

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA:

INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:

INGENIEROS ELECTRÓNICOS

TEMA:

**DESARROLLO DE UN ALGORITMO DE TRAYECTORIA PARA UN
ROBOT SEGUIDOR DE LÍNEA DESTREZA DE COMPETENCIA
MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL**

AUTORES:

CLELIO DANIEL ENDARA SUMBA

EMERSON JOAO MAIGUA YÁNEZ

TUTORA:

CARMEN JOHANNA CELI SÁNCHEZ

Quito, marzo 2021


CESIÓN DE DERECHOS DE AUTOR

Nosotros Clelio Daniel Endara Sumba y Emerson Joao Maigua Yáñez, con documentos de identificación N° 1723598197 y N° 1717983702 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE UN ALGORITMO DE TRAYECTORIA PARA UN ROBOT SEGUIDOR DE LÍNEA DESTREZA DE COMPETENCIA MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL, mismo que ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Clelio Daniel Endara Sumba
C.I. 1723598197



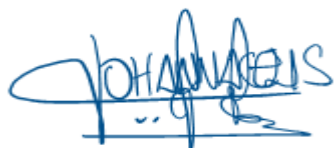
Emerson Joao Maigua Yáñez
C.I. 1717983702

Quito, marzo 2021.

DECLARATORIA DE COAUTORÍA DE LA DOCENTE TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, DESARROLLO DE UN ALGORITMO DE TRAYECTORIA PARA UN ROBOT SEGUIDOR DE LÍNEA DESTREZA DE COMPETENCIA MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL, realizado por Clelio Daniel Endara Sumba y Emerson Joao Maigua Yáñez, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo 2021.



Carmen Johanna Celi Sánchez

C.I. 1717437808

DEDICATORIA

A Dios por brindarme salud, bendiciones y crear resiliencia en mi espíritu; a mi madre Rosa Sumba y a mi padre Clelio Endara quienes me dieron la vida, alimentación, educación, un hogar; desde siempre con su gran entrega, dedicación, compromiso, consejos y apoyo incondicional con lo cual me han demostrado su amor en todo su esplendor para forjarme en la persona que soy en la actualidad; a mi hermano Francisco Endara que a pesar de nuestra diferencias desde siempre nos hemos apoyado; a mi tía Rosita “mamita del campo” mis primas/o Ximena, Nora, Jenny, y Fernando que con su respaldo y cariño me han inspirado para seguir adelante.

Clelio Daniel Endara Sumba

Es de trascendental importancia dedicar este trabajo de titulación a toda mi familia en especial a mis padres y adorada hija Valentina por ser mi fuente de motivación e inspiración para poderme superar cada día más, por último, pero igual importante a mis amigos Miguel, Willy, Eve, Andrea y David quienes con su apoyo me permitió culminar mi carrera, una mención especial a Clelio amigo y compañero académico. Este logro es para ustedes.

Emerson Joao Maigua Yáñez

AGRADECIMIENTO

La vida está llena de retos y gratas experiencias, es por ello que agradezco a la Universidad Politécnica Salesiana sede Quito campus Sur por brindarme las herramientas necesarias para mi formación profesional, a todos los docentes que a lo largo de mi carrera universitaria me han transmitido su sabiduría y conocimiento, en especial a mi tutora Mgtr. Carmen Johanna Celi Sánchez por su confianza, orientación y guía puesta en este trabajo de titulación; a la Secretaría Técnica de Ecosistema de Emprendimiento e Innovación por darme la oportunidad de desarrollar el primer proyecto de investigación en los laboratorios de fabricación StartLabs, a mis compañeros universitarios por su amistad brindada en todos aquellos momentos de jornadas académicas, actividades extracurriculares y de diversión, en especial a la Srta. Mishel Pazmiño; así como también a mi familia y a todas aquellas personas que estuvieron en los momentos más adecuados brindándome sus consejos para motivarme a culminar mi carrera profesional.

Clelio Daniel Endara Sumba

Agradezco a la Universidad Politécnica Salesiana, que abrió sus puertas del conocimiento para mí, a todos los docentes en especial a mi tutora Mgtr. Carmen Johanna Celi Sánchez, que a lo largo de mi carrera me brindaron su apoyo y conocimientos suficientes para mi formación profesional, una mención especial a mis amados padres Pedro y Susana por haberme dado el legado más importante de la vida que es la educación.

Emerson Joao Maigua Yáñez

ÍNDICE DE CONTENIDO

RESUMEN	xii
ABSTRACT	xiii
INTRODUCCIÓN	xiv
CAPÍTULO 1	1
ANTECEDENTES.....	1
1.1 Planteamiento del problema	1
1.2 Justificación del proyecto	2
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 Robot Autónomo.....	4
2.3 Sistema de locomoción de un robot	4
2.4 Robot móvil Seguidor de Línea Destreza.....	4
a) Reglamento para la categoría Seguidor de Línea Destreza	5
b) Robot móvil seguidor de línea destreza con arquitectura diferencial	6
c) Pistas para el robot móvil seguidor de línea destreza.....	7
2.5 Visión Artificial	8
2.6 Estado del arte de los sistemas de navegación en robots móviles.	9
2.7 Inteligencia Artificial	10
2.8 Redes Neuronales	11
a) Neuronas	11
2.9 Arquitectura de una red neuronal.....	11
a) Arquitectura MobileNet.....	12

2.10 Tipos de capas de una red neuronal	12
a) Capa de entrada	13
b) Capas intermedias	13
c) Capa de salida	13
2.11 Etapas de construcción y manejo de una red neuronal.....	13
a) Fase de conceptualización	13
b) Fase de diseño	13
c) Fase de implementación	14
d) Fase de mantenimiento.....	14
2.12 Deep Learning.....	14
2.13 Backpropagation	14
2.14 Aprendizaje de una red neuronal.....	15
2.15 Métodos de Aprendizaje	15
a) Aprendizaje Supervisado	15
b) Aprendizaje no supervisado.....	15
c) Aprendizaje por refuerzo	15
CAPÍTULO 3.....	16
DISEÑO E IMPLEMENTACIÓN DEL ROBOT MÓVIL SEGUIDOR DE	
LÍNEA DESTREZA	16
3.1 Selección de la tarjeta de desarrollo	16
a) NRF52-DK.....	16
b) Tarjeta Raspberry Pi 3	17
c) Tarjeta DECA	17
d) Tarjeta OpenMV	18
e) Placa TMDSDC3359.....	18
3.2 Diseño electrónico del robot móvil seguidor de línea destreza	20
a) Sistema Sensorial	21

b) Sistema de control.....	21
c) Interfaz.....	23
d) Micromotores y ruedas.....	24
e) Fuente de alimentación.....	25
f) Arranque del robot.....	26
3.3 Diseño estructural del robot móvil seguidor de línea destreza	26
a) Modelado 3D del subconjunto de piezas	27
b) Laminado 3D del subconjunto de piezas.....	28
c) Impresión 3D del subconjunto de piezas	29
d) Post Procesado del subconjunto de piezas.....	30
e) Ensamblaje del robot móvil seguidor de línea destreza	30
3.4 Red Neuronal para el robot móvil seguidor de línea destreza	32
a) Entrada del Sistema	32
b) Capa Oculta	32
c) Salida del Sistema	32
3.5 Entrenamiento para la red neuronal	33
a) Configuración y entrenamiento de la Red Neuronal	34
b) Algoritmo del robot móvil.....	36
c) Adquisición de imagen	37
d) Procesamiento de imagen.....	38
3.6 Funcionamiento de la red neuronal	39
3.7 Control del shield para motores	39
3.8 Resultado final del robot móvil seguidor de línea destreza.....	40
3.9 Resumen de costos	41
CAPÍTULO 4.....	42
PRUEBAS – RESULTADOS	42
4.1 Pruebas en pista con un escenario controlado	42

CONCLUSIONES	47
RECOMENDACIONES.....	49
REFERENCIAS.....	50
ANEXOS	55

ÍNDICE DE FIGURAS

Figura 2.1 Cinemática de un robot móvil con arquitectura diferencial	7
Figura 2.2 Pista de competencia categoría seguidor de línea destreza CER 2013	8
Figura 2.3 Funcionamiento general de una neurona.....	11
Figura 2.4 Arquitectura general de una red neuronal	12
Figura 2.5 Etapas de una red neuronal.....	13
Figura 3.1 Tarjeta NRF52-DK	16
Figura 3.2 Tarjeta Raspberry Pi 3.....	17
Figura 3.3 Tarjeta DECA	17
Figura 3.4 Tarjeta OpenMV	18
Figura 3.5 Tarjeta TMS320C3359.....	18
Figura 3.6 Esquema de proceso destacado para las tarjetas de desarrollo	19
Figura 3.7 Cámara Cam H7.....	21
Figura 3.8 Tarjeta OpenMV conectada con shield para motores	23
Figura 3.9 Pantalla LCD	23
Figura 3.10 Rueda Pololu Tipo D.....	25
Figura 3.11 Batería Turnigy 800mAh.....	25
Figura 3.12 Esquema de alimentación	26
Figura 3.13 Conexión del módulo de arranque	26
Figura 3.14 Piezas del robot móvil seguidor de línea destreza	27
Figura 3.15 Vistas del prototipo ensamblado.....	28
Figura 3.16 Slicing en Objet Studio.....	28
Figura 3.17 Impresión 3D en Stratasys Objet 30 Pro	29
Figura 3.18 Post Procesado de las piezas del robo móvil	30
Figura 3.19 Prototipo final	31
Figura 3.20 Vistas del prototipo final	31
Figura 3.21 Diagrama de bloques de la arquitectura de la red neuronal.....	32
Figura 3.22 Interfaz Edge Impulse	33
Figura 3.23 Interfaz Edge Impulse configuración de la red neuronal	34
Figura 3.24 Interfaz Edge Impulse de la salida del entrenamiento	35
Figura 3.25 Archivos generados en Edge Impulse	35
Figura 3.26 Diagrama de flujo del algoritmo en la tarjeta OpenMV.....	36
Figura 3.27 Ejemplos de la adquisición de imagen	37

Figura 3.28 Funciones para el procesamiento de imagen de entrada	38
Figura 3.29 Programación de la red neuronal en Open MV IDE	39
Figura 3.30 Programación para el control del shield para motores	40
Figura 3.31 Prototipo Implementado	40
Figura 4.1 Pistas en escenario controlado	42
Figura 4.2 Pista con líneas de tendencia	46

ÍNDICE DE TABLAS

Tabla 3.1 Resumen característico de las tarjetas electrónicas.....	20
Tabla 3.2 Micromotores motorreductores.....	24
Tabla 3.3 Resumen de Costos del Proyecto	41
Tabla 4.1 Resultados en la pista, red neuronal con 5 neuronas en la capa oculta	43
Tabla 4.2 Resultados en la pista, red neuronal con 10 neuronas en la capa oculta	43
Tabla 4.3 Resultados en la pista, red neuronal con 12 neuronas en la capa oculta	44
Tabla 4.4 Resultados en la pista, red neuronal con 15 neuronas en la capa oculta	44
Tabla 4.5 Resultados en la pista, red neuronal con 20 neuronas en la capa oculta	45

RESUMEN

Con el avance tecnológico y la fácil adquisición de nuevos elementos favorables a las técnicas de visión artificial, además de sus aplicaciones mediante el uso de inteligencia artificial sumado ante el escaso desarrollo de prototipos robóticos móviles que a partir de dicha tecnología en las competencias nacionales de robótica, nace una oportunidad de emplear, ensayar e implementar algoritmos inteligentes que permitan el control en la trayectoria de un prototipo robot móvil seguidor de línea destreza mediante tecnologías de vanguardia.

El presente trabajo se ha dirigido hacia el desarrollo de un robot móvil seguidor de línea destreza con arquitectura diferencial fabricado con tecnología SLA en conjunto con una red neuronal con visión artificial, basándose en estudios de software libre para esta aplicación, en función del reglamento de competencia y características técnicas del prototipo; se determinó que la tarjeta OpenMV Cam H7 con núcleo ARM Cortex A8 Sitara AM335x es idónea la cual alberga la red neuronal para controlar la trayectoria del prototipo robot móvil implementados para el desarrollo del trabajo de investigación.

A través de la creación del prototipo en conjunto con redes neuronales para el control de la trayectoria mediante visión artificial, se obtuvo como resultado una efectividad del 100 % en la aplicación de una red neuronal con 12 neuronas en la capa oculta en un escenario controlado con alto grado de dificultad.

Palabras claves: robot móvil, prototipo, seguidor de línea destreza, visión artificial, algoritmo inteligente, trayectoria.

ABSTRACT

Throughout time, artificial vision technology has developed rapidly, promoting various applications. However, in our country, the development of dexterity line follower robot prototypes based on artificial vision is scarce during national competitions. This has created the opportunity to employ, test, and implement intelligent algorithms that allow the control in the trajectory of a prototype robot follower of the dexterity line using cutting-edge technology.

This work of research and experimentation is directed towards the development of a dexterity line follower mobile robot with differential architecture manufactured with SLA technology in conjunction with a neural network with artificial vision, based on different studies of free software for this application, considering the different characteristics and regulations of the prototype in a determined category. It was determined that the OpenMV CamH7 card with an ARM Cortex A8 Sitara AM335X core was suitable to allocate the intelligent algorithms such as neural networks to control the trajectory of the prototype robot for the development of the case study.

Through the creation of the prototype in conjunction with neural networks to control the trajectory through artificial vision, as a result, 100% effectiveness was obtained in the application of a neural network with 12 neurons in the hidden layer in a controlled scenario with a high degree of difficulty.

Keywords: mobile robot, prototype, the follower of skill line, artificial vision, intelligent algorithm, trajectory.

INTRODUCCIÓN

El presente trabajo de titulación se basó en la aplicación de visión e inteligencia artificial, y como han ayudado de gran manera a solucionar problemas en diversos ámbitos; en la actualidad se cuenta con trabajos relacionados a nuestra investigación como por ejemplo: Obtención de rutas de navegación óptimas, algoritmos de búsqueda, Aprendizaje del Comportamiento Humano en el Espacio Articular del Robot con el uso de Redes Neuronales (Hossian, y otros, 2018), (Aguilar Campos & Sandoval Suquillo, 2018), (Ramírez Díaz, 2019); y aplicación de visión artificial como: Navegación Autónoma de Robots, Navegación autónoma basada en maniobras bajo estimación de posturas humanas, Navegación Autónoma de un vehículo Pequeño en Interiores (Castañeda Bermúdez & Salguero Lozano, 2017), (Barahona Guamani, 2019), (Buendia Ríos, 2017).

En el Capítulo 1 se detalla el planteamiento del problema, la justificación del proyecto, así como también los objetivos propuestos.

En el Capítulo 2 se contempla temas teóricos de hardware y software del prototipo, donde se enfoca en los conceptos generales sobre la Inteligencia Artificial, sus algoritmos inteligentes mediante Visión Artificial, sus funciones para la adquisición y procesamiento de imágenes, además de una descripción general de los requerimientos que engloba a la categoría seguidor de línea destreza.

En el Capítulo 3 se describen las características y funciones principales de las tarjetas electrónicas para el desarrollo de aplicaciones mediante visión artificial de manera que se acople a los requerimientos del prototipo; se presentan las propiedades electrónicas y mecánicas que componen al robot móvil, su diseño estructural, el dimensionamiento tanto electrónico como mecánico, además se describe la arquitectura de la red neuronal con visión artificial empleada para el control de la trayectoria del prototipo.

En el último Capítulo se enuncian las conclusiones y recomendaciones establecidas acorde al propósito del presente trabajo de titulación, a través de la metodología experimental donde se procedió al análisis de pruebas realizadas en un escenario controlado con las cuales se logró concluir que el trabajo de titulación cumplió con los objetivos planteados.

CAPÍTULO 1

ANTECEDENTES

1.1 Planteamiento del problema

Actualmente la robótica de competencia ha sido de gran acogida a nivel mundial en diversas categorías, por lo tanto; Ecuador no es la excepción ya que cada año se realizan torneos de robótica donde los participantes han evolucionado los diseños mecánicos, electrónicos y software de los prototipos robóticos, en ese sentido los integrantes del grupo ASU del área académica club de robótica "Robotics UPS UIO" de la Universidad Politécnica Salesiana sede Quito campus Sur, han desarrollado prototipos robóticos tradicionales de competencia para diferentes categorías.

Para la categoría Seguidor de Línea Destreza que consiste en seguir un camino con bifurcaciones desde un punto inicial a un punto final cuyo escenario tiene un fondo blanco con líneas de color negro, los integrantes del club han desarrollado robots móviles con un diseño mecánico, electrónico y de control tradicional, haciendo uso de sensores reflectivos (QTR-8A o su equivalente CNY70) o de sensores infrarrojos digitales que tienen respuesta inmediata, los mismos que a su vez forman una hilera de dos o más sensores (uno a cada lado de la línea) por lo general se los instala en el frente del chasis y paralelamente al piso permitiendo que el robot móvil seguidor de línea destreza pueda seguir la línea.

Estos robots seguidores de línea destreza presentan errores aleatorios al momento de la competencia debido a factores como: las variaciones del voltaje de la batería ya sea por descarga o sobrecarga, el aumento en la velocidad de desplazamiento, el ruido que se ve representado por el cambio de luminosidad percibida por los sensores durante todo el circuito de prueba, las vibraciones mecánicas que se generan al momento de entrar en funcionamiento, las perturbaciones que se interpretan como los cambios de la trayectoria y las impurezas en la pista de competencia; debido a estos factores el prototipo presenta altos tiempos de recorrido, pérdida de la trayectoria parcial o total en la pista de competencia, ocasionando más trabajo para el competidor ya que el prototipo requiere de procesos de recalibración constantes debido al entorno cambiante, por lo que la dificultad principal de este tipo de prototipos es como seguir la línea sin problemas, con precisión y completar el circuito en el menor tiempo posible para mejorar las participaciones en concurso de robótica de dicha categoría.

1.2 Justificación del proyecto

El presente Proyecto se enfocará en el desarrollo de un algoritmo de trayectoria mediante inteligencia y visión artificial para el aprendizaje de un robot móvil seguidor de línea en la categoría destreza sobre un escenario controlado, ya que con la visión artificial se tiene la ventaja de analizar la ruta óptima de navegación en la pista de competencia mediante adquisición de imágenes reales del entorno, obteniendo información más relevante para realizar un control más efectivo.

La mayoría de los robots que participan en los concursos de robótica de competencia a nivel nacional, específicamente en la categoría seguidor de línea destreza, obtienen datos de su entorno por medio de sensores infrarrojos o de proximidad, los cuales tienen limitada información para un control de la trayectoria más desarrollada, razón por la cual nace la necesidad de aprovechar las nuevas tecnologías como son la visión e inteligencia artificial orientando a solventar las limitaciones que presentan los prototipos fabricados de forma tradicional.

Actualmente en los concursos nacionales no se manejan prototipos robóticos con este tipo de tecnología, por lo que se pretende encaminar al grupo ASU del área académica club de robótica "Robotics UPS UIO" de la Universidad Politécnica Salesiana sede Quito campus Sur, hacia una nueva etapa tecnológica y desarrollo de prototipos que puedan generar investigación. Además, contribuirá a la innovación y motivación para los integrantes de dicho club con nuevos conocimientos favorables para el desempeño en futuras participaciones de concursos de robótica nacionales e internacionales.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un algoritmo de trayectoria para un robot móvil seguidor de línea en la categoría de destreza mediante visión e inteligencia artificial utilizando software libre.

1.3.2 Objetivos específicos

- Analizar el estado del arte de sistemas de navegación de robots móviles para la identificación de las variables que intervienen en el proceso.

- Diseñar el hardware para el robot móvil seguidor de línea destreza utilizando los lineamientos reglamentarios de dicha categoría.
- Diseñar un algoritmo de trayectoria para el robot móvil seguidor de línea destreza mediante visión artificial con un lenguaje de programación de software libre.
- Implementar un prototipo para las pruebas del algoritmo inteligente utilizando una tarjeta ARM en un escenario controlado.
- Realizar pruebas con el prototipo implementado para la validación de su funcionamiento a través de un escenario controlado bajo las reglamentaciones respectivas de una competencia real.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Introducción

Se presentan las propiedades y características que engloban a un robot móvil seguidor de línea destreza, partiendo desde sus requerimientos básicos hasta el análisis para la implementación estructural, electrónica y mecánica; así como también una descripción general de la reglamentación que regulan los requerimientos técnicos del prototipo, las especificaciones técnicas de las pistas, el desarrollo de la competencia y la determinación de los ganadores. Además, se realizó una presentación y estado del arte del control a través de algoritmos inteligentes mediante la visión e inteligencia artificial, analizando sus conceptos y principales características que se presentarán a lo largo del capítulo para llegar al procedimiento de la implementación de estos para el caso de estudio.

2.2 Robot Autónomo

Son dispositivos que generalmente están constituidos por una estructura mecánica, sistema de energía, sistema de accionamiento, sistemas sensoriales y sistema de control, con la capacidad de adquirir datos de su entorno para procesarlos con el objetivo de realizar una función específica en ambientes simples o complejos sin la necesidad de estar constantemente supervisados, guiados o controlados por operadores humanos (Vera Arenas & Alejandro Proaño, 2016).

2.3 Sistema de locomoción de un robot

El sistema de locomoción es la capacidad de movimiento que tiene un prototipo robótico, el cual le permite trasladarse en el espacio desde un punto inicial a un punto final; para el caso de estudio en particular se empleará un robot móvil con una arquitectura del tipo diferencial con ruedas, debido a que es ideal por sus ventajas en diseño, fabricación, control y costo (Vera Arenas & Alejandro Proaño, 2016).

2.4 Robot móvil Seguidor de Línea Destreza

Es un robot móvil de tipo diferencial completamente autónomo, tiene como objetivo seguir una línea con bifurcaciones desde un punto inicial a un punto final en el menor tiempo posible, la línea generalmente es de color negro cuyo fondo es de color blanco, el diseño del escenario varía de acuerdo con el criterio de los organizadores del torneo.

Tradicionalmente este tipo de prototipo robótico usa una hilera de al menos dos o más sensores reflectivos o infrarrojos (uno a cada lado de la línea) que por lo general se los instala en el frente del chasis y paralelamente al piso permitiendo que el robot móvil siga la línea. La arquitectura mecánica, electrónica y de control para la fabricación del robot móvil seguidor de línea destreza es libre de acuerdo con los criterios del participante (Tapiero Santa, 2019).

Este tipo de robot está presente en los diferentes concursos de robótica de competición que se organizan anualmente por las instituciones de educación superior en todo el Ecuador y en ciertos torneos a nivel mundial, donde se fomenta el desarrollo de la robótica de competencia e innovación para la categoría Seguidor de Línea Destreza siendo cada vez más competitiva (Robot Games ZL7, 2019).

a) Reglamento para la categoría Seguidor de Línea Destreza

En los concursos de robótica que se desarrollan en todo el Ecuador y a nivel mundial existen reglamentaciones que generalmente regulan los requerimientos técnicos del prototipo, las especificaciones técnicas de las pistas, el desarrollo de la competencia y la determinación de los ganadores para cada una de las categorías dentro de la robótica de competencia (Robot Games ZL7, 2019).

En el XV Concurso Ecuatoriano de Robótica “CER 2019” organizado por la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia “CEDIA” junto con la Universidad de Cuenca en el año 2019, a continuación, se citan los puntos más relevantes del reglamento para la categoría seguidor de línea destreza.

- El robot debe presentar completa autonomía.
- Las dimensiones y el peso del prototipo no tienen restricciones.
- Los participantes deben diseñar y fabricar sus prototipos de acuerdo con la presente reglamentación.
- El tipo de control del sistema es libre.
- Los escenarios serán diseñados por los organizadores del evento, y estos tienen un fondo blanco con bifurcaciones de color negro.
- El tiempo que tiene cada participante es de 3 minutos cronometrados, tiempo en el cual el prototipo debe seguir la trayectoria de la pista, cuando el robot llega a la meta el tiempo finaliza con la participación del competidor.

- De principio a fin el robot móvil debe encontrarse dentro del área de pista cumpliendo la trayectoria marcada.
- Si el prototipo pierde la trayectoria de la pista y retorna al mismo punto no se finaliza el tiempo.
- Cuando el prototipo pierda la trayectoria parcial, completamente o presenta inmovilidad durante 5 segundos, finaliza su participación.
- Cuando el prototipo salga de la pista o tome un atajo dentro de la trayectoria marcada, se declara como intento fallido.
- Cada competidor dispone de 2 participaciones para cumplir la trayectoria de la pista, tomando como resultado el mejor tiempo obtenido.
- Se determina la etapa final con los 5 mejores tiempos del primer cronometraje.
- Se declara ganador de la competencia al robot móvil que haya completado la trayectoria en el menor tiempo posible, además de haber cumplido las reglamentaciones estipuladas.

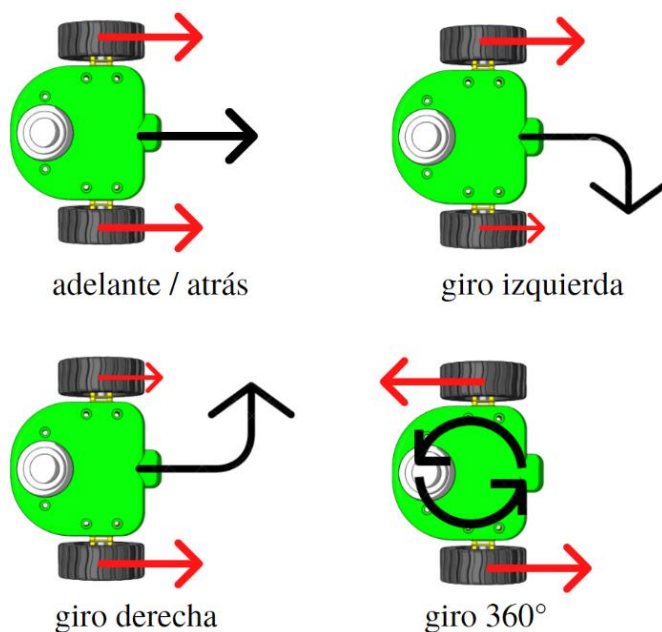
La reglamentación para la categoría seguidor de línea destreza como para las demás categorías evolucionan y están sujetas a continuos cambios, debido a que toman como referencia reglamentos presentados por instituciones de educación superior y competiciones de robótica (CEDIA, 2019, págs. 1-2).

b) Robot móvil seguidor de línea destreza con arquitectura diferencial

El robot móvil seguidor de línea destreza tiene como tipo de arreglo cinemático más común y eficiente el de arquitectura o tracción diferencial presente en la Figura 2.1, esta configuración es la más frecuente en robots móviles para moverse dentro de interiores, este prototipo presenta dos ruedas con sus respectivos motores de forma paralela entre sí y una rueda de apoyo sin tracción como tercer punto de apoyo ubicada en el centro del lado opuesto del lugar donde se encuentran ubicadas las ruedas (Baturone, 2001).

La traslación y rotación de este tipo de locomoción diferencial se determinan por el movimiento independiente de cada una de las ruedas modificando la velocidad, esto permite tener ventajas en cuanto a su movilidad como por ejemplo le permite girar sobre su propio eje, así como también le permite realizar distintos tipos de giros para una mejor respuesta durante la trayectoria del robot móvil (Cortéz, 2011).

Figura 2.1 Cinemática de un robot móvil con arquitectura diferencial



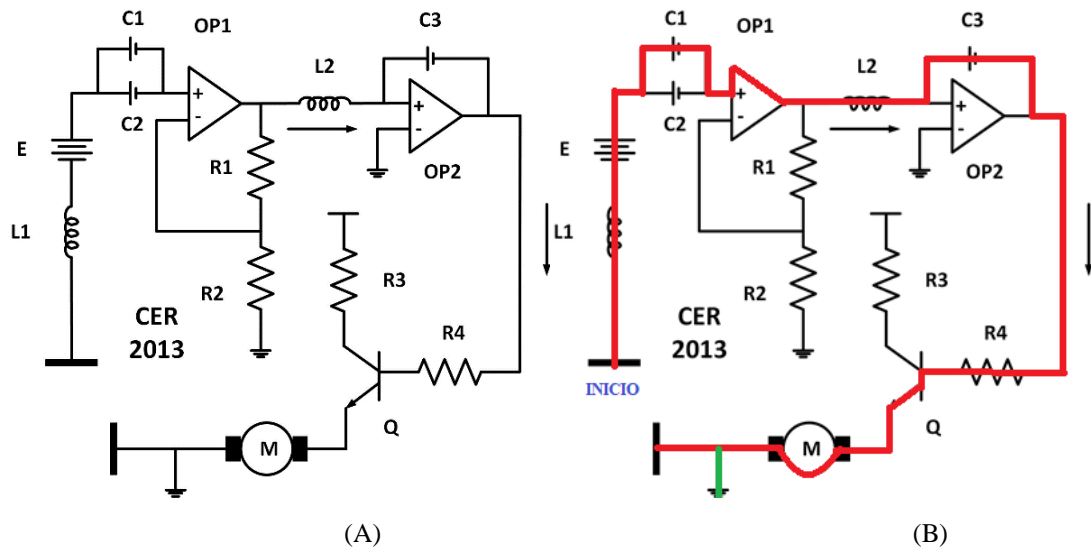
La disposición del movimiento se realiza modificando la velocidad de las ruedas izquierda y derecha independientemente, Elaborado por: Clelio Endara y Emerson Maigua.

c) Pistas para el robot móvil seguidor de línea destreza

Las pistas o escenarios para el seguidor de línea destreza por lo general tienen un fondo blanco y líneas con bifurcaciones de color negro que pueden ser curvas, ángulos rectos, discontinuidades, entre otras; son diseñadas con un alto grado de dificultad basado a los reglamentos y criterios de los organizadores de cada competencia que regula este diseño para que los participantes puedan agregar o tomar en cuenta ciertas decisiones al momento del desarrollo de la competencia, como por ejemplo, obedecer a la tendencia de línea recta al aparecer un cruce, no girar, sino seguir el camino, o al contrario en cruces solo obedecer a un lado dominante (izquierda o derecha), todas estas consideraciones están sujetas siempre al reglamento de cada competencia.

Para el Concurso Ecuatoriano de Robótica en el año 2013 (CER 2013), la pista de competencia que se muestra en la Figura 2.2 (A) demuestra la prioridad del sentido de giro hacia la izquierda, mientras que en la Figura 2.2 (B) se puede observar que la trayectoria ideal es a través de la línea color rojo cumpliendo esta decisión; además de observar que la línea color verde presenta un retorno donde el robot debe girar y volver por el mismo camino obedeciendo la decisión de ir a la izquierda y terminar su trayectoria en el menor tiempo posible.

Figura 2.2 Pista de competencia categoría seguidor de línea destreza CER 2013



(A) Vista superior de la pista categoría seguidor de línea destreza en el concurso CER 2013, (B) Vista superior de la trayectoria ideal para que el robot seguidor de línea destreza pueda completar el circuito de prueba de la Figura 2.1 (A), Elaborado por: Clelio Endara y Emerson Maigua.

2.5 Visión Artificial

Este tipo de tecnología tiene como objetivo tratar de emular artificialmente el sentido de la vista humana a través de la adquisición, procesamiento e interpretación de imágenes capturadas desde dispositivos electrónicos ópticos como son las cámaras y el procesamiento a partir de tarjetas de desarrollo o computadores compatibles con dicha tecnología (Benavent Pla, 2017).

Los sistemas de visión artificial tienen procesos para el análisis de las imágenes como son: la capacitación de imágenes, el almacenamiento de la información captada, el procesamiento de imágenes y la interpretación de los resultados; estos procesos logran resultados precisos, confiables y repetibles.

Entre las principales aplicaciones de la visión artificial se tiene:

- Identificación de objetos, formas y personas.
- Análisis de color y variación de luz.
- Análisis de la cinemática de objetos.
- Guía en la ubicación y orientación en un espacio de 2D O 3D.
- Medición entre dos puntos o ubicaciones geométricas de un objeto.

2.6 Estado del arte de los sistemas de navegación en robots móviles.

El problema de la navegación de robots móviles en diferentes entornos ha sido un tema de interés para los investigadores en las dos últimas décadas. Sin embargo, el desafío actual es la navegación autónoma en ambientes que contienen obstáculos estáticos y móviles, especialmente cuando los objetos que ocupan el entorno tienen una posición y movimiento desconocido (Suárez Romero & del Pozo Quintero, 2016).

Desde la inclusión del robot móvil en la industria, la navegación de este ha venido siendo un caso de estudio donde han surgido tecnologías de toda índole, para nuestro tema de trabajo de titulación dirigimos nuestro estudio e investigación en la resolución de los problemas cinemáticos de robots móviles a través de algoritmos inteligentes como son las redes neuronales.

El área de utilización de los robots móviles es cada vez más amplia, debido a que la tarea de navegación cobra mayor importancia en aplicación de áreas como la industria, exploración espacial, educación, entretenimiento, asistencia médica, seguridad, entre otras (Suárez Romero & del Pozo Quintero, 2016).

En la actualidad ha ido aumentando la tecnología a la cual va dirigido nuestro estudio como es la inteligencia y visión artificial, existe una gran variedad de softwares libres orientados al desarrollo de algoritmos programables basados en lenguaje como son: Basic, C, Ensamblador y Pascal de modo que los fabricantes crean sus plataformas de programación y a la par producen tarjetas de desarrollo complementarias a los mismos, que permiten crear mejores algoritmos inteligentes para el control de la navegación de los robots móviles.

Una de las soluciones que ha tenido mayor acogida es el área de la visión artificial que en conjunto con un algoritmo inteligente han dado mitigación a un sin número de problemas como por ejemplo: Deep Learning para la detección de peatones y automóviles, Desarrollo de algoritmos para la exploración submarina mediante vehículos autónomos con visión artificial monocular, Vehículo autónomo con visión artificial utilizando OpenCV, Sistema de detección y conteo de vehículos utilizando visión artificial (Vizcaya Cárdenas, 2018), (Trabes, 2018), (Peña Merino, 2017), (Benavent Pla, 2017).

La gran cantidad de problemáticas resueltas en la navegación de robots móviles es gracias al uso de diferentes tipos de control como son: PID, lógica difusa, sistemas embebidos, redes neuronales, entre otros; por esta razón se orientó el tema de investigación hacia las redes neuronales debido a su gran impacto en el uso de la visión artificial en la generación de nuevos conocimientos para generaciones posteriores.

2.7 Inteligencia Artificial

La inteligencia artificial es “la habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana” (Rouhiainen, 2018). Pero; para tener una definición más clara y detallada, se puede definir que la Inteligencia Artificial es la capacidad que poseen las máquinas para imitar el comportamiento inteligente humano a través de la utilización de algoritmos, en donde esta máquina aprende los datos tomados de su entorno para luego utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano.

Actualmente el crecimiento exponencial de los procesos, han creado la necesidad que los dispositivos y softwares informáticos lleguen a tener un aprendizaje dirigido a la toma de decisiones autónomas, motivo por el cual la inteligencia artificial está diseñada para resolver distintas situaciones cotidianas y complejas a través de dispositivos tecnológicos orientados a la emulación inspirada en el comportamiento y accionamiento que realizaría un ser humano.

La Inteligencia Artificial (IA) con aplicaciones orientadas al beneficio y ayuda significativa hacia los seres humanos, brindan una mejor calidad de vida resolviendo problemas presentes en casi todas las situaciones actuales del hombre moderno (Rouhiainen, 2018).

En la actualidad este tipo de tecnologías se las puede aplicar en:

- En varias áreas de la industria se aplica el reconocimiento, clasificación y etiquetado de imágenes.
- Una herramienta fundamental en sectores industriales que requieren un mantenimiento de tipo predictivo en sus maquinarias y equipos.
- Aplicaciones industriales donde se requiere detectar y clasificar objetos como es el caso en la industria de vehículos autónomos.

2.8 Redes Neuronales

Las redes neuronales en la robótica son una parte de la inteligencia artificial que tienen un objetivo bioinspirado como es simular un cerebro humano con sus respectivas neuronas y conexiones. En la actualidad debido a la creciente tecnología las redes neuronales han podido sobrepasar sus expectativas, son capaces de aprender, entrenarse y realizar operaciones igual o mejor que un operador humano llegando así a crear una inteligencia artificial “real”, capaz de tomar decisiones complejas en tiempo real (Suárez Romero & del Pozo Quintero, 2016).

a) Neuronas

El componente mínimo presente en la red neuronal se llama neurona o también elemento de procesamiento, cada una de estas neuronas presentes en la red neuronal tienen una capacidad limitada de cómputo o procesamiento restringidas por un conjunto de instrucciones, además posee una pequeña memoria para almacenar pesos y activaciones (Matich, 2001). Para simplificar el funcionamiento general de una neurona se puede verlo a través de la Figura 2.3 donde se puede apreciar que una neurona se reduce finalmente a un sumador, que tiene sus propias entradas, pesos y funciones de activación que se van interrelacionando con otras neuronas a través de la arquitectura de la red neuronal.

Figura 2.3 Funcionamiento general de una neurona

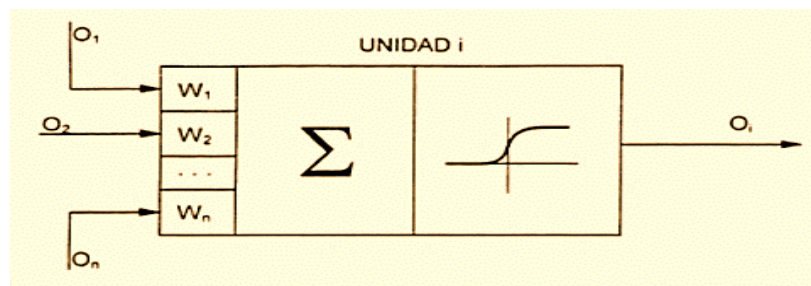


Ilustración didáctica de una neurona, Elaborado por: Clelio Endara y Emerson Maigua.

2.9 Arquitectura de una red neuronal

En una red neuronal no solo se expone sus componentes mínimos con su respectivo comportamiento individual, también se realiza un análisis en la interconexión que existe entre ellas. Se agrupan las neuronas en capas y cada una de estas tiene un número variable de neuronas en un conjunto con un comportamiento similar, por lo que así se van creando varias capas en la red neuronal.

Cada una de estas capas está conectada con la siguiente en su totalidad o parcialmente excepto en la última de estas capas, que es donde se encuentra la salida total de la red neuronal (Hossian, y otros, 2018).

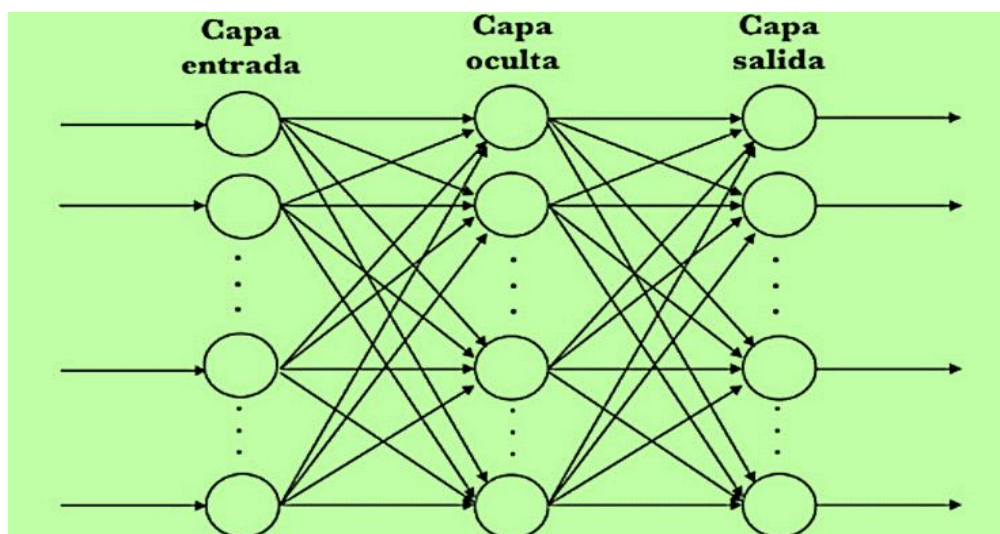
a) Arquitectura MobileNet

Es una nueva arquitectura de red neuronal que está específicamente diseñada para entornos móviles y con recursos limitados, mejora el rendimiento de vanguardia de los modelos móviles en múltiples tareas y puntos de referencia, así como en un espectro de diferentes tamaños de modelos. Las MobileNets se basan en una arquitectura optimizada que utiliza convoluciones separables en profundidad para construir redes neuronales profundas y ligeras (Sadler, Howard, Zhu, Zhmogimov, & Chen, 2019). Entre las variantes de esta arquitectura existe la versión 2 con base de un Backpropagation por medio de Deep Learning haciendo uso del método de aprendizaje supervisado, este tipo de arquitectura logra un equilibrio óptimo entre precisión y rendimiento.

2.10 Tipos de capas de una red neuronal

Una red neuronal tiene distintas formas de distribuir, conectar e interrelacionar sus capas. Existen tres tipos de capas y cada una con sus respectivas neuronas, de manera general se puede entender su arquitectura en la Figura 2.4 donde se pueden apreciar de manera genérica la distribución de las capas y sus conexiones.

Figura 2.4 Arquitectura general de una red neuronal



Correlación entre capa y capa presentes en una red neuronal, Fuente: (Torres, 2018).

a) Capa de entrada

Depende de los datos o variables que genere el problema a resolver ya que con estos se puede definir el número y tipo de neuronas que la constituyen.

b) Capas intermedias

Depende de la complejidad del problema que la red va a resolver, por esta razón se puede necesitar más de una capa. En un análisis y tratamiento de estas capas se pueden obtener propiedades interesantes para la red neuronal como son generalización, extracción de características, adaptabilidad, etc.

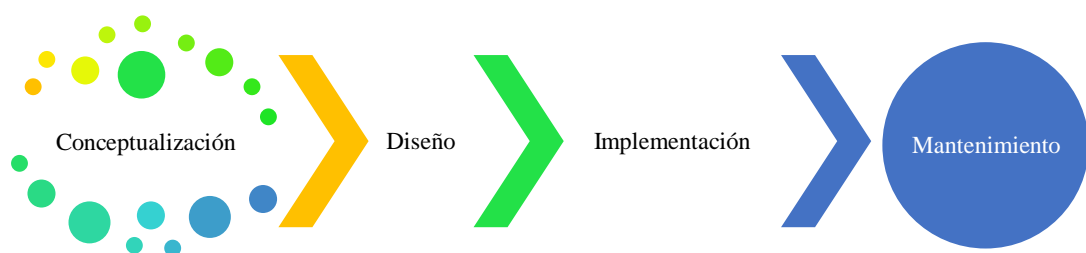
c) Capa de salida

Dependiendo del formato de nuestra arquitectura y la aplicación que tendrá la red neuronal se va colocando el número de neuronas de salida.

2.11 Etapas de construcción y manejo de una red neuronal

Las redes neuronales generalmente en su construcción y manejo tienen cuatro etapas o fases a seguir mismas que se muestran en la Figura 2.5.

Figura 2.5 Etapas de una red neuronal



Construcción y manejo de una red neuronal, Elaborado por: Clelio Endara y Emerson Maigua.

a) Fase de conceptualización

Se estudia y analiza un modelo de red neuronal que se ajuste al problema a resolver, partiendo de una observación caracterizada del problema.

b) Fase de diseño

En esta parte se determina la estructura que tendrá la red, el tipo de elemento de procesamiento y el algoritmo de aprendizaje que utilizará la misma.

c) Fase de implementación

Para que la red neuronal logre resolver un problema, esta deberá ser entrenada para su correcta ejecución; este proceso se denomina entrenamiento que es una configuración óptima realizada mediante la introducción de ejemplos.

d) Fase de mantenimiento

En esta parte se realiza un seguimiento del funcionamiento de la red neuronal mediante el monitoreo y adquisición de datos para tener nuevos aprendizajes o descartar otros ejemplos obsoletos (Matich, 2001).

2.12 Deep Learning

Actualmente es una de las aplicaciones de mayor crecimiento y rendimiento en el campo de la inteligencia artificial, además que es un subcampo del aprendizaje supervisado donde su utilización está orientado a problemas de alta complejidad que involucren una gran cantidad de datos.

Mediante el uso de redes neuronales se realiza el aprendizaje profundo, donde los datos reconocen sus respectivas relaciones y patrones complejos que se van organizando en capas. Es necesario una gran capacidad de procesamiento y enormes conjuntos de información para su aplicación (Rouhiainen, 2018).

2.13 Backpropagation

Es una red neuronal, parte del aprendizaje supervisado y actualmente la más popular para aplicaciones de visión artificial, posee un conjunto definido de entradas y salidas, el cual emplea solo entrenamientos supervisados donde los ejemplos de la red neuronal de entrada – salida correctos deben ser abundantes. “No existe una ley matemática que permita calcular cuantos ejemplos deben ser utilizados para el entrenamiento de la red neuronal, solo se posee reglas empíricas o experimentales sujetas al problema donde se estiman la cantidad de ejemplos” (San Juan, Jamett, Kaschel, & Sánchez, 2015).

Partiendo de esta red se desarrolla un algoritmo aún más específico para el procesamiento de la red neuronal, además facilita de mayor forma la aplicación en específico para el caso de estudio en la realización de un algoritmo inteligente que se implementará al robot móvil seguidor de línea destreza.

2.14 Aprendizaje de una red neuronal

Esta es la característica más importante en una red neuronal, debido a que esta parte es un factor que permite determinar las ventajas y también los inconvenientes de nuestro sistema; para casos sencillos, los pesos o entradas se pueden agregar o asignar de una manera manual, pero para casos más complejos es necesario utilizar algún algoritmo para llevar a cabo el proceso de entrenamiento.

2.15 Métodos de Aprendizaje

Existen tres métodos de aprendizaje los cuales son:

- Supervisado.
- No supervisado.
- Reforzado.

a) Aprendizaje Supervisado

En este método de manera general se presenta los vectores de entrada y salida deseada. La salida que es manejada por la red neuronal se compara con la salida deseada, y para reducir el error cometido se van modificando los pesos de la red, motivo por el cual se repite de forma iterada hasta que este error tenga un porcentaje de confiabilidad aceptable.

b) Aprendizaje no supervisado

Este método es uno de los más cercanos al sistema biológico de aprendizaje, no se tiene como en el anterior método mencionado el vector de la salida esperada, el único vector que continua es la entrada en el conjunto de entrenamiento. El algoritmo hace que las salidas sean consistentes modificando los pesos, es decir que a entradas muy similares la red neuronal compute la misma salida. Las salidas se asocian a las entradas acorde al proceso de entrenamiento.

c) Aprendizaje por refuerzo

Este aprendizaje tiene como primicia que aprende a través de la experiencia, en palabras más generales se debe darle un refuerzo positivo en cada acierto de la red. Este algoritmo se puede entender de forma sencilla como por ejemplo el entrenamiento a un perro cuando se le da una recompensa por aprender algún truco, de la misma manera es en la red neuronal he ahí su nombre refuerzo positivo (Rouhiainen, 2018).

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL ROBOT MÓVIL SEGUIDOR DE LÍNEA DESTREZA

El objetivo principal del caso de estudio es el control de la trayectoria de un robot móvil seguidor de línea destreza mediante visión e inteligencia artificial, para lo cual se requiere del uso de procesamiento de imágenes en tiempo real y de un control inteligente, donde el sistema debe ser portable para que el prototipo robótico albergue todos estos procesos. Además de un chasis o estructura idónea para el robot, que permita una ubicación óptima para los periféricos que tendrán una comunicación como: una cámara, driver controlador de motores y entradas digitales, los cuales permitan el accionamiento y control del robot móvil.

3.1 Selección de la tarjeta de desarrollo

Para una selección de una placa controladora ideal, se optó entre diversas tarjetas de desarrollo disponibles en el mercado ecuatoriano, motivo por el cual la selección de una placa para el caso de estudio se la realizó mediante un análisis con criterios que beneficien tanto hardware como software en la implementación del proyecto, las cuales se presentan a continuación.

a) NRF52-DK

La placa desarrollada por Nordic Semiconductor (Figura 3.1), posee un solo núcleo 32-bit ARM Cortex M4F con una memoria de programa de 512 kB (flash con cache) y una memoria RAM de 64kB, ventajas como compatibilidad con Arduino en uso de módulos a través de sus periféricos, comunicación UART para conexión a una cámara con un software libre; su costo es de 40 USD disponible en el mercado ecuatoriano (Nordic Semiconductor, 2017).

Figura 3.1 Tarjeta NRF52-DK



Tarjeta de desarrollo NRF52-DK, Fuente: (Nordic Semiconductor, 2017).

b) Tarjeta Raspberry Pi 3

Raspberry es una tarjeta electrónica (Figura 3.2), orientada al desarrollo que posee un núcleo ARM, tiene un procesador Cortex de 1.2 GHz, además de una memoria RAM de 1 GB, trabaja con el sistema operativo Linux, disponibilidad de conexión con dispositivos externos y su costo ronda en los 90 \$ (Raspberry Pi, 2015).

Figura 3.2 Tarjeta Raspberry Pi 3



Tarjeta de desarrollo Raspberry Pi 3, Fuente: (Raspberry Pi, 2015).

c) Tarjeta DECA

La placa DECA (Figura 3.3), por su arquitectura se la clasifica en el grupo de FPGA, posee un núcleo Nios II Gen2, tipo RISC de 32 bits, una memoria RAM de 512MB, dispone de conectividad USB y ethernet, posee periféricos de entrada y salida, además trabaja en sistema operativo Windows 7 en adelante, esta tarjeta ronda un costo de 240 USD en el mercado estadounidense, no se encuentra disponible en el mercado ecuatoriano actualmente (Arrow Development Tools, 2015).

Figura 3.3 Tarjeta DECA

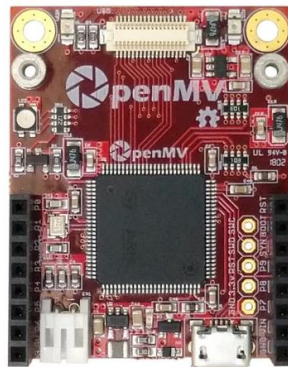


Tarjeta de desarrollo DECA, Fuente: (Arrow Development Tools, 2015).

d) Tarjeta OpenMV

La placa OpenMV de la Figura 3.4, posee un núcleo STM32H743VI ARM Cortex M7, una memoria SRAM de 1 MB y 2 MB de flash, extensible hasta 32 GB (dependiendo del modelo), dispone de conectividad USB, I2C y SPI, posee periféricos de entrada y salida, además trabaja con software libre, esta tarjeta ronda un costo de 65 USD en el mercado estadounidense, no se encuentra disponible en el mercado ecuatoriano actualmente (Agyeman & Abdelkader, OpenMV, 2018).

Figura 3.4 Tarjeta OpenMV



Tarjeta de desarrollo OpenMV, Fuente: (OpenMV, 2018).

e) Placa TMDSDC3359

La TMDSDC3359 (Figura 3.5), de la familia A8 Sitara AM335x, tiene un núcleo ARM Cortex y un coprocesador NEON SIMD, memoria RAM de 512 MB, comunicación ethernet, etherCAT, Profibus, Profinet y más, además de poseer periféricos de entrada y salida, trabaja con software desarrollado por Texas Instruments (pagado), el costo de la tarjeta ronda los 740 USD en Estados Unidos, actualmente se encuentran disponibles en el mercado ecuatoriano (Texas Instruments, 2011).

Figura 3.5 Tarjeta TMDSDC3359



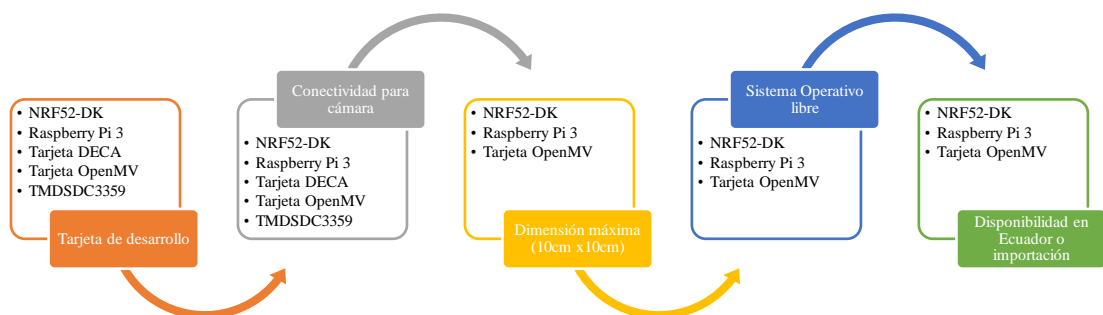
Tarjeta de desarrollo OpenMV, Fuente: (Texas Instruments, 2011).

Para realizar una correcta selección de la tarjeta de desarrollo se dispuso de la realización de un esquema de proceso destacado con el objetivo de clasificar las tarjetas dependiendo de las necesidades requeridas por el robot móvil; la tarjeta debe ser capaz de conectarse a una cámara, tener la capacidad de manejar redes neuronales, controlar módulos externos y cumplir un espacio máximo de 10 cm x 10 cm por razones de estética del hardware.

También se tiene en cuenta que su programación sea accesible a través de un software libre y para finalizar que pueda estar disponible en el mercado ecuatoriano o en su defecto que se pueda adquirir por medios de envío internacional.

El esquema de proceso destacado se muestra en la Figura 3.6, donde se puede tener una idea más simple de la selección de la tarjeta acorde a las características y funciones requeridas para el robot seguidor de línea destreza mediante visión e inteligencia artificial.

Figura 3.6 Esquema de proceso destacado para las tarjetas de desarrollo



Esquema de proceso destacado para la selección de la tarjeta de desarrollo para el robot móvil,
Elaborado por: Clelio Endara y Emerson Maigua.

A partir del esquema de proceso destacado se seleccionaron: la NRF52-DK, Raspberry Pi 3 y la tarjeta OpenMV; para proceder a una elección más específica se realizó un cuadro comparativo en la Tabla 3.1, en la cual se determinó que la tarjeta de desarrollo OpenMV es la adecuada debido a sus características como: procesador, memoria, tamaño y portabilidad, también tiene un sistema de módulo de cámara extraíble que permite interactuar con sensores, además la posibilidad de admitir dispositivos externos como controlador de motores con un bajo costo.

Tabla 3.1 Resumen característico de las tarjetas electrónicas

Tarjeta de desarrollo	Núcleo	Memoria RAM	Conexión para cámara	Sistema Operativo Libre	Precio en su mercado de origen	Disponibilidad en Ecuador
NRF52-DK	ARM Cortex M4F	64kB	✓	✓	40 USD	✓
Raspberry Pi 3	ARM x64 Cortex	1GB	✓	✓	90 USD	✓
Tarjeta DECA	Nios II Gen2	512MB	✓	✗	240 USD	✓
Tarjeta OpenMV	STM32H743VI ARM Cortex M7	4GB	✓	✓	65 USD	✗
Placa TMDSDC3359	ARM Cortex A8	512MB	✓	✗	740 USD	✗

Resumen característico de las tarjetas electrónicas, Elaborado por: Clelio Endara y Emerson Maigua.

Teniendo en cuenta lo mencionado en el párrafo anterior y las necesidades del proyecto, se puede ratificar que la tarjeta OpenMV es la adecuada para ejecutar redes neuronales con una velocidad de reacción eficiente para la aplicación del control de la trayectoria debido a su procesador que la hace básicamente una mini computadora, además su software de programación es libre y gratuito que se lo puede encontrar en la página oficial de la organización OpenMV y permite utilizar herramientas basadas de otros softwares como Keras o Python.

3.2 Diseño electrónico del robot móvil seguidor de línea destreza

Para el reconocimiento de imagen se usará la cámara CAM H7, la cual es compatible en la tarjeta OpenMV en donde se realizará el algoritmo de visión artificial, para el control de la trayectoria del robot se hace uso de un shield para motores de la organización OpenMV el cual controlará la locomoción del robot a partir del algoritmo inteligente que dará ejecución la tarjeta OpenMV.

Por otro lado, estos módulos de visión artificial de la organización OpenMV al compartir una configuración compacta para su fácil comunicación y la alimentación, internamente dispone de un control para la regulación del voltaje tanto para la tarjeta como para el driver de los motores y otros elementos electrónicos que nos permiten controlar los pines de E/S en el mundo real; para entender de manera general la composición electrónica del robot se lo ha dividido en las siguientes partes:

a) Sistema Sensorial

Para la adquisición de la imagen de la línea la cual el robot móvil deberá seguir, se utilizó la cámara Cam H7 (OV7725) compatible con la tarjeta OpenMV con una comunicación y control directos a través de la interfaz del bus de control de cámara en serie (SCCB), de la Figura 3.7, que como característica tiene que el controlador puede modificar su saturación, brillo, contraste y tamaño; el esquema electrónico se adjunta en el Anexo 1. A continuación se menciona las características técnicas más importantes:

- Compatibilidad con OpenMV.
- Resolución de imágenes hasta VGA en escala de grises de 640x480 de 8 bits o imágenes RGB565 de 640x480 de 16 bits a 75 FPS.
- Alta sensibilidad para funcionamiento con poca luz.
- Funciones de control automático de imagen.
- Controles de calidad de imagen.

Figura 3.7 Cámara Cam H7



Cámara Cam H7 conectada a tarjeta OpenMV, Fuente: (Agyeman & Abdelkader, OpenMV, 2018).

b) Sistema de control

Para el control de la trayectoria y de la programación del algoritmo inteligente mediante la visión artificial con la tarjeta OpenMV, se detalla a continuación las características destacadas de la tarjeta OpenMV Cam H7 (Agyeman & Abdelkader, OpenMV, 2018).

- Arm Cortex-M7 de 32 bits funcionando a 400MHz.

- Alto ancho de banda 1 MB SRAM / 2 MB FLASH.
- Unidad de coma flotante de doble precisión (FPU).
- Instrucciones completas de DSP, codificación JPEG de hardware. 2 UART, 2 I2C, 1 SPI, 1 CAN, 3 TIM / PWM.
- 1 USB de máxima velocidad (FS) para programación.
- 1 LED RGB y 2 LED IR integrados.
- 1 toma de tarjeta μ SD (admite hasta 64 GB).
- Regulador de conmutación de alta eficiencia (salida 1A).
- LDO de bajo ruido para alimentación analógica del sensor.
- Conector de batería Lipo. Consumo de energía inferior a 150 mA.
- El diseño de sensor modular admite múltiples sensores:
 - OV7725 640x480.
 - MT9V034 (Sensor de obturador global).
 - Sensores de imagen térmica FLIR 1,2 y 3.

Para el control de motores se eligió un driver diseñado especialmente para la tarjeta OpenMV, este shield permite controlar hasta 2 motores. Cuenta con el controlador de motor de corriente continua un TB6612 con circuito de apagado térmico incorporado y circuito de detección de bajo voltaje. Entre las características técnicas más notables se tiene:

- Controlador TB6612.
- Controla hasta 2 motores de CC.
- Circuito de apagado térmico incorporado.
- Circuito detector de baja tensión.
- Modos de función CW / CCW / freno corto / parada.
- Corriente de espera baja.
- Rango de voltaje de suministro operativo de 6 V a 9 V.

La gran ventaja de este shield es su fácil conexión con la tarjeta Open MV, ofrece una estética compacta e ideal para el diseño estructural del robot en la Figura 3.8 se puede observar la conexión entre la tarjeta y el shield, además el esquema electrónico se adjunta en el Anexo 1.

Figura 3.8 Tarjeta OpenMV conectada con shield para motores

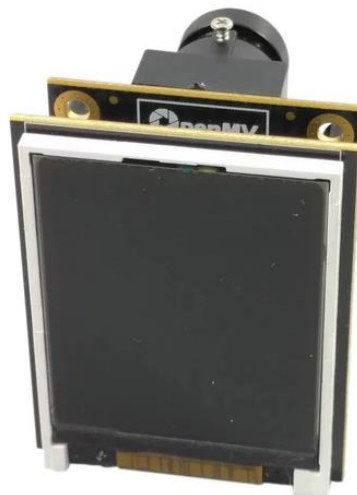


Shield para motores conectado a tarjeta OpenMV con cámara CAM H7, Fuente (Agyeman & Abdelkader, OpenMV, 2018).

c) Interfaz

Para la adquisición de datos que se implementan en el algoritmo de navegación inteligente de la red neuronal, los ejemplos de entrenamiento necesitan ser visualizados en tiempo real para una correcta adquisición de datos, por lo que se hace uso de una pantalla LCD que es otro módulo que amplía la funcionalidad con una conectividad directa de conexión simple a los periféricos de la tarjeta OpenMV como se puede ver en la Figura 3.9, además el esquema electrónico se adjunta en el Anexo 1.

Figura 3.9 Pantalla LCD



Conexión Pantalla LCD con Tarjeta OpenMV, Fuente: (Agyeman & Abdelkader, OpenMV, 2016).

Entre las principales características de esta pantalla están:

- Controlador ST7735R LCD.
- Una dimensión de 1.8 x 1.49 pulgadas.
- Resolución de 128x160 pixeles, Color 16-bit.
- Pin control de luz de fondo.

d) Micromotores y ruedas

Para la selección de los micromotores hay que tener en cuenta parámetros como la velocidad y torque; debido a que la categoría seguidor de línea destreza opera con una velocidad baja y un torque intermedio en los prototipos de competencia, esto para evitar efectos de derrape o deslizamiento especialmente al momento de seguir una trayectoria con curva muy cerrada.

El prototipo con el dimensionamiento de sus componentes electrónicos y mecánicos presentaría un peso aproximado de 500 gramos, casi la mayoría de los micromotores DC pueden mover dicho peso; como se menciona en la Tabla 3.2.

Tabla 3.2 Micromotores motorreductores

Motor	Voltaje de trabajo	RPM	Relación	Disponibilidad en Ecuador
A) Pololu	6V 9V 12V	310 RPM	100:1 1,7Kg.cm	✓
B) Fingertech	9V	9100RPM	50:1 2Kg.cm	✓
C) Maxon	9V 12V	13000RPM	50:1 3,3Kg.cm	✗

Características generales motores: A) (Pololu, 2019), B) (Fingertech Robotics, 2013) y C) (Maxon Group, 2020) , Elaborado por: Clelio Endara y Emerson Maigua.

En función de los datos anteriormente mencionados; se optó por los micromotores del fabricante Pololu debido a su alto torque y baja velocidad, de la misma manera la elección de las ruedas de goma del fabricante Pololu 42 x 19 mm con un orificio tipo D que se acopla al eje de los micromotores, además posee un labrado de caucho como se puede observar en la Figura 3.10, ideal para evitar derrapes, deslizamientos y brindar una tracción eficiente asegurando un agarre al momento en que el robot seguidor de línea destreza se moviliza en la pista de competencia brindando estabilidad al robot.

Figura 3.10 Rueda Pololu Tipo D



Rueda Pololu 42 x 19mm tipo D con motor, Fuente: (Pololu Robotics & Electronics, 2018).

e) Fuente de alimentación

Para la alimentación del robot se utilizó una batería tipo Lipo (Figura 3.11), de dos celdas es decir 7,4 V DC con capacidad de 800mAh (miliamperios hora); batería la cual el fabricante (HobbyKing, 2020) informa que se tiene una autonomía de aproximadamente 2 horas de uso continuo en altas condiciones.

Figura 3.11 Batería Turnigy 800mAh

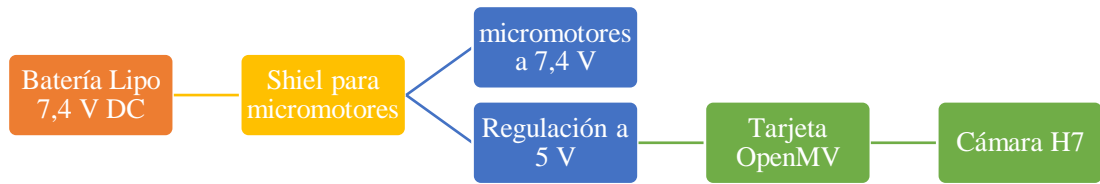


Turnigy 800mAh 40C 2S Lipo Pack, Fuente: (HobbyKing, 2020).

Debido al diseño electrónico de la tarjeta OpenMV y el shield para motores, solo se dispondrá de una conexión directa debido a que la regulación de voltaje es interna aprovechando espacio y rendimiento de la batería.

Para tener una idea general del funcionamiento de la alimentación del robot en la Figura 3.12 se observa el esquema de conexiones y la regulación de voltaje que se da a través de los dispositivos.

Figura 3.12 Esquema de alimentación



Esquema de alimentación del robot, Elaborador por: Clelio Endara y Emerson Maigua.

f) Arranque del robot

El módulo de arranque es una parte indispensable de un prototipo dentro de la robótica de competencia; en el diseño se utiliza dicho módulo de arranque de la compañía Ja-Bots, utilizando un control remoto el cual envía una señal al módulo receptor un estado en alto (1 lógico) o un estado en bajo (0 lógico) para que inicie el algoritmo de control de la trayectoria y de igual forma detenerlo de manera remota, en la Figura 3.13 se puede ver la conexión simple de este módulo hacia la tarjeta Open MV.

Figura 3.13 Conexión del módulo de arranque

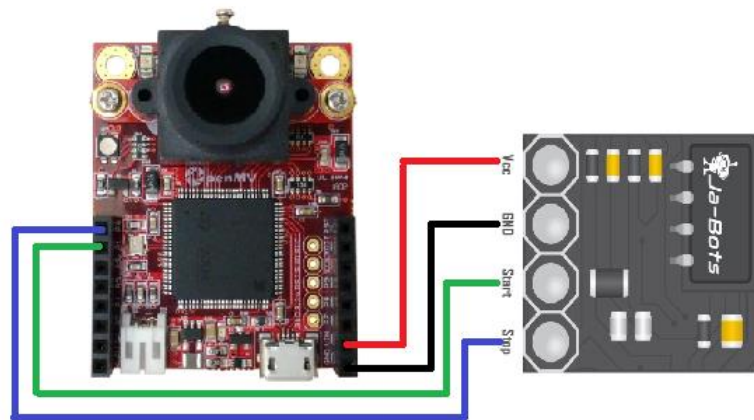


Diagrama de conexión entre el módulo de arranque y tarjeta open MV, Elaborado por: Clelio Endara y Emerson Maigua.

3.3 Diseño estructural del robot móvil seguidor de línea destreza

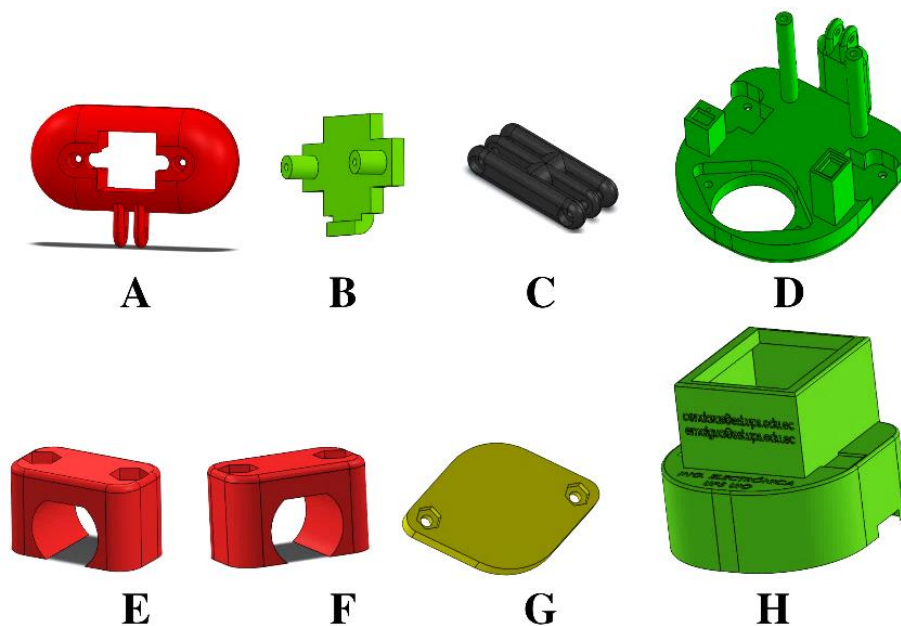
El diseño estructural se lo plantea a partir del análisis de factores importantes como son las dimensiones de cada uno de los elementos electrónicos, electromecánicos y móviles, además del diseño de la arquitectura del robot móvil de tipo diferencial, este análisis determina las posiciones en las cuales cada uno de los elementos que conforman el prototipo deben ser ubicados y diseñados dentro del chasis, por lo que abarca cinco etapas.

a) Modelado 3D del subconjunto de piezas

El modelado de cada una de las piezas contemplando las características técnicas y reglamentaciones requeridas para el robot móvil seguidor de línea destreza; mediante el software SolidWorks que es especializado en el diseño mecánico, se realizó el modelado 3D de cada una de las piezas que conforman el chasis del prototipo robótico en la que se instalará todos los elementos electrónicos (Tarjeta de control OpenMV, Cam H7, motor shield, lcd, módulo de arranque, switch, batería Lipo y cableado), elementos electromecánicos (micromotores) y elementos móviles (llantas, rueda loca), en el Anexo 2 se detalla el plano de subconjunto de cada una de las piezas.

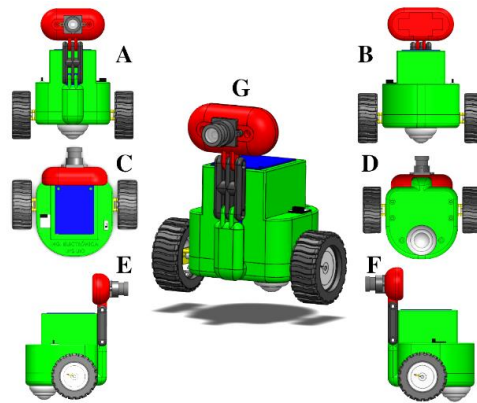
El diseño estructural del robot seguidor de línea destreza cuenta con 8 piezas como se observa en la Figura 3.14; en la Figura 3.15 se observa el ensamblaje de dichas piezas además de la ubicación de la cámara CAM H7 (OV7725) se tuvo en cuenta factores como la luminosidad, altura de la cámara respecto a la pista, el ángulo de enfoque y el tamaño de imagen. Se escogió un diseño articulable como se observa en la Figura 3.15, para que la cámara tenga libertad de movimiento en encontrar un ángulo de inclinación ideal que oscila en un rango de 0° a 90° grados con respecto a la unión del chasis con el brazo articulable y una altura entre 56.4 mm a 122.9 mm con respecto al piso.

Figura 3.14 Piezas del robot móvil seguidor de línea destreza



(A) Estuche de la cámara, (B) Soporte de cámara, (C) Tapa de la cámara, (D) Chasis, (E) Soporte micromotor 1, (F) Soporte micromotor 2, (G) Tapa de la rueda Loca, (H) Tapa del Chasis, Elaborado por: Clelio Endara y Emerson Maigua.

Figura 3.15 Vistas del prototipo ensamblado

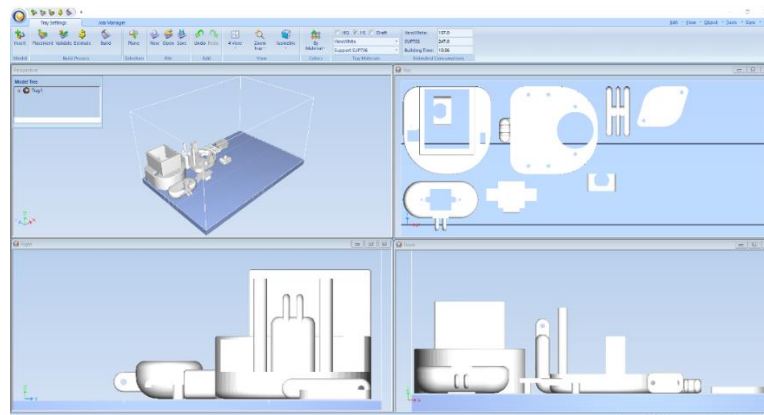


Robot móvil seguidor de línea de tipo diferencial diseñado en SolidWorks; (A) Vista frontal, (B) Vista trasera, (C) Vista superior, (D) Vista inferior, (E) Vista lateral izquierdo, (F) Vista lateral derecho, (G) Vista isométrica, Elaborado por: Clelio Endara y Emerson Maigua.

b) Laminado 3D del subconjunto de piezas

Cada una de las piezas se las exporta desde SolidWorks en formato de archivo STL (*.stl), estos archivos pasan por una etapa de Slicing o laminado que se importan en el software Objet Studio en el cual se configura todos los parámetros necesarios como la resistencia del soporte, la resolución de impresión en HS (capas de 0,28 micrones) y la selección de materiales (Vero White & Support SUP706) como se observa en la Figura 3.16, además Objet Studio tiene la ventaja de ordenar automáticamente las piezas dentro del área de trabajo para una eficiente impresión, para luego convertir estas configuraciones en un archivo de código máquina, el mismo permite que la impresora entienda las instrucciones e imprima el modelo 3D.

Figura 3.16 Slicing en Objet Studio



Configuración de parámetros de impresión 3D para las piezas que conforman el robot móvil seguidor de línea destreza en el software Objet Studio, Elaborado por: Clelio Endara y Emerson Maigua.

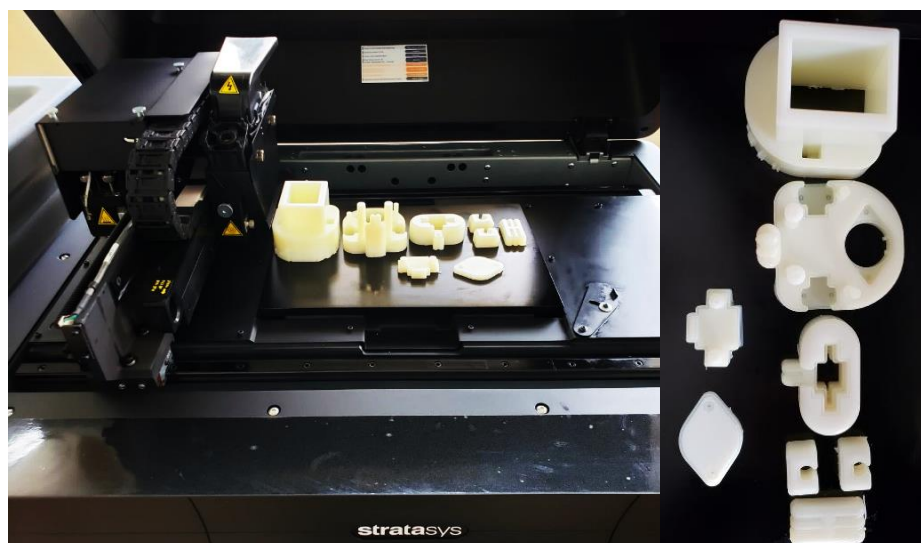
c) Impresión 3D del subconjunto de piezas

Para fabricar el prototipo se utilizó una impresora 3D Stratasys Objet 30 Pro que pertenece a los laboratorios de fabricación StartLabs de la Universidad Politécnica Salesiana sede Quito. StartLabs, es un espacio colaborativo que promueve el desarrollo de proyectos a través de la fabricación e implementación de prototipos rápidos con fines investigativos, innovación tecnológica y de emprendimiento (Salesiana, 2020).

La Stratasys Objet 30 Pro usa tecnología de impresión 3D SLA (estereolitografía) para la creación prototipos 3D de alta gama debido a su resolución de impresión con acabados suaves y precisos, utiliza dos tipos de materiales líquidos simultáneamente, el material modelo utilizado para la impresión de las piezas 3D usa tecnología PolyJet Vero el cual es un fopolímero líquido resistente que proporciona un excelente detalle visual y un material de soporte soluble SUP706; ambos materiales se combinan entre sí para crear la pieza 3D. Estos materiales que llegan al cabezal de impresión crean capas de 0,28 micrones que se configuraron en la etapa de laminado 3D y al mismo tiempo se secarán utilizando luz ultravioleta.

En la Figura 3.17 se observa las ocho piezas impresas dentro del área de impresión, el material modelo utilizado fue de 220 gramos y el material soporte 235 gramos, empleando un tiempo de 28 horas de impresión 3D con tecnología SLA.

Figura 3.17 Impresión 3D en Stratasys Objet 30 Pro

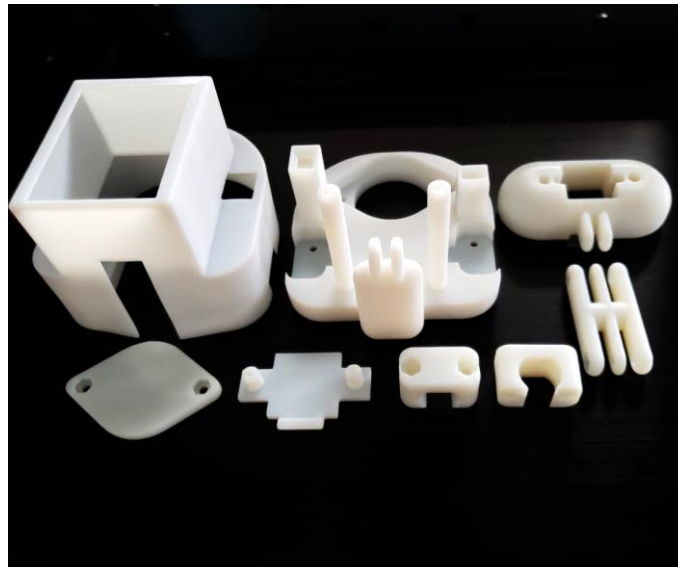


Impresión 3D con tecnología SLA del subconjunto de piezas que conforman el robot móvil seguidor de línea destreza, Elaborado por: Clelio Endara y Emerson Maigua.

d) Post Procesado del subconjunto de piezas

En esta etapa se procede a la eliminación del material soporte que se encuentra adherido a las paredes de cada una de las piezas impresas, dicho material tiene la ventaja de que es soluble en una solución de metasilicato de sodio con agua. Las piezas estarán inmersas en esta solución entre 6 a 8 horas para eliminar el material de soporte, por lo que se recomienda utilizar un cepillo con cerdas suaves para eliminar los restos del material soporte que aún pueden permanecer entre los espacios de las piezas; finalmente se enjuaga con agua limpia y se la deja secar las piezas al aire libre como se observa en la Figura 3.18.

Figura 3.18 Post Procesado de las piezas del robo móvil



Eliminación del material de soporte de las piezas impresas en 3D, Elaborado por: Clelio Endara y Emerson Maigua.

e) Ensamblaje del robot móvil seguidor de línea destreza

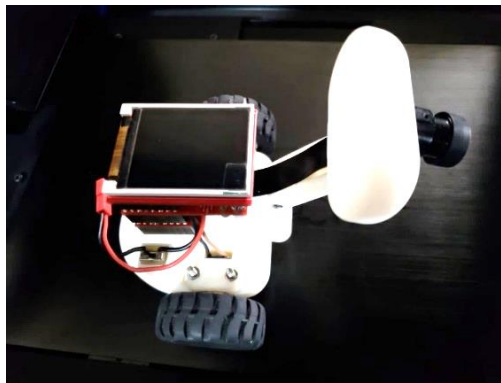
Para el ensamblaje del robot móvil seguidor de línea destreza se procede a clasificar cada una de las piezas que conforman el chasis, por lo que se va armando por secciones debido a que siguen un orden específico al momento de hacer encajar las piezas electrónicas, electromecánicas y móviles como se observa en el Anexo 3.

Antes de realizar el ensamblaje específicamente al módulo de cámara Cam H7 se debe realizar una limpieza al sensor utilizando una tela suave juntamente con alcohol isopropílico para luego ser montado en la pieza denominado estuche de cámara, se conectan un cable extensor a dicho módulo y se coloca la tapa de cámara; se ensambla

cada uno de los 2 micromotores haciéndole coincidir con el soporte del micromotor respectivamente para finalmente colocar las llantas.

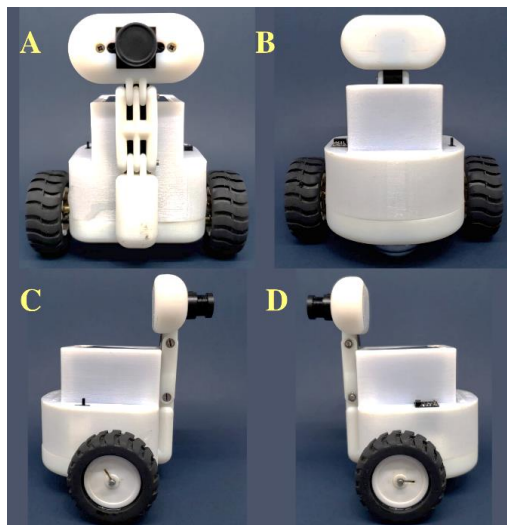
Los módulos de control OpenMV, el shield de motores y la lcd se conectan entre si haciendo encajar sus respectivos pines, para luego conectar al extensor del cable que se dirige al Cam H7, se coloca la rueda loca en el chasis juntamente con su tapa, para finalmente colocar todos los elementos dentro del chasis, cabe mencionar que se usa tornillo de 3 mm para ajustar cada una de las piezas. En la Figura 3.19 se observa en ensamble del prototipo final, así como también en la Figura 3.20 se aprecian las vistas del robot móvil seguidor de línea de destreza.

Figura 3.19 Prototipo final



Ensamblaje de la parte electrónica, electromecánica y móvil del robot seguidor de línea, Elaborado por: Clelio Endara y Emerson Maigua.

Figura 3.20 Vistas del prototipo final



Robot móvil seguidor de línea de tipo diferencial; (A) Vista frontal, (B) Vista trasera, (C) Vista lateral derecho, (D) Vista lateral izquierdo, Elaborado por: Clelio Endara y Emerson Maigua.

3.4 Red Neuronal para el robot móvil seguidor de línea destreza

Para el desarrollo e implementación de la red, dispone de tres bloques para su funcionamiento, En la Figura 3.21 de manera general se puede observar el funcionamiento de esta arquitectura de red, donde la entrada del sistema será una matriz imagen que pasará a través de una capa oculta para posteriormente obtener en nuestro caso cinco salidas.

Figura 3.21 Diagrama de bloques de la arquitectura de la red neuronal



Diagrama de bloques general de la arquitectura para la red neuronal, Elaborado por: Clelio Endara y Emerson Maigua.

a) Entrada del Sistema

En la entrada la red neuronal se tiene una matriz imagen previamente procesada por la tarjeta Open MV, los ejemplos de la imagen adquirida y procesada que pueden llegar a la entrada de la red neuronal se pueden ver en el Anexo 4.

b) Capa Oculta

Para la capa oculta se tiene la arquitectura MobileNet, en donde se realizan comparaciones profundas con los ejemplos previamente entrenados para obtener una respuesta a la salida.

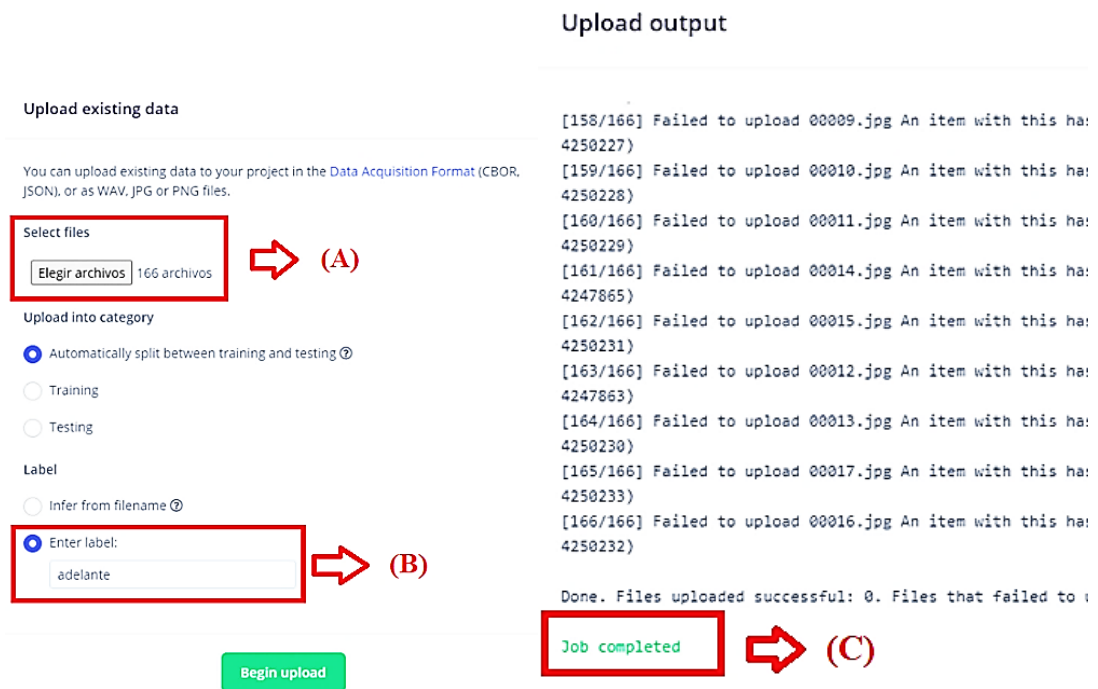
c) Salida del Sistema

La salida de la red neuronal tiene cinco salidas para sus casos de control, los cuales son: adelante, giro a la derecha, giro a la izquierda, giro a la derecha con radio y giro a la izquierda con radio. Esta salida es un vector de dos columnas una con la etiqueta del nombre de cada uno de los posibles casos de control y otra con el porcentaje de confianza de esa salida.

3.5 Entrenamiento para la red neuronal

Para el entrenamiento de la red neuronal, se utiliza una plataforma de desarrollo online, líder para el aprendizaje automático que es completamente gratuita para los desarrolladores de software, ingenieros y expertos en dominios para resolver problemas reales utilizando el aprendizaje automático en dispositivos como son Edge Impulse y Keras (EdgeImpulse, 2020), los cuales a través de su interfaz de subida de datos de los ejemplos previamente almacenados en un formato admisible para el software como se mostrará más adelante en el procesamiento y adquisición de imagen; son seleccionados como se muestra en la Figura 3.22 (A), donde se almacenan los datos de imagen de entrada con su respectiva salida mediante una etiqueta como se puede ver en la Figura 3.22 (B), posteriormente el software muestra un mensaje de completado el trabajo de subida mostrado en la Figura 3.22 (C).

Figura 3.22 Interfaz Edge Impulse



(A) Selección de los ejemplos para la red neuronal, (B) Asignación de nombre para la etiqueta de salida de los ejemplos correspondientes, (C) Mensaje de subida de datos completa, Elaborado por:

Clelio Endara y Emerson Maigua.

El número de ejemplos tomados es de forma experimental para el caso de esta red neuronal, se promedia generalmente 350 datos, pero no menos de 200 datos para un correcto funcionamiento.

a) Configuración y entrenamiento de la Red Neuronal

Para el entrenamiento y configuración de las características de la red neuronal, la interfaz de Edge Impulse permite configurar mediante el lenguaje de Keras directamente aplicando la arquitectura de red neuronal como se muestra en la Figura 3.23 (A).

Además de la configuración de parámetros mostrada en la Figura 3.23 (B), con el objetivo de buscar experimentalmente los parámetros ideales para su óptimo funcionamiento y rendimiento, manipulando variables como los ciclos de entrenamiento y el índice de confianza que tendrá la salida de la red neuronal, estas variables alteran de manera positiva la velocidad de procesamiento de la tarjeta de desarrollo utilizada.

Figura 3.23 Interfaz Edge Impulse configuración de la red neuronal

The image shows a screenshot of the Edge Impulse web interface. On the left, under the heading "Neural network architecture", there is a code editor with Python code for a Keras model. The code defines a MobileNetV2 model with specific weights, a learning rate of 0.0005, and an Adam optimizer. On the right, under the heading "Neural Network settings", there are several configuration options: "Number of training cycles" set to 15, "Learning rate" set to 0.0005, "Data augmentation" which is unchecked, and "Minimum confidence rating" set to 0.80. The code editor is labeled (A) and the settings panel is labeled (B).

```
1 import tensorflow as tf
2 from tensorflow.keras import Model
3 from tensorflow.keras.models import Sequential
4 from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D,
5   Flatten, Reshape, MaxPooling1D, BatchNormalization, Conv2D,
6   GlobalMaxPooling2D, Lambda
7 from tensorflow.keras.optimizers import Adam, Adadelta
8 from tensorflow.keras.losses import categorical_crossentropy
9 INPUT_SHAPE = (48, 48, 3)
10
11 base_model = tf.keras.applications.MobileNetV2(
12     input_shape=INPUT_SHAPE, alpha=0.05,
13     weights='./transfer-learning-weights/edgeimpulse/MobileNetV2.0_05
14     .96x96.color.bsiz64.lr_0_05.epoch_574.val_loss_4.22.hdf5',
15     include_top=True
16 )
17 base_model.trainable = False
18
19 model = Sequential()
20 model.add(InputLayer(input_shape=INPUT_SHAPE, name='x_input'))
21 model.add(Model(inputs=base_model.inputs, outputs=base_model.layers[-3]
22     .output))
23 model.add(Dense(12))
24 model.add(Dropout(0.1))
25 model.add(Flatten())
26 model.add(Dense(classes, activation='softmax'))
27
28 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
29     loss='categorical_crossentropy',
30     metrics=['accuracy'])
```

(A) Ingreso de la Arquitectura de la red neuronal, (B) Interfaz de configuración de parámetros de la red neuronal, Elaborado por: Clelio Endara y Emerson Maigua.

Una vez realizado las respectivas configuraciones se procede a entrenar la red neuronal, donde al completar exitosamente el programa, genera un aviso en su interfaz como se puede ver en la Figura 3.24.

Figura 3.24 Interfaz Edge Impulse de la salida del entrenamiento

Training output

```

17/17 - 1s - loss: 0.1434 - accuracy: 0.9643 - val_loss: 0.2717 - val_accuracy:
0.9173
Epoch 9/10
17/17 - 1s - loss: 0.1434 - accuracy: 0.9643 - val_loss: 0.2717 - val_accuracy:
0.9173
Epoch 10/10
17/17 - 1s - loss: 0.1497 - accuracy: 0.9586 - val_loss: 0.2716 - val_accuracy:
0.9173
Finished training

Saving best performing model...
Still saving model...
Still saving model...
Still saving model...
Still saving model...
Still saving model...
Still saving model...
Still saving model...
Still saving model...
Still saving model...
Converting TensorFlow Lite float32 model...
Converting TensorFlow Lite int8 quantized model with float32 input and output...
Converting TensorFlow Lite int8 quantized model with int8 input and output...
Calculating performance metrics...

Model training complete

Job completed
    
```

Interfaz de la salida de entrenamiento, mensaje de “Job completed (trabajo completado)” donde se indica que la red neuronal ya fue entrenada, Elaborado por: Clelio Endara y Emerson Maigua.

Terminado el entrenamiento de la red neuronal el software Edge Impulses genera dos archivos, un archivo de texto con las etiquetas de las salidas y la red neuronal entrenada en un comprimido de formato Tensor Flow Lite (TFLITE) mostrados en la Figura 3.25 compatible para la presencia y función en el algoritmo de control de la trayectoria del robot programado en el software OpenMV IDE.

Figura 3.25 Archivos generados en Edge Impulse

Name	Size	Packed	Type	Modified	CRC32
..			Carpeta de archivos		
labels.txt	56	40	Documento de tex...	14/11/2020 6:20	E2C3D43E
trained.tflite	243.776	120.173	Archivo TFLITE	14/11/2020 6:20	004D68C7

Archivos generados, etiquetas en archivo de texto y la red neuronal en formato Tensor Flow Lite, Elaborado por: Clelio Endara y Emerson Maigua.

b) Algoritmo del robot móvil

En la Figura 3.26 se describe el diagrama de flujo del algoritmo que ejecutará la tarjeta programada en el software libre OpenMV IDE, en el inicio de este se realiza la importación de librerías tanto para el funcionamiento como la comunicación con la cámara y el módulo para motores, además de la declaración de las variables necesarias para: el arranque del robot, funcionamiento de la red neuronal y el control de la locomoción del robot. En condiciones iniciales la cámara y motores están apagados a la espera de la inicialización del módulo de arranque donde una vez seteada la variable asignada para el control del arranque, empieza la inicialización de la cámara y con ello la captura de imagen.

Figura 3.26 Diagrama de flujo del algoritmo en la tarjeta OpenMV

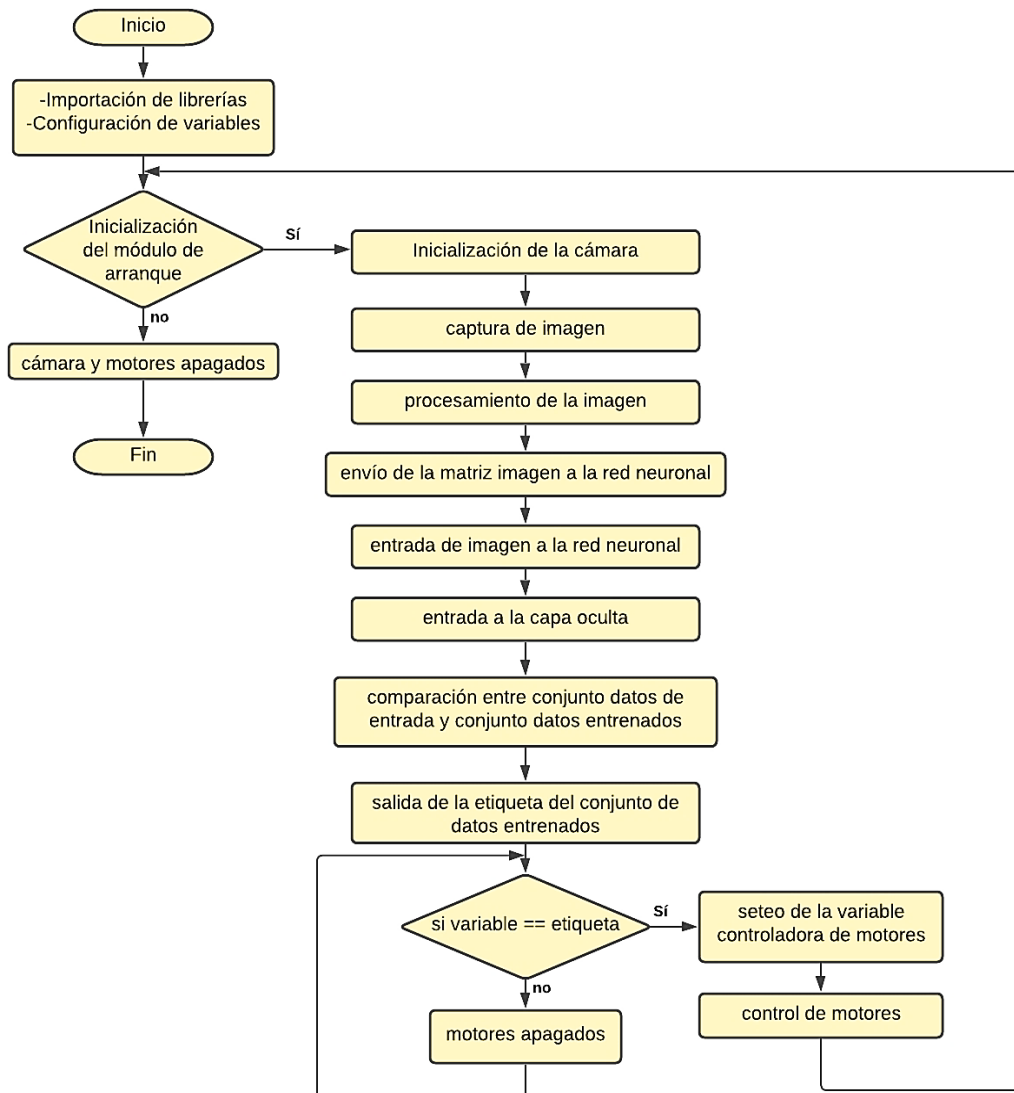


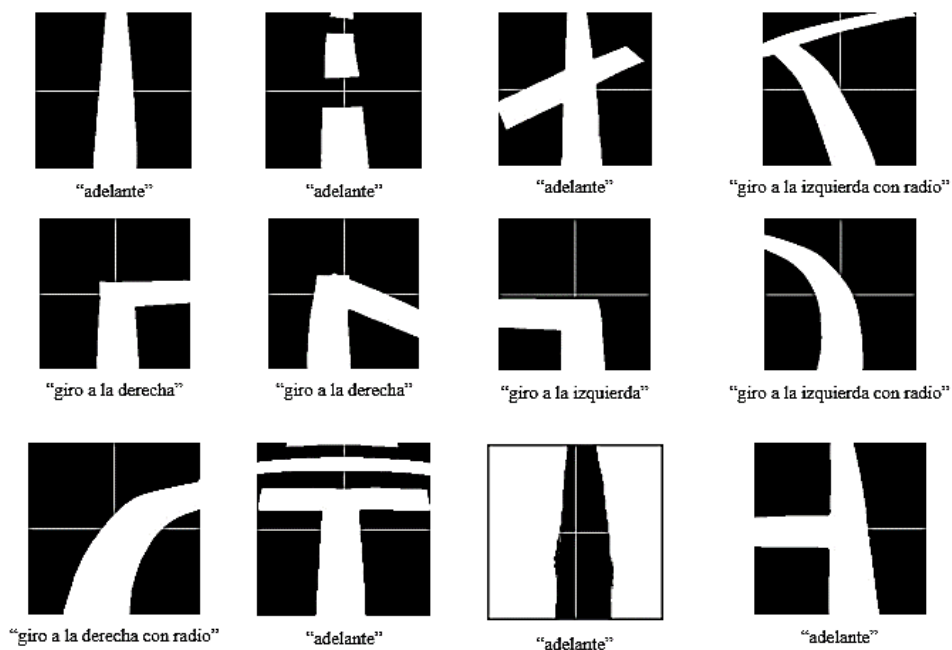
Diagrama de flujo del algoritmo implementado en la tarjeta de desarrollo OpenMV., Elaborado por:
Clelio Endara y Emerson Maigua.

Las imágenes pasan por un procesamiento donde se adecúa para ser enviada como una matriz imagen a la red neuronal en donde se tiene presente la capa oculta en el que empieza a realizar una comparación entre los datos de entrada y los datos entrenados para que después la red neuronal envíe una de las cinco salidas previamente asignadas en la arquitectura de la red neuronal que se detallan a continuación: adelante, giro sobre su propio eje a la derecha, giro sobre su propio eje a la izquierda, giro con radio a la izquierda y giro con radio a la derecha; con esta variable de salida ejercer el accionamiento y control para los motores dependiendo del caso brindado por la salida de la red neuronal en un número infinito de iteraciones hasta que el robot reciba la señal de paro o cero lógico por parte del módulo de arranque para detener los micromotores del robot móvil seguidor de línea destreza.

c) Adquisición de imagen

Para la adquisición de datos se utiliza el procesamiento explicado en el tema anterior con el cual se realizan capturas directamente en el software OpenMV IDE con su herramienta de guardar imagen; cada ejemplo se almacenará en una carpeta para al momento de su subida a la red neuronal se puedan etiquetar los ejemplos necesarios para el caso correspondiente, como se puede ver en la Figura 3.27 se organiza los ejemplos con la finalidad de ingresar las imágenes tomadas con su respectiva etiqueta.

Figura 3.27 Ejemplos de la adquisición de imagen



Imágenes adquiridas con su respectiva etiqueta, Elaborado por: Clelio Endara y Emerson Maigua.

d) Procesamiento de imagen

Se deben tomar en cuenta cuatro aspectos importantes que se requieren de manera obligatoria tanto la entrada como los ejemplos de entrenamiento de la red neuronal, los cuales se especifican a continuación:

- Dimensión de la imagen: la imagen de entrada a la red neuronal debe ser cuadrada, por tanto, debe tener un proceso de recorte, para este caso en específico se utilizará la dimensión de 240 x 240 píxeles, debido al que el software utilizado para el entrenamiento de la red neuronal obligatoriamente debe cumplir la dimensión antes mencionada, toda esta acción se resume en la línea de código que se muestra en la Figura 3.28 (A).
- Escala de color: debido a los colores generales de la pista para evitar el procesamiento de colores, se transforma a escala de grises (GRAYSCALE) las imágenes de entrada del sistema como se muestra en la Figura 3.28 (B).
- Formato de píxeles: Para complementar la escala de grises, a partir de una función (BINARY) que se muestra en la Figura 3.28 (C), donde se obtiene un mapa de bits en formato binario, en el cual cada píxel es un bit, y las imágenes de entrada a la red neuronal tiene la ventaja de adaptarse a distintas condiciones de luz, como también facilitar la toma de ejemplos para el entrenamiento de la red neuronal.
- Trazo de ejes de referencia: se dibujan dos ejes de referencia correspondientes al plano XY, mediante la función de dibujo de líneas (draw_line) en la imagen de entrada mostrada en la Figura 3.28 (D), especificando su posición en el centro de la imagen dibujando virtualmente cuatro cuadrantes iguales.

Figura 3.28 Funciones para el procesamiento de imagen de entrada

```
(A) sensor.set_windowing((240, 240))  
(B) sensor.set_pixformat(sensor.GRAYSCALE)  
(C) blobs = img.binary([(0,80)])  
(D) y=img.draw_line((120,0,120,240),color=(200,0,0))  
    x=img.draw_line((0,120,320,120),color=(200,0,0))
```

(A) Código del dimensionamiento de la imagen, (B) Código de función salida tipo formato de píxeles en escala de grises, (C) Código de conversión binaria de la imagen, (D) Código para dibujar líneas virtuales en la imagen, Elaborado por: Clelio Endara y Emerson Maigua.

3.6 Funcionamiento de la red neuronal

Para la programación en el OpenMV IDE se hace uso de sus funciones propias como son crear un bucle “*for*” para que tenga infinitas iteraciones como se aprecia en la Figura 3.29 (A) donde se indica que `tf.classify` es una función que relaciona la red neuronal con la imagen de entrada captada por la cámara haciendo que cada imagen captada entre en la red neuronal, donde se hace uso de los archivos generados en el entrenamiento de la red, en el que una variable llamada `predictions_list` tendrá el dato de la salida de la red neuronal como se ve en la Figura 3.29 (B), a continuación de esto se tendrá un “*for*” anidado el cual tiene la función de crear iteraciones infinitas para la asignación continua de la nueva variable de salida enviada por la red neuronal apreciado en la Figura 3.29 (C).

Figura 3.29 Programación de la red neuronal en Open MV IDE

```
(A) for obj in tf.classify(net, img):  
  
(B)     predictions_list = list(zip(labels, obj.output()))  
  
(C)     for i in range(len(predictions_list)):
```

(A) Función de relación entre imagen de entrada y red neuronal, (B) Almacenaje de la variable de salida de la red neuronal, (C) Iteraciones para actualizar el valor de la salida almacenada, Elaborado por: Clelio Endara y Emerson Maigua.

3.7 Control del shield para motores

En la parte de control del shield para motores, se parte con el uso de la variable de salida que otorga la red neuronal la cual entra en una simple sentencia *if* para cuando llegue la salida se lo almacena a una variable la cual tendrá el valor del porcentaje de confiabilidad de esa salida como se puede ver en la Figura 3.30 (A), con este valor almacenado se procede a realizar el control de los motores donde con otro *if* se compara para saber si el valor supera el 80% de confiabilidad obtenido de la red neuronal, debido a que experimentalmente al bajar el rango del valor del porcentaje se llegó a este valor promedio para ejecutar de manera rápida el control hacia los motores.

Con una función `motor.set_speed` se controla el sentido de los motores y para el PWM se tiene un valor mínimo para moverse al cual se multiplica por una variable global *velocidad* de forma experimental hallar la velocidad más alta con la que pueda

movilizarse correctamente el robot móvil, esto se puede apreciar en las líneas de código de la Figura 3.30 (B) donde se aprecia el control de ambos motores.

Figura 3.30 Programación para el control del shield para motores

```
(A) if(predictions_list[i][0] == 'adelante'):  
    a = predictions_list[i][1]  
  
(B) if (a > 0.8):  
  
    m1.set_speed(20 * velocidad) # Motor 1  
    m2.set_speed(-20 * velocidad) # Motor 2
```

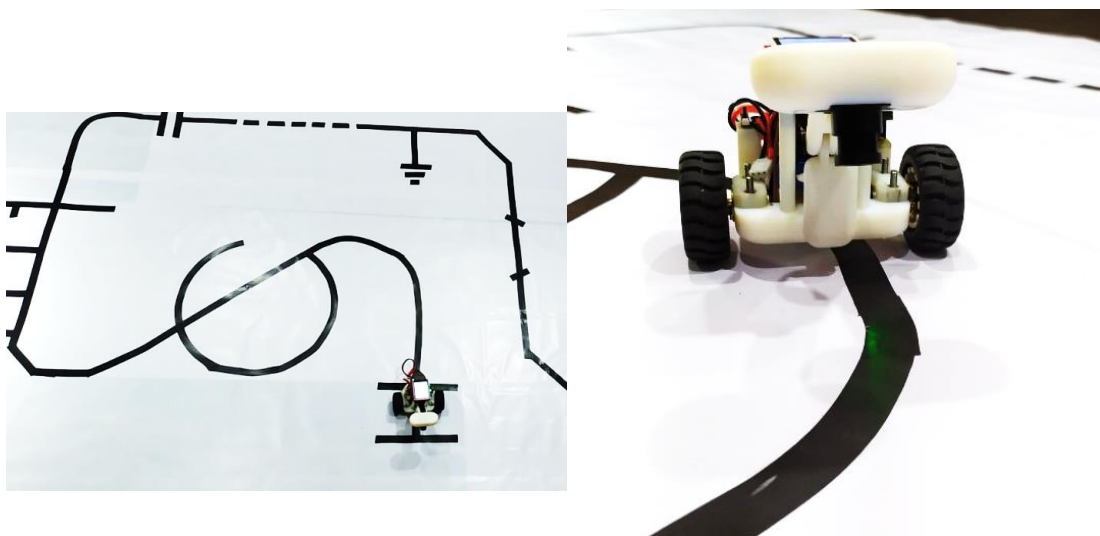
(A) Sentencia *if* para la verificación de etiquetas, (B) Sentencia de activación motor 1 y motor 2,

Elaborado por: Clelio Endara y Emerson Maigua.

3.8 Resultado final del robot móvil seguidor de línea destreza

Una vez realizado el estudio de los parámetros y reglamentaciones que debe tener el prototipo se fue desarrollando su hardware y software, además se diseñó una pista de competencia con las bifurcaciones más comunes que se pueden encontrar en los concursos de robótica; el cual culminó teniendo como resultado el prototipo mostrado en la Figura 3.31 y en el Anexo 3 se detalla el ensamblable; con este robot móvil seguidor de línea destreza se realizó todas las posteriores pruebas mostradas en el siguiente capítulo.

Figura 3.31 Prototipo Implementado



Prototipo implementado en pista de competencia con un escenario controlado, Elaborado por: Clelio

Endara y Emerson Maigua.

3.9 Resumen de costos

Una vez realizado todo el dimensionamiento tanto en hardware y software, se detalla en la Tabla 3.3 todos los costos requeridos para la fabricación e implementación del robot móvil seguidor de línea destreza.

Tabla 3.3 Resumen de Costos del Proyecto

TIPO	ITEM	DESCRIPCIÓN	CANT.	P. U	P.T
E Q U I P O S	Software OpenMV	OpenMV IDE V 2.4.0	1	0.0	0.0
	Laptop	Laptop hp Win 10	1	0.0	0.0
	Cargador Batería	Cargador batería lipo Imax B6 80W	1	45.00	45.00
	Pista	Pistas de prueba	1	30.00	30.00
M A T E R I A L E S	OpenMV Cam H7	OMV-Cam H7 V1	1	65.00	65.00
	Motor Shield	OMV-Shield Motor	1	25.00	20.00
	Camera Module Extensión Cable	OMV-Cab fpc V1	2	10.00	20.00
	LCD Shield	OMV-Shield Lcd V1	1	25.00	20.00
	Robot Start Module	Módulo receptor	1	10.00	10.00
	Control Remoto	Módulo emisor on / off	1	30.00	30.00
	Batería Turnigy	Batería 7.4V 800 mAh	1	22.00	22.00
	Pololu MicroMotor	6V DC Motor 50:1 HP	2	15.00	30.00
	Llantas Pololu	Llanta 42 x 19 mm tipo D	2	4.50	9.00
	Rueda Loca	Rueda loca metálica	1	3.00	3.00
	Chasis Robot	Impresión del Chasis	1	241.00	241.00
Elementos Varios	Electrónicos, tornillos	20	0.25	5.00	
H E R R A M I E N T A S	Pinzas	Pinzas puntas redondas	1	7.00	7.00
	Destornillador plano	Para tornillos de 3 mm y 5 mm	2	4.00	8.00
	Destornillador estrella	Para tornillos de 3 mm y 5 mm	2	4.00	8.00
	Cautín	Cautín de 60W	1	12.00	12.00
	Estaño	Para Soldar	1	5.00	5.00
	Pasta	Para soldar	1	3.00	3.00
	Pegamento	Loctité	1	3.00	3.00
	Alcohol Isopropílico	500 ml.	1	15.00	15.00
Otros	Ingeniería	Hora de trabajo de ingeniería	2	460.00	920.00
	Importación de tarjeta de desarrollo y shields	Envío internacional, impuesto SENAE, tramite nacionalización DHL	1	129.00	129.00
VALOR TOTAL SOLO HARDWARE Y SOFTWARE					611,00\$
VALOR TOTAL DE TRABAJO DE INGENIERÍA E IMPORTACIÓN					1049,00\$
VALOR TOTAL DE TRABAJO					1660,00\$

Tabla de costos detallada para la fabricación e implementación del robot móvil seguidor de línea destreza, Elaborado por: Clelio Endara y Emerson Maigua.

CAPÍTULO 4

PRUEBAS – RESULTADOS

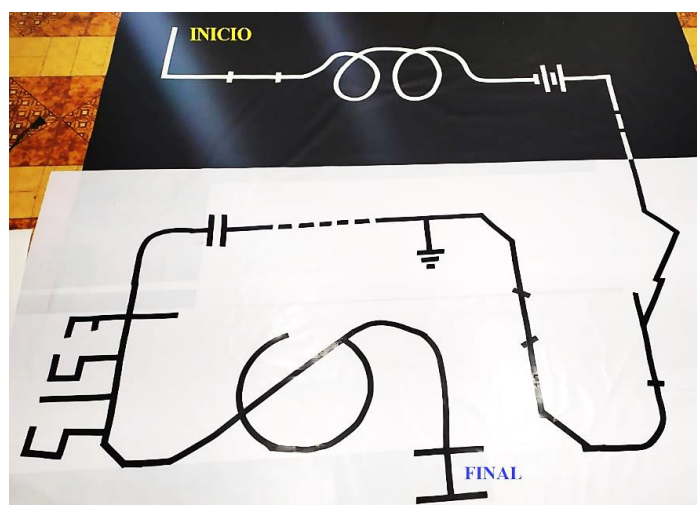
Para realizar la validación del sistema, tanto en su parte de hardware como software se realizaron pruebas en un escenario controlado, con el objetivo de encontrar experimentalmente fallas o mejoras durante el desarrollo de estas, analizando su comportamiento en la pista mediante; mediciones de tiempo, acompañados del porcentaje de avance de pista realizado, para calcular la efectividad del sistema implementado.

4.1 Pruebas en pista con un escenario controlado

Para la comprobación de la efectividad del robot móvil seguidor de línea destreza se creó un escenario controlado adecuado a las condiciones y características reales que tendrá una pista en un concurso de robótica real como se observan en la Figura 4.1 que es una pista de nivel de dificultad alta, con dimensiones aproximadas de dos metros por dos metros, donde se tomarán los datos de 10 repeticiones registradas en sus respectivas tablas.

Teniendo en cuenta que el robot tendrá una velocidad general que está presente en todas las pruebas como de igual forma el ángulo de posición de la cámara que de manera experimental se obtuvo tanto la altura de 8 cm de alto, como la inclinación de la cámara aproximadamente 60° .

Figura 4.1 Pistas en escenario controlado



Pista de prueba en un escenario controlado para el robot móvil seguidor de línea destreza, Elaborado por: Clelio Endara y Emerson Maigua.

A través de la experimentación, los resultados obtenidos se visualizan en las siguientes tablas de las pruebas en sentido horario de la pista en función del número de neuronas en la capa oculta.

Tabla 4.1 Resultados en la pista, red neuronal con 5 neuronas en la capa oculta

CAPA OCULTA CON 5 NEURONAS			
Número de intento	Registro del cronómetro (s)	Trayectoria fallida	Porcentaje de pista recorrida
1	65,06 s	X	100 %
2	64,49 s	X	100 %
3	65,14 s	X	100 %
4	64,97 s	X	100 %
5	65,17 s	X	100 %
6	65,01 s	X	100 %
7	64,55 s	X	100 %
8	64,84 s	X	100 %
9	64,73 s	X	100 %
10	65,01 s	X	100 %
Promedio total (s)	64,897 s		

Resultados de la red neuronal con 5 neuronas, Elaborado por: Clelio Endara y Emerson Maigua.

De los resultados obtenidos en Tabla 4.1, se pudo determinar que el robot no tiene pérdida de trayectoria alguna con la red neuronal de 5 neuronas en la capa oculta, el cual brinda un porcentaje de pista recorrida del 100%, y un promedio de tiempo de 64,897 s.

Tabla 4.2 Resultados en la pista, red neuronal con 10 neuronas en la capa oculta

CAPA OCULTA CON 10 NEURONAS			
Número de intento	Registro del cronómetro (s)	Trayectoria fallida	Porcentaje de pista recorrida
1	59,47 s	X	100 %
2	59,21 s	X	100 %
3	58,82 s	X	100 %
4	58,95 s	X	100 %
5	59,02 s	X	100 %
6	59,11 s	X	100 %
7	58,91 s	X	100 %
8	59,07 s	X	100 %
9	58,89 s	X	100 %
10	58,97 s	X	100 %
Promedio total (s)	59,042 s		

Resultados de la red neuronal con 10 neuronas, Elaborado por: Clelio Endara y Emerson Maigua.

De la Tabla 4.2 se muestra un porcentaje de pista recorrida del 100% de la red neuronal con 10 neuronas y un promedio de tiempo de 59,042 s.

Tabla 4.3 Resultados en la pista, red neuronal con 12 neuronas en la capa oculta

CAPA OCULTA CON 12 NEURONAS			
Número de intento	Registro del cronómetro (s)	Trayectoria fallida	Porcentaje de pista recorrida
1	54,89 s	X	100 %
2	54,57 s	X	100 %
3	54,67 s	X	100 %
4	55,02 s	X	100 %
5	54,68 s	X	100 %
6	54,96 s	X	100 %
7	55,02 s	X	100 %
8	54,88 s	X	100 %
9	54,89 s	X	100 %
10	54,93 s	X	100 %
Promedio total (s)	54,851 s		

Resultados de la red neuronal con 12 neuronas, Elaborado por: Clelio Endara y Emerson Maigua.

En la Tabla 4.3 muestra que la red neuronal con 12 neuronas tiene un porcentaje de pista recorrida del 100% y un promedio de tiempo de 54,851 s.

Tabla 4.4 Resultados en la pista, red neuronal con 15 neuronas en la capa oculta

CAPA OCULTA CON 15 NEURONAS			
Número de intento	Registro del cronómetro (s)	Trayectoria fallida	Porcentaje de pista recorrida
1	55,45 s	X	100 %
2	55,78 s	X	100 %
3	55,90 s	X	100 %
4	55,64 s	X	100 %
5	55,73 s	X	100 %
6	55,81 s	X	100 %
7	55,92 s	X	100 %
8	55,89 s	X	100 %
9	55,77 s	X	100 %
10	55,59 s	X	100 %
Promedio total (s)	55,748 s		

Resultados de la red neuronal con 15 neuronas, Elaborado por: Clelio Endara y Emerson Maigua.

De los resultados obtenidos en Tabla 4.4, se tiene un porcentaje de pista recorrida del 100% de la red neuronal con 15 neuronas y un promedio de tiempo de 55,748 s.

Tabla 4.5 Resultados en la pista, red neuronal con 20 neuronas en la capa oculta

CAPA OCULTA CON 20 NEURONAS			
Número de intento	Registro del cronómetro (s)	Trayectoria fallida	Porcentaje de pista recorrida
1	57,03 s	X	100 %
2	56,58 s	X	100 %
3	56,43 s	X	100 %
4	29,27 s	✓	40 %
5	57,11 s	X	100 %
6	56,93 s	X	100 %
7	57,01 s	X	100 %
8	56,95 s	X	100 %
9	29,53 s	✓	40 %
10	56,89 s	X	100 %
Promedio total con pérdida de pista (s)	50,753 s		Promedio: 88%
Promedio total sin pérdida de pista (s)	56,854 s		

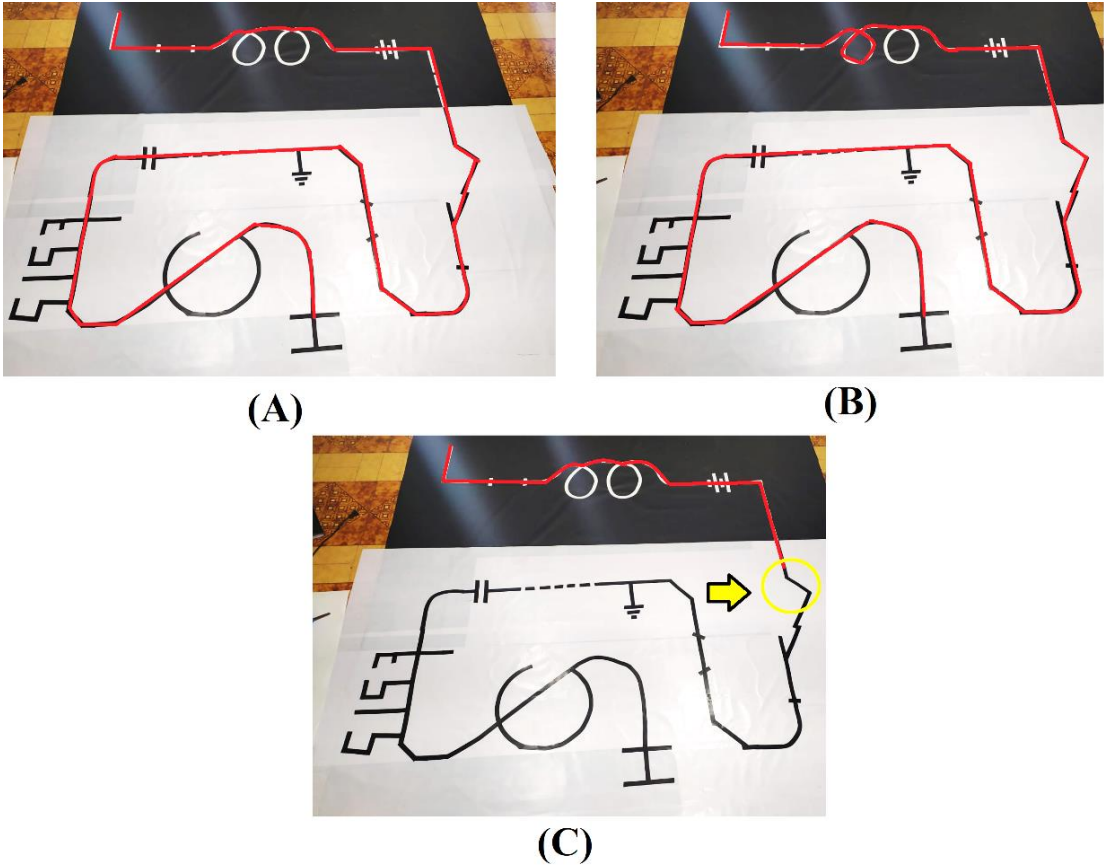
Resultados de la red neuronal con 20 neuronas, Elaborado por: Clelio Endara y Emerson Maigua.

La Tabla 4.5 muestra que la red neuronal con 20 neuronas tiene un porcentaje de pista recorrida del 88% y un promedio de tiempo de 56,854 s.

Se puede observar que al trabajar con 12 neuronas el comportamiento del prototipo es mucho mejor, y con 20 neuronas en la capa oculta tiene ciertas pérdida de trayectoria debido a que el tamaño de memoria RAM de buffer de cuadros que ocupa la red neuronal ocasionando una ralentización de procesamiento en especial al momento de invertir los colores de pista, esto se pudo visualizar en la tendencia donde se perdió la trayectoria del robot como se pudo observar en la Tabla 4.5, la pérdida de trayectoria se da aproximadamente al 40% del recorrido como se muestra en la Figura 4.2 (C).

Además, con todas las pruebas se comprobó la trayectoria de tendencia del robot, donde la tendencia de la Figura 4.2 (A) tiene mayor probabilidad de cumplirse, seguido de la tendencia mostrada en la Figura 4.2 (B).

Figura 4.2 Pista con líneas de tendencia



Detalle de pista de prueba con líneas de tendencia, Elaborado por: Clelio Endara y Emerson Maigua.

CONCLUSIONES

Se logró implementar un algoritmo de trayectoria para un robot seguidor de línea destreza de competencia mediante la selección de la tarjeta controladora OpenMV con uso de visión artificial acompañado de redes neuronales, que a través del software OpenMV IDE se pudo programar una red neuronal con la cual el robot móvil presenta una buena eficiencia en la aplicación de la misma, software que no necesita pago de licencias, por lo que es una gran ventaja frente a otros y permite que el robot diseñado con una arquitectura diferencial pueda seguir una trayectoria en línea blanca con superficie negra o línea negra con superficie blanca; el sistema se validó con métodos experimentales a partir de pruebas de campo en un escenario controlado con un diseño de una pista de alta complejidad, obteniendo como resultados el 100% de efectividad, pudiendo validar que la red neuronal con visión artificial controlan la trayectoria del robot móvil seguidor de línea destreza satisfactoriamente.

Se identificó a través del estado del arte que los sistemas de navegación para robots móviles mediante algoritmos inteligentes aplicados a la visión artificial, obtuvieron los mejores resultados con la arquitectura MobileNet, además de un buen procesamiento con un formato de píxeles en negativo con los cuales permite al robot controlar su trayectoria con mayor rapidez y fluidez reduciendo tiempo de procesamiento y espacio de memoria, dado que esta arquitectura reduce el espacio de memoria RAM de búfer de cuadros requerida debido a que la tarjeta de desarrollo OpenMV tiene aproximadamente 31 MB de RAM de búfer de cuadros. Dicho esto, ejecutar un modelo en cualquier lugar cercano a ese tamaño será extremadamente lento.

Con las pruebas en un escenario controlado se validó el funcionamiento del sistema, mismas que permitieron de manera experimental contemplar fallos y mejoras a nivel de hardware en la fabricación del subconjunto de piezas que conforman el chasis del robot móvil seguidor de línea destreza. En un principio se utilizó una impresora 3D con tecnología FDM, que no fue la mejor opción debido a que al momento del ensamblaje del chasis presentó fisuras en ciertas piezas afectando la locomoción del prototipo, motivo por el cual se optó utilizar una impresora 3D con tecnología SLA que soluciona estos problemas, generando piezas con alta precisión, superficies más lisas y alta resistencia mecánica. De la misma manera con respecto al software se tuvo problemas en la memoria de procesamiento que está directamente relacionado con la

red neuronal por su número de neuronas presente en la capa oculta, error que se rectificó mediante el uso de la arquitectura MobileNet en su segunda versión; de manera que al subsanar estos problemas se logra una estabilidad y confiabilidad alta, donde las pruebas mostraron un rango de porcentaje del 100% de efectividad.

En función de las 10 pruebas realizadas de la arquitectura de red neuronal con 12 neuronas en la capa oculta, el prototipo presenta su punto de mayor efectividad, donde los resultados arrojaron menores tiempos con un promedio de 54,851 segundos con el cumplimiento del 100% de la trayectoria, de manera opuesta al elevar el número de neuronas en la capa oculta a 20 se tuvo un incremento en el valor del tiempo con un promedio de 56,854 segundos y pérdidas de trayectoria en dos ocasiones, se justifica este resultado porque la red se torna más grande con mayor número de interconexiones en la arquitectura que produce alto consumo de memoria RAM generando un retraso en el enfoque de la cámara especialmente en el cambio de colores de la superficie, esto conlleva a generar una adquisición de imagen errónea por dicho retraso dando un control equivocado y como resultado la pérdida de trayectoria del prototipo.

RECOMENDACIONES

Con el fin de garantizar un entrenamiento eficaz, en la toma de ejemplos para la red neuronal, se debe contemplar la modalidad del reglamento dado es el caso ante bifurcaciones el reglamento obliga una decisión de trayectoria al robot, con esto tener un algoritmo que evite problemas de conflicto en su salida, también el tamaño de imagen el software Edge Impulse trabaja con imágenes cuadradas por tanto se debe recortar las imágenes para el ingreso de los datos, caso contrario la red neuronal mostrará errores al momento de compilar el programa.

Para garantizar una correcta adquisición y procesamiento de imagen en la cámara CAM H7 (OV7725), se debe tener en cuenta para nuestro caso el tamaño del archivo con formato *.tflite* debido a que el mismo ocupa cierta cantidad de memoria volátil afectando en el enfoque de la cámara, mismo que puede provocar lecturas erróneas.

La red neuronal debe ser entrenada en un proceso off line con un número mínimo de 200 ejemplos, dicho valor se obtuvo de manera experimental para garantizar un buen funcionamiento, caso contrario el robot móvil seguidor de línea destreza tiende a tomar decisiones erróneas afectando su trayectoria. Una opción para solventar los problemas a nivel de software que se presentó en este trabajo de titulación, se recomienda el uso de una tarjeta de desarrollo con mejores características, como por ejemplo la tarjeta de desarrollo Jetson Xavier NX la cual está diseñada específicamente para aplicaciones de inteligencia artificial hacia la creación de proyectos innovadores.

REFERENCIAS

- Aguilar Campos, J. P., & Sandoval Suquillo, H. I. (2018). *Desarrollo de un robot móvil diferencial controlado mediante un algoritmo de búsqueda con redes neuronales*. Quito: Universidad Politécnica Salesiana - Carrera de Ingeniería Electrónica.
- Agyeman, K., & Abdelkader, I. (02 de Noviembre de 2016). *OpenMV*. Obtenido de OpenMV: <https://openmv.io/collections/shields/products/proto-shield>
- Agyeman, K., & Abdelkader, I. (20 de Agosto de 2018). *OpenMV*. Obtenido de OpenMV: <https://cdn.shopify.com/s/files/1/0803/9211/files/OpenMV-H7.pdf?10726770933929369108>
- Agyeman, K., & Abdelkader, I. (20 de Agosto de 2018). *OpenMV*. Obtenido de OpenMV: https://cdn.shopify.com/s/files/1/0803/9211/files/OpenMV_Motor_Shield.pdf?10726770933929369108
- Aldaz Andrade, L. A., & Orellana Torres, R. D. (2017). *Desarrollo de un prototipo de robot móvil de competencia multi-categoría*. Quito: Universidad Politécnica Salesiana - Carrera de Ingeniería Electrónica.
- Arrow Development Tools. (21 de Mayo de 2015). *arrow.com*. Obtenido de arrow.com: https://static6.arrow.com/aropdfconversion/efb389d1f9c390cf04346a67dbcd75f31d01f43/deca_user_manual.pdf
- Barahona Guamani, E. S. (2019). *Navegación autónoma basada en maniobras bajo estimación de posturas humanas para un robot omnidireccional Kuka YouBot*. Ambato: Universidad Técnica de Ambato, Ingeniería en Sistemas, Electrónica e Industrial.
- Baturone, A. O. (2001). *Robótica: Manipuladores y Robots Móviles*. Barcelona: Marcombo BOIXAREU.
- Benavent Pla, A. (2017). *Vehículo autónomo con visión artificial utilizando OpenCV*. Valencia: Universitat Politècnica de València. Departamento de Ingeniería Electrónica - Departament d'Enginyeria Electrònica.
- Buendia Ríos, A. A. (2017). *Navegación Autónoma de un vehículo Pequeño en Interiores Empleando Visión Artificial y Diferentes Sensores*. Texcoco: Universidad Autónoma del Estado de México.

- Caballero Barriga, E. R. (2017). *Aplicación Práctica de la Visión Artificial para el Reconocimiento de Rostros en una Imagen, Utilizando Redes Neuronales y Algoritmos de Reconocimiento de Objetos de la Biblioteca OPENCV*. Bogotá: Universidad Distrital Francisco José de Caldas - Facultad de Ingeniería.
- Castañeda Bermúdez, J. S., & Salguero Lozano, Y. A. (2017). *Ajuste de Algoritmo de Identificación Visual para su Uso en Aplicaciones de Navegación Autónoma de Robots*. Bogotá: Universidad Distrital Francisco José de Caldas - Facultad de tecnología Politécnica.
- CEDIA. (28 de Noviembre de 2019). *CER 2019*. Recuperado el 14 de Enero de 2021, de Reglamento Seguidor De Linea Destreza: <https://cedia.edu.ec/dmdocuments/CER/2019/Reglamentos/ReglamentoSeguidorDeLineaDestreza.pdf>
- Cortéz, F. R. (2011). *Robótica Control de Robots Manipuladores*. México: Alfaomega.
- Díaz Saravia, M. W., & Trejo Peraza, J. M. (2019). *Diseño de Prototipo de Vehículo Autónomo Utilizando Redes Neuronales*. La Libertad: Escuela Especializada en Ingeniería ITCA-FEPADE.
- EdgeImpulse. (6 de Diciembre de 2020). *Edge Impulse*. Obtenido de EdgeImpulse Inc.: <https://www.edgeimpulse.com/>
- Fingertech Robotics. (Abril de 2013). *fingertechrobotics.com*. Obtenido de fingertechrobotics.com: https://www.fingertechrobotics.com/prodimages/drivemotors/ss_torque_kg-cm_table.jpg
- García Cedeño, M. G. (2013). *Desarrollo de un Kit Educativo de Robótica Basado en Arduino (Robot de Destreza)*. Guayaquil: Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Networking y Telecomunicaciones. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/11472>
- González Linch, A. J. (2013). *Aplicaciones prácticas de microcontroladores a través de la plataforma de programación Matlab*. Guayaquil: Carrera de Ingeniería en Telecomunicaciones, Universidad Católica de Santiago de Guayaquil.
- Guerrero Aguirre, Á., & Ramos Giraldo, P. J. (2014). *Sistema embebido de bajo costo para visión artificial*. Pereira: Scientia et Technica Año XIX.

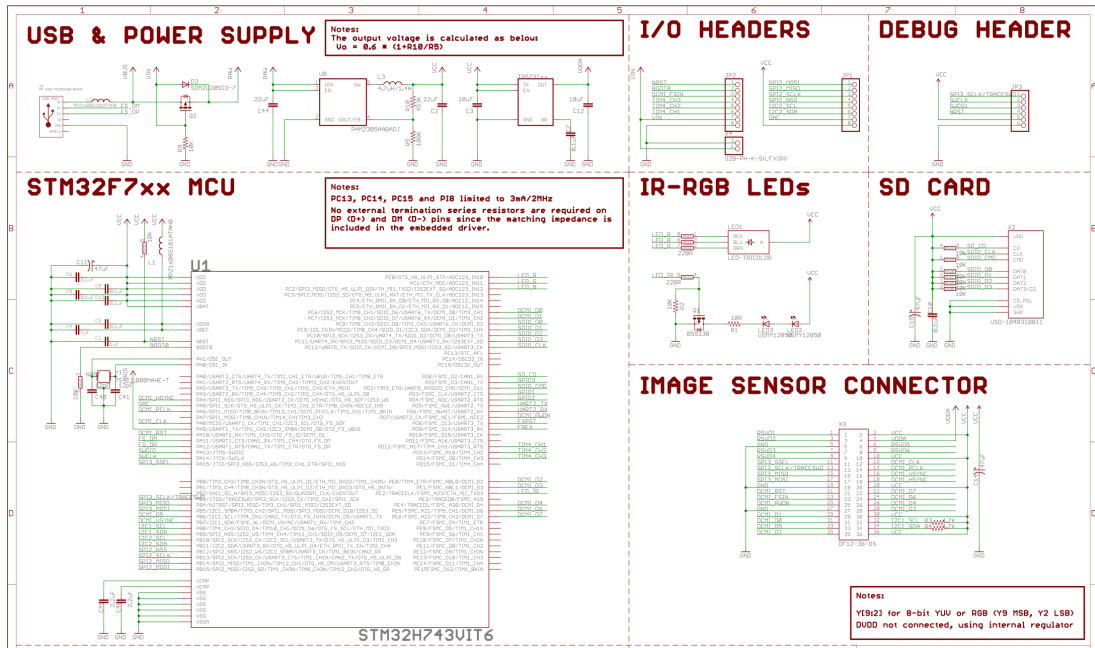
- Hayes, P. (1974). *Some Problems and Non-Problems in Representation Theory AISB*. Córdoba: Sussex.
- Hernández, A., Rayo, H., & Torres, J. (2011). *Prototipo de Montacargas Automático*. Bogotá: Universidad de San Buenaventura. Obtenido de http://45.5.172.45/bitstream/10819/1294/1/Prototipo_monta_cargas_automatizado_Rayo_2011.pdf
- HobbyKing. (Febrero de 2020). *hobbyking.com*. Obtenido de hobbyking.com: https://hobbyking.com/es_es/turnigy-800mah-2s-20c-lipo-pack-parkzone-compatible-pkz1032.html?__store=es_es
- Hossian, A., Cejas, L., Carbajal, R., Echeverría, C., Alveal, M., & Olivera, V. (2018). *Obtención de rutas de navegación óptimas de robot móvil mediante la aplicación de Redes Neuronales Artificiales (RNA) y Algoritmos de Búsqueda Local (ABL) en entornos estructurados*. Tucumán: Grupo de Investigación de Robótica - Universidad Tecnológica Nacional – Facultad Regional del Neuquén.
- Malásek, J. (11 de Diciembre de 2013). *Pololu Robotics & Electronics*. Obtenido de Pololu Robotics & Electronics: <https://www.pololu.com/blog/247/powerful-new-u3v50x-boost-regulators>
- Matich, D. J. (2001). *Redes Neuronales: Conceptos básicos y aplicaciones*. Rosario: Universidad Tecnológica Nacional.
- Maxon Group. (Septiembre de 2020). *maxongroup.es*. Obtenido de maxongroup.es: https://www.maxongroup.es/medias/sys_master/root/8846330724382/20-ES-109.pdf
- Nordic Semiconductor. (Agosto de 2017). *nordicsemi.com*. Obtenido de nordicsemi.com: <https://www.nordicsemi.com/-/media/Software-and-other-downloads/Product-Briefs/nRF52-DK-product-brief.pdf?la=en&hash=01C23BC9A39B642EF150702065B7673E430EBE05>
- OpenMV. (Agosto de 2018). *openmv.io*. Obtenido de openmv.io: <https://cdn.shopify.com/s/files/1/0803/9211/files/OpenMV-H7.pdf?10726770933929369108>
- Peña Merino, J. M. (2017). *Sistema de detección y conteo de vehículos utilizando visión artificial*. Piura: Universidad Nacional de Piura.

- Pololu. (Diciembre de 2019). *Pololu Robotics & Electronics*. Obtenido de Pololu Robotics & Electronics: <https://www.pololu.com/file/0J1487/pololu-micro-metal-gearmotors-rev-4-2.pdf>
- Pololu Robotics & Electronics. (Abril de 2018). *pololu.com*. Obtenido de pololu.com: <https://www.pololu.com/product/1090>
- Ramírez Díaz, J. A. (2019). *Aprendizaje del Comportamiento Humano en el Espacio Articular del Robot Utilizando Redes Neuronales*. Ciudad de México: Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Unidad Zacatenco, Departamento de Control Automático.
- Raspberry Pi. (Abril de 2015). *raspberrypi.org*. Obtenido de raspberrypi.org: <https://static.raspberrypi.org/files/product-briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf>
- Robot Games ZL7. (1 de Junio de 2019). *robotgameszerolatitud*. Obtenido de robotgameszerolatitud: <https://robotgameszerolatitud.jimdofree.com/>
- Rouhiainen, L. (2018). *Inteligencia artificial*. Madrid: Alienta.
- Sadler, M., Howard, A., Zhu, M., Zhmogimov, A., & Chen, L.-C. (2019). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Estados Unidos: Google.Inc.
- Salazar Logroño, F. W., & Pico Aponte, G. M. (2019). *Sistema avanzado de asistencia al conductor empleando visión artificial en vehículos de transporte público*. Ambato: Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería Electrónica y Comunicaciones.
- Salesiana, V. d. (18 de Octubre de 2020). *Ecosistema Universidad Politécnica Salesiana*. Obtenido de <https://www.investigacion.ups.edu.ec/startlabs/>
- San Juan, E., Jamett, M., Kaschel, H., & Sánchez, L. (2015). Sistema de reconocimiento de voz mediante wavelets, predicción lineal y redes backpropagation. *Ingeniare*, 11.
- Sparrold, S., & Stairiker, M. (07 de Febrero de 2017). *Edmund Optics*. Obtenido de Edmund Optics: <https://www.edmundoptics.com/document/download/489641>
- STMicroelectronics . (17 de Abril de 2019). *STlife.augmented*. Obtenido de STlife.augmented: <https://www.st.com/resource/en/datasheet/stm32h743vi.pdf>

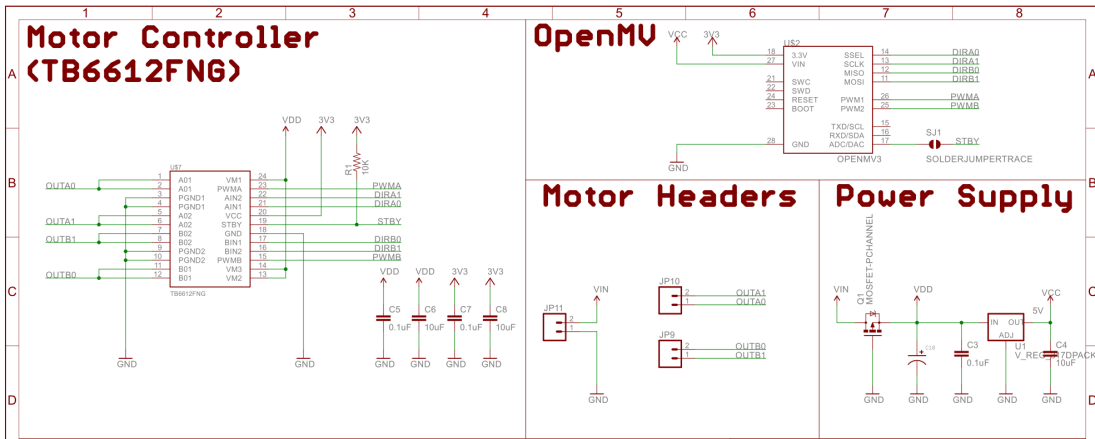
- Suárez Romero, A., & del Pozo Quintero, A. (2016). Estrategia de navegación para robots móviles mediante redes neuronales. 1.
- Tapiero Santa, Y. D. (2019). *Diseño e implementación de un robot seguidor de línea de competencia para la categoría velocista*. Ibagué.
- Texas Instruments. (Octubre de 2011). *ti.com*. Obtenido de ti.com: https://www.ti.com/lit/ds/symlink/am3359.pdf?ts=1607099653568&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FAM3359
- Torres, J. (2018). *Deep Learnig, Introducción práctica con Keras (PRIMERA PARTE)*. Barcelona: Kindle Direct Publishing.
- Trabes, E. (2018). *Desarrollo de algoritmos para la exploración submarina mediante vehículos autónomos con visión artificial monocular*. Bahía Blanca: Universidad Nacional del Sur.
- Vera Arenas, J. A., & Alejandro Proaño, E. A. (2016). *Diseño e implementación de dos robots seguidores de línea modalidad velocista y destreza para participaciones en concursos de robótica*. Guayaquil: Universidad Católica de Santiago de Guayaquil. Obtenido de <http://repositorio.ucsg.edu.ec/handle/3317/5445>
- Vizcaya Cárdenas, R. (2018). *Deep Learning para la Detección de Peatones y Vehículos*. Ciudad de México: Universidad Autónoma del Estado de México.

ANEXOS

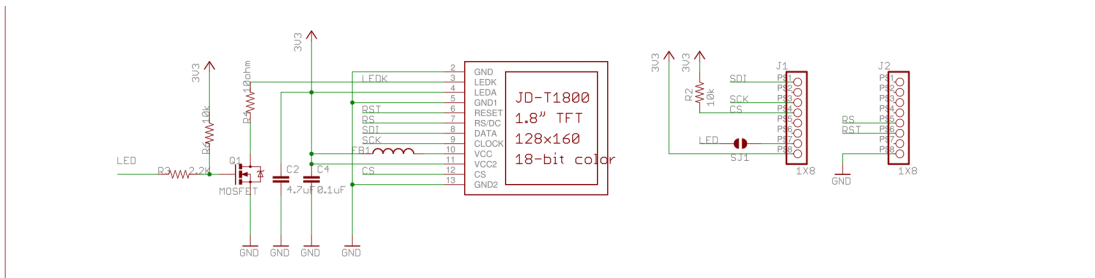
Anexo 1 Esquema electrónico de OpenMV Cam H7, Motor Shield & LCD Shield.



Esquema electrónico de OpenMV Cam H7, Fuente: (Agyeman & Abdelkader, OpenMV, 2018).

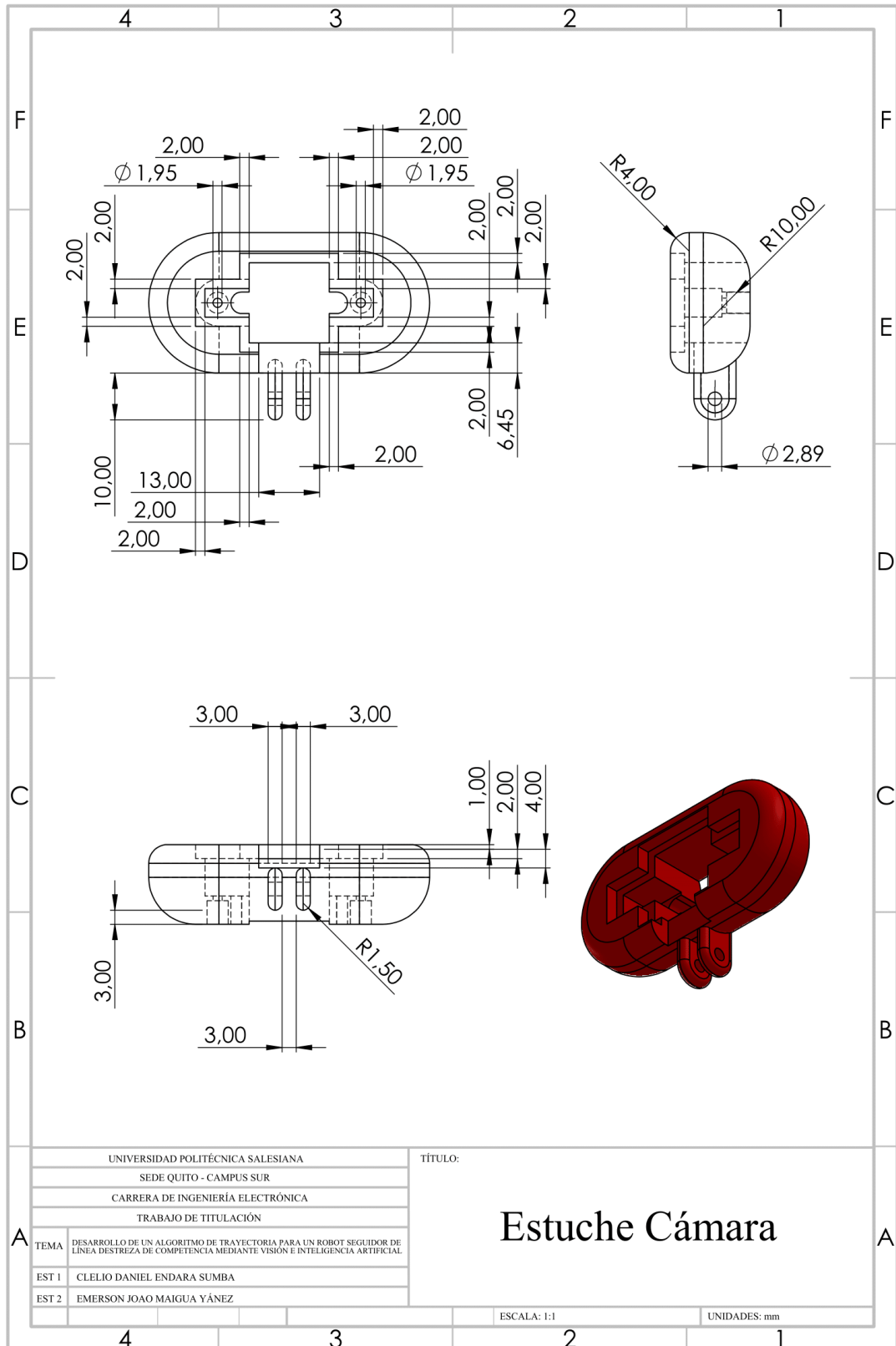


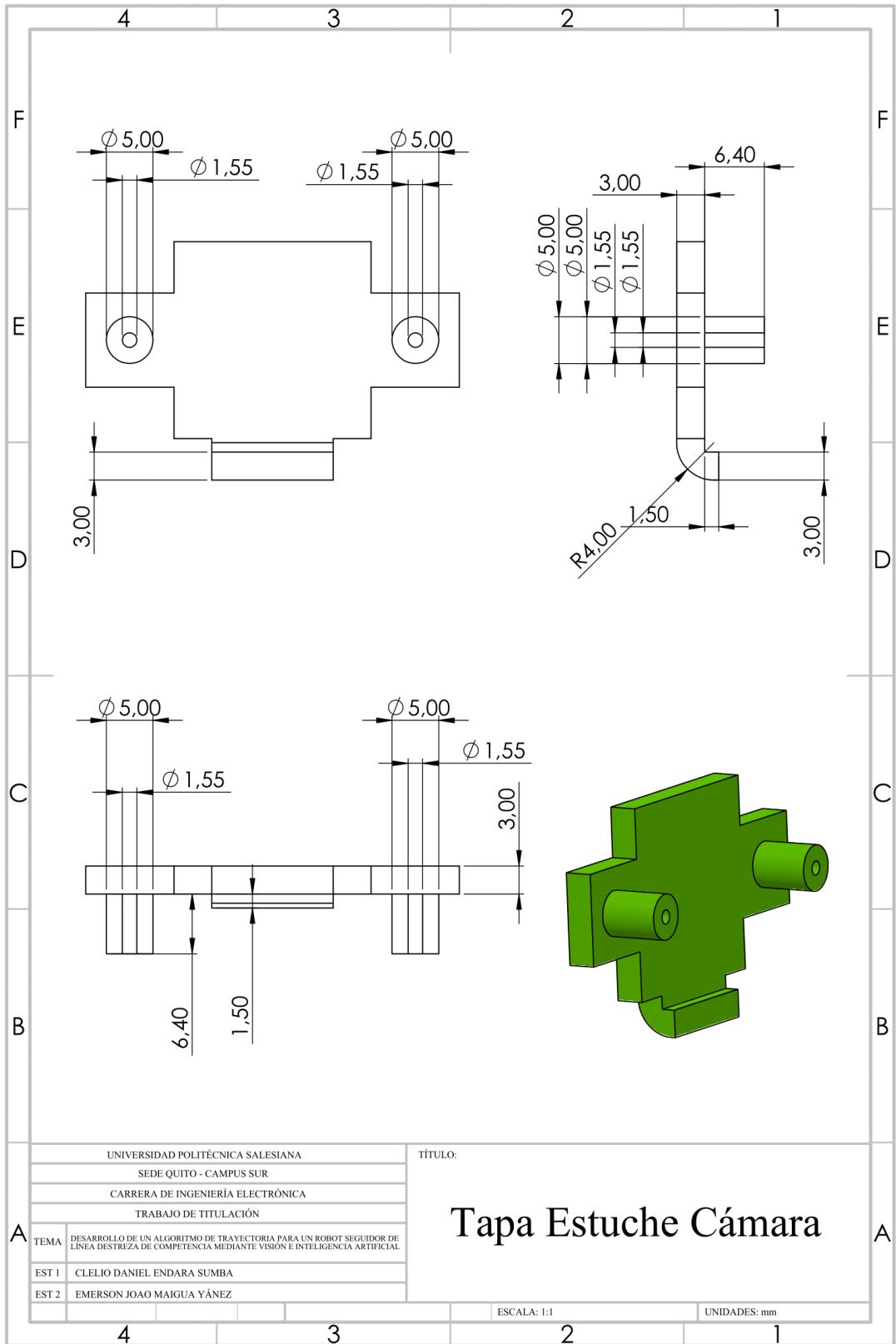
Esquema electrónico del Motor Shield, Fuente (Agyeman & Abdelkader, OpenMV, 2018).



Esquema electrónico de la LCD Shield, Fuente: (Agyeman & Abdelkader, OpenMV, 2018).

Anexo 2 Planos de detalle del subconjunto de piezas que conforma el robot móvil seguidor de línea destreza.



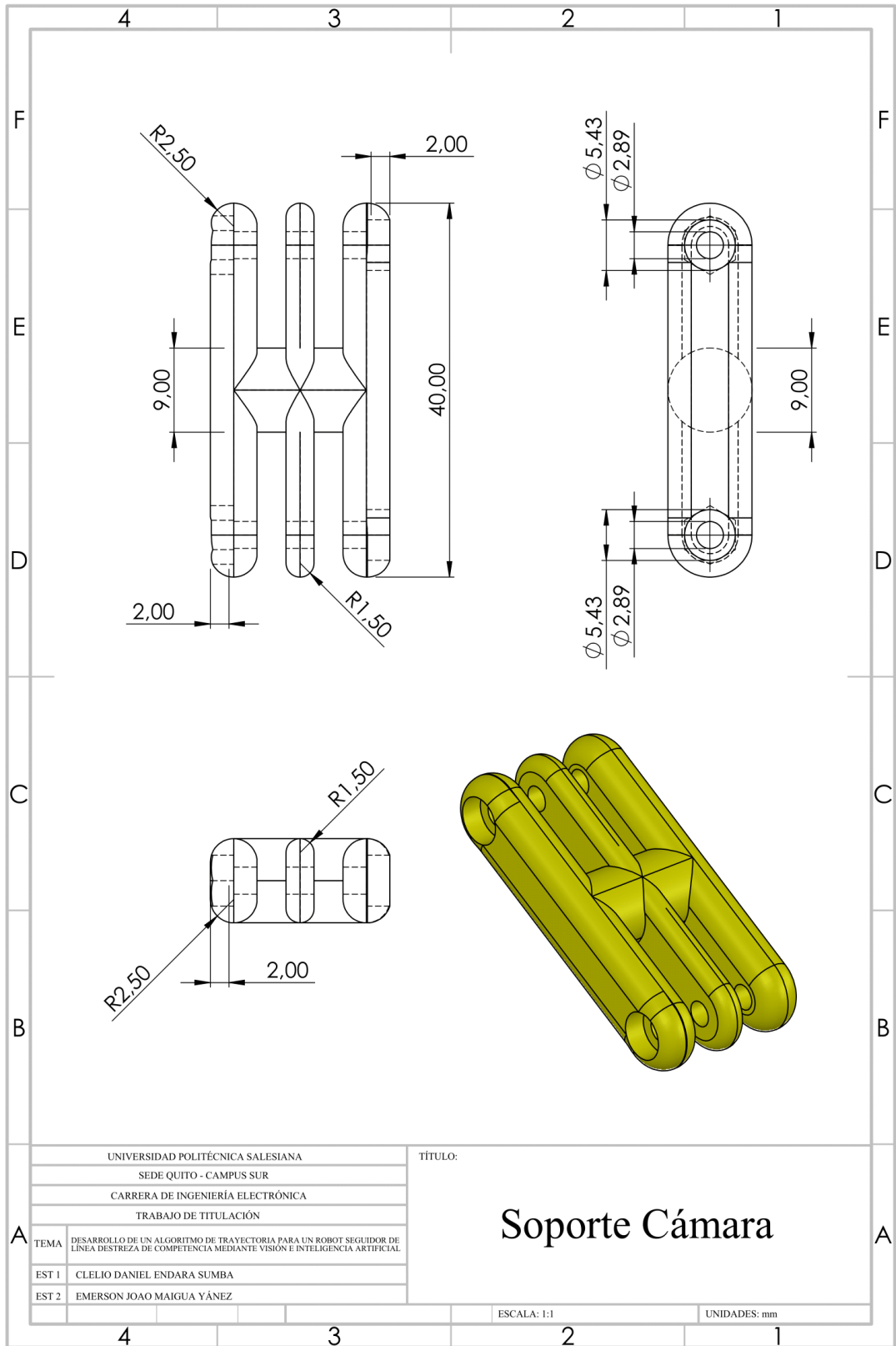


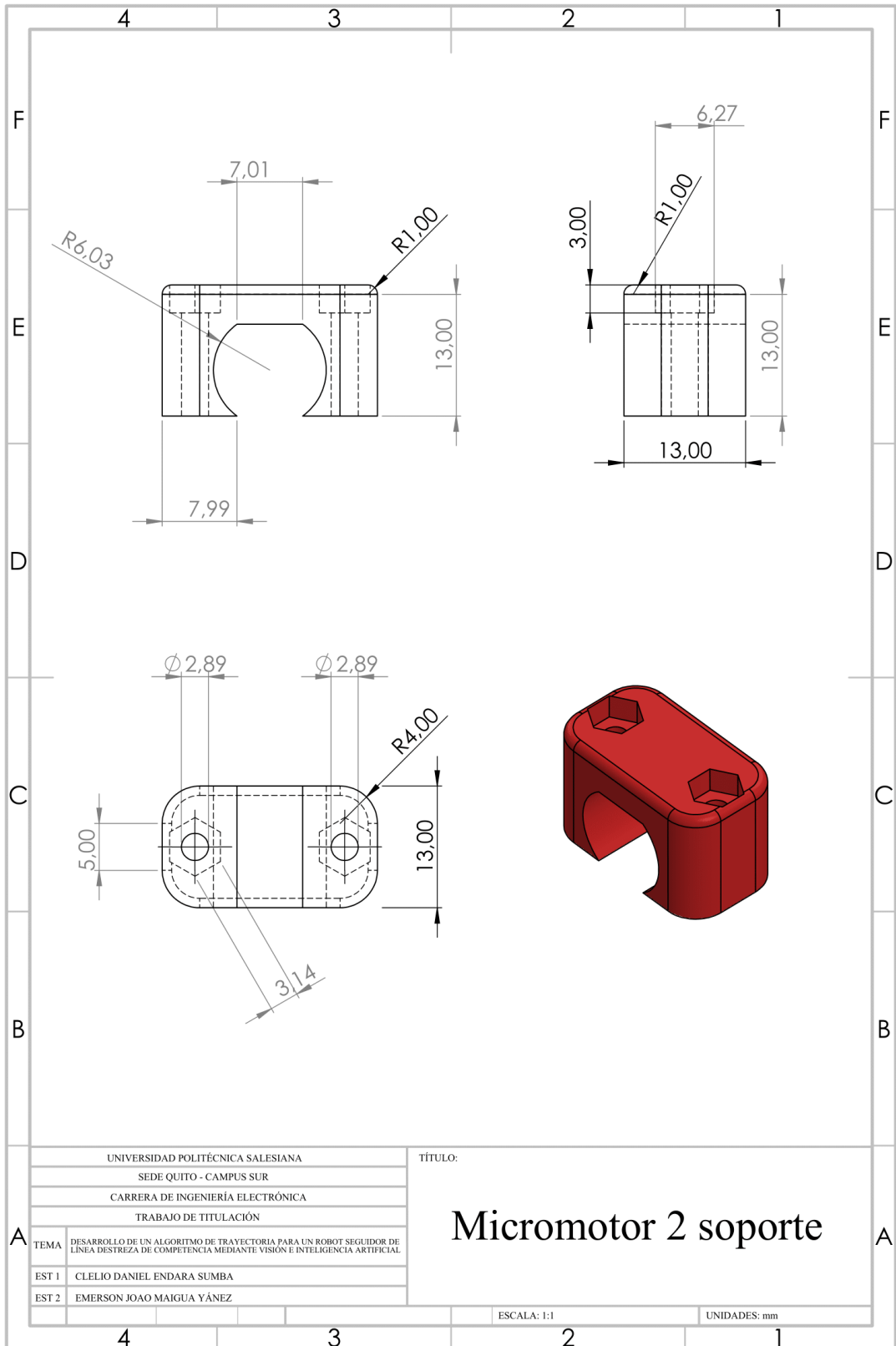
UNIVERSIDAD POLITÉCNICA SALESIANA	
SEDE QUITO - CAMPUS SUR	
CARRERA DE INGENIERÍA ELECTRÓNICA	
TRABAJO DE TITULACIÓN	
TEMA	DESARROLLO DE UN ALGORITMO DE TRAYECTORIA PARA UN ROBOT SEGUIDOR DE LINEA DESTREZA DE COMPETENCIA MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL
EST 1	CLELIO DANIEL ENDARA SUMBA
EST 2	EMERSON JOAO MAIGUA YÁNEZ

TÍTULO:

Tapa Estuche Cámara

ESCALA: 1:1 UNIDADES: mm



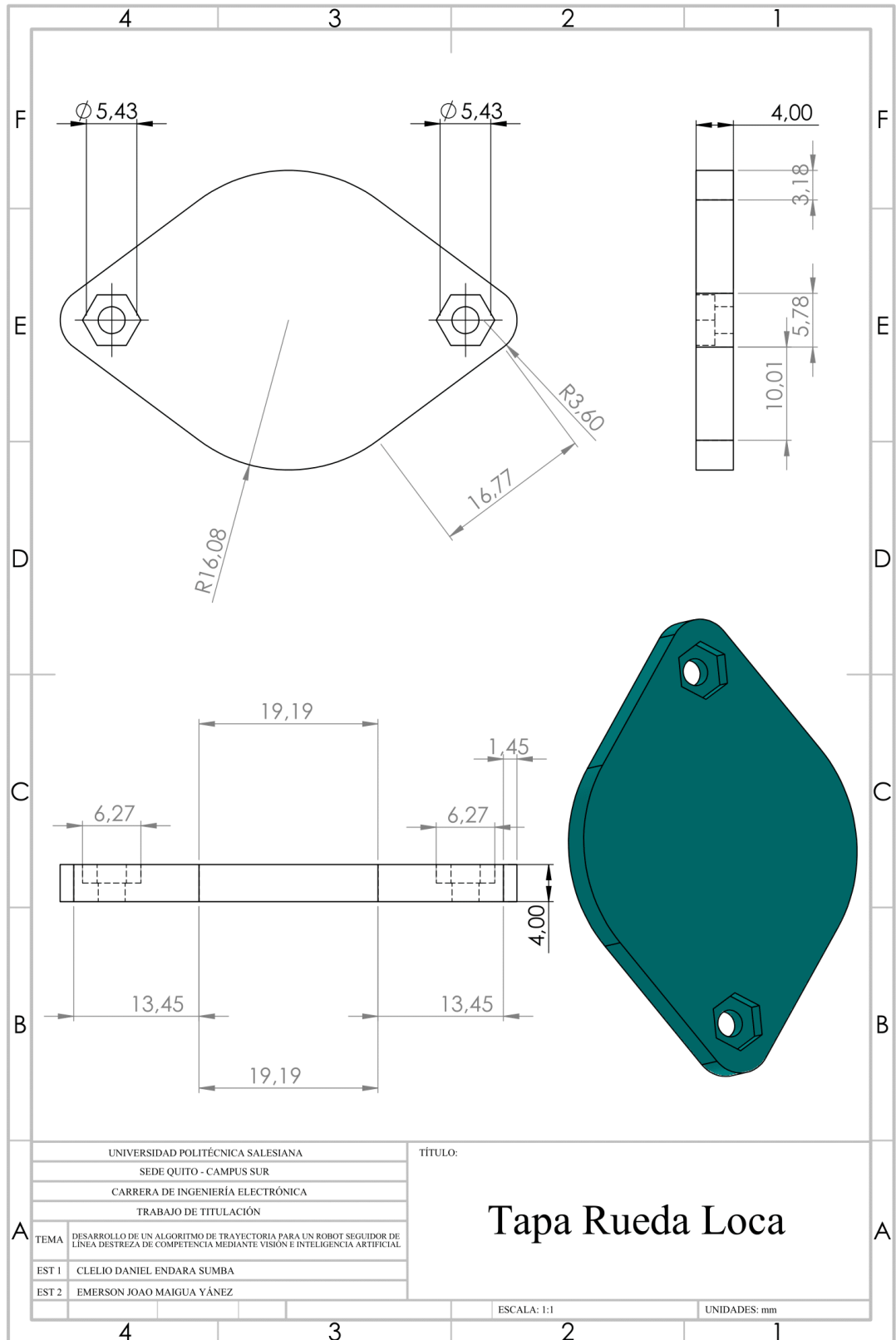


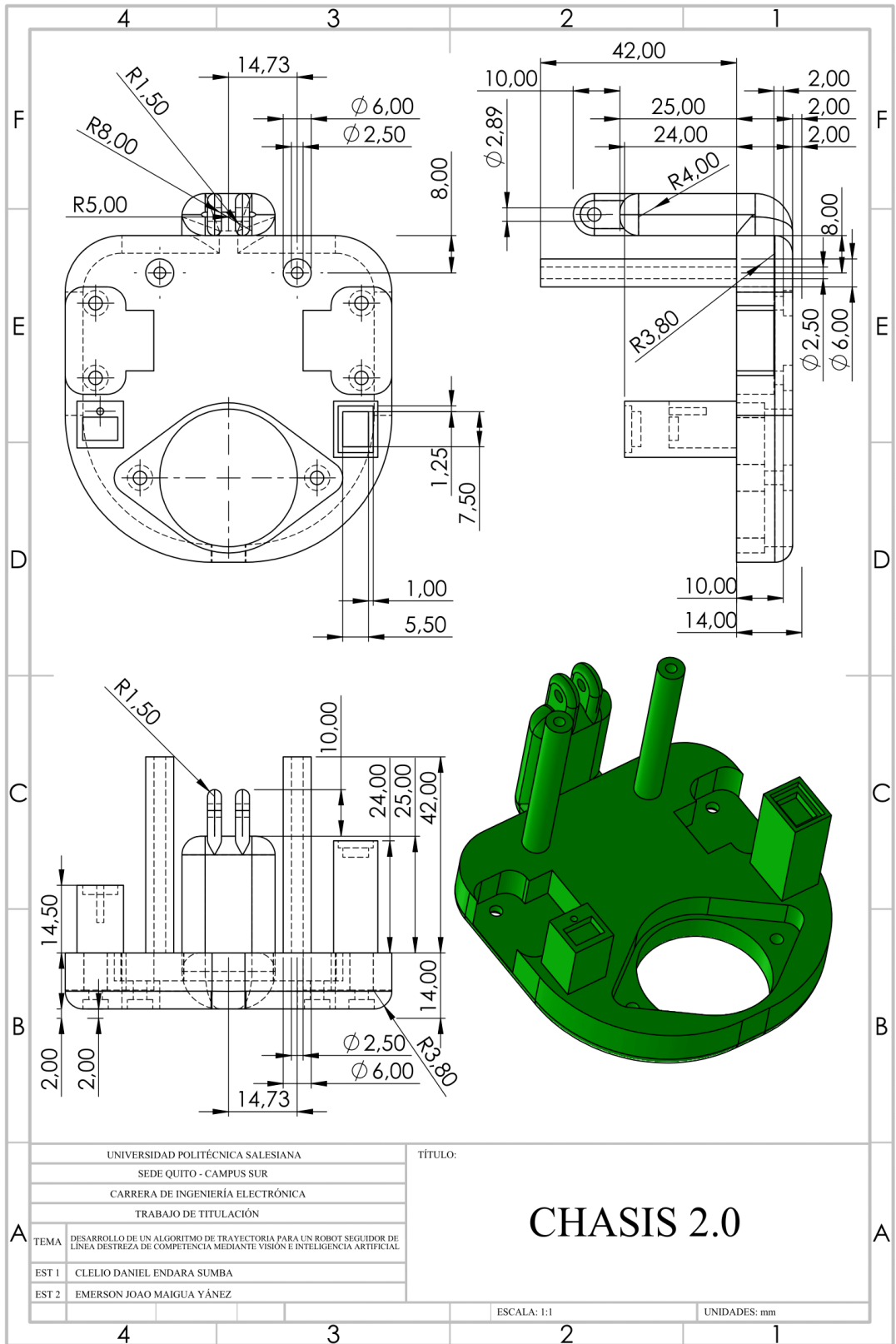
UNIVERSIDAD POLITÉCNICA SALESIANA	
SEDE QUITO - CAMPUS SUR	
CARRERA DE INGENIERÍA ELECTRÓNICA	
TRABAJO DE TITULACIÓN	
TEMA	DESARROLLO DE UN ALGORITMO DE TRAYECTORIA PARA UN ROBOT SEGUIDOR DE LÍNEA DESTREZA DE COMPETENCIA MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL.
EST 1	CLELIO DANIEL ENDARA SUMBA
EST 2	EMERSON JOAO MAIGUA YÁNEZ

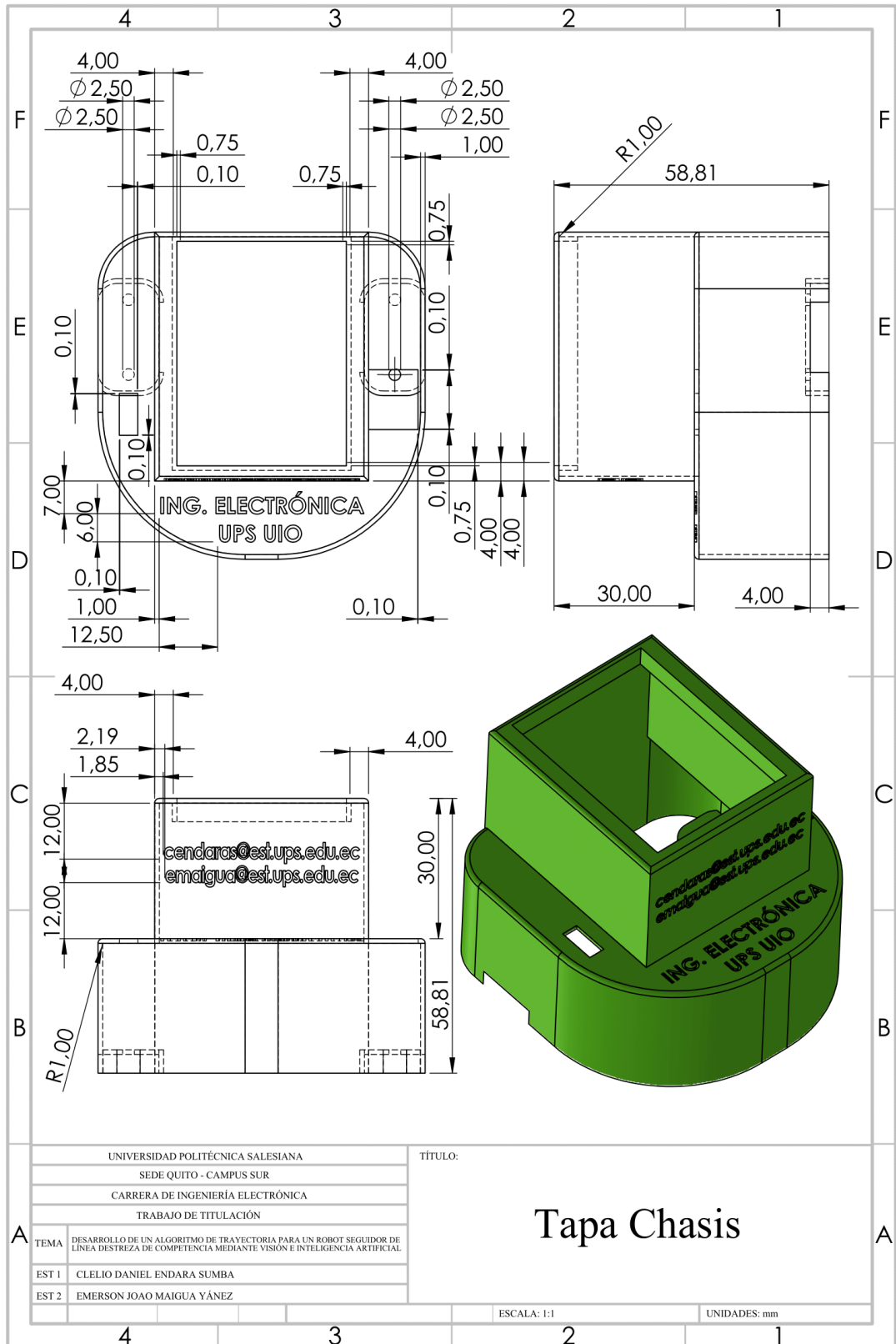
TÍTULO:

Micromotor 2 soporte

ESCALA: 1:1 UNIDADES: mm







UNIVERSIDAD POLITÉCNICA SALESIANA	
SEDE QUITO - CAMPUS SUR	
CARRERA DE INGENIERÍA ELECTRÓNICA	
TRABAJO DE TITULACIÓN	
TEMA	DESARROLLO DE UN ALGORITMO DE TRAYECTORIA PARA UN ROBOT SEGUIDOR DE LÍNEA DESTREZA DE COMPETENCIA MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL
EST 1	CLELIO DANIEL ENDARA SUMBA
EST 2	EMERSON JOAO MAIGUA YÁNEZ

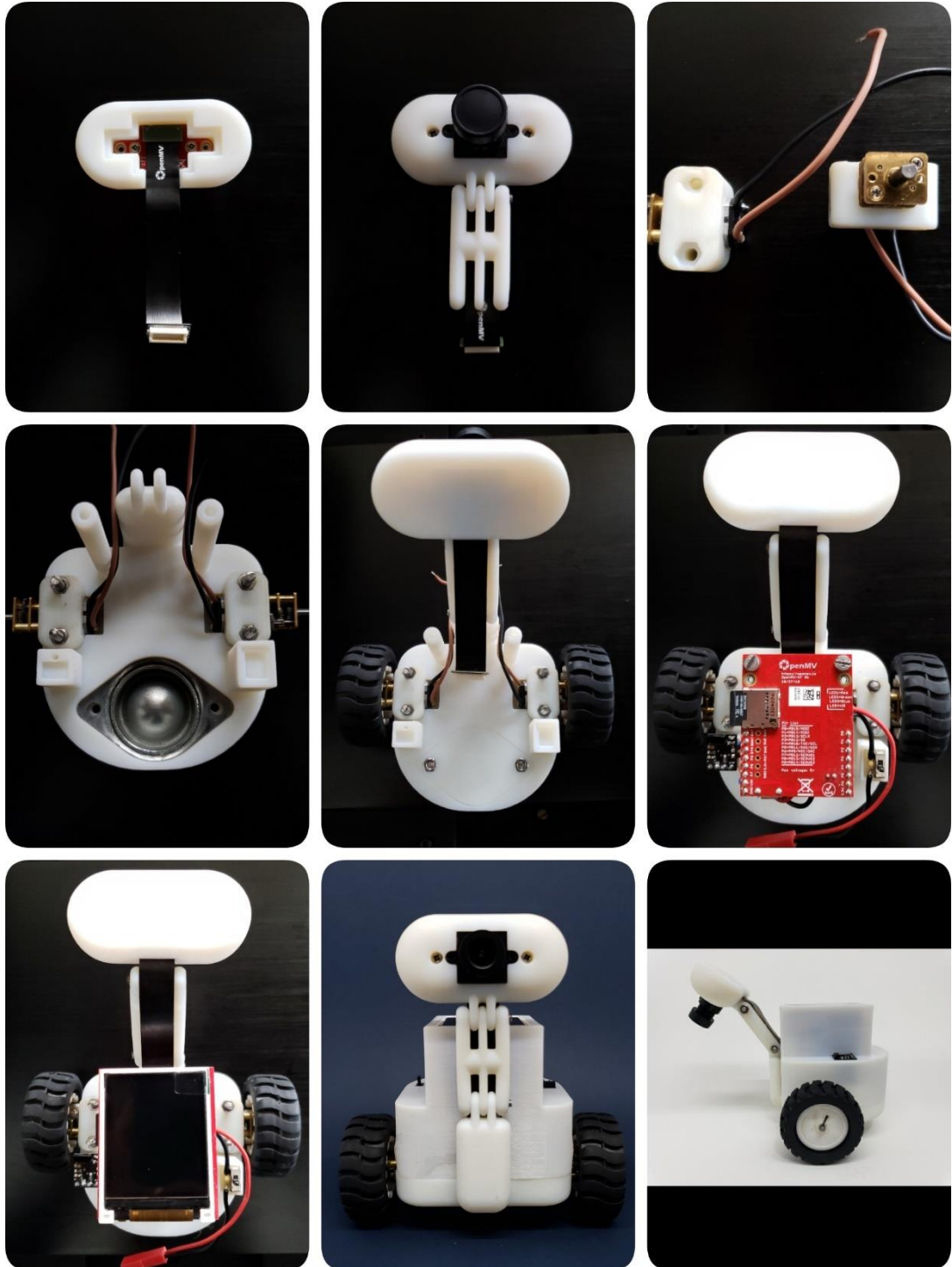
TÍTULO:

Tapa Chasis

ESCALA: 1:1

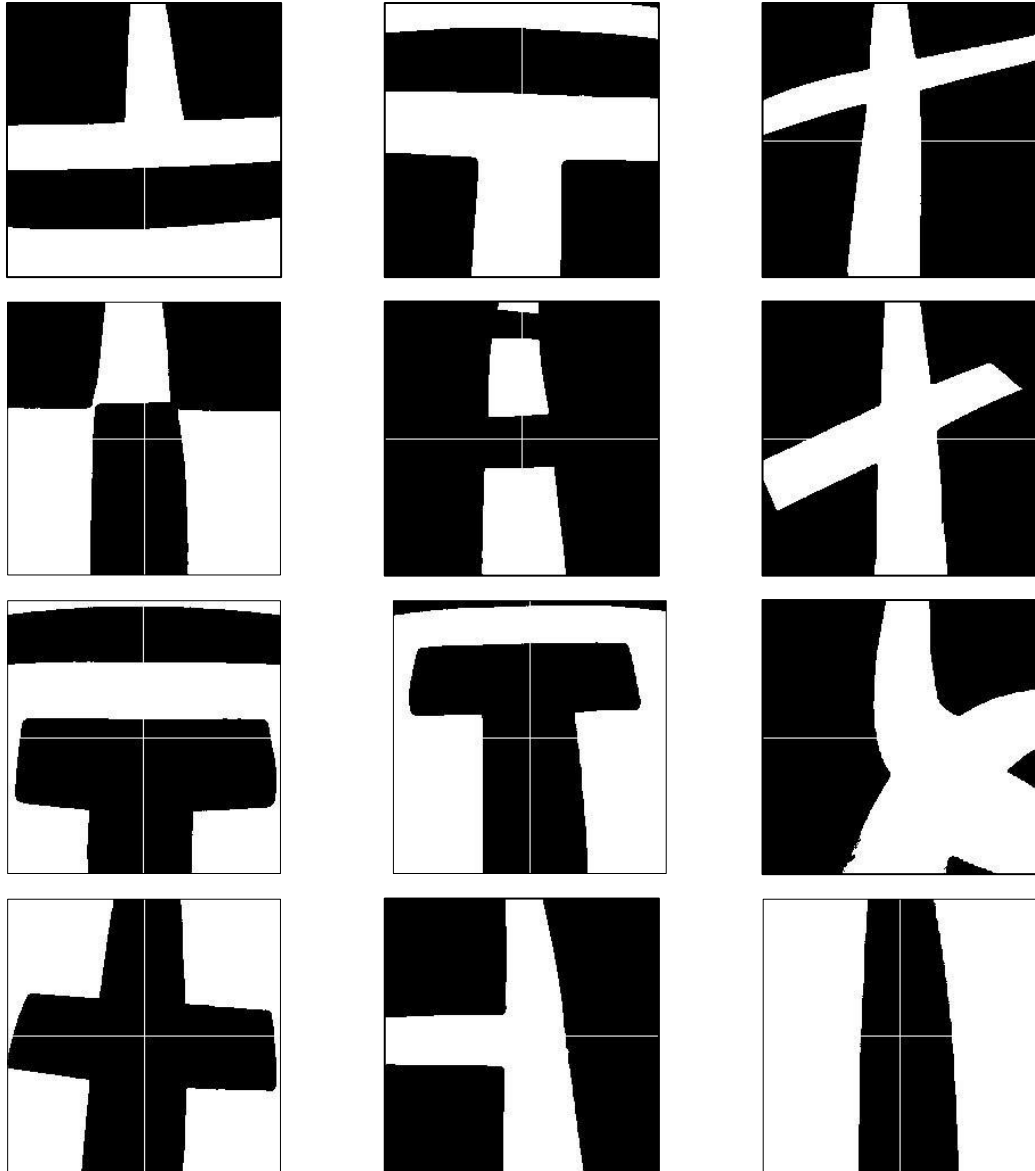
UNIDADES: mm

Anexo 3 Ensamblaje del robot móvil seguidor de línea destreza.

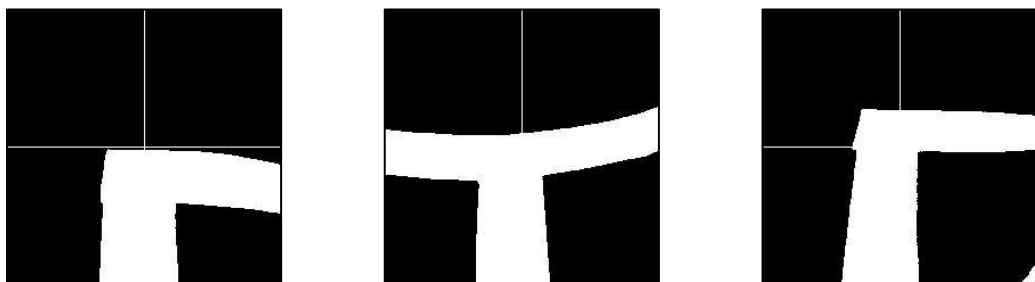


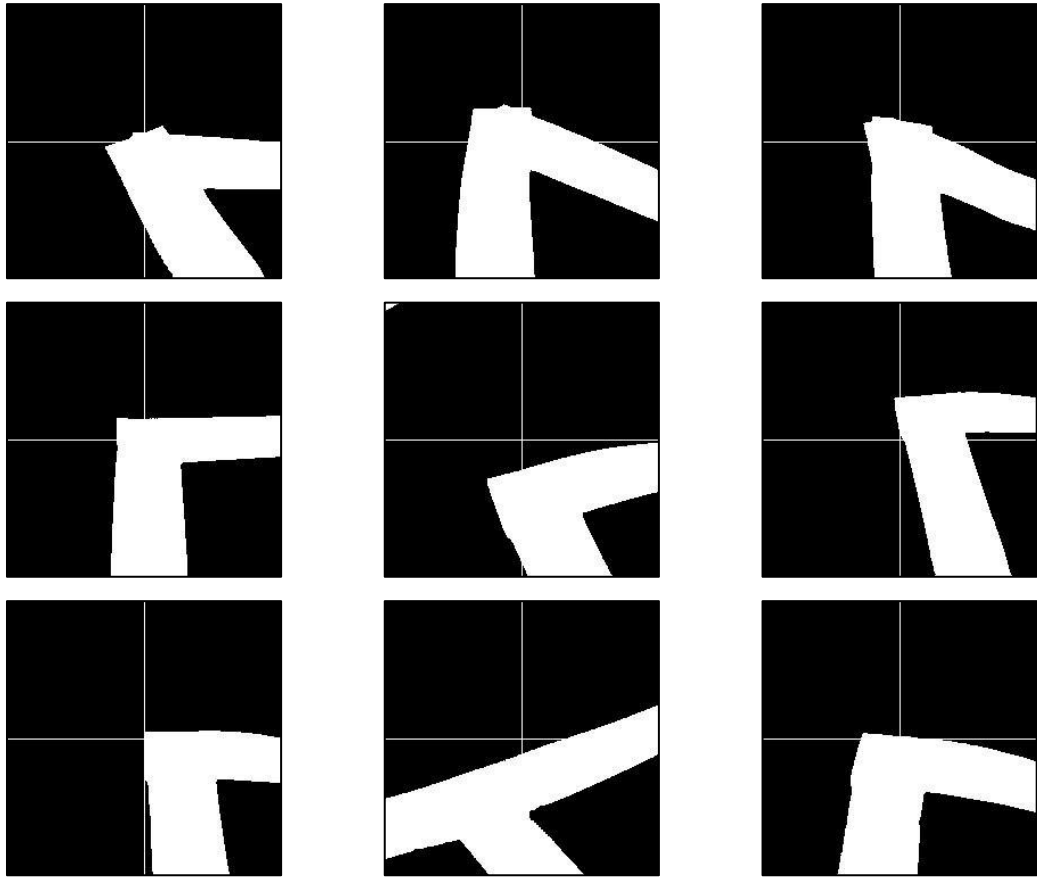
Anexo 4 Ejemplos relevantes para las entradas de la red neuronal.

Ejemplos relevantes caso “adelante”.

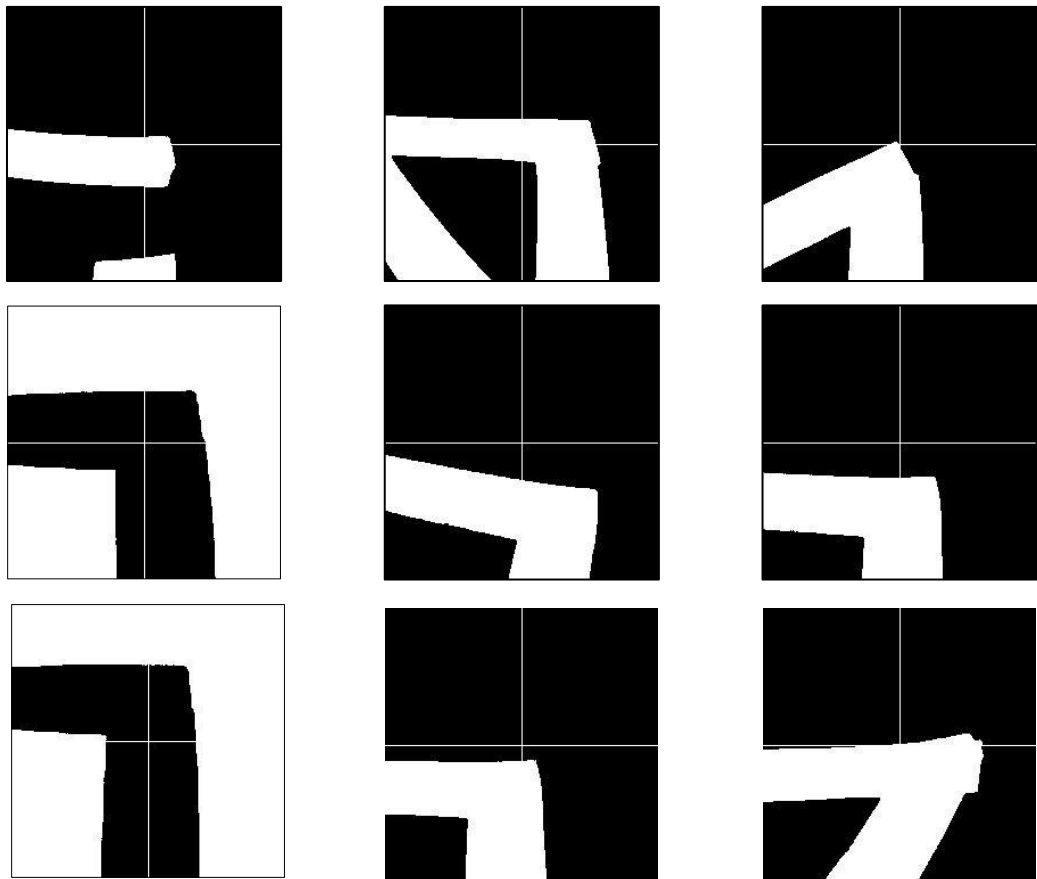


Ejemplos relevantes caso “Giro a la derecha”.

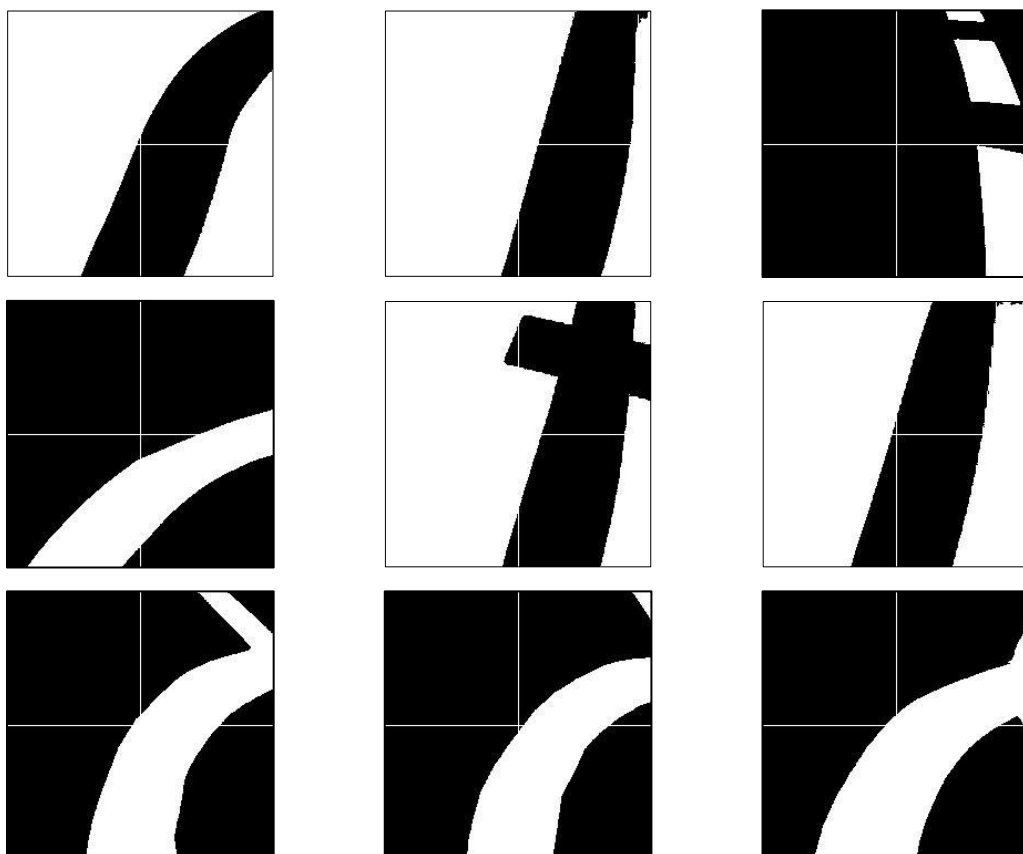




Ejemplos relevantes caso “Giro a la izquierda”.



Ejemplos relevantes caso “Giro con radio a la derecha”.



Ejemplos relevantes caso “Giro con radio a la izquierda”.

