33 Programación del robot para trabajo manual parte 3.....	44
Figura 3.34 Programación del robot para trabajo manual parte 4.....	45
Figura 3.35 Programación del robot para trabajo manual parte 4.....	46
Figura 4.1 Ejemplo de Surco con brotes altos.....	49
Figura 4.2 Detección de Surco	53
Figura 4.3 Semillas depositadas	53
Figura 4.4 Robot durante el proceso de siembra.....	54

ÍNDICE DE TABLAS

Tabla 3.1 Resumen de clasificación de las tarjetas de desarrollo	18
Tabla 3.2 Características motores	21
Tabla 3.3 Resumen de costos del proyecto	47
Tabla 4.1 Resultados con 5 neuronas en la capa oculta	50
Tabla 4.2 Resultados con 8 neuronas en la capa oculta	50
Tabla 4.3 200 Muestras	51
Tabla 4.4 350 Muestras	52
Tabla 4.5 700 muestras.	52
Tabla 4.6 Pruebas de posicionamiento y siembra del robot.....	55

RESUMEN

Con el avance tecnológico y la fácil adquisición de nuevos elementos enfocados en la inteligencia artificial, se ha planteado la incógnita en el mundo de la ciencia de hasta donde es posible llegar para solventar de manera más sencilla y organizada los problemas más básicos de los seres humanos.

Tomando en cuenta el área del aprendizaje, se nota que existen muchas falencias y vacíos, los cuales podrán ser solventados con un buen material de estudio y apoyos didácticos para el estudiante.

El presente trabajo se ha dirigido hacia el desarrollo de un robot móvil autónomo, capaz de sembrar semillas basándonos en un algoritmo de visión artificial, el cual permitirá al robot identificar los surcos realizados en el terrero en el cual va a realizar su tarea, se ha determinado el uso de una tarjeta Raspberry PI Pico, quien albergará el algoritmo y será el cerebro de dicho robot.

Será acoplado a un módulo OpenMV Cam H7 para dotar al robot de visión artificial para que pueda identificar los surcos por donde realizará el trabajo de siembra.

Por medio de la implementación del prototipo en conjunto con un algoritmo inteligente se logró que el robot pueda realizar su tarea de manera idónea, creando así un apoyo didáctico para los estudiantes de electrónica y automatización que estén interesados en ampliar sus conocimientos sobre redes neuronales e inteligencia artificial.

Palabras claves: robot, prototipo, sembrador, visión artificial, algoritmo inteligente, campo.

ABSTRACT

With technological progress and the easy acquisition of new elements favorable to artificial vision techniques, in addition to its applications using artificial intelligence, added to the scarce development of mobile robotic prototypes that, based on said technology in national competitions, was born an opportunity to use, test and implement intelligent algorithms that allow control in the industry that allows the technification of processes such as planting.

The present work has been directed towards the development of an autonomous mobile robot, capable of sowing seeds based on an artificial vision algorithm, which will allow the robot to identify the furrows made in the field in which it is going to carry out its task, we have determined the use of an Raspberry PI Pico card, which will house the algorithm and will be the brain of said robot.

An Open MV Cam H7 module will be attached to provide the robot with artificial vision.

Through the implementation of the prototype in conjunction with an intelligent algorithm we achieve that the robot can perform its task in an ideal way.

Keywords: robot, prototype, sower, artificial vision, intelligent algorithm, field.

INTRODUCCIÓN

El presente trabajo de titulación ha basado su campo de estudio en la visión e inteligencia artificial, estas áreas han evolucionado mucho a través de los años y se han vuelto más accesibles para el público en general, de esta manera se ayudará a solucionar problemas y reducir los tiempos de trabajo en ciertas áreas.

Actualmente se cuenta con varios trabajos relacionados a la investigación.

El planteamiento del problema, justificación y los objetivos que se han planteado en el presente proyecto, se encuentran en el Capítulo 1.

En el Capítulo 2 se describen los temas y bases en los cuales se ha fundamentado la obtención de conocimientos necesarios para la elaboración del robot móvil, se detallan conceptos generales, la inteligencia artificial, creación de algoritmos mediante visión artificial, el tipo de tarjetas utilizadas para la adquisición y procesamiento de imágenes y una descripción breve de la ciencia de la robótica y sus complementos.

En el Capítulo 3 se detallan los materiales y los métodos utilizados para la implementación física del robot móvil, se explica el algoritmo de inteligencia artificial, el cual es el encargado de realizar las funciones móviles y el trabajo del robot.

Se explica brevemente el proceso de selección para lograr decidir qué tarjeta es la adecuada para cargar con la programación de la neurona y realizar la tarea de cerebro.

Finalmente, en el Capítulo 4, detallan las conclusiones y análisis de resultados obtenidos durante la realización del presente trabajo de titulación mediante el análisis experimental que a través de pruebas en un espacio controlado permite concluir en que se alcanzaron los objetivos planteados.

CAPÍTULO 1

ANTECEDENTES

1.1 Planteamiento del problema

En la carrera de Electrónica y Automatización de la Universidad Politécnica Salesiana, no se cuenta con un laboratorio equipado ni dirigido específicamente para la materia de sistemas inteligentes y para la materia de robótica móvil, en cuanto se trata de un robot móvil autónomo, el desconocimiento de tecnologías actualizadas genera un precedente negativo en la actividad académica de los participantes en instituciones de educación superior. (Katerine & Rodr, 2019);(Caverzasi et al., 2019)

La creación de una red de inteligencia artificial no puede quedarse únicamente en la teoría, ya que para estos temas es obligatorio y muy necesario la práctica y su implementación sobre un robot que se va a desenvolver en el campo real. (García S & Caranqui S, 2017)

En la última década, según (Bernal et al., 2020), especifica que la tecnificación en el área de la agricultura presenta un panorama desalentador en el Ecuador, esto se debe al crecimiento poblacional, el desempleo, falta de recursos desde el sector público y privado. Toda esta situación genera costos elevados en la producción agrícola tradicional y una posible afeción al derecho alimentario.

1.2 Justificación del proyecto

Un robot móvil autónomo evita la dependencia de un operador que dirija todos los movimientos que debe realizar el robot, creando un sistema automatizado el cual deja de depender de la manipulación del ser humano. (Cruz, 2018)

Un robot móvil sembrador, es útil para el apoyo lúdico e indiscutiblemente para la sinergia entre la parte teórica y la parte práctica, favoreciendo a operarios, estudiantes, profesores e investigadores para futuros desarrollos, esto se menciona debido a que se abarcan diferentes temas dentro de la materia de robótica móvil y sistemas inteligentes.

Según la página del ministerio de agricultura del Ecuador, el proceso de tecnificación en el campo es un área que cada día va en aumento, ayudando a los agricultores a mejorar sus productos y reducir los costos que conllevan el proceso de siembra en el país.

Al contar con un robot sembrador autónomo, se estima beneficios en ahorro de recursos, largas jornadas de trabajo redundante, costos de producción, etc. Es así como este trabajo de titulación pretende desarrollar un robot inteligente con características de hardware desmontable y software abierto. Además, este prototipo contará con visión artificial para la identificación de surcos en el traslado y realizar el sembrado de semillas según las instrucciones del agricultor. (Industria & Rob, 2021)

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un robot móvil autónomo para la siembra de semillas en el campo mediante inteligencia artificial.

1.3.2 Objetivos específicos

- Investigar los sistemas de locomoción, inteligencia artificial y estructuras de robot móvil para el desarrollo de un prototipo autónomo mediante la recopilación del estado del arte.
- Diseñar un robot móvil con una estructura útil para el desplazamiento sobre terrenos de sembrío mediante un software CAD.
- Desarrollar un algoritmo inteligente que permita la identificación de surcos en un terreno mediante visión artificial.
- Integrar el hardware y software en un robot móvil para la colocación de semillas de grano grande a través de una perforación sobre el terreno utilizando inteligencia artificial.

- Realizar las pruebas de funcionamiento del robot en siembra y desplazamiento para la validación de resultados mediante experimentos en un entorno controlado.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Introducción

En el presente capítulo se explican los conceptos básicos de robótica en los cuales se encuentra incluido un robot móvil autónomo, tomando como base sus requerimientos iniciales hasta realizar un análisis estructural para realizar la implementación de sus partes electrónicas y mecánicas.

También se cuenta con una descripción general de los conceptos utilizados en la elaboración del algoritmo inteligente aplicado a la visión artificial, el cual dotará al robot de la capacidad de realizar su trabajo de manera más eficiente ya que puede analizar el entorno en el que se encuentra.

2.2 Estado del arte

En Juchitán de Zaragoza, Oaxaca. 16 de marzo de 2016 se creó un robot sembrador el cual es dirigido y operado a distancia por una persona con una aplicación bluetooth. Sin embargo, este prototipo no asegura que el agricultor se aleje del robot debido a posibles desconexiones de la comunicación inalámbrica. (Carlos et al., 2019)

En el año 2016 el estudio realizado por el ingeniero Héctor Valencia muestra el desarrollo y comparación de diferentes métodos de locomoción para diferentes tipos de robot, se decidió por la locomoción esférica ya que era la que mejores resultados dio para el terreno en el cual se realizaron las pruebas. Se utilizó un terreno plano, pero con diferentes obstáculos y texturas; en estas pruebas el grado de inclinación del terreno no llegaba a afectar de manera directa en el desplazamiento del robot, por lo que se decidió no tomar este factor como determinante al momento de emitir un resultado; al contrario del torque obtenido por el coeficiente de rozamiento entre la rueda y el suelo (Benavides, 2019) Pese a esto (Alay & Moreno, 2017), mencionan en su estudio que cuando se trata de una superficie compuesta por rocas y lodo estos son un impedimento para la tracción delantera y llantas de menor radio, este problema se solucionó con el método de orugas; pero a la vez contaba con la desventaja de ser muy pesado y no brindar una movilidad omnidireccional.

2.3 Robot Móvil

El robot móvil autónomo es una máquina automática capaz de trasladarse en su entorno sin fijarse en ninguna ubicación física; es decir realiza una conexión inteligente entre las actividades cognitivas y de acción. De esta manera determina su comportamiento y le permite alcanzar objetivos programados en un entorno con determinadas irregularidades. (Cruz, 2018)

El grado de autonomía depende en gran medida de la capacidad del robot para abstraer el entorno y convertir la información obtenida en comandos y asegura el desempeño efectivo de su tarea.

A continuación, se enumeran las dos principales características que lo distinguen de cualquier otro tipo de vehículo:

Percepción: El robot móvil debe ser capaz de determinar su relación con su entorno de trabajo a través del sistema sensorial a bordo. La capacidad de percepción del robot móvil conduce a la síntesis de toda la información proporcionada por los sensores para crear mapas globales y locales del entorno bajo diferentes niveles de control. (García S & Caranqui S, 2017)

Razonamiento: Un robot móvil debe ser capaz de decidir las acciones necesarias en cualquier momento dependiendo del estado del robot y su entorno para lograr su objetivo.

El dominio de razonamiento de los robots móviles conduce a la planificación de trayectorias globales seguras y la capacidad de modificarlas en presencia de obstáculos inesperados (control de trayectoria local) para permitir que el robot logre su objetivo asignado. (Caverzasi et al., 2019)

Algunos de los ejemplos más representativos de robots móviles son aquellos utilizados en el área aeroespacial y de investigación del espacio. La NASA se ha encargado de crear varios Rovers a lo largo de la historia que han sido enviados en misiones de investigación a la luna. (Darley Leon, Milton Tenesca, 2019) Estos Rovers contaban con muchas características esenciales, inteligencia y visión artificial, fuente de energía renovable (paneles solares) lo que los dotaba de la capacidad de guiarse de manera autónoma en su misión y explorar una gran cantidad de terreno; contaban con varios

instrumentos como brazos robóticos los cuales le servían para recoger muestras de terreno y analizarlas al instante, sensores para evitar colisiones, termocuplas para determinar la temperatura ambiente y cámaras capaces de enviar video hacia la tierra. (Darley Leon, Milton Tenesca, 2019)

2.4 Sistema de locomoción

Un sistema de locomoción son los métodos que utilizan los robots para transportarse de un lugar a otro, estos muchas veces intentan simular el movimiento generado por los seres vivos, sean humanos o animales. (Alay & Moreno, 2017)

Existen diversos métodos de locomoción para robots, estos están dirigidos a diferentes tipos de suelo, ya que para elegir el método correcto se debe conocer las características del terreno en el que el robot se va a desplazar. (Benavides, 2019)

Para determinar las consideraciones del diseño se analiza su maniobrabilidad, controlabilidad, tracción, estabilidad, etc. Hay varios tipos de locomoción, estos se clasifican dependiendo el tipo de instrumento para el desplazamiento usado, entre ellas se encuentran las ruedas u orugas y las patas.

2.4.1 Ruedas/ cintas de deslizamiento

Cuando se usa ruedas u orugas los tipos de locomoción pueden ser:

Diferencial: no tiene ruedas directrices, se cambia de dirección modificando la velocidad de las ruedas izquierda y derecha

Síncrona: El control del robot está garantizado mecánicamente a los dos lados, moviendo al robot con control de velocidad entre cada motor.

Triciclo: Cuenta con tracción trasera y la rueda delantera se encarga del direccionamiento del robot.

Ackerman: El sistema se basa en dos ruedas traseras tractoras que se montan de forma paralela en el chasis principal del vehículo, mientras que las ruedas delanteras son del tipo direccionamiento, y se utilizan para seguir la trayectoria del robot.

Omnidireccionales: Cuenta con 3 o 4 llantas las cuales pueden moverse en cualquier dirección, variando el sentido y velocidad de estas, garantizan un movimiento libre del robot. (Alay & Moreno, 2017).

Patas: Estos métodos de locomoción se clasifican por el número de patas que cuente el robot, si es bípedo, cuadrúpedo, hexápodo, etc. Generalmente se utilizan para el desplazamiento en terrenos accidentados ya que al contar con un desplazamiento mecánico el cual consta de una elevación de las extremidades, se pueden evitar obstáculos que con otros métodos no se puede.

2.5 Visión artificial

Es una disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina. (Romero et al., 2017)

Un sistema de visión artificial intenta emular una escena (imagen), para entenderla y actuar referente al entorno que se le presente, es decir, la máquina puede percibir lo que hay a su alrededor para automatizar el proceso de percepción visual mediante el tratamiento de imágenes, esto se lleva a cabo mediante 6 procesos detallados a continuación. (García S & Caranqui S, 2017)

2.5.1 Captura: Proceso donde se obtiene una imagen a través de un dispositivo de cámara digital.

2.5.2 Preprocesamiento: Proceso de filtración el cual sirve para la reducción de ruido, mejoramiento del contraste y la nitidez de la imagen.

2.5.3 Segmentación: Proceso donde se divide una imagen en objetos o regiones (grupo de píxeles), que sean el interés de estudio.

2.5.4 Descripción: Proceso donde se obtienen características relevantes convenientes para distinguir un objeto de otro.

2.5.5 Reconocimiento: Proceso que clasifica en categorías los objetos presentes en la imagen utilizada en el proceso de descripción.

2.5.6 Interpretación: Proceso que da sentido o un significado a las categorías de objetos para entender la escena.

2.6 Inteligencia artificial (IA)

Se encarga del estudio de la inteligencia programada y creada por el ser humano para parecerse lo más posible tanto en comportamiento como en pensamiento.

En el punto de vista de la ingeniería la IA pretende construir sistemas y máquinas que ayuden a resolver problemas sin la necesidad de ser supervisada por un operador, aprendiendo y adaptándose a su entorno. (Cruz, 2018)

2.7 Redes neuronales

Una red neuronal se define como un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas. (Izaurieta & Saavedra, 2019)

Los elementos que componen una red neuronal artificial son:

Función de entrada (Input function): La neurona trata de muchos valores de entrada como si fuera uno solo esto recibe el nombre de vector de entrada global donde el objetivo principal es combinar estas simples entradas en una sola.

Función de activación (Activation function): Una neurona biológica puede estar activa o inactiva de la misma forma que una neurona artificial, calcula el estado de actividad de una neurona transformando la entrada global en un estado (“1” ACTIVO, “0” INACTIVO).

Función de salida (output function): La función de salida determina qué valor se transfiere a las neuronas vinculadas

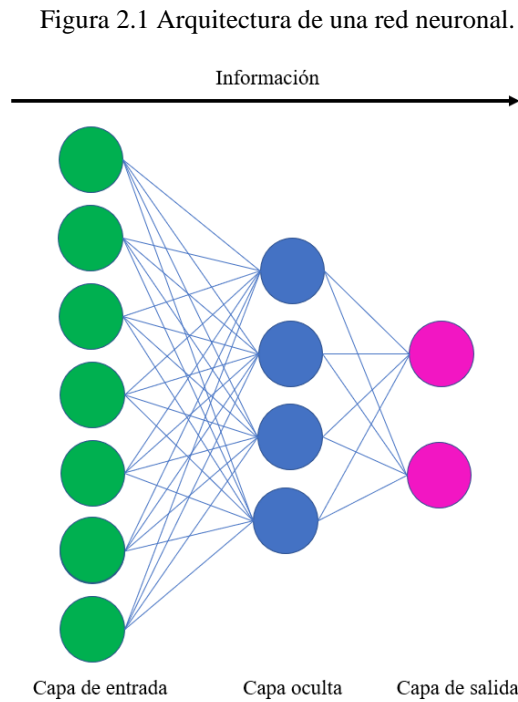
2.8 Arquitectura de una red neuronal

Se denomina arquitectura a la forma en cómo está estructurada la red neuronal, es denominada por su topología o su tipo de conexión.

Una red neuronal artificial está conectada por medio de nodos que realizan un tipo de conexión sináptica, este tipo de conexión crea patrones de dirección de un solo sentido; el cual parte desde la zona presináptica a la post sináptica. (Larrañaga, 2020).

2.9 Capas de una red neuronal

la red neuronal se compone de distintas unidades de procesamiento las cuales son denominadas capas, generalmente se compone de 3 capas principales:



Ejemplo de arquitectura unidireccional con 3 capas. Elaborado por: David Quiroga, Boris Rubio.

2.9.1 Capa de entrada

Está compuesta por neuronas que se encargan de recibir los datos del entorno, esta también es denominada la capa sensorial. (Larrañaga, 2020)

2.9.2 Capa de salida

Está compuesta por neuronas que se encargan de enviar un resultado generado por la red neuronal.(Larrañaga, 2020)

2.9.3 Capa oculta

Esta se encarga de proporcionar los grados de libertad a la red neuronal, no se conecta directamente con el entorno ni a ninguno de los sensores o actuadores que estén conectados a esta, las unidades contenidas dentro de esta capa no son observables por el usuario. (Larrañaga, 2020)

2.10 Red neuronal convolucional

Es un tipo de arquitectura de red diseñada para el entrenamiento profundo mediante la cual la red neuronal aprende directamente de los datos proporcionados en la fase de entrenamiento, sin la necesidad de sacar características de forma manual.

Esta red es generalmente usada en aplicaciones de reconocimiento de objetos y en visión artificial. (Larrañaga, 2020)

Procesan datos con una tipología conocida como cuadrícula, estos datos son analizados y una vez transformados en matrices, se realizan convoluciones de la señal de entrada con respecto a una función de transferencia ubicada a la salida de la neurona, esta se compone de 3 capas, la capa convolucional, capa pooling y capa Fully-conected. (Jalomo, 2019).

a) Capa convolucional

Esta capa es considerada como la más importante de una red neuronal convolucional, se divide en 2 capas diferenciadas por el tipo de conexión que lleva cada neurona.

En la primera capa convolucional las neuronas no se encuentran conectadas a cada pixel de la imagen de entrada si no, únicamente a los pixeles que estén en su campo receptivo.

En la segunda capa convolucional las neuronas se conectan a las neuronas localizadas en una zona específica de la primera capa, esto permite que la red se enfoque en características de bajo nivel, esta ensambla un modelo superior con más datos y los pasa a la siguiente capa oculta, este proceso se repite sucesivamente.

b) Capa Pooling

Esta capa ayuda a reducir la carga computacional al procesar una imagen de entrada, optimiza el uso de memoria y el número de parámetros a identificar, aquí las neuronas están conectadas a una salida específica de neuronas en capas previas, obtiene un valor máximo o promedio de todo este conjunto de datos.

c) Capa Fully-connected

En esta capa todas las neuronas de la capa oculta se conectan a todas las neuronas de la capa anterior y siguiente.

Figura 2.1 Diagrama de capas

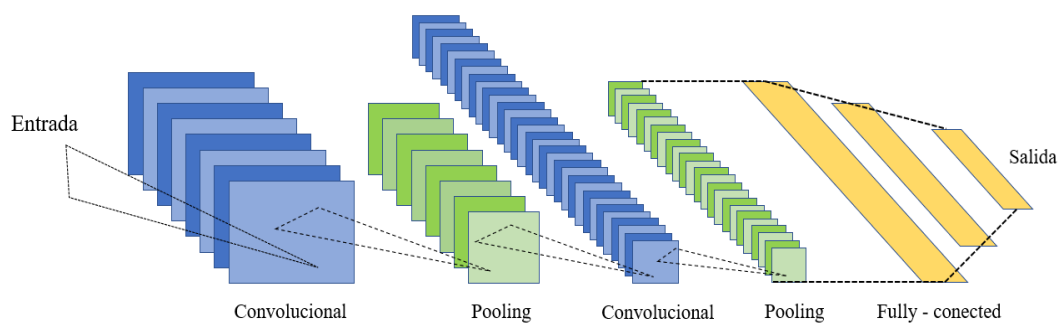


Diagrama de las capas de una red neuronal convolucional. Elaborado por: David Quiroga, Boris Rubio.

2.11 Entrenamiento de una red neuronal

El entrenamiento de una red neuronal consiste en ajustar cada uno de los parámetros de entrada de todas las neuronas que forman parte de la red neuronal, para que entregue respuestas que se ajusten a datos conocidos por el usuario.

Para que una red neuronal sea capaz de identificar un dato en específico es necesario realizar iteraciones de entrenamiento con imágenes, es importante utilizar una gran cantidad de imágenes para que la red neuronal asocie de mejor manera los datos y los resultados sean lo más preciso posible. (Issn, 2018)

2.12 Entrenamiento profundo

También conocido como Deep learning, es un subconjunto del aprendizaje automático que intenta asemejar la manera en cómo funciona el cerebro humano, se lo llama profundo ya que cuenta con varias capas de neuronas con distinto peso las cuales ayudan a que la red neuronal logre tomar decisiones. (Issn, 2018)

2.12.1 Entrenamiento

En esta fase el programador define manualmente el número de capas y de neuronas que tendrá la red neuronal, de esta manera la red neuronal puede aprender por sí misma a diferenciar entre datos válidos e inválidos. Aquí se debe mostrar a la red neuronal una gran cantidad de imágenes con un subconjunto de etiquetas o parámetros que ésta deberá aprender para lograr un reconocimiento más preciso, una vez finalizada esta etapa tendremos como resultado una red neuronal “capacitada”.(Issn, 2018)

2.12.2 Inferencia

En este proceso la red neuronal debe evaluar nuevas imágenes para generar una respuesta con cierto grado de seguridad.(TELEDYNE, 2020)

2.13 Pasos para realizar un aprendizaje profundo

El desarrollo de una aplicación de aprendizaje profundo se logra en 5 pasos.

2.13.1 Identificación de la función de aprendizaje

En este paso debemos clasificar varias funciones para que la neurona sea capaz de aprender con visión artificial.(TELEDYNE, 2020)

a) Clasificación

Es considerada la función fundamental de aprendizaje ya que aquí es donde la red neuronal es capaz de clasificar imágenes en diferentes grupos según sus características en común.

b) Detección y localización

Con esta función la red neuronal es capaz de identificar características en una imagen y delimitar en un cuadro la ubicación y tamaño.

c) Segmentación

Esta función brinda a la neurona la capacidad de identificar objetos en específico en una imagen y saber los pixeles que corresponden a cada uno de ellos.

2.13.2 Selección de la plataforma

Cuando el programador seleccione qué función de aprendizaje profundo utilizará para entrenar a la red neuronal, necesitará de una plataforma que cuente con las herramientas adecuadas para que se adapte a las necesidades del programa de manera correcta.

Para seleccionar la plataforma de forma adecuada hay que tomar en consideración la complejidad de la tarea que va a realizar y la velocidad de respuesta que se necesita en la aplicación, así como también el número de capas y de neuronas que necesitará.(TELEDYNE, 2020).

2.13.3 Preparación de datos

En este paso el programador seleccionará un banco de imágenes que cuenten con la mayor cantidad de características que el usuario necesite utilizar en su aplicación y estas deben mostrarse de manera apropiada.

Se debe tomar en consideración que el nivel de entrenamiento de la red neuronal está relacionado de manera directamente proporcional al número de imágenes que ésta recibió durante su fase de entrenamiento, es decir a mayor número de imágenes más exactos serán los resultados generados por la red neuronal. (TELEDYNE, 2020).

2.13.4 Entrenamiento y validación

Una vez que se hayan preparado todos los datos necesarios para el entrenamiento de la red neuronal, se necesitará probar y validar la precisión de esta, aquí se debe configurar y ejecutar scripts hasta que el proceso de entrenamiento brinde niveles de precisión aceptables para la aplicación específica.(Issn, 2018)

También se puede utilizar una red neuronal entrenada previamente para que realice una transferencia de aprendizaje, es decir no es necesario volver a entrenar una red neuronal desde cero, sino que se puede añadir características extras solo a algunas capas de esta. (TELEDYNE, 2020)

2.13.5 Implementación de la red neuronal

En este paso se implementará la red neuronal entrenada en el hardware seleccionado previamente para poder evaluar su desempeño y recoger todos los datos. (Issn, 2018).

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DE ROBOT SEMBRADOR

El objetivo principal del caso de estudio es la implementación de un robot autónomo destinado a la siembra de semillas en el campo, mediante visión e inteligencia artificial, esta tarea requiere el uso y el procesamiento de imágenes en tiempo real.

Cuenta con una estructura diseñada para que el robot albergue todos los periféricos de manera óptima para que no dificulten su tarea y lo puedan poner en correcto funcionamiento.

3.1 Selección del controlador

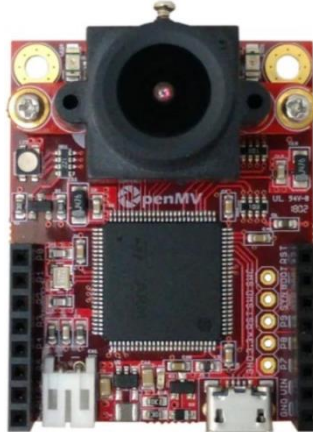
Para seleccionar el controlador de manera correcta se optó por la investigación y prueba entre diversas tarjetas. Por lo que se llevó a cabo un proceso de selección con criterios que ayuden a la implementación del proyecto tanto en hardware como en software.

a) OpenMV Cam H7

Es una placa microcontroladora pequeña y de baja potencia que le permite implementar aplicaciones utilizando visión artificial de manera sencilla.

Cuenta con un procesador ARM Cortex M7 que funciona a 480 MHz con 1 MB de SRAM y 2 MB de flash. Todos los pines de E / S producen 3.3V y son tolerantes a 5V. Las ventajas con las que cuenta esta tarjeta es que se puede programar en scripts de Python de alto nivel lo que facilita el manejo de los resultados complejos de los algoritmos de visión artificial y el trabajo con estructuras de datos de alto nivel, su costo es de 100 USD disponible en el mercado ecuatoriano. (Agyeman Ibrahim; Abdelkader Kwabena., 2019)

Figura 3.1 Tarjeta OpenMV Cam H7



Tarjeta de desarrollo Open-MV Cam H7, Fuente (Agyeman Ibrahim; Abdelkader Kwabena., 2019)

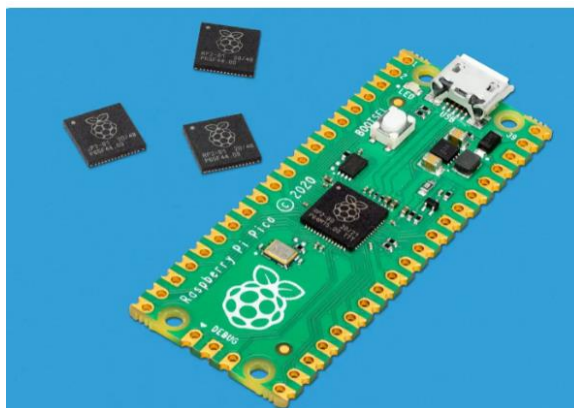
b) Raspberry Pico

Es una tarjeta de desarrollo basada en el microcontrolador RP2040, diseñada para ser una plataforma de desarrollo muy flexible y de bajo costo.

presenta un procesador ARM Cortex-M0 + de doble núcleo con RAM interna de 264kB y soporte para hasta 16MB de flash fuera de chip. Una amplia gama de opciones de E / S flexibles incluye I2C, SPI y, de forma exclusiva, E / S programables (PIO).

Su costo es de 12.50 USD en el mercado ecuatoriano.(Gp et al., 2018)

Figura 3.2 Tarjeta Raspberry Pi Pico



Tarjeta de desarrollo Raspberry Pi Pico, Fuente (Gp et al., 2018)

c) Arduino mega 2560

Es una placa de desarrollo basada en el microcontrolador atmega 2560 el cual cuenta con 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, jack para alimentación DC, conector ICSP, y un botón de reseteo. La placa Mega 2560 es compatible con la mayoría de shields compatibles para Arduino UNO. (Mellis, 2011)

Figura 3.3 Tarjeta Arduino Mega 2560



Tarjeta de desarrollo Arduino Mega, fuente (Mellis, 2011)

Para seleccionar la tarjeta de desarrollo se realizó una tabla con el objetivo de clasificar cada una de las tarjetas basado en sus características y dependiendo de las necesidades del robot sembrador.

Hay que tomar en cuenta que la tarjeta debe ser capaz de conectarse a una cámara y manejar redes neuronales para que sea factible controlar componentes externos como los motores tanto para su desplazamiento, para el elevador y para el dispensador.

A continuación, se muestran en la Tabla 3.1 las características de cada tarjeta para que se vea de manera simple y sea más fácil tomar una decisión de con cuál tarjeta se va a trabajar.

Tabla 3.1 Resumen de clasificación de las tarjetas de desarrollo

Tarjeta de desarrollo	Soporte de conectividad para cámara	Soporte de redes neuronales	Disponibilidad en Ecuador	Bajo costo
Open MV Cam H7	X	X	X	
Raspberry Pi Pico		X	X	X
Arduino mega 2560			X	X

Tabla de clasificación, Realizado por David Quiroga, Boris Rubio.

Una vez identificadas las características de cada una de las tarjetas se verifican que las tarjetas más aptas para el robot son la Raspberry Pi Pico y la Open MV Cam H7.

Se eligieron las 2 tarjetas ya que las características de cada una las permiten trabajar juntas y ambas son compatibles con Python de alto nivel.

En la parte del costo la Raspberry Pi Pico y la Arduino Mega 2560 están bajo los 50 USD lo que las hace accesibles, pero debido a las características de conectividad, soporte para redes neuronales e inteligencia artificial y estar fabricada exclusivamente para ser acoplada a una cámara se decidió utilizar la Open MV Cam H7 y dejar de lado al Arduino Mega 2560 ya que es una tarjeta que no soporta redes neuronales y tampoco tiene un módulo de cámara es capaz de analizar las imágenes en tiempo real.

3.2 Diseño electrónico

La tarjeta Open MV Cam H7 es quien se encargará del reconocimiento de imágenes y en esta se realizará el algoritmo de visión artificial.

El módulo Cam H7 va acoplado a la Raspberry, quién se encarga de realizar todo el trabajo de locomoción y ejecución de tareas, tanto de perforación como de dosificación de semillas; lleva acoplados dos módulos L298N que sirven como driver para los motores.

Un driver controla los motores de las ruedas y el otro controla el motor de la broca quién realiza la perforación en el suelo, cuenta además con un módulo de alimentación, que entrega el voltaje necesario a cada componente instalado sobre el chasis del robot. (Electronics, 2021)

a) Sistema sensorial

El robot cuenta con diferentes módulos para la adquisición de datos del entorno.

Para la obtención de datos gráficos del campo utiliza el módulo cam H7 compatible con la tarjeta OpenMV, con una comunicación y control directos a través del interfaz del bus de la cámara, de la Figura 3.1

b) Sistema de control

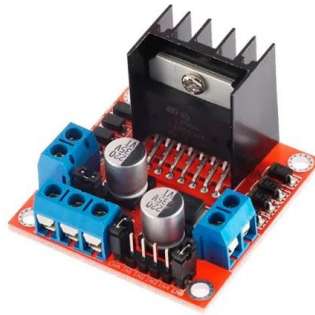
Para el control de desplazamiento y de la programación del algoritmo de inteligencia artificial, se utilizó la tarjeta de desarrollo Raspberry Pico, de la Figura 3.2.

Para el control de motores se eligió el módulo L298N, de la Figura 3.4; se utilizaron 2 ya que cada módulo es capaz de controlar únicamente 2 motores, un driver controla los 2 motores de desplazamiento del robot y el otro driver controla el motor que se encarga de realizar las perforaciones en el suelo.

A continuación, características del módulo controlador de los motores:

- Tiene la capacidad de controlar hasta 2 motores DC o un motor paso a paso bipolar.
- Puede controlar motores entre 5V y 35V.
- Capacidad máxima de hasta 2A
- Es compatible con casi cualquier microcontrolador, ya que no tiene una librería establecida.
- Cuenta con diferentes modos de alimentación dependiendo del voltaje que requiera el motor que va a ser utilizado.

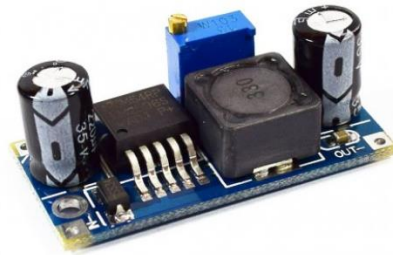
Figura 3.4 Módulo controlador de motores L298N.



Módulo driver L298N, Fuente (Electronics, 2021)

Se utilizó un módulo LM2596 (Figura 3.5) para la regulación de voltaje a 5V ya que las tarjetas de desarrollo trabajan con 5 voltios y no con la alimentación de 12V que entrega la batería. (Electronics, 2021)

Figura 3.5 Regulador LM2596



Módulo regulador para 5V LM2596, Fuente: (Naylamp Mechatronics, 2021)

c) Motores.

Para la selección de los motores hay que tomar en cuenta varios parámetros, siendo los más importantes la velocidad y el torque requeridos para el trabajo, al ser un robot de siembra no requiere una velocidad muy alta, necesita un torque alto, ya que al trabajar en un terreno accidentado como lo es el campo se necesita fuerza para que se pueda desplazar sin que sufra problemas en su trayectoria generados por factores externos del terreno; se eligió el motor JGB37-520 cuyas características son mencionados en la Tabla 3.2.

Para el dispositivo elevador y para el dosificador de semillas se eligió usar un motor a pasos ya que este permite tener una trayectoria controlada con mucha precisión y así evitamos daños en el soporte, se utilizó el motor a pasos 28BYJ-48, cuyas

características son mencionadas en la Tabla 3.2. Para realizar las perforaciones sobre el terreno se utilizó una broca plana de 7/8 con un eje de 8 mm, el cual se conecta al motor con un acople de aluminio; ya que el robot está diseñado para perforar superficies accidentadas de las cuáles no se tiene un previo conocimiento de su dureza, se optó por utilizar un motor ZGB37RG el cual tiene una caja reductora que brinda mayor torque, las características de este motor se mencionan en la Tabla 3.2.

Tabla 3.2 Características motores

Motor	Función	Voltaje de trabajo	RPM motor	Relación	Disponible en Ecuador
JGB37-520	Desplazamiento	12 V	66 RPM	0.12 / 7Kg.cm	Sí
ZGB37RG	Perforación	12V	285 RPM	1/714 /7.2 Kg.cm	Sí
28BYJ-48	Elevador, dispensador	5V	-	1/64	Sí

Tabla de características de motores utilizados, Realizado por David Quiroga, Boris Rubio.

d) Ruedas.

Para el desplazamiento del robot se eligieron las ruedas Dagu Wild Thumper Wheel de 120x60mm, están fabricados de goma blanda con púas lo que brinda mayor tracción lo que las convierte en un modelo A/T de alta fiabilidad en terreno accidentado, de la Figura 3.6.

Figura 3.6 Ruedas Dagu Wild Thumper Wheel



Ruedas Dagu Wild Thumper Wheel, Pololu, Fuente (Pololu Robotics & Electronics, n.d.)

e) Fuente de alimentación.

Para la alimentación del robot se utilizó una batería de la marca First Power (Figura 3.7) de 12V y 2.8 Ah (amperios hora) batería con la que se tiene una autonomía de aproximadamente 3 horas de uso continuo.

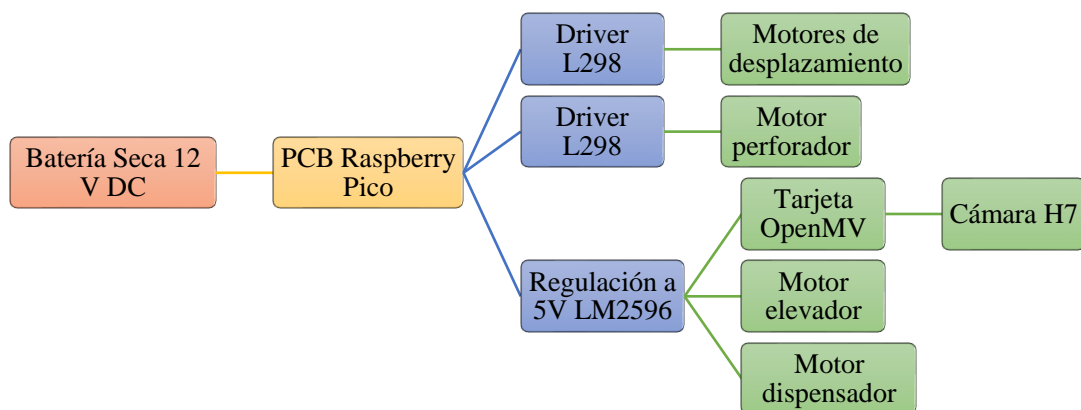
Figura 3.7 Batería First Power 12V / 2.8 A



Batería First Power FP 1228 recargable, Fuente: (Duro, n.d.)

3.3 Esquema de alimentación.

Figura 3.8 Esquema de alimentación del robot



Esquema de alimentación robot sembrador, Elaborado por David Quiroga, Boris Rubio.

3.4 Diseño estructural

El diseño estructural se planteó a partir de uno de los objetivos, el cual era utilizar elementos reciclados para hacer al robot amigable con el medio ambiente y tratar de reducir costos en la parte económica.

Las dimensiones del robot tuvieron que ser tomadas en consideración a través de factores como el tamaño de los surcos a los cuales está destinada a la siembra, brindar el espacio necesario para todos los componentes.

a) Diseño del chasis

El chasis se realizó a partir de tubería PVC (Figura 3.9), ya que ésta es ligera y resistente, lo que reduce el peso total del robot permitiendo un margen más alto en la capacidad de carga de este mismo.

Para la base en la cual se montarán los componentes electrónicos (tarjeta de control Open MV Cam H7, Raspberry Pi Pico, batería, cableado), elementos electromecánicos (motores) y todos los elementos móviles (ruedas, rueda seguidora).

En la parte inferior del chasis se colocó 2 codos PVC de 1 in, los cuales sirven de soporte para los motores que se encargan del desplazamiento del robot.

Figura 3.9 Chasis del robot con la base



Chasis del robot con la base de madera, Elaborado por: David Quiroga, Boris Rubio

b) Diseño del elevador

El elevador está fabricado de madera, este tiene una estructura tipo riel que consta de 2 piezas, una móvil que se desliza sobre una pieza fija, la cual va acoplada al chasis mediante 2 soportes metálicos y pernos auto perforantes.

En cada una de las piezas se acoplaron componentes diferentes los cuales son detallados a continuación:

Cara A pieza fija: Se cortó un rectángulo de MDF de 0.9 mm de espesor, las medidas son de 200 x 100 mm, se acopló una cremallera impresa en 3D, se muestra en la Figura 3.10

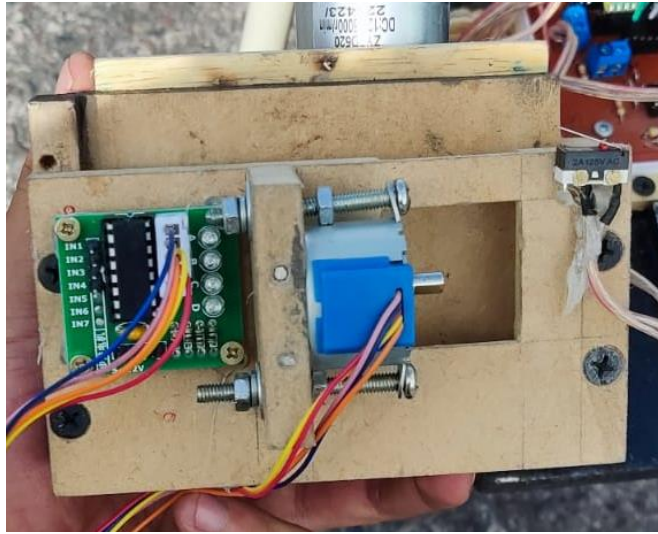
Figura 3.10 Pieza fija Cara A



Cremallera pegada sobre la Cara A de la pieza fija, Elaborado por: David Quiroga, Boris Rubio

Cara A pieza móvil: Se colocó un motor a pasos, el cual lleva un engranaje dentado diseñado en 3D para que suba y baje sobre su cremallera.

Figura 3.11 Pieza móvil cara A



Motor a pasos con su controlador acoplado en la cara A de la pieza móvil, Elaborado por: David Quiroga, Boris Rubio.

Cara B pieza móvil: Se fijó un motor acoplado a una broca que se encarga de realizar las perforaciones en el suelo.

Figura 3.12 Pieza móvil cara B



Motor perforador con su acople de aluminio fijado a la cara B de la pieza móvil, Elaborado por: David Quiroga, Boris Rubio.

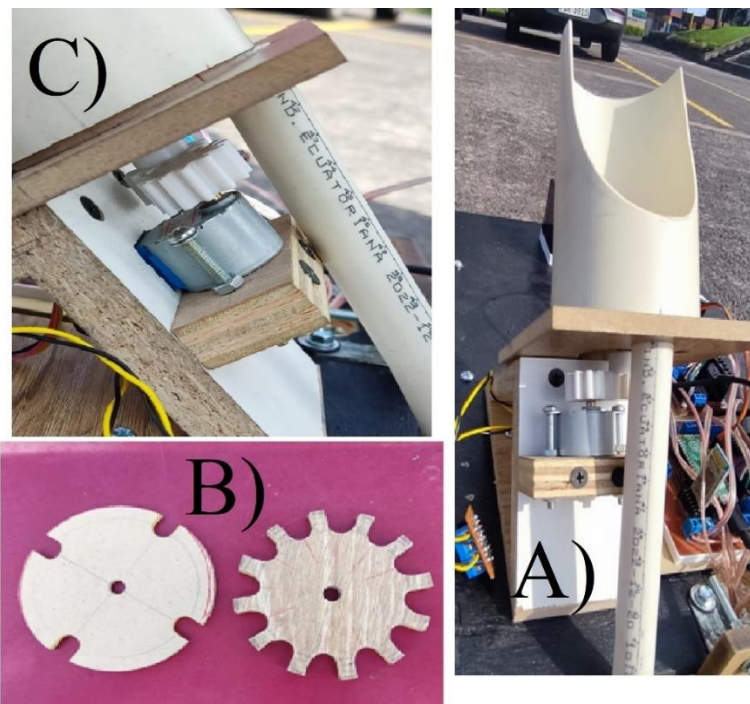
c) Diseño del dispensador de semillas

Para el dispensador se utilizó un tubo PVC de 3 in con un corte lateral en el cual se cargarán las semillas.

En la base del tubo PVC (Figura 3.13) se acopló un trozo de madera con una perforación de 12 mm de diámetro y un mecanismo de dosificación fabricado con aglomerado de 8 mm de espesor, con perforaciones en “U” de 2 cm en los 4 puntos cardinales.

Este sistema se montó en una base de madera en triángulo, sobre ella se encuentra otra base que soporta un motor a pasos acoplado con un engranaje que mueve el mecanismo de dosificación y de esta manera se depositan en el campo las semillas, de 2 a 3 unidades ya que así requiere la jornada de siembra.

Figura 3.13 Dispensador de Semillas



(A)Dispensador de semillas, (B)Piezas rotatorias, (C)Engranajes y motores a paso, Elaborado por: David Quiroga, Boris Rubio.

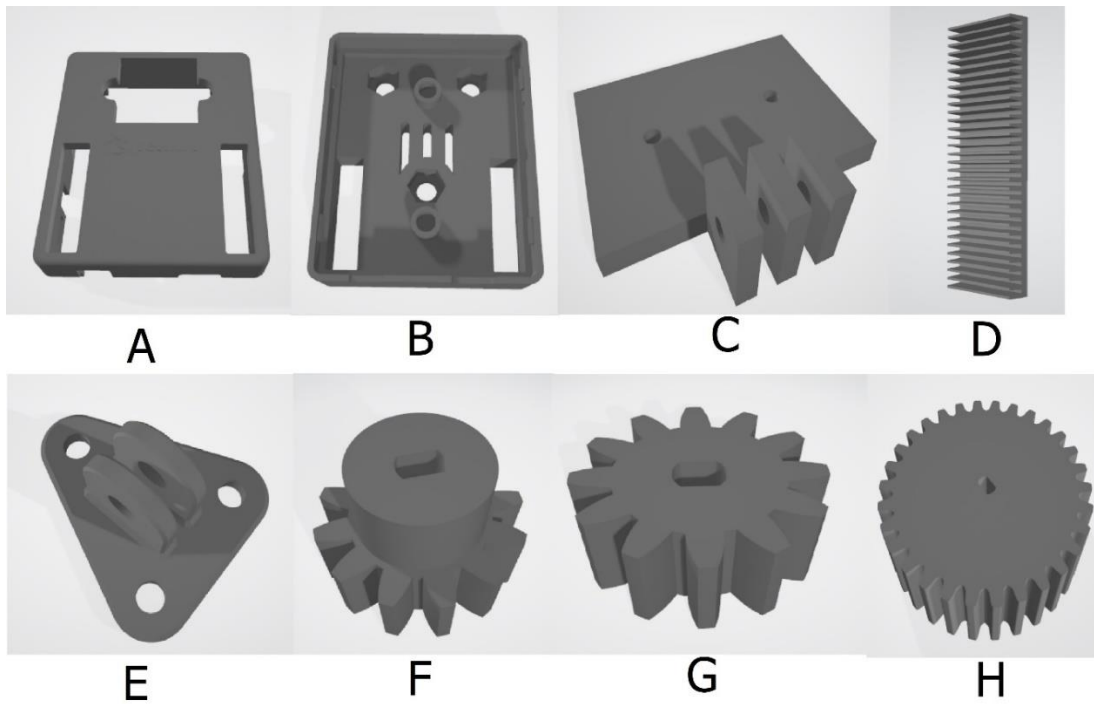
d) Modelado 3D de las piezas

Se realizó un modelado 3D (Figura 3.14) de los engranajes que van acoplados a los ejes de los motores correspondientes al trabajo que van a realizar; así como también

se modeló en 3D la cremallera sobre la cual se va a deslizar el engranaje que realizará el movimiento para subir y bajar la broca al momento de realizar las perforaciones en el suelo donde se depositarán las semillas.

Se modeló el estuche para la cámara H7 y el soporte que lo ancla al chasis del robot se muestran en la Figura 3.14.

Figura 3.14 Piezas diseñadas en 3D



(A)Tapa del estuche de la cámara, (B)Parte inferior del estuche de la cámara, (C) Acople de la cámara para el chasis, (D) cremallera del elevador, (E) soporte para la cámara, (F) y (G) engranajes para el dispensador, (H) engranaje para el elevador; Elaborado por: David Quiroga, Boris Rubio.

e) Impresión de las piezas

Se utilizó una impresora Anycubic Kobra Plus, esta impresora tiene un volumen de impresión muy alto con compatibilidad para filamento PLA que brinda mayor resistencia y dureza a las piezas.

Puede imprimir a 100mm por segundo, lo que da una velocidad alta y reduce el tiempo en el que cada pieza estará lista.(Calentada et al., n.d.).

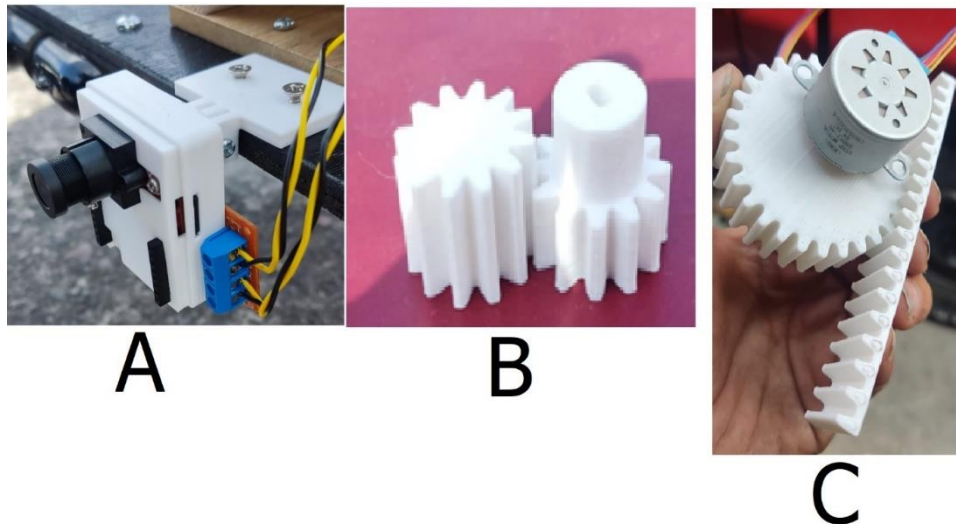
Se imprimió en lotes diferentes para realizar el montaje de cada parte de manera secuencial, así se podría modificar y reimprimir en caso de que alguna salga con alguna falla sin afectar un lote grande.

Se imprimió primero la cremallera con su engranaje (Figura 3.15), la cantidad de material utilizado para imprimir estas 2 piezas fue de 135 gramos, con un tiempo de 4 horas y 7 minutos.

El segundo lote de impresión fue para los 2 engranajes para el sistema dispensador de semillas (Figura 3.15), se utilizó 120 gramos de material con 100% de relleno, para evitar que queden cavidades huecas dentro de cada engranaje, el tiempo empleado en esta impresión fue de 3 horas con 52 minutos.

El último lote de impresión fue para el estuche y soportes de la cámara H7 (Figura 3.15), se imprimió usando 225 gramos de material, con un tiempo de 5 horas con 28 minutos.

Figura 3.15 Piezas impresas



(A) Estuche y soportes para la cámara H7, (B) Engranajes para el dispensador, (C) Cremallera y engranaje del elevador. Elaborado por David Quiroga y Boris Rubio.

f) Ensamblaje del robot móvil sembrador

Para el ensamblaje del robot sembrador se procede a unir los tubos PVC de $\frac{1}{2}$ in, dando una forma rectangular como se muestra en la Figura 3.9, se fijó la base sobre la cual van todos los demás componentes del robot; se unió los motores a las ruedas usando los acoples que venían incluidos con estas (Figura 3.16), y se unió al robot usando como soportes, neplos pegables de PVC de 1 in. (Figura 3.16)

Figura 3.16 Ruedas y motores de desplazamiento



A



B

(A) Acople de las ruedas a los motores, (B) Motores y ruedas fijados en el chasis del robot. Elaborado por David Quiroga, Boris Rubio.

A continuación, se montó el dispositivo elevador (mostrado en las Figuras 3.10, 3.11 y 3.12), el dispositivo dispensador de semillas (Figura 3.13) sobre el chasis y se procedió a fijarlos con sus respectivas bases, una vez montados estos dispositivos, se fijó la batería al chasis y se colocó todos los elementos electrónicos y de control. Figura 3.17

Una vez montados todos los componentes tanto mecánicos como electrónicos se procede a realizar el cableado de estos, tomando en consideración que el mecanismo elevador requiere una longitud de cable extra y suelto para que pueda realizar su desplazamiento sin enredarse o jalar un cable.

Cuando el robot está ensamblado en su totalidad, se puede pasar a la fase de acondicionamiento físico, es decir detalles y terminados estéticos, que brindarán un mejor acabado visual del robot.

Y así presentar el prototipo final, Figura 3.18

Figura 3.17 Elementos electrónicos



Parte superior del robot, elementos electrónicos, placas de desarrollo y batería. Elaborado por David Quiroga, Boris Rubio.

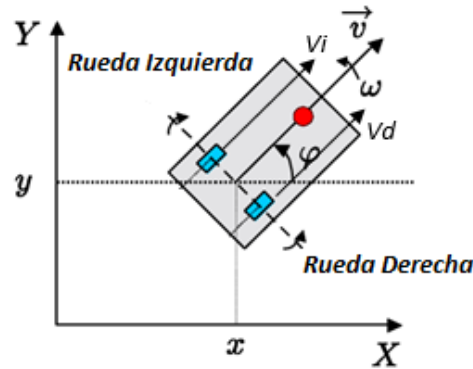
Figura 3.18 Prototipo Final



Ensamblaje final del robot, con todos sus componentes electromecánicos. Elaborado por David Quiroga, Boris Rubio.

3.5 Cinemática del robot inteligente

Figura 3.19 Modelo cinemático



Modelo Cinemático Elaborado por: David Quiroga, Boris Rubio.

Para el accionamiento y trayectoria del robot se utilizaron las ecuaciones de cinemática directa como se muestra a continuación:

$$\begin{bmatrix} \dot{x} \\ w \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2b} & \frac{r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\varphi}d \\ \dot{\varphi}i \end{bmatrix} \quad (1)$$

$$\dot{x} = \frac{r}{2}(\dot{\varphi}d - \dot{\varphi}i) \quad (2)$$

$$w = \frac{r}{2b}(\dot{\varphi}d - \dot{\varphi}i) \quad (3)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ w \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ 0 & 0 \\ \frac{1}{b} & \frac{1}{b} \end{bmatrix} \begin{bmatrix} \dot{\varphi}d \\ \dot{\varphi}i \end{bmatrix} \quad (4)$$

Donde r es la distancia de gira hacia el ICR, b es la distancia desde el centro del robot al centro de la llanta, w es la velocidad angular, \dot{x} es la velocidad lineal y θ es el ángulo de giro.

Las ecuaciones que se utilizan para calcular la velocidad con respecto al sistema del robot se simplifican de la siguiente manera:

$$\dot{x} = \frac{r}{2}(\dot{\varphi}d \cos\theta - \dot{\varphi}i \cos\theta) \quad (5)$$

$$\dot{y} = 0 \quad (\text{robot no holonómico}) \quad (6)$$

$$w = \frac{r}{2b}(\phi d - \phi i) \quad (7)$$

Para el control del robot se tomó como entrada la velocidad de la llanta derecha Vd e izquierda Vi se asume que la $Vd=Vi$, velocidades lineales para cada rueda se obtienen con las siguientes ecuaciones:

$$Vx = \frac{Vd + Vi}{2} \text{Cos } \theta \quad (8)$$

$$Vy = \frac{Vd + Vi}{2} \text{Sen } \theta \quad (9)$$

$$W = \frac{Vd - Vi}{L} \quad (10)$$

Con las ecuaciones en Vx , Vy y W respecto a la velocidad de la rueda izquierda y derecha se toma el centro del chasis hasta el centro de la rueda $L= 2b$

$$Vx = V \text{cos } \theta \quad (1)$$

$$Vy = V \text{sen } \theta \quad (12)$$

Se utiliza el método de igualación con la ecuación (8) y (11) como se muestra a continuación

$$Vx = \frac{Vd + Vi}{2} \text{Cos } \theta = V \text{cos } \theta \quad (13)$$

$$\frac{Vd + Vi}{2} = V \quad (14)$$

$$Vd = 2V - Vi \quad (15)$$

$$Vi = 2V - Vd \quad (22) \quad (16)$$

El método de sustitución de la ecuación 14 en la ecuación 10 nos da como resultado las velocidades en cada llanta.

$$WL = 2V - Vi - Vi \quad (17)$$

$$WL = 2V - 2Vi \quad (18)$$

$$Vi = \frac{2V + WL}{2} \quad Vi = V - \frac{WL}{2} \quad (19)$$

$$Vd = \frac{2V + WL}{2} \quad Vd = V + \frac{WL}{2} \quad (20)$$

Tomando como parámetros $W=0$ la velocidad sea constante se puede evidenciar que la $Vi=Vd$, las dimensiones físicas del robot móvil implementado $L=56 \text{ cm} = 0.56 \text{ m}$ y con las velocidades lineal determinada $V=13.23 \text{ cm/s}$ y la velocidad angular $W=0$ para que el robot se dirija hacia adelante.

$$Vi = 13.23 \frac{\text{cm}}{\text{s}} - \frac{0(56 \text{ cm})}{2} = 13.23 \text{ cm/s} = Vd \quad (21)$$

Para cuando el robot móvil gire la velocidad Angular $W=$ con la misma velocidad $V=13.23 \text{ cm/s}$

$$Vi = 13.23 \text{ cm/s} - \frac{4^\circ/\text{s}(56 \text{ cm})}{2} = 98.77 \text{ cm/s} \quad (22)$$

Las ecuaciones de la cinemática inversa se implementan en el controlador de velocidad y posición del robot móvil, determinando una velocidad lineal hacia adelante \dot{x} y una velocidad angular w constantes, ya que no es un robot velocista se puede mantener estas variables constantes.

3.6 Red neuronal

Se utilizó una red neuronal convolucional, para la programación y la implementación de la red neuronal se tienen 3 bloques que se encargan de su funcionamiento, está compuesta de 3 bloques principales (Figura 3.19)

La entrada del sistema será un cúmulo de imágenes que se enviarán a través de una capa oculta para después obtener las salidas.

Figura 3.20 Diagrama de bloques de la arquitectura de la red neuronal.

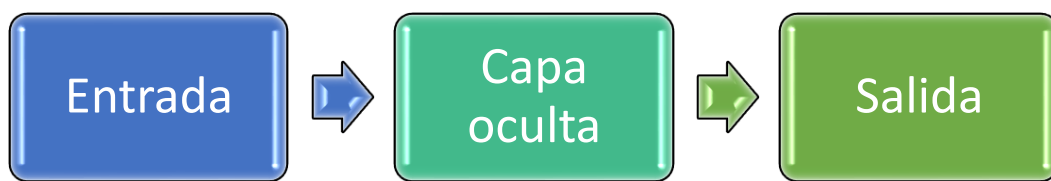


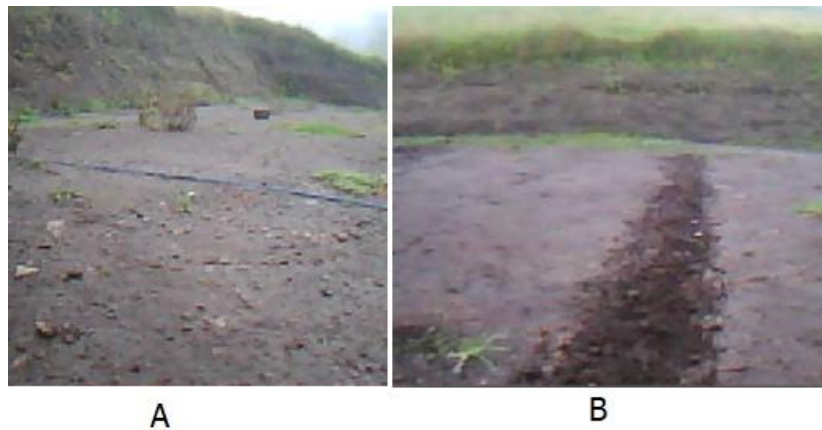
Diagrama general de la arquitectura de la red neuronal. Elaborado por David Quiroga, Boris Rubio.

a) Entrada del sistema

Para la entrada de datos de la red neuronal se tiene una matriz imagen con datos recopilados por la tarjeta OpenMV Cam H7, para el entrenamiento final de la red neuronal se cargaron 754 imágenes como muestra para que la red sea capaz de identificar cuáles son surcos y cuáles no. Figura 3.20

Cada imagen cargada a la red ingresa con un tamaño de 48x48 que, al transformarlo en un vector lineal, son 2304 neuronas a la entrada.

Figura 3.21 Ejemplos de imágenes



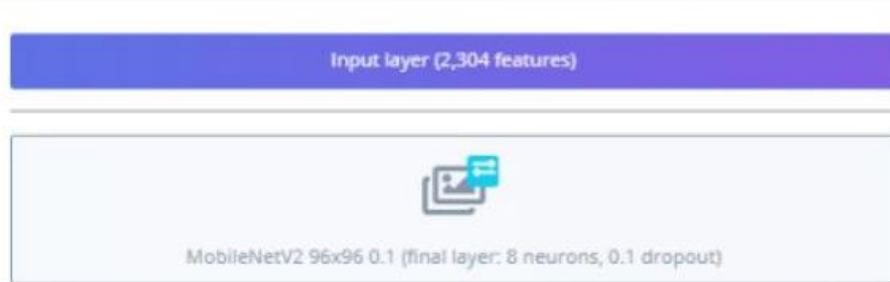
Ejemplos de las imágenes cargadas para el entrenamiento de la red neuronal (A) no es surco, (B) sí es surco. Elaborado por: David Quiroga, Boris Rubio

b) Capa oculta

Se utilizó la arquitectura Mobile Net, para el entrenamiento de la red neuronal y realizar comparaciones entre las imágenes adquiridas por la cam H7 e identificar los surcos de siembra y así obtener una salida adecuada para el proceso.

La red neuronal cuenta con 1 capa oculta que está conformada por 8 neuronas; se utilizó la función de activación ReLU ya que esta anula los valores negativos y deja únicamente los valores positivos exactamente como ingresan a la capa. Figura 3.21

Figura 3.22 Capa oculta y sus neuronas



Número de capas neuronas presentes en la capa oculta. Elaborado por David Quiroga, Boris Rubio

c) Salida del sistema

En la salida del sistema se obtienen 2 respuestas posibles, estas son el reconocimiento de los surcos por parte de la red neuronal con 2 valores comparados entre: sí es surco

o no es surco, dependiendo del resultado inicia o no, el proceso de siembra. Figura 3.22

Figura 3.23 Robot posicionado correctamente.



Robot sembrador ubicado junto al surco para realizar su labor de siembra. Elaborado por David Quiroga, Boris Rubio.

3.7 Entrenamiento de la red neuronal

Para el entrenamiento de la red neuronal se utilizó una plataforma on line gratuita para el público en general, la cual brinda las herramientas necesarias para realizar el entrenamiento de la neurona con varias imágenes que el programador selecciona y carga a la plataforma de Edge Impulse. Figura 3.23

Figura 3.24 Interfaz de Edge Impulse

The screenshot shows the Edge Impulse web interface. It has three main sections: 'Select files' with a button 'Elegir archivos' and the text 'Ninguno archivo selec.'; 'Upload into category' with three radio buttons: 'Automatically split between training and testing' (selected), 'Training', and 'Testing'; and 'Label' with three radio buttons: 'Infer from filename' (selected), 'Leave data unlabeled', and 'Enter label:'. Below the 'Enter label:' radio button is a text input field with the placeholder 'Enter a label'. At the bottom right is a green button labeled 'Begin upload'.

Plataforma de Edge impulse para subir archivos al servidor. Elaborado por David Quiroga, Boris Rubio.

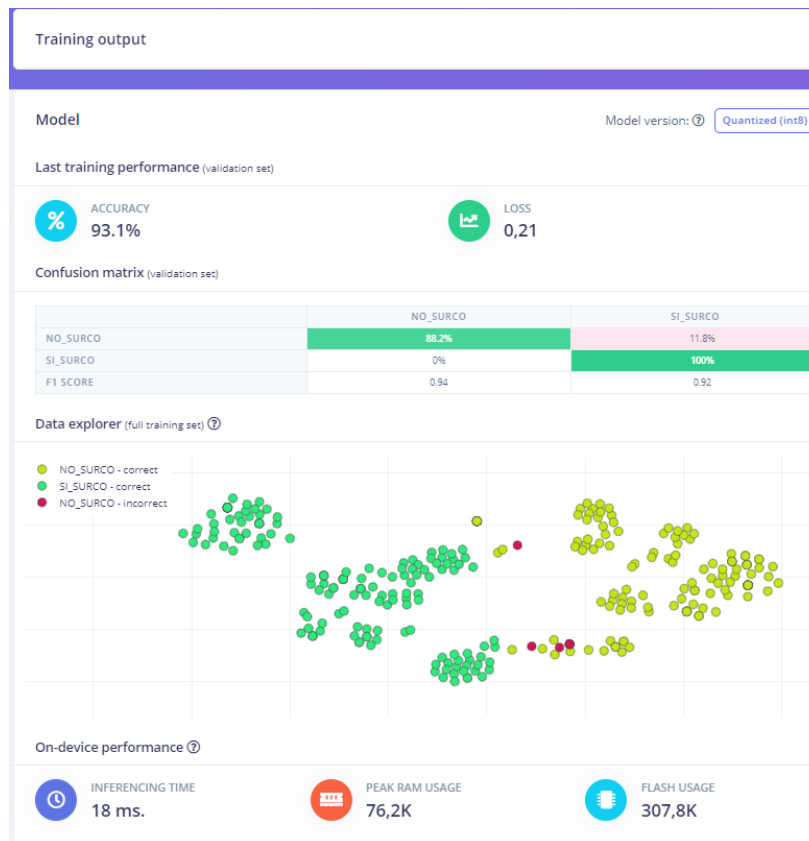
Una vez cargados los archivos el programa comenzará con el entrenamiento de la red neuronal, finalizado el entrenamiento generará los archivos compatibles con el software de OpenMV IDE el cual entrega un archivo comprimido de Tensor Flow, en los formatos txt, tflite y py. Como se observa en la Figura 3.24.

Figura 3.25 Archivos generados por Edge Impulse

 ei_image_classification.py	17/1/2023 3:45	Archivo de origen ...
 labels.txt	17/1/2023 3:45	Documento de te...
 trained.tflite	17/1/2023 3:45	Archivo TFLITE

Lista de los 3 archivos generados al extraer el comprimido de Tensor Flow, Elaborado por David Quiroga, Boris Rubio.

Figura 3.26 Resultados del entrenamiento



Resultados del entrenamiento de la red neural en el programa Edge Impulse. Elaborado por David Quiroga y Boris Rubio.

3.8 Algoritmo inteligente.

El algoritmo que se encarga de recoger los datos por parte de la red neuronal e interpretarlos para que el robot realice su trabajo, fue diseñado en el software OpenMV IDE, el cual brinda una interfaz sencilla y completa para revisar los datos y añadir o modificar las líneas de código. En la Figura 3.27 y Figura 3.28 se observa el código comentado para un mejor entendimiento.

Figura 3.27 Algoritmo Inteligente comentado primera parte

```
1 # Edge Impulse - OpenMV Image Classification Example
2
3 import sensor, image, time, os, tf #importo las librerias necesarias
4 import pyb
5 from pyb import Pin
6 import utime
7 #creo una variable para controlar los leds internos de la placa
8 red_led = pyb.LED(1) #Variable para controlar el color rojo del LED
9 green_led = pyb.LED(2) #Variable para controlar el color verde del LED
10 blue_led = pyb.LED(3) #Variable para controlar el color azul del LED
11 ir_leds = pyb.LED(4) #Variable para controlar el led ir de la placa
12 p_outg = Pin('P7', Pin.OUT_PP)#Variable para controlar el pin P7 como salida de 1
13
14 sensor.reset() # Reset and initialize the sensor.
15 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRAYSCALE)
16 sensor.set_framesize(sensor.QQVGA) # Set frame size to QQVGA (320x240)
17 sensor.set_windowing((240, 240)) # Set 240x240 window.
18 sensor.skip_frames(time=2000) # Let the camera adjust.
19 # Leo los archivos generados en Edge Impulse
20 mobilenet = "trained.tflite"
21 labels = [line.rstrip('\n') for line in open("labels.txt")]
22
```

Algoritmo inteligente, comentarios en cada línea de código. Elaborado por David Quiroga, Boris Rubio.

Figura 3.28 Algoritmo inteligente comentado segunda parte

```
clock = time.clock()
while(True):
    clock.tick()

    img = sensor.snapshot()
    # default settings just do one detection... change them to search the image...
    for obj in tf.classify(mobilenet, img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
        print("*****\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
        img.draw_rectangle(obj.rect(), (0,0,0))
        # This combines the labels and confidence values into a list of tuples
        predictions_list = list(zip(labels, obj.output()))

        for i in range(len(predictions_list)):
            print("%s % %%" % (predictions_list[i][0], predictions_list[i][1]))

    print(clock.fps(), "fps") #imprimo los fps de lo que captura la camara

    if obj.output()[1]>0.9: #Condicion si se detecta un surco del porcentaje al 90% hara:
        green_led.on() #Encendera el led verde de la placa
        p_outg.on() #Pin digital en HIGH
    else:
        green_led.off() #Apagara el led verde de la placa
        p_outg.off() #pin digital en LOW
    if obj.output()[0]>0.5: #Condicion si no se detecta un surco del porcentaje al 50% hara:
        red_led.on() #Encendera el led rojo de la placa
    else:
        red_led.off() #Apagara el led rojo de la placa
```

Algoritmo inteligente, comentarios en cada línea de código. Elaborado por David Quiroga, Boris Rubio.

3.9 Programación de los motores.

En la Figura 3.30 se observa la programación del controlador L298N de los motores, se utilizó el Software Arduino y se cargó a la placa Raspberry Pi Pico.

Esta programación sirve únicamente para el control autónomo de los motores ya que el robot también cuenta instalado un módulo bluetooth, el cual brinda la opción de controlar el robot de manera manual usando una aplicación en su celular.

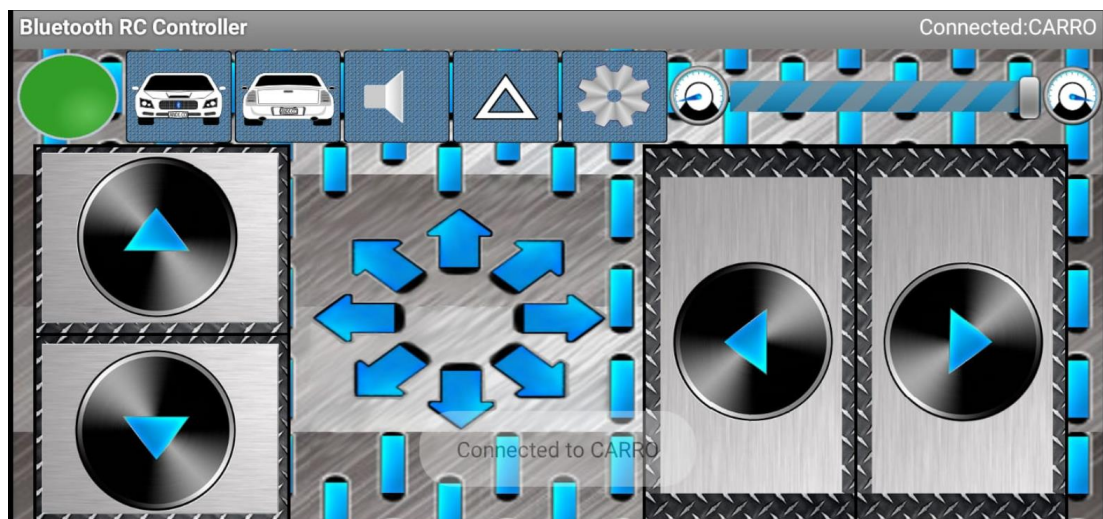
a) App Bluetooth RC Car

Esta opción se usará para trasladar el robot hacia el lugar en donde se lo guardará, evitándole al operador el tener que cargar el robot en el caso de que la ubicación de almacenamiento sea muy distante de la zona de siembra.

Se utilizó la app Bluetooth RC Car, del creador Andi.Co, se obtiene de forma gratuita de la Play Store, para el sistema operativo Android.

Esta aplicación cuenta con soporte para Bluetooth 4.0 y tiene una variedad de botones suficientes para el trabajo del robot, así como código libre para programarla a la necesidad del usuario. Figura 3.29

Figura 3.29 Interfaz de la aplicación Bluetooth RC Car



Interfaz de la aplicación con la cual se controlará el robot de manera manual mediante bluetooth. Elaborado por: David Quiroga, Boris Rubio.

b) Programación Autónoma.

Figura 3.30 Programación de los motores para trabajo autónomo

```
//### ACCIONES QUE DEBERA REALIZAR EL ROBOT###
//1. CAMINAR LINEA RECTA 2 SEG
adelante();delay(2000);alto();delay(1000);
//2. ENCENDER TALADRO y bajar riel
digitalWrite(taladroL, HIGH); digitalWrite(taladroR, LOW);
for(int i=0;i<1400;i++){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, bajada[pasos]);//BAJADA RIEL
    digitalWrite(latchPin, HIGH);
    delay(dt);
    pasos++;
    if(pasos == 4) {pasos=0;}
}
digitalWrite(taladroR, HIGH); digitalWrite(taladroL, LOW);
delay(1000);
while(digitalRead(riel) == LOW){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, subida[pasos]);
    digitalWrite(latchPin, HIGH);
    delay(dt);
    pasos++;
    if(pasos == 4) {pasos=0;}
}
if(digitalRead(riel) == HIGH){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, 0b00000000);
    digitalWrite(latchPin, HIGH);
}
digitalWrite(taladroR, LOW); digitalWrite(taladroL, LOW);
//3. BOTAR SEMILLA
for(int j=0;j<1540;j++){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, bajada2[pasos]);
    digitalWrite(latchPin, HIGH);
    delay(5);
    pasos++;
    if(pasos == 4) {pasos=0;}
}
```

Programación de los motores para trabajo autónomo con comentarios. Elaborado por: David Quiroga, Boris Rubio.

c) Programación Manual

Debido a la extensión del código, se dividió en varias imágenes para explicar las partes de las que éste se compone.

Figura 3.31 Programación del robot para trabajo manual parte 1

```

//*****DEFINIMOS PARA MOVIMIENTO DEL CARRO
#define vel_motor_izq 7 //velocidad izquierdo
#define vel_motor_der 6 //velocidad derecho
#define LED1 25 //LED1
#define LED2 25 //LED2
#define LED 25 //LED
#define IN1 2 //usaremos este pin para dar la señal de ir adelante
#define IN2 3 //usaremos este pin para dar la señal de ir atras
#define IN3 4 //usaremos este pin para dar la señal torcer la direccion hacia la izquierda
#define IN4 5 //usaremos este pin para dar la señal de torcer la direccion a la derecha
char blue; //creamos una variable para guardar las letras que nos enviara a app al apretar
//*****DEFINIMOS PARA MOVIMIENTO DE LOS MOTORES A PASOS (RIEL Y DISPENSADOR DE SEMILLAS)
#define latchPin 10
#define clockPin 11
#define dataPin 12
#define riel 0
bool estadoriel;
int pasos;
//*****DEFINIMOS PARA MOVIMIENTO DEL TALADRO
#define taladroR 13
#define taladroL 14
#define vel_taladro 15
byte bajada[4] = {0b11000000,
                  0b01100000,
                  0b00110000,
                  0b10010000};

byte subida[4] = {0b00110000,
                  0b01100000,
                  0b11000000,
                  0b10010000};

byte bajada2[4] = {0b00001100,
                  0b00000110,
                  0b00000011,
                  0b00001001};

byte subida2[4] = {0b00000011,
                  0b00000110,
                  0b00001100,
                  0b00001001};

```

Definición de variables para el control manual del robot. Elaborado por: David Quiroga, Boris Rubio.

Figura 3.32 Programación del robot para trabajo manual parte 2

```
void setup() {
  Serial2.begin(9600);s.begin(9600);//fijamos la velocidad de coneccion con el bluetooth
  attachInterrupt(digitalPinToInterrupt(22), modo, RISING);//Interrupcion para elegir el control
  //aquí definimos para el control PID
  Setpoint = 0;
  myPID.SetMode(AUTOMATIC);
  myPID.SetOutputLimits(-1023,1023);
  //en esta parte le decimos que nuestros pines seran de salida ya que emitiran una señal
  pinMode (IN1, OUTPUT);
  pinMode (IN2, OUTPUT);
  pinMode (IN3, OUTPUT);
  pinMode (IN4, OUTPUT);
  pinMode (vel_motor_izq, OUTPUT);
  pinMode (vel_motor_der, OUTPUT);
  pinMode(LED, OUTPUT);pinMode(LED1, OUTPUT);pinMode(LED2, OUTPUT);
  //analogWrite(vel_motor_izq, 1023);analogWrite(vel_motor_der, 1023);
  digitalWrite(vel_motor_izq, HIGH);digitalWrite(vel_motor_der, HIGH);
  //*****DEFINIMOS PARA MOVIMIENTO DE LOS MOTORES A PASOS (RIEL Y DISPENSADOR DE SEMILLAS)
  pinMode(latchPin,OUTPUT);
  pinMode(clockPin,OUTPUT);
  pinMode(dataPin,OUTPUT);
  pinMode(riel,INPUT);
  //*****DEFINIMOS PARA MOVIMIENTO DEL TALADRO
  pinMode(taladroR,OUTPUT);
  pinMode(taladroL,OUTPUT);
  pinMode(vel_taladro,OUTPUT);
  analogWrite(vel_taladro, 1023);
  //*****DEFINIMOS PARA DETECCION DE LEDS

  //Aseguramos que todo este apagado y en la pocsion correcta
  digitalWrite(taladroR, LOW);digitalWrite(taladroL, LOW);
  while(digitalRead(riel) == LOW){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, subida[pasos]);
    digitalWrite(latchPin, HIGH);
    delay(dt);
    pasos++;
    if(pasos == 4) {pasos=0;}
  }
}
```

Definición de entradas y salidas para el control manual del robot. Elaborado por: David Quiroga, Boris Rubio.

Figura 3.33 Programación del robot para trabajo manual parte 3

```
void avanzar(int speedL, int speedR){
    analogWrite(vel_motor_izq, speedL);analogWrite(vel_motor_der, speedR);
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
}

void adelante(){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void atras(){
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void izquierda(){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}

void derecha(){
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void adelantederecha(){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}
```

Definición para el control de movimiento del robot. Elaborado por: David Quiroga, Boris Rubio.

Figura 3.34 Programación del robot para trabajo manual parte 4

```
void adelanteizquierda() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}
void abajoizquierda() {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}
void abajoderecha() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}
void alto() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}
void focolencendidos() {
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
}
void focolapagado() {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
}
void foco2encendidos() {
    digitalWrite(LED, HIGH);
}
void foco2apagado() {
    digitalWrite(LED, LOW);
}
}
```

Definición para el control de movimiento del robot. Elaborado por: David Quiroga, Boris Rubio.

Figura 3.35 Programación del robot para trabajo manual parte 4

```

for(int i=0;i<1400;i++){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, bajada[pasos]); //BAJADA RIEL
    digitalWrite(latchPin, HIGH);
    delay(dt);
    pasos++;
    if(pasos == 4) {pasos=0;}
}
digitalWrite(taladroR, HIGH); digitalWrite(taladroL, LOW);
delay(1000);
while(digitalRead(riel) == LOW){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, subida[pasos]);
    digitalWrite(latchPin, HIGH);
    delay(dt);
    pasos++;
    if(pasos == 4) {pasos=0;}
}
if(digitalRead(riel) == HIGH){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin,clockPin,LSBFIRST, 0b00000000);
    digitalWrite(latchPin, HIGH);
}
digitalWrite(taladroR, LOW); digitalWrite(taladroL, LOW);
}
if(blue=='w'){ //en esta parte se compara si blue es igual a S de ser asi se ejecuta la accion de abajo
    //la app envia la letra S cuando no se apreta ningun boton para que el robot se detenga y apague todas
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
}
if(blue=='U'){ //en esta parte se compara si blue es igual a S de ser asi se ejecuta la accion de abajo
    //la app envia la letra S cuando no se apreta ningun boton para que el robot se detenga y apague todas
    for(int j=0;j<1540;j++){
        digitalWrite(latchPin, LOW);
        shiftOut(dataPin,clockPin,LSBFIRST, bajada2[pasos]);
        digitalWrite(latchPin, HIGH);
        delay(5);
        pasos++;
        if(pasos == 4) {pasos=0;}
    }for(int j=0;j<1540;j++){
        digitalWrite(latchPin, LOW);
        shiftOut(dataPin,clockPin,LSBFIRST, bajada2[pasos]);
        digitalWrite(latchPin, HIGH);
        delay(5);
        pasos++;
        if(pasos == 4) {pasos=0;}
    }
}

```

Programación de los motores para trabajo manual con comentarios. Elaborado por David Quiroga, Boris Rubio.

3.10 Resumen de costos

Una vez terminado el trabajo se detallan en la Tabla 3.3 todos los costos requeridos para la implementación del robot.

Tabla 3.3 Resumen de costos del proyecto

	DESCRIPCIÓN	CANT.	P. U	P.T
Software OpenMV	OpenMV Ide v2.4.0	1	\$ -	\$ -
Software Arduino INE	Arduino INE	1	\$ -	\$ -
Laptop	Laptop Lenovo win 10	1	\$ -	\$ -
Amoladora	Jumbo, inalámbrica	1	\$ -	\$ -
Taladro	Jumbo inalámbrico	1	\$ -	\$ -
OpenMV Cam H7	OpenMV cam h7	1	\$ 100,00	\$ 100,00
Camera Module Extension Cable	OpenMV cab fpc	1	\$ 20,00	\$ 20,00
Raspberry Pi Pico	Raspberry pico	1	\$ 12,50	\$ 12,50
Batería	Batería 12V 1000mA	1	\$ 15,00	\$ 15,00
Motor ruedas	JGB37-520	2	\$ 25,00	\$ 50,00
Motor perforador	ZGB37RG	1	\$ 26,00	\$ 26,00
Motor a pasos	28BYJ-48	2	\$ 7,50	\$ 15,00
Llantas	Rueda Pololu AT 12.5 cm	2	\$ 25,00	\$ 50,00
Rueda Loca	Rueda loca metálica	1	\$ 20,00	\$ 20,00
Chasis Robot	Tubería PVC	1	\$ 50,00	\$ 50,00
Acople para motor	Acople para broca 4mm	1	\$ 10,00	\$ 10,00
Dispensador	Cortes láser	1	\$ 7,00	\$ 7,00
Elevador	Cortes láser	1	\$ 7,00	\$ 7,00
Base de madera	Tabla triplex 50x28 cm	1	\$ 2,00	\$ 2,00
	Aglomerado 0.02x1x1 m	1	\$ 4,00	\$ 4,00
Engranajes	Impresión 3D	2	\$ 10,00	\$ 20,00
Cremallera	Impresión 3D	1	\$ 5,00	\$ 5,00
Elementos Varios	Electrónicos, tornillos	50	\$ 0,25	\$12,50
Estaño	Para Soldar	1	\$ 3,00	\$ 3,00
Cautín	Cautín de 100W	1	\$ 5,00	\$ 5,00
Pasta	Para soldar	1	\$ 2,00	\$ 2,00

Pegamento	Soldadura fría	6	\$ 0,50	\$ 3,00
Cola	Pegamento para madera	1	\$ 0,80	\$ 0,80
Broca hoyadora	Broca para elevador 1 in	1	\$ 10,00	\$ 10,00
Disco dentado	Disco para amoladora	1	\$ 6,00	\$ 6,00
Brocas	4,5 mm; 3/16 in cobalto	3	\$ 2,00	\$ 6,00
	Hora de trabajo de ingeniería	240	\$ 2,40	\$ 576,00
VALOR TOTAL SOLO HARDWARE Y SOFTWARE				
VALOR TOTAL DE TRABAJO DE INGENIERÍA				\$ 576,00
VALOR TOTAL DE TRABAJO				\$ 1.325,20

Tabla de costos detallada, Elaborado por: David Quiroga, Boris Rubio.

CAPÍTULO 4

PRUEBAS – RESULTADOS

Para realizar la validación del sistema tanto en hardware como software se realizaron pruebas en un escenario controlado, esto con la finalidad de encontrar cualquier error y analizar el comportamiento del robot al detectar los surcos de siembra y después calcular la efectividad del sistema.

4.1 Pruebas de reconocimiento de surco.

Para comprobar que el robot sea capaz de detectar un surco primero se procedió a realizar pruebas con surcos ya sembrados, ya que estos cuentan con factores que sirven de distracción para el robot, al tener surcos con brotes altos se dificulta la diferenciación de cada uno y así el robot puede aprender a distinguir surcos de siembra en condiciones adversas. Figura 4.1

Figura 4.1 Ejemplo de Surco con brotes altos



Surcos previamente sembrados con brotes altos. Elaborado por David Quiroga, Boris Rubio.

Se realizaron pruebas de detección de surcos con diferentes números de neuronas las cuales arrojaron los siguientes resultados detallados a continuación en las siguientes tablas:

Tabla 4.1 Resultados con 5 neuronas en la capa oculta

Prueba #	Sí Surco	No Surco	Trabajo
1	0.22	0.81	No
2	0.66	0.503	No
3	0.48	0.61	No
4	0.27	0.73	No
5	0.64	0.27	Sí
6	0.23	0.32	No
7	0.58	0.49	Sí
8	0.12	0.58	No
9	0.17	0.26	No
10	0.94	0.06	Sí
Promedio	0.423	0.483	No

Registro de resultados, red neuronal con 5 neuronas, Elaborado por David Quiroga, Boris Rubio.

Con los resultados obtenidos en la Tabla 4.1, se puede determinar que los comparadores de salidas no detectan los surcos con total eficiencia, y ya que al no tener las suficientes unidades de procesamiento cualquier factor puede influir en que el surco no sea detectado, tal es el caso que la mayoría de los intentos arrojan un resultado negativo a la hora de la identificación.

Tabla 4.2 Resultados con 8 neuronas en la capa oculta

Prueba #	Sí Surco	No Surco	Trabajo
1	0.31	0.83	No
2	0.89	0.22	Sí
3	0.73	0.12	Sí
4	0.67	0.25	Sí
5	0.88	0.11	No
6	0.73	0.34	No
7	0.81	0.09	No
8	0.95	0.036	Sí
9	0.87	0.009	Sí
10	0.99	0.005	Sí
Promedio	0.783	0.2091	Sí

Registro de resultados, red neuronal con 8 neuronas, Elaborado por David Quiroga, Boris Rubio.

Con los resultados obtenidos en la Tabla 4.2, se puede determinar que la red neuronal ahora si es capaz de detectar los surcos con mayor facilidad, ya que identifica el camino que debe seguir, aunque éste cuente con ciertos distractores, es así como con 8

neuronas podemos observar que la red neuronal aumentó su capacidad de procesamiento de datos y puede realizar el trabajo con mayor eficiencia.

4.2 Pruebas con diferente número de muestras

Para la identificación de surcos se utilizó diferente número de muestras al momento de entrenar la neurona, siendo más eficiente el trabajo de identificación al aumentar el número de muestras.

a) Pruebas con 200 muestras

Con 200 muestras se observa una precisión del 92.9% y una pérdida del 0.27, así como se muestra en la Tabla 4.3

Tabla 4.3 200 Muestras



Resultado de precisión de identificación de surcos con 200 muestras. Elaborado por David Quiroga, Boris Rubio.

b) Pruebas con 350 Muestras

Con 350 muestras se observa una precisión del 93.1% y una pérdida del 0.21, así como se muestra en la Tabla 4.4

Tabla 4.4 350 Muestras



Resultado de precisión de identificación de surcos con 350 muestras. Elaborado por David Quiroga, Boris Rubio.

c) Pruebas con 700 Muestras

Con 700 muestras se observa una precisión del 98.7% y una pérdida del 0.13, así como se muestra en la Tabla 4.5

Tabla 4.5 700 muestras.



Resultado de precisión de identificación de surcos con 700 muestras. Elaborado por David Quiroga, Boris Rubio.

Una vez que el robot ha pasado el proceso de entrenamiento y de detección de surcos de manera satisfactoria, (Figura 4.2) se procede a probar en terreno apto para siembra,

es decir con surcos arados para las dimensiones del robot y con una textura suave donde la semilla germine de manera normal. (Figura 4.3)

Figura 4.2 Detección de Surco



Led indicador en verde al detectar un surco de siembra. Elaborado por David Quiroga, Boris Rubio.

Figura 4.3 Semillas depositadas



Semillas depositadas en el punto de siembra por parte del robot. Elaborado por David Quiroga, Boris Rubio

4.3 Pruebas de funcionamiento del robot en campo

Para verificar el funcionamiento del robot se realizaron 30 pruebas las cuales se detallan en la Tabla 4.6, con pruebas satisfactorias y no completadas por el robot; las pruebas satisfactorias son aquellas en las que el robot se posicionó correctamente junto al surco y realizó el proceso de siembra con normalidad; las pruebas no completadas por el robot son aquellas en las que no se posicionó correctamente o no detectó el surco de manera adecuada.

El proceso de siembra le toma al robot aproximadamente 1 minuto entre punto de siembra, el trabajo consiste en posicionarse, activar el mecanismo elevador de la broca, perforar el agujero y depositar las semillas. Figura 4.4

Figura 4.4 Robot durante el proceso de siembra



Robot terminando un punto de siembra, se observan las semillas en el agujero. Elaborado por David Quiroga, Boris Rubio

Tabla 4.6 Pruebas de posicionamiento y siembra del robot.

Nº	Satisfactoria	No completada	Observaciones
1	X		
2	X		
3	X		
4	X		
4	X		
5	X		
6		X	No detectó surco
7	X		
8		X	Se posicionó antes del surco
9	X		
10	X		
11		X	Se posicionó antes del surco
12		X	Se posicionó antes del surco
13		X	Se posicionó antes del surco
14	X		
15	X		
16	X		
17		X	No detectó surco
18	X		
19	X		
20	X		
21	X		
22	X		
23	X		
24		X	Se posicionó antes del surco
25	X		
26	X		
27	X		
28	X		
29	X		
30		X	No detectó surco

Tabla de pruebas del robot con descripción del error. Elaborado por David Quiroga, Boris Rubio

CONCLUSIONES

Se logró implementar un robot móvil sembrador de semillas de grano grueso utilizando visión e inteligencia artificial, herramientas que le permiten al robot seguir una trayectoria bien definida e identificar los surcos sobre los cuales realiza el proceso de siembra.

Al usar un tipo de red neuronal convolucional específicamente diseñada para el aprendizaje profundo, se identificó patrones en los surcos para así aprender los datos sin necesidad de mostrar características especiales manualmente, sumado a la activación mediante la función ReLU, se obtuvo un modelo preciso capaz de reconocer un surco de siembra para así comenzar con su trabajo sin dificultad.

El número de datos que se cargan a la red neuronal al momento de su entrenamiento son directamente proporcionales al porcentaje de aciertos en las pruebas de funcionamiento finales, es decir, mientras más imágenes se provean a la red neuronal, más aciertos tendrán en los resultados finales. Con 200 imágenes se obtuvo un porcentaje de aciertos del 92.9% y con 350 imágenes un porcentaje del 93.1%, es decir que se mejoró los aciertos en un 0.2%; al cargar 700 imágenes, el doble de la prueba anterior, se mejoró el porcentaje de aciertos a 98.7%, lo que se traduce en una mejora significativa respecto a la prueba anterior.

El uso del programa INVENTOR permitió realizar el modelado en 3D de piezas que requerían un alto grado de exactitud para acoplarse de manera correcta a los ejes de los motores, el software de modelado 3D son una herramienta que brinda una ventaja significativa a la hora de ensamblar proyectos de ingeniería. Para imprimir los modelos 3D se optó por una impresora Anycubic Kobra Plus con tecnología de capa por filamento PLA debido a la rigidez y durabilidad, que brinda a las piezas, haciéndolas resistentes y livianas por lo que no tendrán problemas al exponerlas al contacto superficial que se ejerce al trabajar con engranajes.

Mediante la selección de tarjeta controladora se determinó que la OpenMV Cam H7, es ideal para trabajar con algoritmos inteligentes y redes neuronales, al contar con un software libre, no necesita añadir ningún otro software o librerías adicionales, lo que

brinda mucha facilidad al usuario al momento de programar una red neuronal de alto nivel y el robot presenta un gran rendimiento al ejercer sus tareas.

La arquitectura de la red neuronal muestra un desempeño significativo a partir de la presencia de 8 neuronas en la capa oculta, logra detectar los surcos pese a que éstos estén cubiertos por maleza o no muestren una forma totalmente definida, esto garantiza que el robot realiza un trabajo eficiente.

Una desventaja del uso de un robot sembrador en el campo es el tiempo que este necesita para realizar su trabajo, 1 minuto por punto de siembra aproximadamente mientras que una persona bien entrenada en esta labor puede realizar el mismo trabajo en unos 45 segundos promedio, entre perforar el agujero y contar las semillas que debe depositar en cada uno; no obstante, el robot puede realizar una jornada completa sin tomar descansos, optimizando el tiempo de los agricultores en otras actividades del campo que no conlleven lesiones por la postura o por exponerse por horas al sol directamente.

RECOMENDACIONES

Al momento de realizar el ensamblaje del dispensador de semillas, se debe considerar el tamaño de cada grano que se va a sembrar, es necesario que se realicen los cortes en forma de U en los discos giratorios 2 mm más grandes que el diámetro de la semilla más grande que se encuentre, esto para garantizar que no se atoren las semillas entre los espacios del disco giratorio y el tubo que lleva las semillas a la perforación del suelo.

El modelado 3D debe ser realizado en escala de milímetros, ya que de esta manera se consigue una impresión más detallada y así no se deberá realizar modificaciones en la pieza en la fase de post procesado, se debe utilizar un calibrador para sacar las medidas exactas del eje del motor al que será acoplado cada engranaje, cada pieza debe entrar sin presión excesiva y mantener una fijación alta para evitar que se den movimientos no planificados en el elevador.

Es necesario acoplar un disipador de calor capaz de refrigerar el motor a pasos del elevador, este motor realiza un gran esfuerzo al subir la plataforma móvil con la broca y bajarla con suficiente fuerza para que pueda perforar el terreno, aunque este se encuentre duro o tenga ciertos componentes pedregosos bajo la superficie.

Para garantizar que el robot no tenga ningún inconveniente al momento de desplazarse, es necesario verificar el peso que este llevará en su chasis y según ese dato, adquirir los motores que más se ajusten a las necesidades del robot.

Al ser un robot diferencial, la rueda loca se coloca en la parte posterior, esta solamente sigue la trayectoria que le darán los motores situados a los lados del robot y para evitar pérdidas de tracción se recomienda que los motores de desplazamiento estén colocados en la parte delantera del robot.

Para evitar que la tarjeta CAM H7 sufra problemas de sobre carga en el procesamiento de la red neuronal, lo óptimo es trabajar con un máximo de 10 neuronas, este número es suficiente para brindar una capacidad de procesamiento alta y evita que se consuma toda la capacidad de memoria volátil RAM.

La red neuronal debe ser entrenada con un mínimo de 500 imágenes, con efectos de iluminación diferentes, esto para garantizar un reconocimiento ágil y confiable en cualquier tipo de tiempo al que se someta el robot.

REFERENCIAS

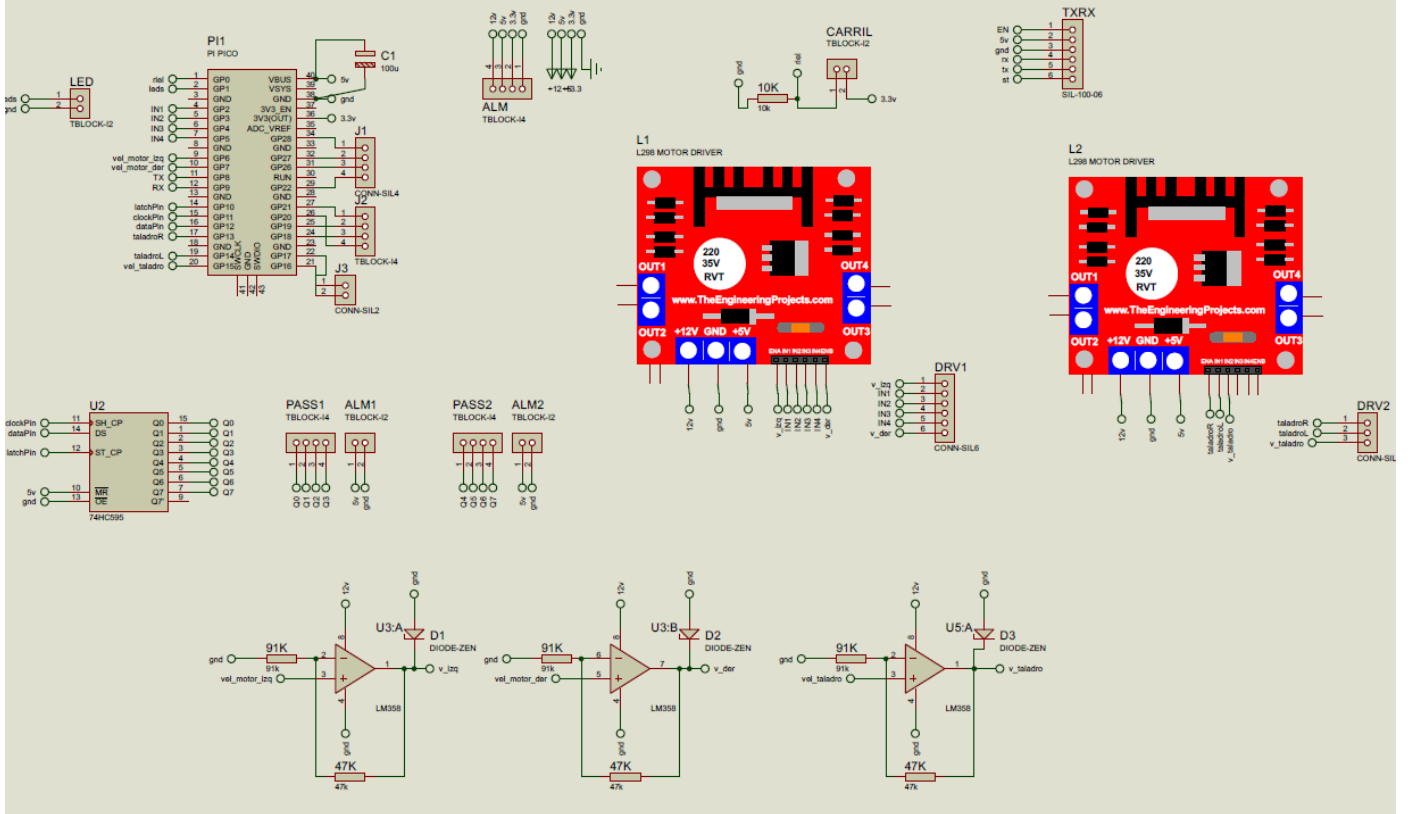
- Agyeman Ibrahim; Abdelkader Kwabena. (2019). *Raspberry Pi Pico*.
<https://openmv.io/products/openmv-cam-h7>
- Alay, A., & Moreno, F. (2017). *INVESTIGACIÓN DE LA LOCOMOCIÓN DE LOS ROBOTS OMNIDIRECCIONALES PARA EL DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO ROBOT RUEDA ESFÉRICA PARA MEJORA DE LA MOVILIDAD EN ESPACIOS REDUCIDOS*.
- Benavides, K. R. (2019). *Locomoción de Robot Móvil*.
- Bernal, J. C., Florez Gonzales, J. C., & Gutierrez Ortiz, J. A. (2020). *Agrobot Ii*. 64.
[https://repository.ucatolica.edu.co/bitstream/10983/3205/4/ROBOT AGROBOT CX.pdf](https://repository.ucatolica.edu.co/bitstream/10983/3205/4/ROBOT%20AGROBOT%20CX.pdf)
- Calentada, C., Anycubic, P., Plus, K., Para, H., & Kobra, A. (n.d.). *Anycubic Kobra Plus*.
- Carlos, J., González, F., Manuel, V., & Pabón, C. (2019). *Diseño En Autocad De Robot Sembrador De Semillas De Fresa: Fresabot Iii Juan Carlos Florez Gonzales, Victor Manuel Castellanos Pabón*.
- Caverzasi, A., Saravia, F., Micolini, O., Mathe, L., & Lichtensztein, L. F. (2019). Robot móvil autónomo para crear mapas 3D en un ambiente acotado. *2019 IEEE Biennial Congress of Argentina, ARGENCON 2019*, 786–791.
<https://doi.org/10.1109/ARGENCON.2019.6868589>
- Cruz, H. (2018). Monografía robot movil Una Introducción a los Robots Móviles. *Academia*.
- Darley Leon, Milton Tenesca, J. Hu. (2019). Rover Spirit, El Explorador de Marte Rover Spirit, The Martian Explorer. *Revista Científica Perspectivas*, 2019.
https://www.easychair.org/publications/preprint_open/dcb5
- Duro, D. (n.d.). *Batería Recargable Seca Descripción*. 6–9.
- Electronics, A. V. (2021). *Módulo L298N*. 1–6.
- García S, I., & Caranqui S, V. (2017). La visión artificial y los campos de aplicación. *Tierra Infinita*, 11(2602–8131), 94–103.
- Gp, L. E. D., Sda, I. C., Scl, I. C., Sda, I. C., Scl, I. C., Tx, S. P. I., Tx, S. P. I., Tx, S. P. I., & Tx, S. P. I. (2018). *Raspberry Pi Pico*.
<https://www.raspberrypi.com/products/raspberry-pi-pico/>
- Industria, E., & Rob, N. (2021). *Agricultura automatizada, aplicaciones de la robótica en la Agricultura*. 1–5.

- Issn, P. (2018). Entrenamiento De Una Red Neuronal Artificial Usando El Algoritmo Simulated Annealing. *Entrenamiento De Una Red Neuronal Artificial Usando El Algoritmo Simulated Annealing*, 1(24), 13–18.
- Izaurieta, F., & Saavedra, C. (2019). Redes Neuronales Artificiales. *Charlas de Fisica*, 1–15. [https://doi.org/10.1016/S0210-5691\(05\)74198-X](https://doi.org/10.1016/S0210-5691(05)74198-X)
- Jalomo, J. (2019). *Diseño de una red neuronal convolucional para la clasificación de señales biomédicas*. 0–2.
- Katerine, A., & Rodr, B. (2019). *Diseño De Un Robot Móvil Recolector Artificial*.
- Larrañaga, P. (2020). *Redes Neuronales*. 1–19.
- Mellis, D. (2011). Arduino Mega 2560. Retrieved November, 2560. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Arduino+MEGA+2560#0>
- Naylamp Mechatronics. (2021). *Convertidor Voltaje DC-DC Step Down 3A LM2596*. Naylamp Mechatronics Web Site. <https://naylampmechatronics.com/conversores-dc-dc/196-convertidor-voltaje-dc-dc-step-down-3a-lm2596.html>
- Pololu Robotics & Electronics. (n.d.). *Dagu Wild Thumper Wheel 120x60mm Pair with 4mm Shaft Adapters - Metallic Red*. 3, 22–23.
- Romero, J. J., Dafonte, C., Gómez, Á., & Penousal, F. J. (2017). Inteligencia Artificial Y Computación Avanzada. In *Inteligencia Artificial* <http://fmachado.dei.uc.pt/wp-content/papercite-data/pdf/ms07.pdf#page=9>
- TELEDYNE. (2020). *5 Pasos Para Crear E Implementar Redes Neuronales De Aprendizaje Profundo*. <https://www.flir.es/discover/iis/machine-vision/5-steps-for-building-and-deploying-a-deep-learning-neural-networks/>

ANEXOS

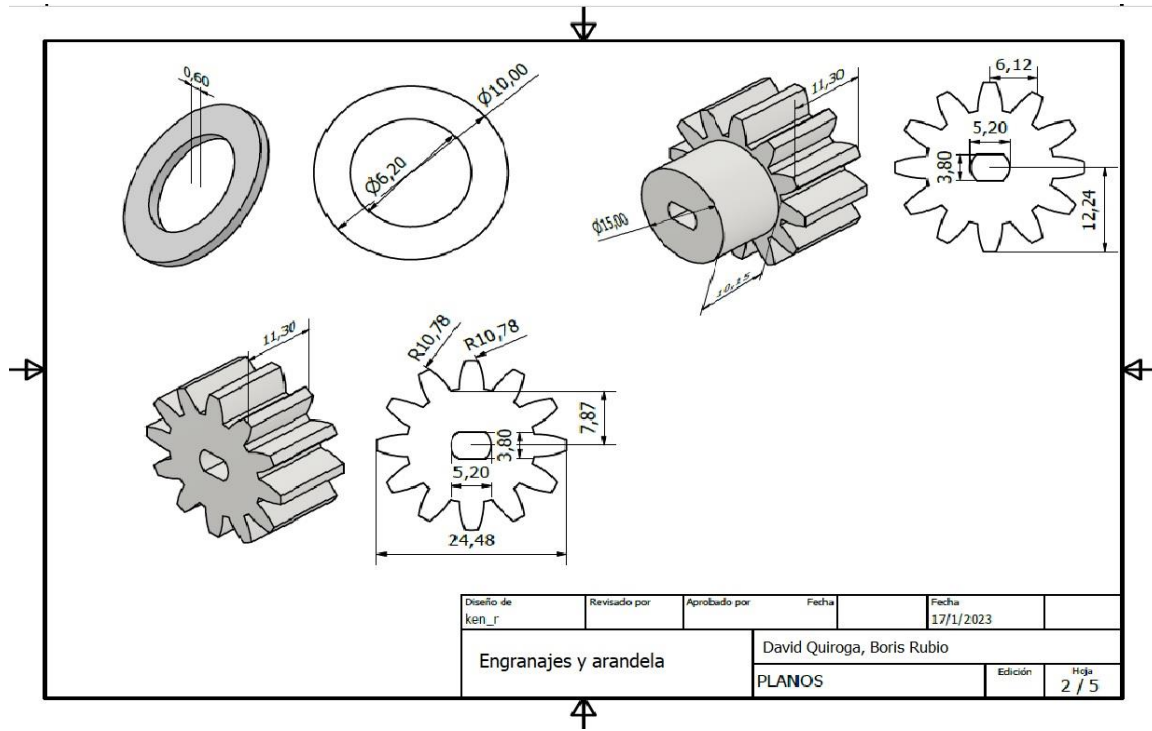
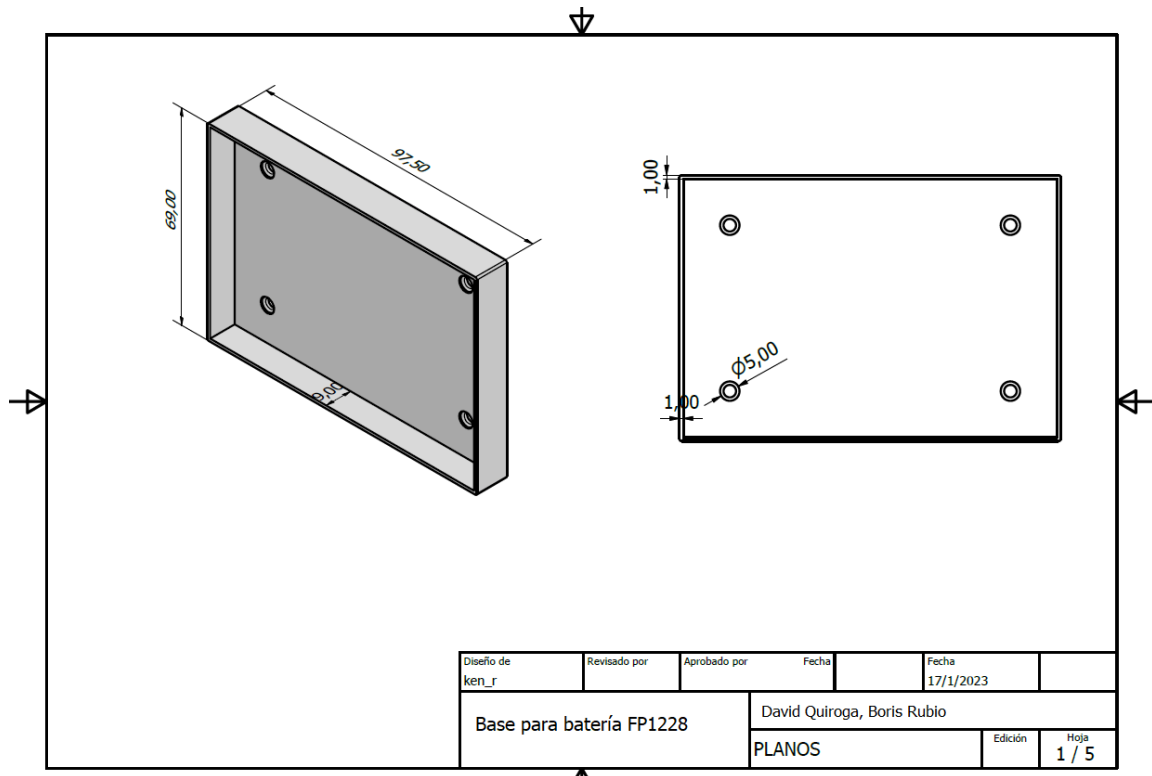
Anexo 1: Esquema electrónico robot sembrador.

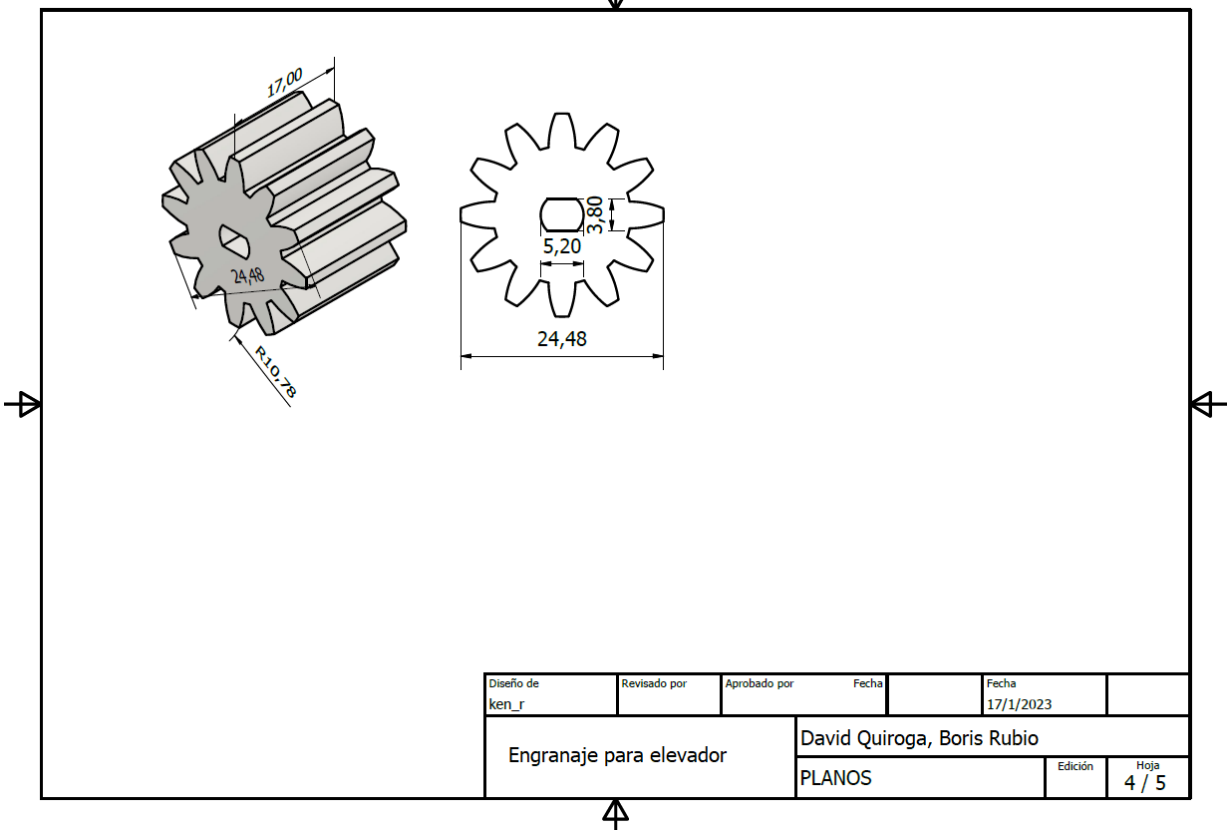
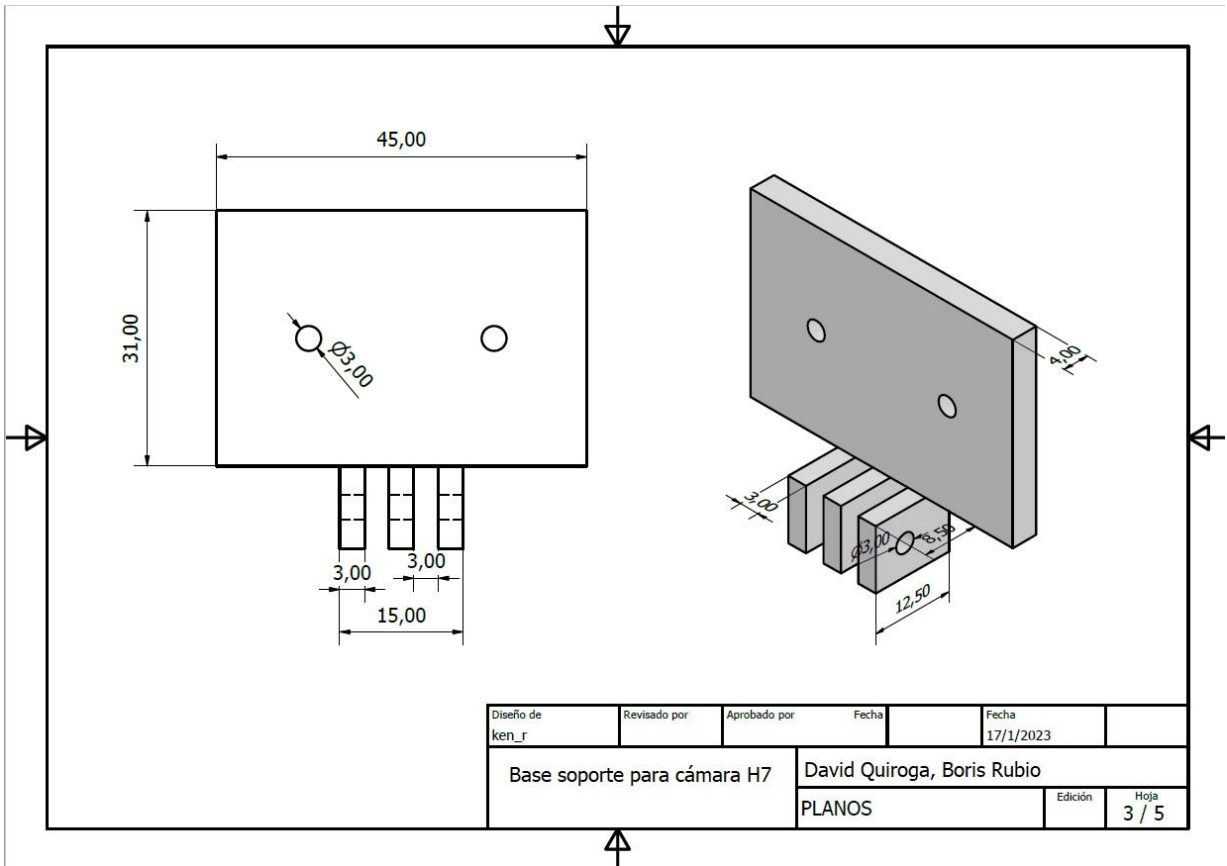
ESQUEMA ELECTRÓNICO (ROBOT SEMBRADOR)

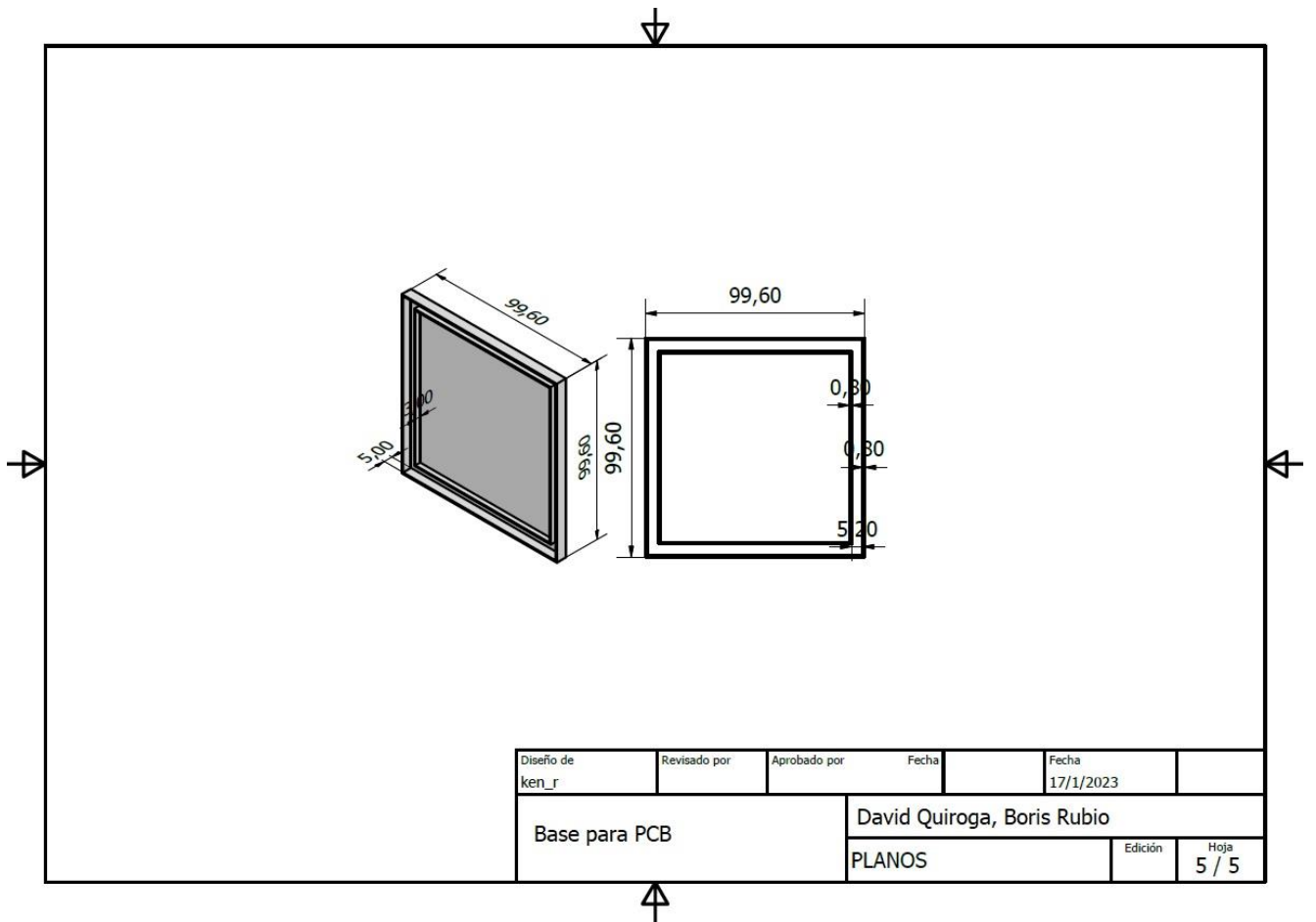


Esquema electrónico, drivers motores, Raspberry pico y entrada Cam H7, Elaborado por David Quiroga, Boris Rubio.

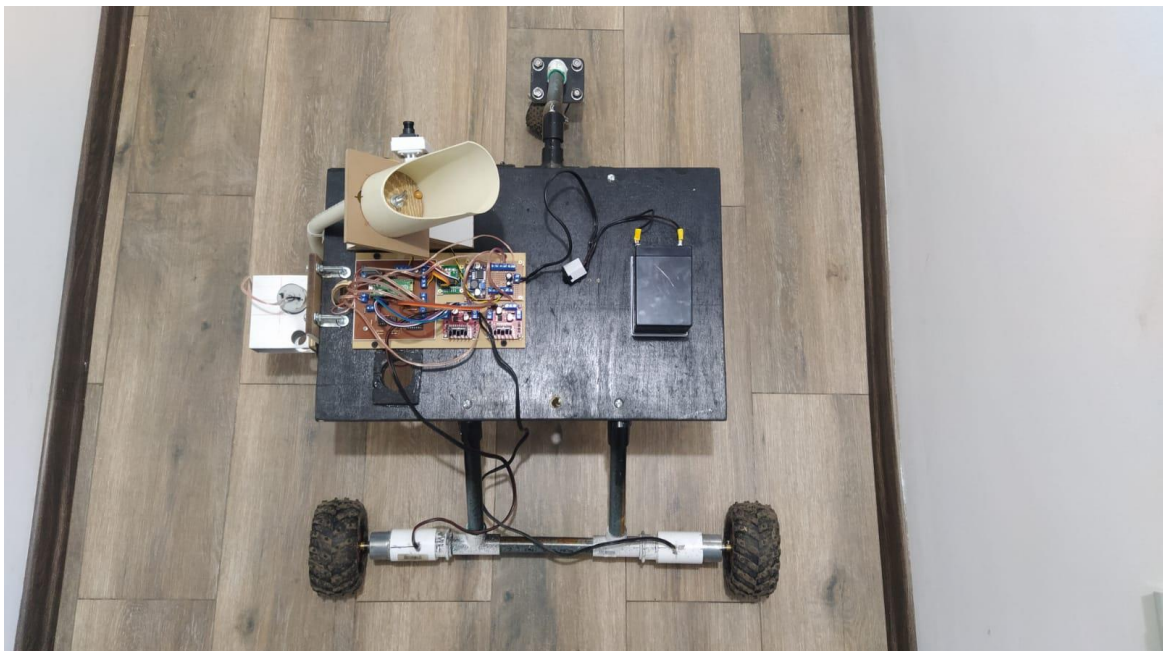
Anexo 2: Planos de las piezas diseñadas en 3D para los componentes móviles del elevador y del dispensador de semillas.



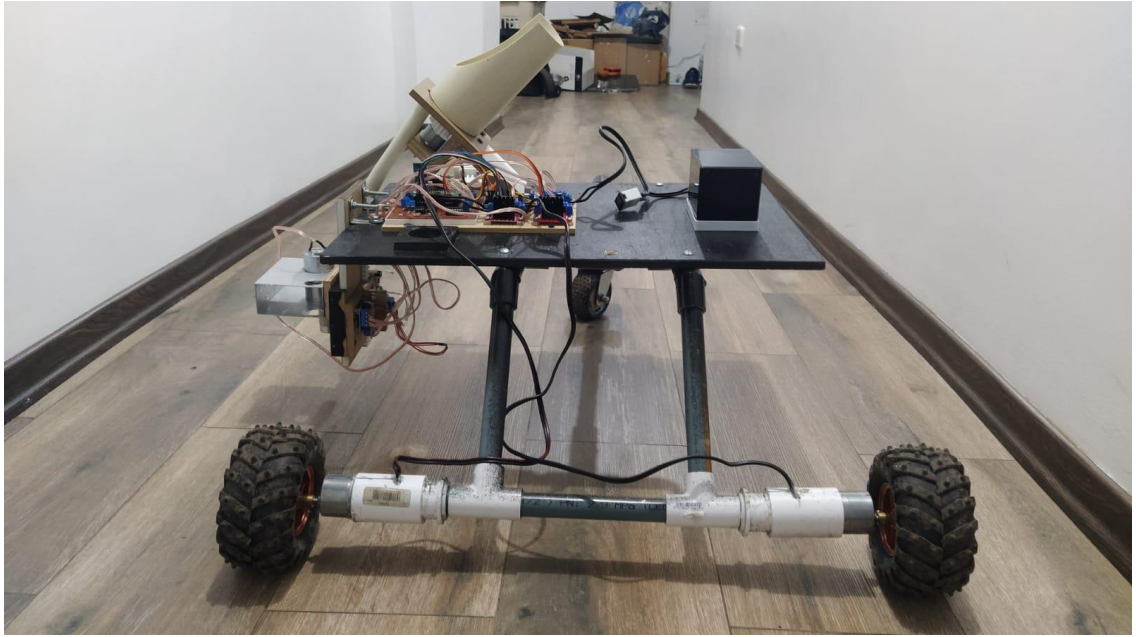




Anexo 3: Robot móvil sembrador ensamblado.



Vista superior, robot sembrador. Elaborado por David Quiroga, Boris Rubio.



Vista posterior robot móvil sembrador. Elaborado por David Quiroga, Boris Rubio.



Vista frontal robot móvil sembrador. Elaborado por David Quiroga, Boris Rubio.