



**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**CARRERA DE TELECOMUNICACIONES**

**TRABAJO DE INTEGRACIÓN CURRICULAR**

previo a la obtención del título de

**INGENIERO EN TELECOMUNICACIONES**

“Desarrollo de técnicas en MediaPipe Development con robot ROSMASTER X3 para determinar la percepción visual equipos y redes inalámbricas en el campo de las telecomunicaciones mediante el software WiFi HeatMap”

**AUTOR**

**JOSUÉ ARTURO VÉLEZ ARIAS**

**DOCENTE TUTOR**

**ING. LUIS MIGUEL AMAYA FARIÑO, MGTR**

**LA LIBERTAD–ECUADOR**

**2024 - 1**

## **AGRADECIMIENTO**

Agradezca de manera muy profunda y especial en primer lugar a Dios, debido a que siempre me dio la fortaleza mental para seguir adelante con este arduo proyecto. Agradecer de manera muy enfática a mi esposa que estuvo conmigo siempre apoyándome en todo momento del proyecto y ayudando el todo lo posible. A mis padres y mi familia que siempre me dieron ánimos suficientes para alcanzar mis objetivos en este proyecto.

## **DECLARACIÓN DE DOCENTE TUTOR**

En mi calidad de Docente Tutor del Trabajo de Integración Curricular, **“Desarrollo de técnicas en MediaPipe Development con robot ROSMASTER X3 para determinar la percepción visual de los equipos de telecomunicaciones y redes inalámbricas en el campo de las telecomunicaciones mediante el software WiFi HeatMap”**, elaborado por el señor **Vélez Arias Josué Arturo**, estudiante de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Península de Santa Elena, previo a la obtención del título de Ingenieros en Telecomunicaciones, me permito declarar que luego de haber dirigido su desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por la cual, apruebo en todas sus partes, encontrándose apto para la evaluación del docente especialista.

**Atentamente**



---

ING. LUIS MIGUEL AMAYA FARIÑO,

MGTR.DOCENTE TUTOR

## **DECLARACIÓN AUTORÍA DEL ESTUDIANTE**

El presente Trabajo de Integración Curricular, con el título **“Desarrollo de técnicas en MediaPipe Development con robot ROSMASTER X3 para determinar la percepción visual de los equipos de telecomunicaciones y redes inalámbricas en el campo de las telecomunicaciones mediante el software WiFi HeatMap”**, declaro que la concepción, análisis y resultados son originales y aportan a la actividad educativa en el área de Telecomunicaciones.

**Atentamente,**



---

Vélaz Arias Josué Arturo  
C.I. 2450566464

## **DECLARATORIA DE RESPONSABILIDAD**

Quien suscribe, VÉLEZ ARIAS JOSUE ARTURO con C.I. 2450566464, estudiante de la carrera de Telecomunicaciones declaro que el Trabajo de Titulación presentado a la unidad de Integración Curricular cuyo tema es **“Desarrollo de técnicas en MediaPipe Development con robot ROSMASTER X3 para determinar la percepción visual de los equipos de telecomunicaciones y redes inalámbricas en el campo de las telecomunicaciones mediante el software WiFi HeatMap”** corresponde y es de exclusiva responsabilidad de los autores y pertenece al patrimonio intelectual de la Universidad Estatal Península de Santa Elena.

**Atentamente,**



---

Vélez Arias Josué Arturo

C.I. 2450566464

## TRIBUNAL DE SUSTENTACIÓN



---

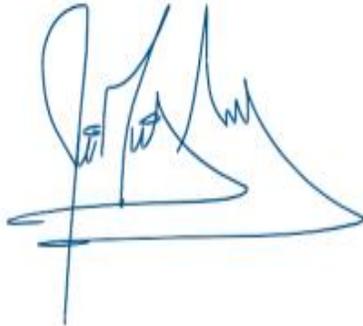
Ing. Ronald Rovira Jurado, Ph.D.  
**DIRECTOR DE LA CARRERA**



Firmado  
digitalmente por  
DANIEL  
ARMANDO  
JARAMILLO  
CHAMBA

---

Ing. Daniel Jaramillo Chamba, M.Sc.  
**DOCENTE ESPECIALISTA**



---

Ing. Luis Amaya Fariño, Mgr.  
**DOCENTE TUTOR GUÍA**



---

Ing. Corina Gonzabay De la A, Mgr.  
**SECRETARIA**

## RESUMEN

La percepción visual en entornos controlados, monitoreo de señales inalámbricas y la ubicación estratégica de equipos en redes inalámbricas se ha convertido en un aspecto crítico para garantizar un rendimiento eficiente de las infraestructuras de telecomunicaciones. Con el robot, se propone emplear herramientas avanzadas como MediaPipe Development y Rosmaster, que se destacan por su capacidad de realizar análisis de percepción visual de manera precisa y detallada. La implementación exitosa de reconocimiento de objetos en tiempo real mediante Mediapipe ha potenciado significativamente la capacidad del robot Rosmaster X3 para interpretar y responder a su entorno de manera dinámica. Este avance permite una detección precisa de objetos en tiempo real, combinada con un mapeo eficiente de los alrededores a través de ROS. La combinación de Mediapipe y del robot Rosmaster X3 ha llevado a resultados satisfactorios, mejorando la autonomía y la toma de decisiones de los robots en entornos cambiantes.

La aplicación de MediaPipe Development permite procesar información visual de manera robusta, facilitando la identificación de patrones y características en entornos de redes inalámbricas. Por otro lado, Rosmaster, al utilizar el enfoque de sistemas operativos de robots (ROS), proporciona una plataforma versátil para la coordinación y control de dispositivos de telecomunicaciones.

La iniciativa se orienta hacia la mejora de la calidad de señal y la eficiencia en la ubicación de equipos mediante la implementación de estrategias que recién están siendo explotadas y exploradas en el campo de las telecomunicaciones. El análisis visual proporcionado por MediaPipe Development contribuirá a la identificación de áreas críticas y a la toma de decisiones para optimizar la infraestructura de red.

Este enfoque se aplicará en un futuro una vez demostrado el funcionamiento total del proyecto en la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, proporcionando un escenario práctico y relevante para la implementación y validación de las propuestas. Con esta investigación, se busca contribuir significativamente al avance y la eficiencia de las redes inalámbricas, especialmente en entornos académicos y educativos.

## **ABSTRACT**

The optimization of signal quality and the strategic placement of equipment in wireless networks has become a critical aspect to ensure efficient performance of telecommunications infrastructures. In this context, it is proposed to use advanced tools such as MediaPipe Development and Rosmaster, which stand out for their ability to perform accurate and detailed visual perception analysis.

The application of MediaPipe Development allows robust processing of visual information, facilitating the identification of patterns and features in wireless network environments. On the other hand, Rosmaster, using the Robot Operating Systems (ROS) approach, provides a versatile platform for the coordination and control of telecommunication devices.

The initiative is geared towards improving signal quality and equipment placement efficiency by implementing innovative strategies. The visual analysis provided by MediaPipe Development will contribute to the identification of critical areas and informed decision-making to optimize the network infrastructure.

This approach will be applied at the School of Systems and Telecommunications of St. Helena Peninsula State University, providing a practical and relevant scenario for the implementation and validation of the proposals. With this research, we seek to contribute significantly to the advancement and efficiency of wireless networks, especially in academic and educational environments.

## INDICE DE CONTENIDO

INTRODUCCIÓN .....	1
CAPÍTULO I .....	2
1. GENERALIDADES DEL PROYECTO .....	2
1.1 Planteamiento del proyecto.....	2
1.2 Antecedentes .....	3
1.3 Descripción del proyecto. ....	4
1.4    Objetivos .....	5
1.4.1    Objetivo general.....	5
1.4.2    Objetivos específicos .....	5
1.5    Justificación .....	5
1.6    Alcance del proyecto.....	6
1.7    Metodología .....	7
1.8    Resultados esperados .....	9
CAPÍTULO 2.....	10
2. MARCO TEÓRICO.....	10
2.1 Introducción .....	10
2.2 Mapeo de redes .....	10
2.3 Optimización de una red .....	11
2.4 Zonas muertas de una red inalámbrica.....	16
- Interferencia electromagnética .....	16
- Características del entorno .....	16
- Rango y potencia de transmisión .....	16
- Protocolos y estándares Wi – Fi.....	17
- Sistemas Mesh y roaming .....	17
- Herramientas de diagnostico .....	17
2.5 Inteligencia artificial .....	17
2.5.1 Aprendizaje automático .....	17
2.5.2 Aplicaciones prácticas de la IA.....	18
2.5.3 Aplicaciones de la IA en el campo de las telecomunicaciones .....	18
2.5.4 El futuro de la inteligencia artificial.....	19

2.5.5 Redes neuronales.....	19
2.6 Lenguajes de programación .....	25
2.6.1 Python 3 .....	26
2.7 Real VNC viewer .....	27
2.8 VMware Workstation pro .....	28
2.9 WiFi Heatmap.....	30
2.10 MediaPipe Development.....	31
2.11 Lidar S1 para Rosmaster X3 .....	33
2.12 Jetson Nano 4 Gb .....	44
2.13 Rosmaster X3.....	45
2.13.1 Depth Cámara .....	47
CAPÍTULO 3.....	49
3. PROPUESTA DEL PROYECTO.....	49
3.1 Descripción general de la propuesta .....	49
3.2 FASE I .....	50
3.2.1 Componentes físicos .....	50
3.2.1.1 Ensamblaje del robot Rosmaster X3 .....	51
3.3 FASE II.....	57
3.3.1 Configuración Jetson Nano.....	57
3.4 Instalación de red neuronal – reconocimiento de objetos .....	61
3.5 FASE III.....	67
3.5.1 Mapeo en 2D en entornos controlados.....	67
3.5.2 Mapeo en 3D en entornos controlados.....	70
3.6 WiFi Heatmap .....	72
3.7 Costos de la implementación de la propuesta .....	78
CAPITULO IV.....	79
1. RESULTADOS ESPERADOS.....	79
4.1 Análisis resultados FASE I .....	79
4.2 Análisis resultados FASE II.....	84
4.3 Análisis de resultados FASE III.....	90
4.3.1 Comparación de resultados pre y post uso del software WiFi Heatmap.....	92
4.4 Resultados finales .....	93
CAPITULO V.....	94
CONFIGURACIÓN INICIAL ROSMASTER X3 Y JETSON NANO 4GB.....	94
5.1 FICHA TECNICA ROSMASTER X3 .....	94

5.1.1 Información general .....	94
5.1.2 Descripción general.....	95
5.1.3 Hardware.....	95
5.1.4 Software .....	95
5.2 Tarjeta de expansión .....	95
5.3 PID del movimiento del robot ROSMASTER X3.....	96
5.4 Ensamblaje del ROSMASTER X3.....	98
5.5 Conexiones de tarjetas .....	111
5.6 Configuración de la Jetson Nano.....	118
5.7 Estructura de ejecución del ROSMASTER X3 .....	124
CONCLUSIONES .....	126
RECOMENDACIONES.....	127
REFERENCIAS.....	127

## TABLA DE FIGURAS

Figura 1 .....	8
Diseño de la metodología.....	8
Figura 2.....	23
Plano de diseño de redes neuronales.....	23
Figura 3 .....	27
Logo PYTHON 3.....	27
Figura 4.....	28
Logo REAL VNC .....	28
Figura 5 .....	30
Logo VMWare.....	30
Figura 6.....	31
Mapeo de cobertura de red en WiFi Heatmap.....	31
Figura 7.....	33
Imagen de referencia MediaPipe Development.....	33
Figura 8.....	33
Diagrama de funcionamiento LIDAR laser.....	33
Figura 9 .....	35
Descripción mecanismo de rotación LIDAR laser.....	35

Figura 10 .....	36
Mecanismo de triangulación laser.....	36
Figura 11 .....	37
Mapeo de entorno mediante LIDAR laser. ....	37
Figura 12 .....	38
Triangulación laser directa.....	38
Figura 13 .....	38
Triangulación laser oblicua.....	38
Figura 14 .....	41
Comparación gráfica del LIDAR laser referentes a la distancia del mismo. ....	41
Figura 15 .....	42
LIDAR laser.....	42
Figura 16 .....	44
Escenarios de aplicación LIDAR laser. ....	44
Figura 17 .....	45
Placa de Jetson Nano 4Gb.....	45
Figura 18 .....	45
Datasheet Jetson Nano de 4Gb.....	45
Figura 19 .....	46
Piezas fuera de caja del ROSMASTER X3. ....	46
Figura 20 .....	47
Imagen de referencia ROSMASTER X3. ....	47
Figura 21 .....	48
Depth cámara del ROSMASTER X3.....	48
Figura 22 .....	51
Ensamble de piezas para el ROSMASTER X3. ....	51
Figura 23 .....	52
Diagrama de conexión de tarjetas del ROSMASTER X3. ....	52
Figura 24 .....	53
Tarjetas ensambladas en el ROSMASTER X3.....	53
Figura 25 .....	53
Imagen de referencia del LIDAR laser en el ROSMASTER X3.....	53
Figura 26 .....	54
Imagen de puesta final del LIDAR en el ROSMASTER X3.....	54
Figura 27 .....	55

Imagen de referencia Depth cámara en el ROSMASTER X3. ....	55
Figura 28 .....	55
Depth cámara montada en el ROSMASTER X3. ....	55
Figura 29 .....	56
Tarjeta Jetson Nano de 4Gb montada en el ROSMASTER X3. ....	56
Figura 30 .....	56
Entorno de inicio de Nvidia para la Jetson Nano. ....	56
Figura 31 .....	58
Nombre de red del ROSMASTER X3. ....	58
Figura 32 .....	58
Interfaz de inicio de RealVNC. ....	58
Figura 33 .....	60
Códigos de descarga para el soporte Jetson Inference. ....	60
Figura 34 .....	60
Códigos de actualización para la red neuronal. ....	60
Figura 35 .....	60
Actualización de paquetes. ....	60
Figura 36 .....	61
Descarga en tiempo real de las actualizaciones. ....	61
Figura 37 .....	61
Finalización de librerías y códigos de Jetson Inference. ....	61
Figura 38 .....	62
Interfaz del banco de imágenes para descargar. ....	62
Figura 39 .....	63
Verificación del archivo descargado del banco de imágenes. ....	63
Figura 40 .....	63
Datasheet de la red neuronal previamente entrenada. ....	63
Figura 41 .....	64
Guía de optimización del código de la red neuronal. ....	64
Figura 42 .....	64
Finalización de la ejecución de comandos para la red neuronal. ....	64
Figura 43 .....	65
Código de ejecución final para la red neuronal. ....	65
Figura 44 .....	65
Llamamiento de librerías para la ejecución. ....	65

Figura 45 .....	66
Detección de prueba con un código sencillo de reconocimiento de manos. ....	66
Figura 46 .....	67
Llamamiento para la identificación del robot, cámara y lidar. ....	67
Figura 47 .....	68
Acceso a la cámara y lidar mediante los siguientes códigos.....	68
Figura 48 .....	68
Mapeo en 2D del ROSMASTER X3. ....	68
Figura 49 .....	70
Comandos para guardar de manera directa el mapeo.....	70
Figura 50 .....	70
Diagrama de funcionamiento del SLAM Lidar en 3D.....	70
Figura 51 .....	71
Comandos para activación de cámara y lidar en mapeo 3D. ....	71
Figura 52 .....	72
Mapeo en 3D del ROSMASTER.....	72
Figura 53 .....	73
Interfaz de inicio WiFi Headmap.....	73
Figura 54 .....	74
Seleccionar los equipos a utilizar en nuestra práctica.....	74
Figura 55 .....	74
Selección de cámara IP. ....	74
Figura 56 .....	75
Plano de casa donde se realizó la práctica. ....	75
Figura 57 .....	76
Zona de cobertura de equipos “mal posicionados”.....	76
Figura 58 .....	77
Optimización y ubicación adecuada de los equipos para mejor cobertura y rendimiento. ....	77
Figura 59 .....	78
Mapa final de la cobertura de red inalámbrica mejorada.....	78
Figura 60 .....	79
Guía de los elementos del ROSMASTER X3. ....	79
Figura 61 .....	80
Primer arranque del ROSMASTER X3.....	80
Figura 63 .....	81

ROSMASTER X3 armado y ensamblado de manera correcta. ....	81
Figura 64 .....	83
Interfaz de inicio en la APP del ROSMASTER X3.....	83
Figura 65 .....	83
Insertar la IP correcta y los puertos de audio y video. ....	83
Figura 66 .....	83
Menú del Rosmaster X3.....	83
Figura 67 .....	84
Verificación del correcto funcionamiento de la APP para el manejo del ROSMASTER X3.....	84
Figura 68 .....	85
Inicio del entorno Nvidia de la Jetson Nano. ....	85
Figura 69 .....	86
Inicio de la Jetson Nano. ....	86
Figura 70 .....	86
Código para la detección de objetos. ....	86
Figura 71 .....	87
Llamamiento de comandos para la ejecución final de la red neuronal. ....	87
Figura 72 .....	88
Reconocimiento de equipos de telecomunicaciones en tiempo real. ....	88
Figura 73 .....	88
Reconocimiento de equipos de telecomunicaciones en tiempo real II.....	88
Figura 74 .....	89
No identifica objetos que no hayan sido programados previo a la ejecución de la red neuronal.....	89
Figura 75 .....	89
No identifica objetos que no hayan sido programados en la red neuronal II. ....	89
Figura 76 .....	90
Mapeo en 2D.....	90
Figura 77 .....	91
Mapeo en 3D.....	91
Figura 78 .....	93
Zona de cobertura optimizada.....	93

**INDICE DE TABLAS**

Tabla 1 .....	34
Especificaciones generales del Laser .....	34
Tabla 2 .....	40
Parámetros de medición LIDAR laser. ....	40
Tabla 3 .....	42
Tabla de entrada y salida de voltaje LIDAR laser. ....	42
Tabla 4 .....	43
Tabla de descripción del motor y núcleo del LIDAR laser.....	43
Tabla 5 .....	79
Tabla de costos de los elementos del proyecto. ....	79
Tabla 6 .....	92
Tabla de valores del rendimiento de la red primera prueba. ....	92
Tabla 7 .....	92
Tabla de valores del rendimiento de la red con la ubicación de equipos optimizados. ....	92

# INTRODUCCIÓN

La introducción a la percepción visual en entornos controlados de la calidad de señal y la ubicación estratégica de equipos en redes inalámbricas se configura como una parte esencial en la evolución de las infraestructuras de telecomunicaciones. En este contexto, la aplicación de herramientas avanzadas como MediaPipe Development y Rosmaster emerge como un enfoque innovador para llevar a cabo análisis de percepción visual de manera precisa y detallada.

La demanda creciente de un rendimiento eficiente en las redes inalámbricas ha impulsado la necesidad de estrategias más sofisticadas y tecnológicamente avanzadas. En este sentido, la utilización de MediaPipe Development se presenta como una solución robusta para procesar información visual, identificando patrones y características clave en el entorno de las redes inalámbricas.

Paralelamente, Rosmaster, basado en el sistema operativo de robots (ROS), proporciona una plataforma versátil para la coordinación y control de dispositivos de telecomunicaciones. Esta combinación de tecnologías ofrece una sinergia única para abordar desafíos relacionados con la calidad de señal y la ubicación estratégica de equipos en entornos inalámbricos.

Este estudio se enfocará en la aplicación práctica de estas herramientas en la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena. A través de este enfoque, buscamos no solo mejorar la eficiencia de las infraestructuras de red, sino también contribuir al conocimiento y desarrollo de estrategias innovadoras en el ámbito de las telecomunicaciones inalámbricas. Con esta investigación, aspiramos a impulsar avances significativos que beneficien tanto a entornos académicos como a la evolución continua de las redes inalámbricas.

# CAPÍTULO I

## 1. GENERALIDADES DEL PROYECTO

### 1.1 Planteamiento del proyecto

En la actualidad los sistemas de comunicación evolucionan a un ritmo acelerado y no solo es necesario que se pueda establecer comunicaciones de forma tradicional ya surge la necesidad de mejorar y hacerlo de una forma mucho más rápida, así que el hombre por medio de la tecnología ha desarrollado los medios para que lo que tenemos hoy en día sea tan fascinante, como por ejemplo el internet, la telefonía móvil, televisión digital, entre otras. Una de las herramientas que nos permite la comunicación es la implementación de una red inalámbrica que facilita la transferencia de información sin la necesidad de cables en donde la información se convierte en una herramienta fundamental para la obtención y aplicación de nuevas tendencias tecnológica. El avance tecnológico a nivel de Latinoamérica se ha desarrollado notablemente en todas las áreas y campos, cada vez hay mejores servicios y aplicaciones que requieren una mayor capacidad, calidad y seguridad de transmisión, los medios inalámbricos es uno de ellos, ha incrementado su capacidad de transmisión de información tanto en su ancho de banda como en su velocidad, y ha implementado calidad de servicio en sus transmisiones.

En muchos países del mundo y no es la excepción Ecuador, las redes inalámbricas constituyen un elemento clave en la industria de las telecomunicaciones, tanto por su influencia sobre la oferta y calidad de los servicios, como por la importancia que adquieren en los mercados liberalizados. Las redes inalámbricas han sido de gran ayuda en el momento de dar servicios en lugares de difícil acceso, gracias a ellos se han beneficiado un sinnúmero de personas; tal es el caso del Ecuador donde varias empresas han implementado este servicio en zonas rurales, esto se debe a que el tiempo de instalación, así como los costos de implementación de la red son bajos. Con las bandas no licenciadas que operan con WiFi, son la principal fortaleza de este tipo de redes ya que pueden operar libremente, son usadas principalmente en redes locales y personales. Redes como Wifi se las puede encontrar en muchos lugares como en centros comerciales, aeropuertos, campus universitarios, etc., por lo que comercialmente se han vuelto

muy populares, al igual que Bluetooth, debido que, en muchos teléfonos móviles, computadoras personales y muchos dispositivos electrónicos ya incluyen esta clase de chips para la transmisión de datos. El desarrollo vertiginoso de las telecomunicaciones en el Ecuador constituye un factor primordial en el desarrollo tecnológico de las instituciones educativas, aprovechando los distintos medios y recursos posibles que se tiene a disposición, esto ha llevado que las instituciones cuenten con todas las opciones de servicio de comunicación que existen en la actualidad como son las redes inalámbricas.

## 1.2 Antecedentes

Actualmente la tecnología Wi-Fi la se encuentra en todos los lugares. Desde su casa hasta su trabajo; hay lugares donde el acceso es gratuito lo cual facilita que muchas personas lo usen. Uno de los problemas que está enfrentando actualmente esta tecnología dentro de campus universitarios son las interferencias en ciertos puntos, dado por mala ubicación de los equipos, infraestructura de la institución educativa, etc. Existen ahora nuevas alternativas para solucionar estos inconvenientes, una para usuarios comunes y otras para usuarios avanzados que tienen conocimientos en tecnología. Dado los avances tecnológicos con dispositivos electrónicos que tengan conexión a WI-FI cada miembro tiene más de dos dispositivos para conectarse, entre estos pueden ser computadoras, celulares, tabletas, etc. Por lo que la aplicación de esta guía sería efectiva solucionando los inconvenientes de una manera sencilla para cada tipo de usuario.

La cobertura de señal débil es un problema común en las redes inalámbricas, especialmente en áreas grandes o con obstáculos físicos. Uno de los puntos a tener en cuenta son los factores que contribuyen a una señal débil, como la distancia entre el enrutador y los dispositivos, la interferencia y la absorción de señal por parte de obstáculos físicos. También saber explorar técnicas para mejorar la cobertura, como el uso de antenas de mayor potencia, repetidores o el ajuste estratégico de la ubicación de los puntos de acceso.

La interferencia es una problemática común en las redes inalámbricas, especialmente en entornos densamente poblados o con múltiples redes cercanas. Factores a tener en cuenta son

los diferentes tipos de interferencia, como la interferencia co - canal (de otras redes en el mismo canal) y la interferencia adyacente (de redes en canales cercanos). Explorar las técnicas para mitigar la interferencia, como la selección adecuada de canales, el uso de técnicas de mitigación de interferencias y la aplicación de algoritmos de gestión de tráfico.

### 1.3 Descripción del proyecto.

Actualmente la identificación de zonas muertas, de baja latencia, inventario de equipos de los laboratorios de la facultad y distribución de recursos siguen siendo de la forma tradicional, es decir mediante la prueba y error, calculo humano, etc. El siguiente proyecto tiene como finalidad mejorar significativamente como propuesta de desarrollo a futuro la eficiencia en la gestión del inventario de equipos de telecomunicaciones en la facultad de la carrera mediante la aplicación de tecnologías avanzadas. Se combinarán técnicas de percepción visual para evitar zonas muertas en señales inalámbricas, con capacidades de reconocimiento de objetos utilizando Mediapipe Development, con el fin de proporcionar una solución integral y eficaz para la identificación y seguimiento de equipos.

En el primer capítulo se detallará toda la información necesaria de manera general acerca del proyecto.

En el segundo capítulo hará referencia al marco teórico, donde se contextualiza y revisan conceptos, palabras y términos esenciales para el entendimiento de nuestro proyecto.

En el tercer capítulo de manera muy detallada y a fondo explicaremos la propuesta del proyecto.

En el cuarto capítulo se explicará los resultados esperados de nuestro proyecto con referencia nuestro objetivo general y nuestros objetivos específicos.

## 1.4 Objetivos

### 1.4.1 Objetivo general

- Mejorar la planificación y despliegue de redes wi – fi mediante técnicas de MediaPipe Development y el robot Rosmaster X3.

### 1.4.2 Objetivos específicos

- Identificar y analizar los fundamentos teóricos y técnicos relacionados al robot ROSMASTER X3, la tarjeta principal Jetson Nano de 4 Gb y el entorno de desarrollo MediaPipe Development.
- Desarrollar un código de programación para la percepción visual para el reconocimiento de objetos en tiempo real y el mapeo de área en entornos controlados con el ROSMASTER X3
- Evaluar y comparar el desempeño del mapeo y el uso del software WiFi HeatMap propuesto en entornos de red inalámbricas reales, considerando un escenario donde exista mala cobertura y otro donde la cobertura sea óptima.
- Desarrollo de una ficha técnica de manera general para el uso del robot Rosmaster X3 con su tarjeta Jetson Nano.

## 1.5 Justificación

La justificación de este proyecto se basa en la importancia de mejorar la calidad de la señal en redes inalámbricas de alta frecuencia en el campo de las telecomunicaciones. A medida que aumenta la demanda de servicios de comunicación inalámbrica, especialmente en frecuencias de GHz, es crucial garantizar una conectividad confiable y de alta calidad.

El uso de tecnologías como MediaPipe Development y análisis de percepción visual ofrece una oportunidad para abordar los desafíos asociados con la calidad de la señal en redes inalámbricas. Estas tecnologías permiten analizar y evaluar en tiempo real la calidad de la señal, identificación de áreas de interferencia, puntos de accesos débiles a zonas de baja

cobertura. Esto proporcionará información valiosa para la planificación de la red inalámbrica, así como para la toma de decisiones estratégicas en términos de ubicación de puntos de acceso, configuración de la red y mejora de la experiencia de uso.

Al llevar a cabo esta investigación, se espera obtener resultados significativos que contribuyan al conocimiento científico y tecnológico en el campo de las telecomunicaciones. Los resultados pueden ayudar a los profesionales de las telecomunicaciones, operadores de redes y proveedores de servicios a mejorar la calidad de la señal y mejorar el rendimiento de alta frecuencia.

Además, la investigación puede tener un impacto positivo en la sociedad al garantizar una conectividad confiable y de alta calidad en entornos donde se requiere comunicación inalámbrica eficiente, como en aplicaciones industriales, redes IoT, comunicaciones móviles avanzadas y comunicaciones críticas. Una mejora en la calidad de la señal en estas aplicaciones puede impulsar la eficiencia operativa, mejorar la seguridad y brindar una mejor experiencia al usuario.

En resumen, esta investigación se justifica por la necesidad de mejorar la calidad de señal en redes inalámbricas de alta frecuencia, utilizando tecnologías como MediaPipe Development y análisis percepción visual. Los resultados de esta investigación pueden tener un impacto significativo de forma positiva logrando obtener una mejor conectividad y eficiencia de las redes inalámbricas en frecuencias de GHz.

## 1.6 Alcance del proyecto

El principal objetivo del proyecto es lograr obtener un sistema funcional basado en MediaPipe Development y análisis de percepción visual, así como estrategias efectivas para mejorar la calidad de la señal en redes inalámbricas. Los resultados de esta investigación contribuirán al campo de las telecomunicaciones al proporcionar herramientas y enfoques innovadores para futuras optimizaciones en la calidad de la señal y mejorar la experiencia de los usuarios de redes inalámbricas en frecuencias de GHz.

Una vez demostrada la validez del código, en un futuro se espera que sea propuesto, aceptado y aplicado el proyecto en la Universidad Estatal Península de Santa Elena para poder identificar problemas con las redes inalámbricas que funcionan en todo el campus de la universidad y la facultad de sistemas y telecomunicaciones como por ejemplo problemas de cobertura, interferencia o zonas muertas donde se aprovecha al máximo la red de la universidad.

## 1.7 Metodología

En el desarrollo de este proyecto, utilizaremos algunos métodos de investigación en la cuales son:

### **Investigación exploratoria**

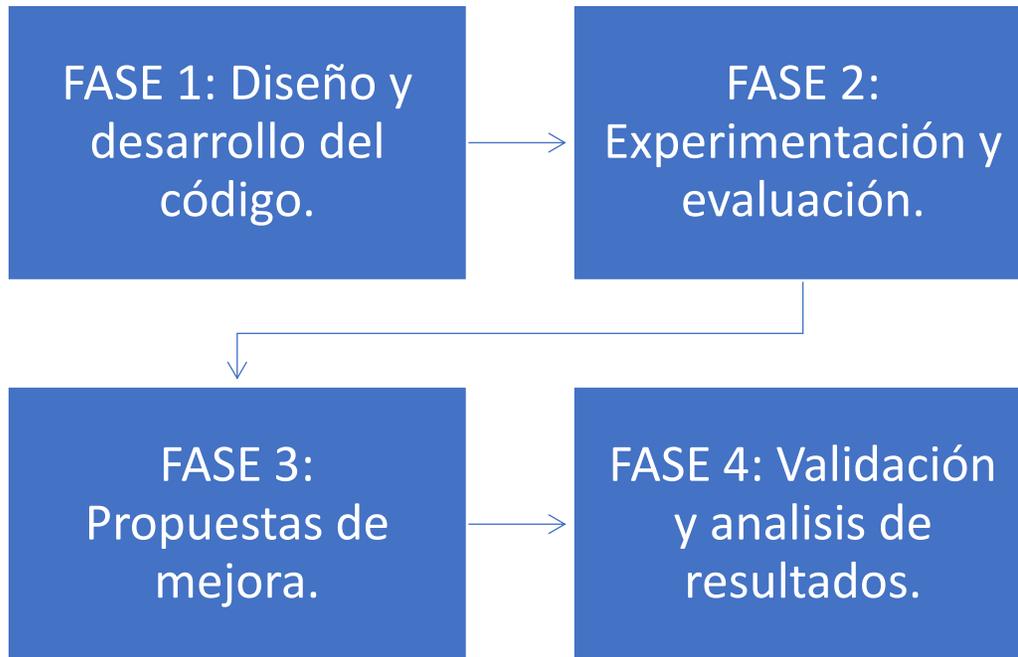
Con el fin de recopilar información acerca de los materiales y equipos a utilizar para identificar, mejorar y eliminar las deficiencias en las redes inalámbricas en el campo de las telecomunicaciones, fue fundamental realizar una investigación preliminar, ya que se tiene la recolección de datos en diferentes fuentes como sitios web, artículos, libros entre otras. El proyecto pretende que de su implementación se usen los equipos por personas que quieran mejorar su conocimiento en el campo de las telecomunicaciones más específicamente en el área de las redes inalámbricas para mejorar las redes.

### **Investigación Aplicada**

Investigación aplicada que es con el fin de poder entender realmente mediante la aplicación de los mismo cada una de las herramientas y equipos que se van a utilizar para esto se ha dividido en cuatro etapas el proyecto como lo muestra la figura 1 con el diseño de la metodología.

**Figura 1**

**Diseño de la metodología**



- ❖ **Diseño y desarrollo del sistema:** Se diseñará y desarrollará un sistema basado en MediaPipe Development que permita el análisis en tiempo real de la calidad de la señal en redes inalámbricas. Se implementarán algoritmos de percepción visual para la identificación de áreas de interferencia, puntos de acceso débiles y zonas de baja cobertura.
- ❖ **Experimentación y evaluación:** Se realizarán pruebas y mediciones en entornos de redes inalámbricas reales para evaluar el desempeño del sistema propuesto. Se recopilará datos de calidad de señal y se compararán con mediciones de referencia.
- ❖ **Propuesta de mejora:** En base a los resultados obtenidos, se propondrán estrategias y soluciones para mejorar la calidad de la señal en áreas problemáticas identificadas. Se realizarán ajustes en la ubicación de los puntos de acceso, configuración de red y/o implementación de soluciones adicionales.
- ❖ **Validación y análisis de resultados:** Se llevará a cabo una validación exhaustiva de las estrategias propuestas, realizando mediciones y pruebas en entornos reales. Se analizarán los resultados obtenidos y demostrando la efectividad de las mejores propuestas en términos de calidad de señal y experiencia de usuario.

## 1.8 Resultados esperados

- Desarrollo del código para el análisis y evaluación de la calidad de la señal: Se desarrollará un sistema basado en MediaPipe Development y análisis de percepción visual que permita identificar áreas de interferencia, puntos de accesos débiles y zonas de baja cobertura en tiempo real. El sistema estará diseñado para ser utilizado en entornos de redes inalámbricas y proporcionará información detallada sobre la calidad de la señal en diferentes ubicaciones.
- Identificación de áreas problemáticas: Mediante el uso del código desarrollado, se identificarán áreas específicas donde la calidad de la señal es deficiente. Estas áreas pueden estar afectadas por interferencias, puntos de accesos débiles o baja cobertura. Los resultados proporcionarán una visión clara de los problemas existentes en la red inalámbrica y permitirán una mejor comprensión de los factores que afectan la calidad de servicio.
- Estrategias de mejora: Con base en los resultados obtenidos, se propondrán estrategias y soluciones para mejorar la calidad de la señal en las áreas problemáticas identificadas. Estas estrategias pueden incluir ajustes en la ubicación de los puntos de acceso, mejora de la configuración de la red, implementación de dispositivos adicionales u otras acciones correctivas. Los resultados demostrarán la efectividad de estas estrategias propuestas para la mejora de la calidad de la señal en redes inalámbricas de alta frecuencia.
- Validación de las mejoras propuestas: Se llevarán a cabo pruebas y mediciones en entornos reales para validar las mejores propuestas. Estas pruebas se realizarán en diferentes ubicaciones y condiciones para evaluar la efectividad de las estrategias implementadas. Los resultados validarán la capacidad del sistema desarrollado y las estrategias propuestas para mejorar la calidad de la señal en redes inalámbricas de alta frecuencia.

## CAPÍTULO 2

### 2. MARCO TEÓRICO

#### 2.1 Introducción

En este capítulo se detallará a fondo cada uno de los términos, definiciones, softwares, herramientas físicas y digitales que fueron de gran ayuda para el entendimiento teórico y práctico del proyecto. Como temas más relevantes especificaremos MediaPipe Development que fue el software en el que nos basamos para el desarrollo de los códigos para el reconocimiento de objetos, Wifi HeatMap software para la ubicación óptima de las herramientas de telecomunicaciones tales como routers, switches, Access point, etc. Open CV que es una biblioteca de visión por computadora de código abierto, el cual también proporciona herramientas y algoritmos para procesar imágenes y videos, que en conjunto a MediaPipe Development nos ayudarán a desarrollar y entender de mejor manera los códigos de reconocimiento de objetos, la tarjeta Jetson Nano de 4Gb desarrollada por Nvidia es nuestro componente esencial de inteligencia artificial compactada en una placa y nuestra principal herramienta, el robot Rosmaster X3 que es un carro apoyado con tres tarjetas principales, una cámara de reconocimiento y un sensor laser que todo eso en conjunto abre la puerta a un sin número de opciones al momento de programación enfocada a la inteligencia artificial.

#### 2.2 Mapeo de redes

El mapeo de redes dentro del contexto de las telecomunicaciones es el proceso de contextualizar y virtualizar los elementos y datos dentro de una red. Este componente es esencial para el diseño y monitoreo constante del tráfico de paquetes que existe en nuestra red. Se usan técnicas como la arquitectura y ensamblaje de la topología, análisis del tráfico de paquetes y enrutamiento, herramientas de mapeo para la detección de dispositivos, conexiones y flujos de datos en una red. Tener la representación de la topología de manera gráfica proporciona una valiosa información para el óptimo rendimiento y detección de

vulnerabilidades en la red. Todo este análisis detallado de la infraestructura de la red contribuye a una gestión más efectiva y segura de los recursos de la red, facilitando la toma de decisiones informadas en el ámbito de las telecomunicaciones.

### 2.3 Optimización de una red

En el contexto de la optimización de redes, se puede abordar varios aspectos para una visión mucho más integral. En cuanto a los aspectos más importantes a tener en cuenta en la optimización de una red, se tienen:

#### **Topología de una red**

Explorar las diferentes topologías de una red, como anillo, malla, bus, estrella y árbol, ofrece una comprensión profunda de cómo los dispositivos están interconectados. La topología de anillo, donde cada dispositivo está conectado al siguiente formando un círculo, facilita la transmisión de datos en una única dirección, mientras que la topología de malla proporciona redundancia con múltiples rutas de conexión, mejorando la confiabilidad y resistencia a fallos.

La topología de bus implica que todos los dispositivos comparten un único canal de comunicación, lo que puede generar congestiones, pero es simple y económica. Por otro lado, la topología de estrella conecta todos los dispositivos a un punto central, facilitando la gestión, pero la caída del nodo central puede afectar toda la red.

La elección de la topología depende de los requisitos específicos de la red, considerando factores como la escalabilidad, la tolerancia a fallos, la facilidad de administración y el rendimiento. Cada topología presenta ventajas y desventajas, y la decisión impacta directamente en la eficiencia y rendimiento de la red, siendo crucial para el diseño y la implementación exitosa de sistemas de comunicación.

## **Protocolos de enrutamiento**

Explorar los protocolos de enrutamiento, como EIGRP (Enhanced Interior Gateway Routing Protocol), BGP (Border Gateway Protocol) y OSPF (Open Shortest Path First), proporciona perspectivas esenciales sobre cómo influyen en la eficiencia y escalabilidad de una red. EIGRP, comúnmente utilizado en redes empresariales, optimiza el enrutamiento dentro de un sistema autónomo, priorizando caminos más rápidos y eficientes.

BGP, un protocolo de enrutamiento interdominio, se utiliza en Internet para determinar las mejores rutas entre sistemas autónomos, brindando flexibilidad y permitiendo la toma de decisiones basada en políticas. Por otro lado, OSPF, un protocolo de enrutamiento de estado de enlace, es especialmente eficaz en entornos empresariales grandes, al calcular rutas basadas en el estado de los enlaces, priorizando la velocidad y la redundancia.

La elección del protocolo de enrutamiento depende de factores como la complejidad de la red, la necesidad de escalabilidad, la confiabilidad y los requisitos de seguridad. Cada protocolo tiene sus ventajas y desventajas, y la implementación adecuada impacta directamente en la eficiencia operativa y la capacidad de escalar la red de manera efectiva.

## **Calidad de servicio (QoS)**

Explorar la implementación de la calidad de servicio (QoS) en una red es esencial para priorizar el tráfico y asegurar un rendimiento óptimo para aplicaciones críticas, como VoIP (Voice over Internet Protocol). QoS permite asignar prioridades a diferentes tipos de datos, asegurando que el tráfico de voz, por ejemplo, tenga una mayor prioridad que el tráfico de datos menos sensible al tiempo.

La utilización de mecanismos de QoS, como la asignación de ancho de banda, la gestión de colas y la marcación de paquetes, contribuye a evitar la congestión y garantiza una entrega eficiente de datos críticos. Además, QoS puede controlar la latencia, la variabilidad

y la pérdida de paquetes, asegurando una experiencia de usuario consistente y de alta calidad.

La elección de políticas de QoS depende de los requisitos específicos de la red y las aplicaciones. La implementación adecuada de QoS no solo optimiza el rendimiento de la red, sino que también mejora la eficiencia en la transmisión de datos, lo que es fundamental en entornos donde la priorización de tráfico es esencial para el funcionamiento adecuado de aplicaciones críticas.

### **Ancho de banda y latencia**

Examinar la relación entre el ancho de banda y la latencia es esencial para comprender cómo estos factores influyen en la optimización del rendimiento de una red. El ancho de banda, que representa la cantidad de datos que pueden transmitirse en un período de tiempo, impacta directamente en la velocidad de transferencia de información.

Una mayor cantidad de ancho de banda puede mejorar la velocidad de transferencia y permitir la transmisión simultánea de más datos. Sin embargo, la latencia, que es el tiempo que tarda un paquete de datos en viajar desde el origen hasta el destino, también es crucial. Una red con alto ancho de banda, pero latencia elevada puede experimentar demoras en la entrega de datos.

La optimización del rendimiento de la red implica equilibrar eficazmente el ancho de banda y la latencia. Además, la implementación de tecnologías como el almacenamiento en caché, la compresión de datos y la priorización de tráfico mediante QoS contribuyen a una utilización más eficiente del ancho de banda, mejorando la velocidad de transmisión y minimizando la latencia. En conjunto, estos factores son fundamentales para lograr un rendimiento óptimo en entornos de red.

### **Optimización de tráfico**

Examinar las técnicas para la optimización del tráfico en una red revela estrategias fundamentales para mejorar la eficiencia y el rendimiento. La compresión de datos, al reducir el tamaño de los archivos antes de la transmisión, minimiza la carga en la red, acelerando la transferencia de información y aprovechando mejor el ancho de banda disponible.

La deduplicación, al eliminar duplicados de datos antes de su transmisión, optimiza el uso del ancho de banda y reduce la redundancia, especialmente valioso en entornos donde la eficiencia es crucial. La optimización WAN (Wide Area Network) utiliza técnicas como la aceleración de protocolos y la gestión eficiente del tráfico para mejorar el rendimiento en redes dispersas geográficamente, reduciendo la latencia y aumentando la velocidad de transmisión.

Estas técnicas trabajan en conjunto para maximizar la eficiencia del tráfico, minimizando los tiempos de espera y mejorando la respuesta general de la red. La implementación cuidadosa de estas estrategias de optimización no solo optimiza la utilización del ancho de banda, sino que también contribuye a una experiencia de usuario más rápida y fluida, esencial en entornos donde la velocidad y la eficiencia son prioritarias.

## **Seguridad de red**

Considerar las medidas de seguridad esenciales directamente vinculadas al rendimiento de una red, como firewalls, sistemas de detección y prevención de intrusiones (IDS/IPS) y la encriptación de datos, destaca la importancia de equilibrar la seguridad sin sacrificar el rendimiento. Los firewalls, al filtrar el tráfico no autorizado, garantizan la integridad de la red, pero una configuración inadecuada puede generar cuellos de botella. Los sistemas IDS/IPS, al monitorear y prevenir intrusiones, pueden afectar la latencia, aunque su configuración precisa puede minimizar este impacto.

La encriptación de datos, fundamental para la privacidad y seguridad, puede ralentizar la velocidad de transmisión, especialmente en redes grandes. Implementar algoritmos de encriptación eficientes y utilizar dispositivos de hardware dedicados puede mitigar este

impacto. Además, la segmentación de red y la aplicación de políticas de acceso garantizan que solo usuarios autorizados accedan a recursos específicos, fortaleciendo la seguridad sin sacrificar el rendimiento.

El equilibrio entre seguridad y rendimiento implica una cuidadosa planificación, configuración y monitoreo constante. La adopción de tecnologías de seguridad avanzadas y la actualización regular de políticas de seguridad son esenciales para mantener una red segura sin comprometer su eficiencia operativa.

## **Escalabilidad**

Abordar estrategias para el diseño de redes escalables implica considerar cuidadosamente la capacidad de adaptarse y crecer con eficacia frente al aumento del número de dispositivos y la demanda de tráfico. La segmentación de red, al dividir la infraestructura en segmentos más pequeños, facilita la gestión y permite escalabilidad al agregar nuevos segmentos según sea necesario. La implementación de tecnologías de enrutamiento dinámico, como OSPF o EIGRP, facilita la expansión de la red de manera eficiente, adaptándose automáticamente a cambios en la topología.

El uso de tecnologías de virtualización, como VLANs y máquinas virtuales, ofrece flexibilidad al separar lógicamente recursos y servicios, permitiendo una expansión más sencilla. La adopción de arquitecturas de nube permite escalar recursos de manera elástica según las necesidades, optimizando la infraestructura de red para cargas variables.

Además, la planificación de direcciones IP y la implementación de IPv6 brindan una amplia gama de direcciones disponibles, evitando limitaciones en el crecimiento de la red. La monitorización continua y la actualización proactiva de hardware y software son cruciales para anticipar y abordar cuellos de botella antes de que afecten el rendimiento.

En resumen, el diseño de redes escalables requiere una combinación de segmentación efectiva, enrutamiento dinámico, virtualización y tecnologías en la nube, junto con una planificación cuidadosa y una gestión proactiva para asegurar una expansión suave y

eficiente a medida que la red evoluciona.

### **Monitoreo y gestión**

Explora herramientas y prácticas para el monitoreo continuo y la gestión proactiva de la red para identificar y resolver problemas antes de que el rendimiento de la red se vea afectado.

#### 2.4 Zonas muertas de una red inalámbrica

Las zonas muertas dentro de una red o también llamadas zonas sin cobertura o puntos muertos, representan áreas dentro de una infraestructura inalámbrica es débil o nula. Estas zonas muertas aparecen debido a varios factores y plantean un gran desafío de manera significativa en términos de rendimiento y conectividad.

- **Interferencia electromagnética**

Examinar como la presencia de otros dispositivos electrónicos, como dispositivos bluetooth, microondas, pueden generar cierta interferencia en la calidad de la señal Wi – Fi.

- **Características del entorno**

El análisis de la arquitectura de los edificios, casas, departamentos, entre otros. La disposición de las paredes y presencia de ciertos elementos metálicos pueden influir en la propagación de las ondas Wi – Fi y crear zonas de baja cobertura o cero coberturas.

- **Rango y potencia de transmisión**

Considerar el rango y la potencia de transmisión de los puntos de acceso inalámbricos

afectan la cobertura y la presencia de zonas muertas.

- **Protocolos y estándares Wi – Fi**

Examinar como los protocolos y estándares wi – fi como la 802.11ac o 802.11ax, pueden ofrecer mejoras en el rendimiento y capacidad de penetración de las señales.

- **Sistemas Mesh y roaming**

Analizar como los sistemas mesh que son redes con topología malla y las capacidades del roaming mejoran la cobertura y la continuidad del servicio al permitir que los dispositivos se conecten de manera fluida a diferentes puntos de acceso.

- **Herramientas de diagnostico**

Explorar las herramientas y técnicas de diagnóstico que ayudan a identificar y gestionar zonas muertas, como software de análisis de espectro y herramientas de mediación para el nivel de la señal.

## 2.5 Inteligencia artificial

La inteligencia artificial representa un campo de estudio y desarrollo de sistemas informáticos que buscan simular la inteligencia humana. Este ámbito abarca una variedad de técnicas y enfoques, desde algoritmos de aprendizaje automático hasta lo que son redes neuronales y procesamiento del lenguaje natural. La IA tiene el enorme potencial de transformar numerosos sectores desde la atención medica hasta la gestión de recursos, impulsando avances significativos en la automatización y la toma de decisiones basadas en datos.

### 2.5.1 Aprendizaje automático

En el corazón de la inteligencia artificial se encuentra el aprendizaje automático, un paradigma que permite a los sistemas mejorar su rendimiento a través de la experiencia. Por otro lado, tenemos a las redes neuronales, inspiradas en la estructura del cerebro humano, han demostrado ser eficaces en tareas complejas como el reconocimiento de patrones y toma de decisiones. La capacidad de las redes neuronales para aprender de grandes conjuntos de datos ha llevado a avances notables en campos como la visión por computadora y el procesamiento de voz.

### 2.5.2 Aplicaciones prácticas de la IA

La IA ha encontrado aplicaciones prácticas en una variedad de industrias. En el sector de la salud, por ejemplo, se ha utilizado para el diagnóstico médico y la personalización de tratamientos. En el ámbito financiero los algoritmos de IA han mejorado la detección de fraudes y optimizado estrategias comerciales. Estos ejemplos ilustran como la IA está impactando positivamente en la eficiencia y la precisión en diversas áreas, impulsando a la innovación y la mejora continua.

### 2.5.3 Aplicaciones de la IA en el campo de las telecomunicaciones

Dentro del campo de las telecomunicaciones la IA ha marcado un hito en la evolución de las redes y servicios de comunicación. La IA aporta capacidades avanzadas en la automatización, análisis de datos y toma de decisiones, permitiendo una gestión más eficiente y dinámica de las redes. Desde la optimización de la calidad de servicio hasta la detección proactiva de fallos, la IA redefine la forma en la que operan y evolucionan la infraestructura de las telecomunicaciones.

#### - Aplicaciones prácticas de las telecomunicaciones

En el ámbito de las telecomunicaciones, la IA se ha convertido en un facilitador clave para mejorar la experiencia del usuario y la eficiencia operativa. Los sistemas de recomendación basados en IA optimizan las ofertas de servicios según los patrones de uso, mientras que

en los algoritmos de aprendizaje automático ajustan dinámicamente los parámetros de red para garantizar un rendimiento óptimo. Además, la detección y mitigación automática de amenazas de seguridad, respaldadas por la IA, fortalecen la seguridad informática en las comunicaciones, asegurando la integridad y confidencialidad de los paquetes y datos transmitidos. La IA en las telecomunicaciones no solo mejoran la eficiencia operativa, sino que también sienta las bases para servicios más personalizados y seguros en la era digital.

#### 2.5.4 El futuro de la inteligencia artificial

A medida que la indagación de la IA avanza a pasos agigantados, se vislumbra un futuro brillante. La combinación de IA con otras tecnologías emergentes, como el internet de las cosas (IoT) y la computación cuántica, promete transformaciones aun mas profundas en la forma en que interactuamos con la tecnología y abordamos los desafíos globales. La continua exploración y comprensión de la IA son esenciales para aprovechar todo su potencial de manera ética y responsable.

#### 2.5.5 Redes neuronales

Inspiradas en la complejidad de las redes neuronales del cerebro humano, constituyen una piedra angular en el ámbito de reconocimiento de objetos mediante IA. Estas redes, especialmente las redes neuronales convoluciones (CNN), han demostrado una capacidad excepcional para aprender patrones complejos en datos visuales, desempeñando un papel crucial en el reconocimiento y clasificación de objetos. Al simular el proceso de aprendizaje y adaptación de las neuronas biológicas, las redes neuronales artificiales aprovechan capas interconectadas para extraer características jerárquicas y discernir información valiosa de las imágenes. Este enfoque ha llevado a avances significativos en la precisión y eficacia del reconocimiento de objetos, permitiendo en los sistemas de IA comprender y contextualizar entornos visuales con una profundidad impresionante. La aplicación de redes neuronales en el reconocimiento de objetos abarca diversas áreas, desde la vigilancia hasta la atención médica, impulsando la frontera de lo que es posible lograr

con la IA en la interpretación de datos visuales.

- Visión computarizada

La visión computarizada representa la capacidad de las computadoras para extraer información y conocimientos de imágenes y videos. A través de las redes neuronales, los sistemas informáticos pueden discernir e identificar imágenes de manera análoga a la capacidad humana. Las aplicaciones de la visión computarizada son diversas e incluyen:

**Reconocimiento visual en vehículos autónomos:** Permite la identificación de señales de tráfico y otros usuarios en la vía.

**Moderación de contenido:** Automatiza la eliminación de contenidos inseguros o inapropiados en archivos de imágenes y videos.

**Reconocimiento facial:** Identifica rostros y atributos como ojos abiertos, gafas y vello facial.

**Etiquetado de imágenes:** Identifica elementos específicos en las imágenes, como logotipos de marcas, prendas de vestir y equipo de seguridad.

- Reconocimiento de voz

Mediante redes neuronales, se logra analizar el habla humana, superando variaciones en patrones de habla, tono, idioma y acento. Asistentes virtuales como Amazon Alexa y software de transcripción automática utilizan esta tecnología para:

**Asistir a agentes en centros de llamadas:** Clasifica llamadas automáticamente y brinda apoyo.

**Conversión de conversaciones clínicas en documentación en tiempo real:** Facilita la creación de registros médicos.

**Subtitulado preciso de videos y grabaciones de reuniones:** Mejora la accesibilidad al contenido.

- Procesamiento de lenguaje natural (PLN)

El PLN, facilitado por redes neuronales, capacita a las computadoras para procesar texto natural creado por humanos. Se aplica en diversas situaciones, como:

***Chatbots y agentes virtuales:*** Proporcionan respuestas automáticas y asistencia.

***Organización automática de datos escritos:*** Clasifica información en documentos extensos, como correos electrónicos y formularios.

***Análisis de inteligencia empresarial:*** Evalúa documentos largos, como correos electrónicos y formularios, para extraer información relevante.

***Indexación de frases clave:*** Detecta sentimientos, como comentarios positivos y negativos en redes sociales.

***Resumen de documentos:*** Sintetiza información y genera resúmenes para un tema específico.

- Motores de recomendaciones

Las redes neuronales rastrean la actividad del usuario para ofrecer recomendaciones personalizadas y descubrir nuevos productos o servicios de interés. Ejemplos incluyen el servicio de etiquetado inteligente de productos (IPT) de Curalate, que utiliza redes neuronales para automatizar la selección y recomendación de productos relevantes en base a la actividad del usuario en redes sociales. Este enfoque simplifica la experiencia de compra al permitir a los consumidores encontrar productos fácilmente a través del etiquetado automático de productos.

- Funcionamiento

La arquitectura de las redes neuronales encuentra su inspiración en la complejidad del cerebro humano. En el cerebro, las neuronas forman una red intrincada y altamente interconectada, transmitiendo señales eléctricas para facilitar el procesamiento de información. Análogamente, una red neuronal artificial se compone de neuronas

artificiales, representadas como nodos de software, que colaboran para abordar problemas. Estas neuronas artificiales, dispuestas en capas, forman programas de software o algoritmos que emplean sistemas informáticos para realizar cálculos matemáticos.

- Estructura de una red neuronal sencilla

Una red neuronal básica consiste en neuronas artificiales conectadas en tres capas:

*Capa de entrada:* Aquí, la información externa ingresa a la red neuronal artificial. Los nodos de entrada procesan, analizan o clasifican los datos y los transmiten a la capa siguiente.

*Capa oculta:* Estas capas reciben entrada de la capa de entrada u otras capas ocultas. Pueden ser numerosas y cada capa oculta analiza y procesa la salida de la capa anterior antes de enviarla a la siguiente capa.

*Capa de salida:* Proporciona el resultado final del procesamiento de datos de la red neuronal artificial. Puede constar de uno o varios nodos. Por ejemplo, en un problema de clasificación binaria, la capa de salida tendría un nodo que produce un resultado de 1 o 0. En un problema de clasificación multiclase, la capa de salida podría tener múltiples nodos.

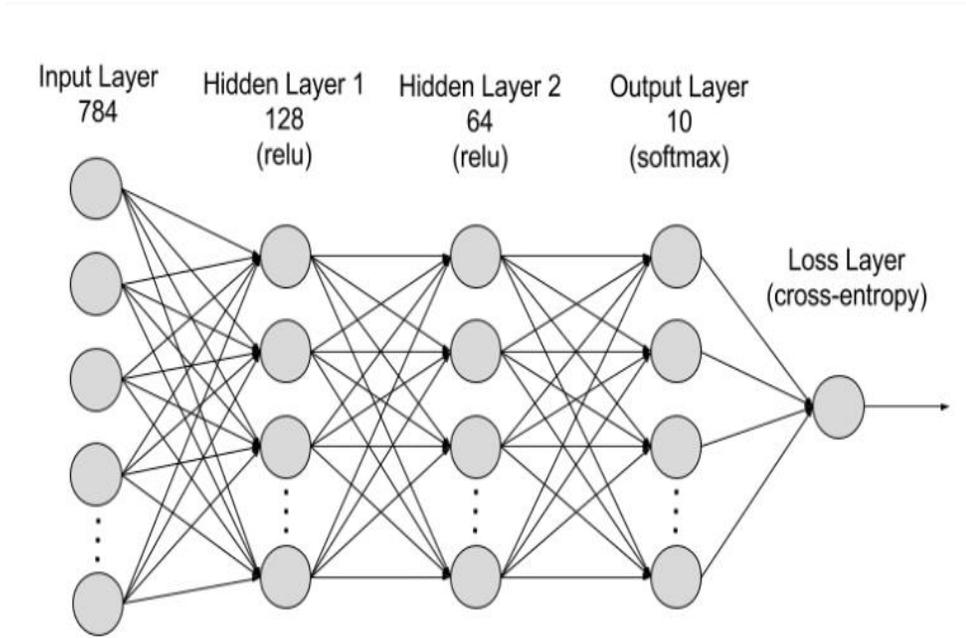
- Estructura de una red neuronal profunda

Las redes neuronales profundas, también conocidas como redes de aprendizaje profundo, poseen numerosas capas ocultas con millones de neuronas artificiales interconectadas. Las conexiones entre nodos están representadas por pesos, siendo positivos si un nodo estimula a otro y negativos si lo inhibe. Los nodos con pesos más altos ejercen una mayor influencia sobre otros nodos. Aunque teóricamente las redes neuronales profundas pueden asignar cualquier entrada a cualquier salida, requieren un extenso entrenamiento con millones de ejemplos de datos, en contraste con las redes más simples que podrían necesitar cientos o miles. En la figura 2 se refiere a la fase en la que se crea y estructura la arquitectura de una red neuronal, este plano implica decisiones fundamentales tales como se organizan y

conectan las neuronas dentro de la red para lograr el rendimiento específico en una tarea asignada.

**Figura 2**

**Plano de diseño de redes neuronales**



- Entrenamiento de redes neuronales

El proceso de entrenamiento de redes neuronales implica enseñar a una red neuronal a llevar a cabo una tarea específica. En esencia, estas redes aprenden al procesar conjuntos extensos de datos, ya sea etiquetados o sin etiquetar, y utilizan estos ejemplos para mejorar su capacidad de procesar entradas desconocidas con mayor precisión.

**En el aprendizaje supervisado:** Los científicos de datos suministran a las redes neuronales conjuntos de datos etiquetados, que contienen las respuestas correctas de antemano. Por ejemplo, en el entrenamiento de una red de aprendizaje profundo para el reconocimiento facial, se procesan inicialmente cientos de miles de imágenes de rostros humanos, cada una asociada con términos que describen aspectos como el origen étnico, el país o la emoción.

A lo largo del entrenamiento, la red neuronal acumula gradualmente conocimiento a partir de estos conjuntos de datos que contienen respuestas previamente conocidas. Una vez completado el entrenamiento, la red comienza a hacer predicciones sobre atributos como el origen étnico o la emoción en nuevas imágenes de rostros humanos que nunca antes ha analizado.

- Aprendizaje profundo en redes neuronales

La inteligencia artificial (IA) constituye el ámbito de las ciencias de la computación dedicado a explorar métodos que permitan a las máquinas ejecutar tareas que demandan inteligencia humana. Dentro de la inteligencia artificial, el machine learning se destaca como una técnica que otorga a las computadoras acceso a extensos conjuntos de datos, enseñándoles a aprender de esta información. El software de machine learning identifica patrones en los datos existentes y los aplica a datos nuevos, capacitándose para tomar decisiones informadas. En este contexto, el aprendizaje profundo surge como un subconjunto del machine learning que emplea redes de aprendizaje profundo para procesar datos.

- Comparación entre machine learning y aprendizaje profundo

Los métodos convencionales de machine learning implican la intervención humana para configurar el software de manera efectiva. Un científico de datos selecciona manualmente el conjunto de características relevantes que el software debe analizar, lo cual conlleva limitaciones en la capacidad del software y complicaciones en su desarrollo y gestión.

En contraste, en el aprendizaje profundo, el científico de datos proporciona únicamente datos sin procesar al software. La red de aprendizaje profundo identifica características de manera autónoma y aprende de forma más independiente. Puede abordar conjuntos de datos no estructurados, como documentos de texto, identificar atributos prioritarios y resolver

problemas más complejos.

Por ejemplo, al entrenar un software de machine learning para reconocer imágenes de mascotas, se seguirían estos pasos:

- Seleccionar y etiquetar manualmente miles de imágenes de mascotas, incluyendo gatos, perros, caballos, hámsters, loros, etc.
- Indicar al software de machine learning las características clave que debe analizar para identificar la imagen, como contar patas, verificar la forma de los ojos, orejas, cola, pelo, etc.
- Evaluar y ajustar manualmente los conjuntos de datos etiquetados para mejorar la precisión del software, corrigiendo posibles sesgos o desequilibrios.
- En cambio, en el aprendizaje profundo, las redes neuronales procesarían automáticamente todas las imágenes, determinando por sí mismas que deben analizar primero el número de patas y la forma del rostro, priorizando ciertos atributos para identificar correctamente el animal en la imagen.

## 2.6 Lenguajes de programación

Los lenguajes de programación son herramientas fundamentales en el ámbito de la informática, permitiendo a los desarrolladores comunicarse con las computadoras para crear software y soluciones tecnológicas. Estos lenguajes actúan como intermediarios, traduciendo las intenciones del programador en instrucciones comprensibles para la máquina.

Existen numerosos lenguajes de programación, cada uno diseñado con propósitos específicos y ventajas particulares. Algunos, como Python y JavaScript, destacan por su simplicidad y versatilidad, facilitando el desarrollo rápido de aplicaciones. Otros, como C++ y Java, se especializan en el rendimiento eficiente y se utilizan en sistemas embebidos, desarrollo de videojuegos y software de alto rendimiento.

La diversidad de lenguajes de programación permite a los desarrolladores seleccionar la herramienta más adecuada para sus objetivos, considerando factores como la eficiencia, la facilidad de mantenimiento y la escalabilidad. Además, el continuo avance tecnológico conlleva a la evolución constante de nuevos lenguajes que se adaptan a las demandas cambiantes de la industria. En última instancia, estos lenguajes son los cimientos sobre los cuales se construye la innovación en la era digital, impulsando el desarrollo de aplicaciones, sistemas y tecnologías que moldean nuestro entorno informático.

### 2.6.1 Python 3

Python 3 es la versión más reciente de Python, un lenguaje de programación de alto nivel ampliamente utilizado. Introducido en 2008, Python 3 presenta mejoras significativas sobre versiones anteriores, centrándose en la claridad del código y la eficiencia. Destaca por su sintaxis sencilla y legible, facilitando el desarrollo de aplicaciones, scripts y proyectos complejos. Python 3 incluye características como la gestión automática de memoria, soporte para programación orientada a objetos y una extensa biblioteca estándar que abarca desde manipulación de archivos hasta networking. Su versatilidad lo convierte en una opción popular para diversas áreas, como desarrollo web, inteligencia artificial, análisis de datos y automatización. Su filosofía, expresada en el "Zen de Python," promueve la elegancia y simplicidad en el código, contribuyendo a su atractivo y adopción generalizada en la comunidad de desarrollo de software. En la figura 3 muestra el logo que representa la última versión de Python, que es Python 3.

**Figura 3**

**Logo PYTHON 3**



## 2.7 Real VNC viewer

VNC Viewer es una aplicación de software que facilita el control remoto de sistemas informáticos. Esta herramienta, compatible con Python 3, permite a los usuarios visualizar y gestionar escritorios de manera remota a través de una conexión VNC (Virtual Network Computing). Diseñado para ser multiplataforma, VNC Viewer es compatible con diversos sistemas operativos, incluyendo Windows, macOS y Linux.

El software brinda una interfaz intuitiva que posibilita la conexión a máquinas remotas mediante el protocolo VNC. Los usuarios pueden acceder y controlar escritorios a distancia, facilitando la administración remota de sistemas y la colaboración en entornos de trabajo distribuidos.

Entre las características destacadas de VNC Viewer se incluyen opciones de seguridad

avanzadas, como la encriptación de la conexión, garantizando la protección de los datos transmitidos durante las sesiones remotas. Asimismo, proporciona herramientas para ajustar la calidad y velocidad de la transmisión, adaptándose a diversas condiciones de red.

En resumen, VNC Viewer ofrece una solución eficiente y segura para el control remoto de sistemas, siendo una herramienta valiosa en entornos donde se requiere acceso remoto a escritorios y la administración eficaz de recursos informáticos. En la figura 4 muestra el logo representativo del software Real VNC.

**Figura 4**

**Logo REAL VNC**



---

## 2.8 VMware Workstation pro

VMware Workstation Pro es una aplicación avanzada de virtualización que permite a los usuarios crear y gestionar máquinas virtuales en sus sistemas locales. Diseñado para entornos profesionales, este software es compatible con sistemas operativos Windows y Linux, y se destaca por su versatilidad y potencia en la ejecución de múltiples sistemas operativos simultáneamente.

La interfaz intuitiva de VMware Workstation Pro facilita la creación y gestión de máquinas virtuales, permitiendo a los usuarios personalizar configuraciones como asignación de recursos, redes y almacenamiento. La herramienta ofrece una variedad de funciones avanzadas, como instantáneas (snapshots) para la captura del estado de una máquina virtual en un momento específico y clonación para la duplicación de máquinas virtuales existentes. VMware Workstation Pro proporciona un entorno de pruebas seguro para desarrolladores y profesionales de TI, permitiendo la simulación de diversos entornos operativos en una sola máquina física. Además, ofrece características de red avanzadas, como la posibilidad de crear entornos virtuales complejos y personalizados.

En términos de seguridad, el software incluye funciones como la protección mediante contraseña y la encriptación de máquinas virtuales para garantizar la integridad de los datos. Su compatibilidad con tecnologías de virtualización líderes, como Intel VT-x y AMD-V, optimiza el rendimiento de las máquinas virtuales.

En resumen, VMware Workstation Pro se presenta como una herramienta esencial para profesionales que requieren virtualización avanzada, proporcionando un entorno robusto y seguro para el desarrollo, prueba y ejecución de múltiples sistemas operativos de forma simultánea en una única máquina física. En la figura 5 muestra el logo representativo de VMware.

**Figura 5**

**Logo VMWare**



## 2.9 WiFi Heatmap

WiFi Heatmap es una herramienta de software especializada en la visualización y análisis de la cobertura de redes inalámbricas. Diseñado para entornos donde la calidad de la señal Wi-Fi es crítica, este software permite a los usuarios mapear la intensidad de la señal y la calidad de la conexión en un área específica.

La interfaz de WiFi Heatmap facilita la creación de mapas de calor que representan visualmente la fuerza y la distribución de las señales Wi-Fi. Este tipo de representación permite identificar áreas con cobertura deficiente, interferencias o posibles problemas de conexión. La herramienta es especialmente útil en entornos empresariales, hogares grandes o espacios públicos donde la eficiencia de la red inalámbrica es esencial.

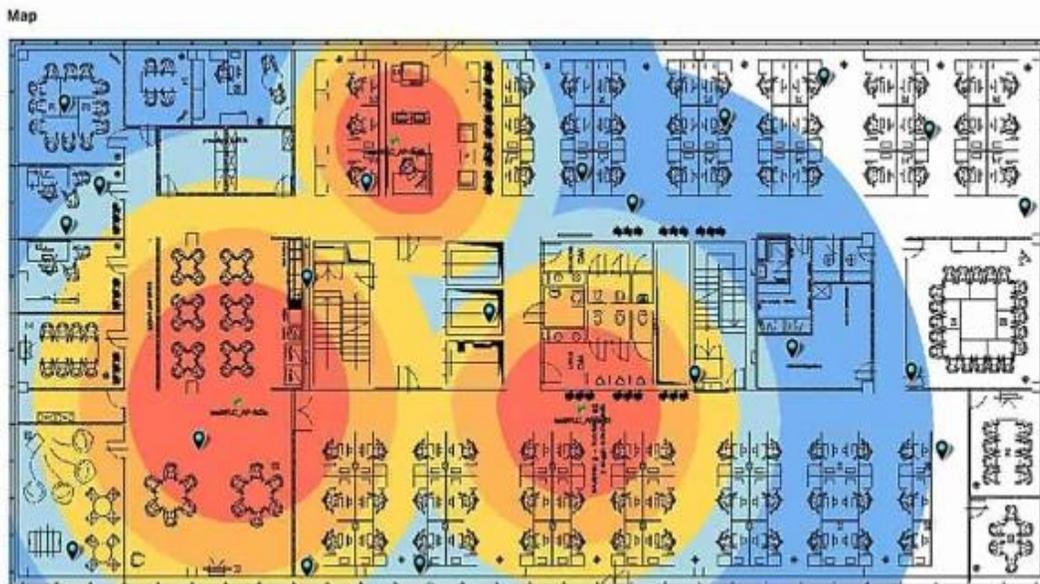
Además de la generación de mapas de calor, WiFi Heatmap proporciona funciones analíticas que permiten a los usuarios evaluar la calidad del servicio, la velocidad de

conexión y otros parámetros clave. Algunas versiones del software también ofrecen la capacidad de simular cambios en la infraestructura de red para optimizar la disposición de puntos de acceso y mejorar la cobertura.

En términos de seguridad, WiFi Heatmap no solo ayuda a identificar áreas con posibles problemas de conexión, sino que también puede ayudar a detectar posibles puntos de acceso no autorizados o interferencias externas.

En resumen, WiFi Heatmap se presenta como una herramienta valiosa para profesionales de redes y administradores de sistemas, ofreciendo una visualización detallada y análisis de la cobertura Wi-Fi en entornos diversos, contribuyendo así a mejorar la eficiencia y confiabilidad de las redes inalámbricas. En la figura 6 muestra una imagen de referencia sobre el mapa de cobertura hecho por el software de WiFi Heatmap.

**Figura 6**  
**Mapeo de cobertura de red en WiFi Heatmap.**



## 2.10 MediaPipe Development

MediaPipe Development es un conjunto integral de herramientas y bibliotecas desarrolladas por Google que se centran en el procesamiento de medios visuales y la detección de movimientos en tiempo real. Este marco de desarrollo proporciona a los desarrolladores las herramientas necesarias para construir aplicaciones de visión por computadora y seguimiento de movimientos de manera eficiente.

La interfaz de programación de aplicaciones (API) de MediaPipe simplifica la implementación de funciones como el reconocimiento facial, la detección de objetos y el seguimiento de gestos en aplicaciones y servicios. Este marco es conocido por su flexibilidad y capacidad para ejecutar algoritmos de procesamiento de imágenes en una variedad de plataformas, incluyendo dispositivos móviles y sistemas de escritorio.

MediaPipe Development incluye módulos pre - entrenados que facilitan tareas específicas, como el seguimiento de manos, la detección de pose y el reconocimiento de objetos, acelerando el desarrollo de aplicaciones de visión por computadora. Además, ofrece la posibilidad de personalizar y adaptar estos módulos según las necesidades específicas de los proyectos.

En términos de aplicaciones prácticas, MediaPipe Development se utiliza en campos como la realidad aumentada, la interacción persona-computadora y el análisis de movimientos. Su enfoque en el desarrollo de aplicaciones en tiempo real lo hace especialmente valioso para casos de uso que requieren respuestas instantáneas a eventos visuales.

En resumen, MediaPipe Development emerge como una herramienta esencial para los desarrolladores que buscan integrar capacidades de procesamiento visual y seguimiento de movimientos en sus aplicaciones, brindando una solución robusta y versátil para la creación de experiencias interactivas y aplicaciones innovadoras. En la figura 7 se tiene una imagen sobre la interfaz de MediaPipe Development.

**Figura 7**

**Imagen de referencia MediaPipe Development.**



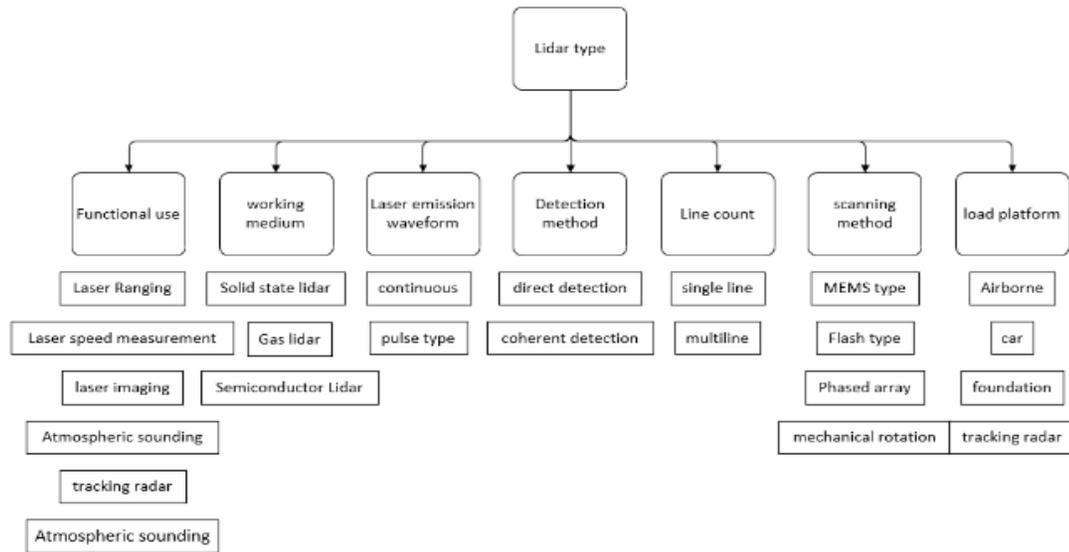
### 2.11 Lidar S1 para Rosmaster X3

El RPLIDAR A1 es una solución de escáner láser 2D de 360 grados de bajo costo desarrollada por SLAMTEC. Este sistema puede realizar un escaneo de 360 grados en un rango de 6 metros. Los datos de la nube de puntos 2D generados se pueden utilizar en mapeo, localización y modelado de objetos y entornos.

La frecuencia de escaneo del RPLIDAR A1 alcanza los 5.5 Hz al muestrear 360 puntos en cada ciclo, y se puede configurar hasta un máximo de 10 Hz. El RPLIDAR A1 es esencialmente un sistema de medición por triangulación láser y puede funcionar de manera excelente en diversos entornos, ya sea en interiores o exteriores, sin verse afectado por la luz solar. En la figura 8 se tiene el diagrama completo de como funciona a exactitud el Lidar Laser.

**Figura 8**

**Diagrama de funcionamiento LIDAR laser.**



- Laser

El láser es el mecanismo emisor de láser del lidar. Durante su funcionamiento, se enciende por impulsos. El Lidar RPLIDAR serie A3 de Silan Technology se enciende y apaga 16.000 veces por segundo. A continuación, se muestra una tabla con las especificaciones de manera general del láser usado por el RPLIDAR.

**Tabla 1**  
**Especificaciones generales del Laser.**

Item	Unit	Min	Typical	Max	Comments
Laser wavelength	Nanometer(nm)	775	785	795	Infrared Light Band
Laser power	Milliwatt (mW)	TBD	3	5	Peak power
Pulse length	Microsecond(us)	TBD	110	300	

- Receptor

Después de que el láser emitido irradie el obstáculo, la luz reflejada se concentrará en el receptor a través del grupo de lentes mediante la reflexión del obstáculo.

- Unidad de procesamiento de señales

La unidad de procesamiento de señales se encarga de controlar la emisión del láser y el procesamiento de la señal recibida por el receptor. A partir de ella, se calcula la información sobre la distancia del objeto objetivo.

- Mecanismo de rotación

Estos tres componentes constituyen el núcleo de la medición. El mecanismo de rotación se encarga de hacer girar los componentes centrales mencionados a una velocidad de rotación estable, con el fin de escanear el plano y generar información del plano en tiempo real. En la figura 9 muestra una imagen de referencia el RPLIDAR con el señalamiento de todos sus componentes y su mecanismo de rotación.

**Figura 9**

**Descripción mecanismo de rotación LIDAR laser**



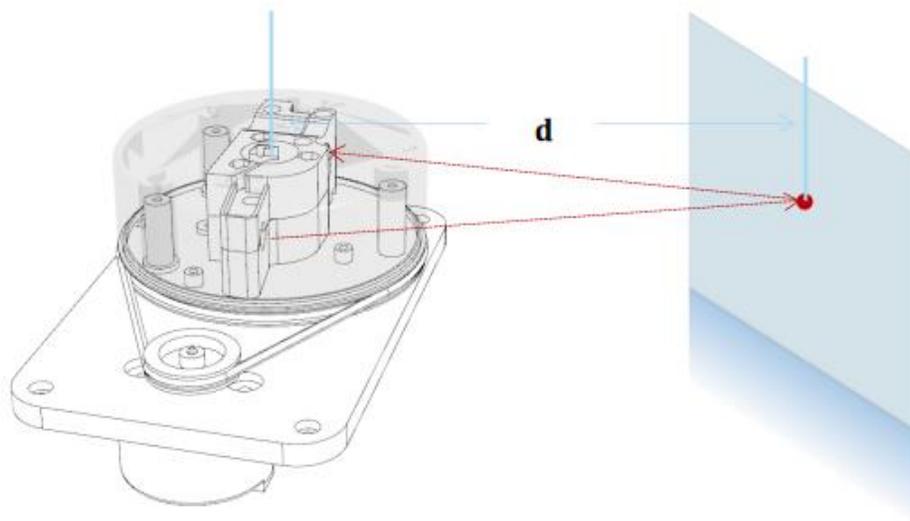
- Mecanismo general

RPLIDAR se basa en el principio de medición por triangulación láser y utiliza hardware de adquisición y procesamiento de visión de alta velocidad desarrollado por SLAMTEC. El sistema mide datos de distancia en más de 2000 veces por segundo y ofrece una salida de distancia de alta resolución (<1% de la distancia).

RPLIDAR emite una señal láser infrarrojo modulada y esta señal láser es luego reflejada por el objeto a detectar. La señal de retorno es muestreada por el sistema de adquisición de visión en RPLIDAR A1, y el DSP incorporado en RPLIDAR A1 comienza a procesar los datos de muestra y a emitir valores de distancia y ángulo entre el objeto y RPLIDAR A1 a través de la interfaz de comunicación. En la figura 10 muestra el mecanismo de triangulación laser y en la figura 11 muestra el mapeo y el uso del lidar dentro de entornos controlados.

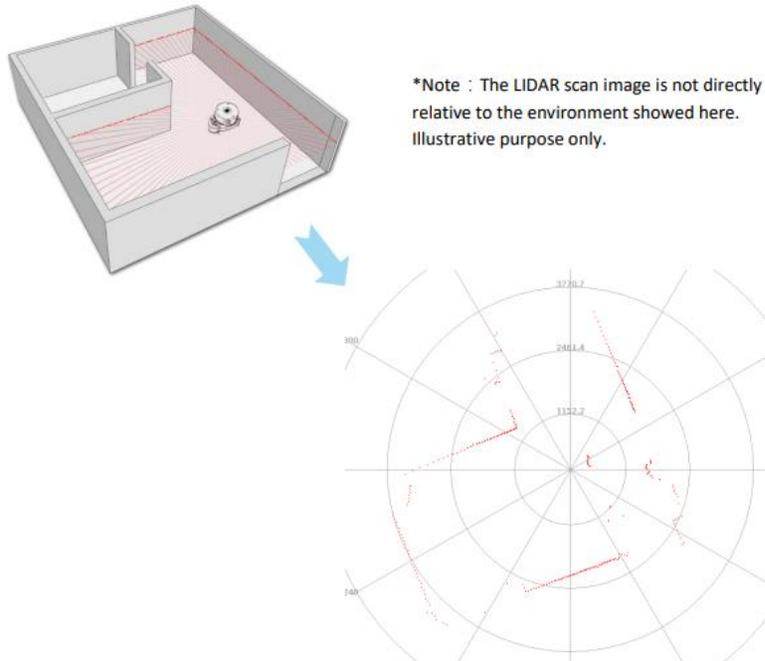
**Figura 10**

**Mecanismo de triangulación laser.**



## Figura 11

### Mapeo de entorno mediante LIDAR laser.



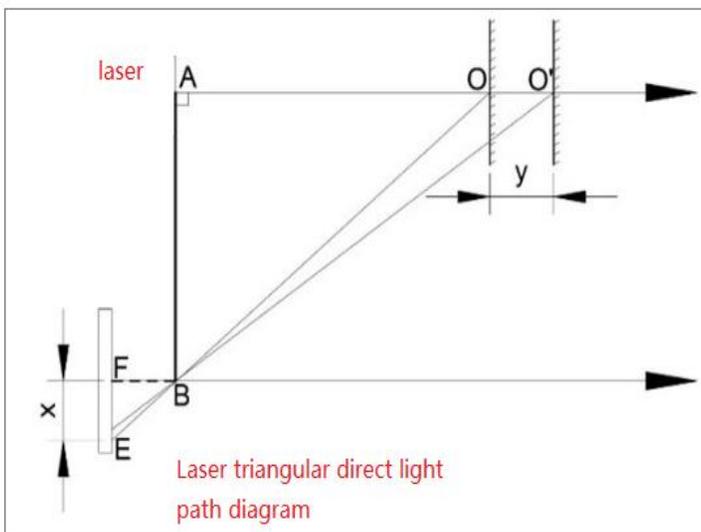
#### - Método de triangulación

El método de triangulación láser utiliza principalmente un haz de luz láser para irradiar el objeto medido con un ángulo de incidencia determinado, el láser se refleja y dispersa en la superficie del objeto, y el láser reflejado es recogido y visualizado por una lente en otro ángulo, y el punto se visualiza en un sensor de posición CCD (dispositivo de carga acoplada, componente de acoplamiento fotosensible). Cuando el objeto medido se mueve a lo largo de la dirección del láser, el punto de luz en el sensor de posición se moverá, y su desplazamiento corresponde a la distancia en movimiento del objeto medido. Por lo tanto, la distancia entre el objeto medido y la línea de base puede calcularse a partir de la distancia de desplazamiento del punto luminoso mediante el diseño de algoritmos. valor. Dado que la luz incidente y la luz reflejada forman un triángulo, se utiliza el teorema del triángulo geométrico para el cálculo del desplazamiento del punto, por lo que el método de medición se denomina método de triangulación láser.

Según la relación angular entre el haz incidente y la normal de la superficie del objeto medido, el método de triangulación láser puede dividirse en dos tipos: de tipo oblicuo y de tipo directo. En la figura 12 muestra la triangulación laser de manera directa, mientras que en la figura 13 muestra la triangulación laser de manera oblicua.

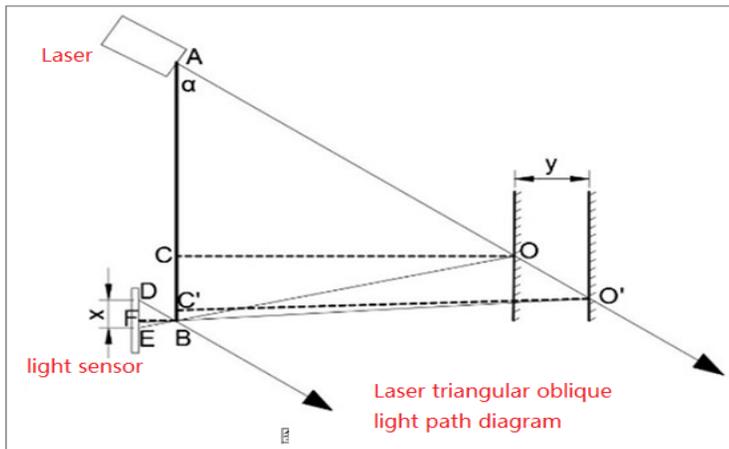
**Figura 12**

**Triangulación laser directa.**



**Figura 13**

**Triangulación laser oblicua.**



- Rendimiento de medición

El LIDAR (Light Detection and Ranging) Laser A1 es un sistema avanzado de teledetección que utiliza pulsos láser para medir distancias con precisión y generar mapas tridimensionales del entorno circundante. El rendimiento de medición del LIDAR Laser A1 destaca por su capacidad para proporcionar datos altamente precisos y detallados en tiempo real.

Este dispositivo utiliza una combinación de tecnologías de escaneo láser y algoritmos avanzados para emitir pulsos de luz hacia el entorno y medir con precisión el tiempo que tarda cada pulso en regresar, calculando así las distancias con una gran exactitud. La velocidad de muestreo y la frecuencia de repetición de los pulsos contribuyen a su rendimiento excepcional, permitiendo la captura rápida y continua de datos.

Además, el LIDAR Laser A1 destaca por su capacidad para funcionar en diversas condiciones ambientales, desde entornos urbanos hasta áreas más amplias, y proporciona una alta resolución espacial. Su capacidad para realizar mediciones precisas en tiempo real lo convierte en una herramienta valiosa para aplicaciones que van desde la cartografía topográfica y la gestión de recursos hasta la conducción autónoma y la robótica.

En resumen, el rendimiento de medición del LIDAR Laser A1 se caracteriza por su precisión, velocidad y versatilidad, ofreciendo soluciones efectivas para una variedad de aplicaciones que requieren información detallada y actualizada sobre el entorno circundante. En la tabla 2 se tiene de manera más específica los parámetros de medición del LIDAR laser. Y en la figura 14 muestra la diferencia de gráficas y curvatura del alcance del láser.

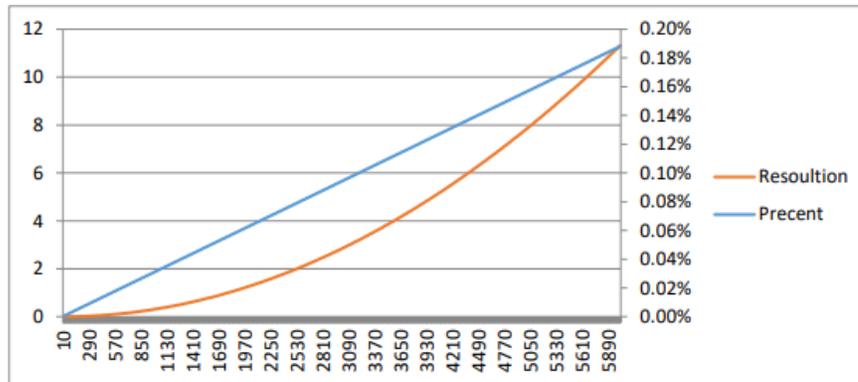
**Tabla 2****Parámetros de medición LIDAR laser.**

Item	Unit	Min	Typical	Max	Comments
Distance Range	Meter(m)	TBD	0.15 - 6	TBD	White objects
Angular Range	Degree	n/a	0-360	n/a	
Distance Resolution	mm	n/a	<0.5 <1% of the distance	n/a	<1.5 meters All distance range*
Angular Resolution	Degree	n/a	≤1	n/a	5.5Hz scan rate
Sample Duration	Millisecond(ms )	n/a	0.5	n/a	
Sample Frequency	Hz	n/a	≥2000	2010	
Scan Rate	Hz	1	5.5	10	Typical value is measured when RPLIDAR A1 takes 360 samples per scan

*La resolución del sistema de rango de triangulación cambia junto con la distancia, y el cambio teórico de resolución de RPLIDAR A1 se muestra a continuación:*

**Figura 14**

**Comparación gráfica del LIDAR laser referentes a la distancia del mismo.**

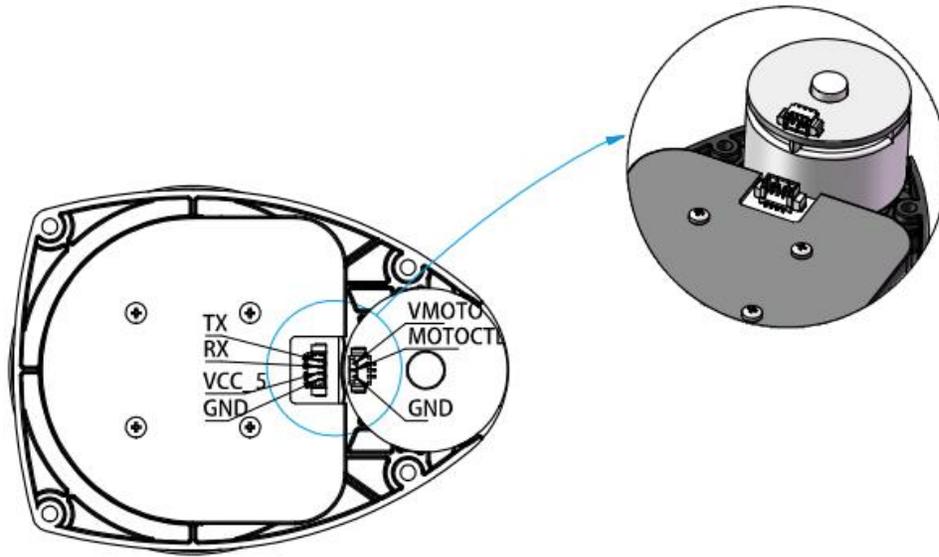


- Interfaz de comunicación

RPLIDAR A1 utiliza un puerto serie (UART) de 3.3V-TTL como interfaz de comunicación. Otras interfaces de comunicación, como USB, pueden personalizarse según los requisitos del cliente. La tabla 3 a continuación describe las especificaciones para la interfaz de puerto serie.

Para la interfaz del motor y la interfaz principal de la versión de lote, se utilizan el conector de paso horizontal PH1.25-3P y el conector de paso vertical PH1.25-4P, respectivamente. Consulte las dimensiones mecánicas en el Capítulo 5 para obtener detalles sobre las señales y los pines correspondientes. Sin embargo, el kit de desarrollo RPLIDAR utiliza el conector de paso PH2.54-7P. en la figura 15 se tiene una imagen de referencia del Lidar Laser. En la tabla 4 muestra el datasheet del núcleo del láser del Lidar.

**Figura 15**  
**LIDAR laser.**



**Tabla 3**  
**Tabla de entrada y salida de voltaje LIDAR laser.**

Item	Unit	Min	Typical	Max	Comments
Band rate	bps	-	115200	-	
Working mode	-	-	8N1	-	8n1
Output high voltage	Volt (V)	2.9	-	3.5	Logic High
Output low voltage	Volt (V)	-	-	0.4	Logic Low
Input high voltage	Volt (V)	1.6*	-	3.5	Logic High
Input low voltage	Volt (V)	-0.3	-	0.4	Logic Low

**Tabla 4****Tabla de descripción del motor y núcleo del LIDAR laser.**

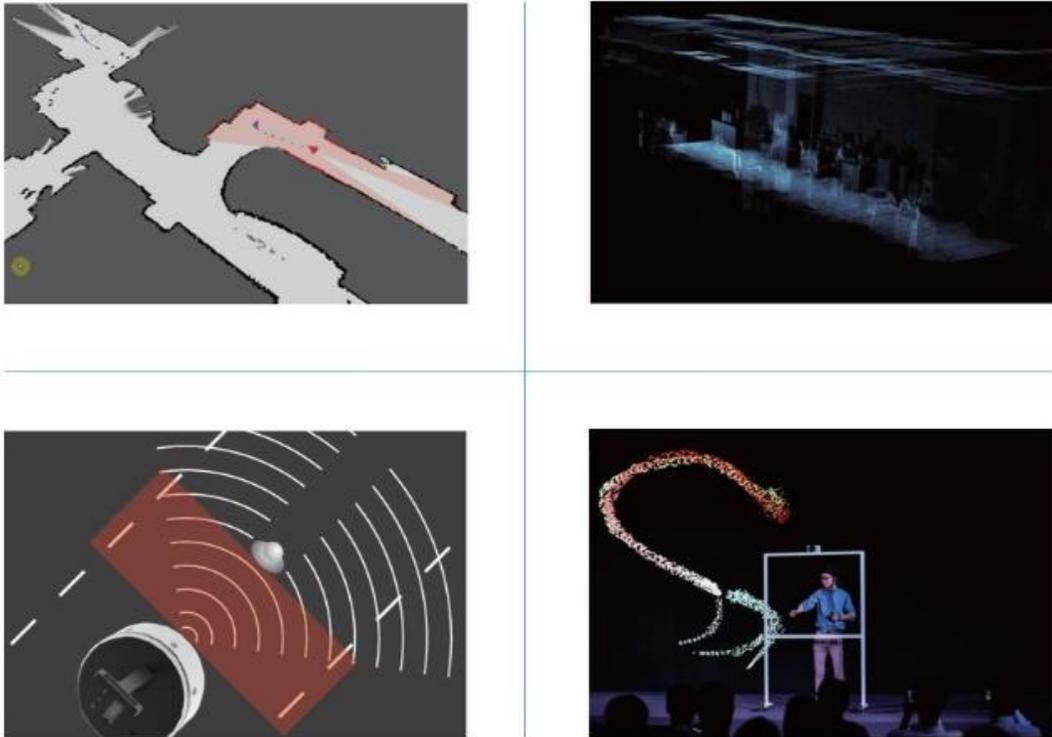
Interface	Signal Name	Type	Description	Min	Typical	Max
Motor Interface	VMOTO	Power	Power for RPLIDAR A1 Motor	-	5V	9V
	MOTOCTL	Input	Enable signal for RPLIDAR A1 Motor/PWM Control Signal	0V	-	VMOTO
	GND	Power	GND for RPLIDAR A1 Motor	-	0V	-
Core Interface	VCC_5	Power	Power for RPLIDAR A1 Range Scanner Core	4.9V	5V	6V
	TX	Output	Serial output for Range Scanner Core	0V	-	5V
	RX	Input	Serial input for Range Scanner Core	0V	-	5V
	GND	Power	GND for RPLIDAR A1 Range Scanner Core	-	0V	V5.0

- Escenarios de aplicación

Gracias a los avances de la tecnología lidar, el radio de medición, la frecuencia de alcance, la resolución de alcance y la resolución angular del lidar se han mejorado enormemente, lo que puede ayudar a diversas aplicaciones a obtener escenas más amplias e información de contorno más rica. Desempeña un papel indispensable e importante en muchos campos, como el posicionamiento autónomo y la navegación de robots, la cartografía del entorno espacial y la seguridad. En la figura 16 muestra los diferentes escenarios donde se usa el Lidar Laser.

**Figura 16**

**Escenarios de aplicación LIDAR laser.**



## 2.12 Jetson Nano 4 Gb

El dispositivo de desarrollo NVIDIA® Jetson Nano™ es un computador compacto y potente diseñado para ejecutar múltiples redes neuronales simultáneamente, brindando capacidades como clasificación de imágenes, detección de objetos, segmentación y procesamiento del lenguaje. Este sistema, que opera con tan solo 5 vatios de energía, se presenta como una plataforma amigable con un entorno Linux familiar, tutoriales accesibles y una comunidad activa de desarrolladores que comparten proyectos de código abierto. Es una herramienta ideal para el aprendizaje práctico dada su facilidad de uso y el respaldo de una comunidad comprometida. En la figura 17 se tiene una imagen de referencia de la placa de la Jetson Nano y en la figura 18 el datasheet de la Jetson Nano.

**Figura 17**

**Placa de Jetson Nano 4Gb.**



**Figura 18**

**Datasheet Jetson Nano de 4Gb.**

<b>GPU</b>	GPU con arquitectura NVIDIA Maxwell™ de 128 núcleos
<b>CPU</b>	Procesador de cuatro núcleos ARM® Cortex®-A57 MPCore
<b>Memoria</b>	LPDDR4 de 4 GB y 64 bits
<b>Almacenamiento</b>	microSD (tarjeta no incluida)
<b>Codificación de vídeo</b>	1x 4K30   2x 1080p60   4x 1080p30   9x 720p30 (H.264/H.265)
<b>Descodificación de vídeo</b>	1x 4K60   2x 4K30   4x 1080p60   8x 1080p30   18x 720p30 (H.264/H.265)
<b>Redes</b>	Gigabit Ethernet, clave E M.2
<b>Cámara</b>	2 conectores de cámara MIPI CSI-2 de 15 pines y 2 vías

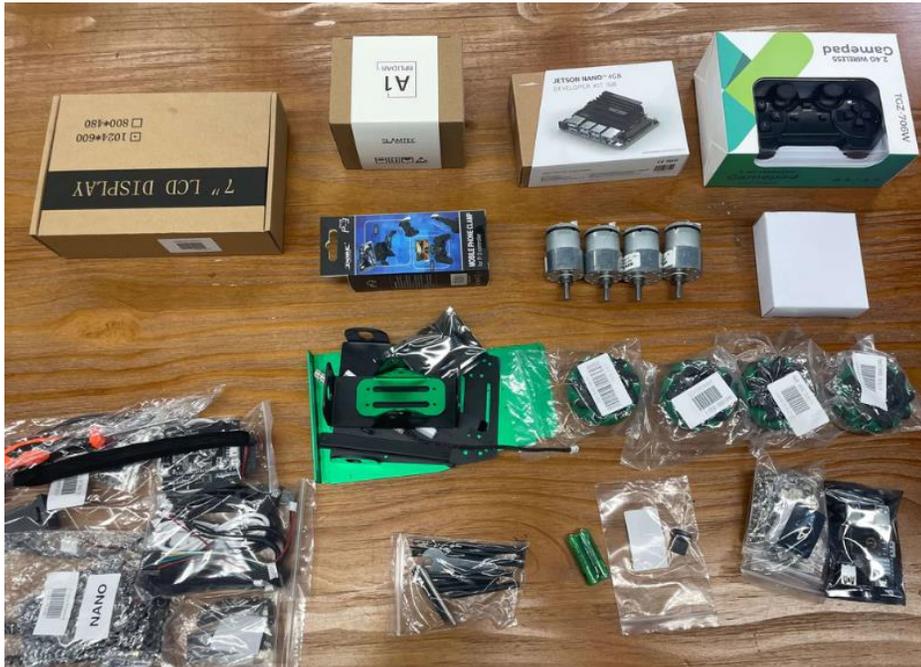
### 2.13 Rosmaster X3

ROSMaster X3 es un robot omnidireccional desarrollado utilizando el sistema operativo para robots ROS (Robot Operating System). Utiliza placas de la serie Jetson y Raspberry

Pi 4B como controladores principales, y cuenta con una configuración de hardware de alto rendimiento que incluye lidar y cámara de profundidad. Esto permite la realización de diversas aplicaciones robóticas, tales como control de movimiento, comunicación remota, construcción de mapas y navegación, evasión de obstáculos, conducción automática y reconocimiento de características del cuerpo humano. La figura 19 muestra las piezas que vienen dentro de la caja del ROSMASTER X3 y en la figura 20 una imagen de referencia del ROSMASTER X3 completamente armado.

**Figura 19**

**Piezas fuera de caja del ROSMASTER X3.**



**Figura 20**

**Imagen de referencia ROSMASTER X3.**



### 2.13.1 Depth Cámara

Esta cámara de profundidad de percepción real para ROS incluye una cámara a color (RGB), una cámara de infrarrojos, un proyector infrarrojo y un procesador de profundidad. Con decenas de funciones como reconocimiento facial, reconocimiento de gestos, reconocimiento del esqueleto humano, medición en 3D, percepción del entorno, navegación con mapas en 3D, etc., puede ser ampliamente utilizada en televisores, teléfonos móviles, robots, drones, realidad virtual/aumentada, hogares inteligentes y otros campos. Diseñamos especialmente un soporte de extensión para facilitar a los usuarios su instalación en varios robots ROS o vehículos inteligentes. En la figura 21 muestra la cámara que utiliza el ROSMASTER X3, utiliza la cámara Astra PRO Depth.

**Figura 21**

**Depth cámara del ROSMASTER X3.**



#### 2.14 Jetson Inference

Jetson Inference representa una biblioteca de inferencia de código abierto específicamente desarrollada para operar en plataformas de hardware NVIDIA Jetson, reconocidas por ser sistemas embebidos de alto rendimiento. Esta herramienta posibilita la ejecución en tiempo real de modelos de aprendizaje profundo en dispositivos Jetson, simplificando la integración de capacidades de visión por computadora en aplicaciones embebidas.

## CAPÍTULO 3

### 3. PROPUESTA DEL PROYECTO

#### 3.1 Descripción general de la propuesta

El siguiente proyecto tiene como finalidad potenciar la eficiencia y la calidad en la planificación y despliegue de redes Wi-Fi mediante la integración de dos tecnologías clave: MediaPipe Development y el robot Rosmaster X3.

La aplicación de técnicas de MediaPipe Development se enfocará en la optimización del reconocimiento de objetos en tiempo real. Esto permitirá una evaluación más precisa del entorno físico y de los elementos que puedan afectar la señal Wi-Fi, como obstáculos y fuentes de interferencia. La utilización de MediaPipe contribuirá a una mayor inteligencia en la toma de decisiones durante la fase de planificación, mejorando la capacidad de anticipar y mitigar posibles problemas de cobertura y rendimiento.

Por otro lado, la incorporación del robot Rosmaster X3 desempeñará un papel crucial en el despliegue práctico de la red Wi-Fi. La capacidad de movilidad del robot facilitará la colocación estratégica de puntos de acceso, antenas y otros dispositivos en ubicaciones óptimas para lograr una cobertura homogénea y una calidad de señal robusta. Además, el Rosmaster X3 permitirá realizar ajustes dinámicos en tiempo real, adaptándose a cambios en el entorno y optimizando continuamente el rendimiento de la red.

En conjunto, esta iniciativa busca no solo mejorar la eficacia técnica de las redes Wi-Fi, sino también elevar la capacidad de respuesta y adaptabilidad de la infraestructura ante condiciones cambiantes. La combinación de MediaPipe Development y el robot Rosmaster X3 ofrece un enfoque integral para enfrentar los desafíos de planificación y despliegue de redes inalámbricas, contribuyendo a un desempeño más eficiente y una experiencia de usuario mejorada.

Para el correcto entendimiento del proyecto se ha dividido en tres fases primordiales las cuales son:

- Fase I; será donde se detallen todos los componentes físicos, herramientas digitales,

softwares de aplicación y desarrollo, ensamblaje de equipos, es decir, la preparación de todos los elementos indispensables al momento de poder llevar a cabo sin ningún problema el proyecto.

- Fase II; será el desarrollo del código de programación para la identificación de objetos específicamente equipos de telecomunicaciones tales como routers, switch, cableado, antenas, cajas de fibra, etc. Todo esto se logrará mediante el uso de MediaPipe Development que lo ejecutaremos en la tarjeta Jetson Nano haciendo uso de Python3, cabe destacar que todos los equipos mencionados anteriormente se encuentran ensamblados en el robot Rosmaster X3. Una vez culminado el código podremos hacer las respectivas prácticas para demostrar el funcionamiento del mismo.
- Fase III; consta del mapeo en entornos controlados donde se verifica el correcto funcionamiento de la cámara Depth y el lidar laser, ya ensamblados en el robot Rosmaster X3. Mediante la codificación, instrucciones y control total del robot se realizarán mapas en 3D los mismos que serán exportados o guardados en formatos conocidos como jpg o png desde una máquina virtual (VMware) para llevarlos al software WiFi Heatmap que nos ayudará a la ubicación correcta de nuestros dispositivos y equipos de telecomunicaciones dándonos un mapa de calor de las zonas que abarca los dispositivos inalámbricos.

Una vez finalizada cada una de las fases y comprobadas cada una de ellas con su correcto funcionamiento y especificaciones podremos juntar los conocimientos adquiridos en las fases para que la propuesta pueda ser llevada a cabo en facultad de Sistemas y Telecomunicaciones en los diferentes edificios, laboratorios, aulas y decanato de la misma.

## **3.2 FASE I**

### **3.2.1 Componentes físicos**

Los componentes físicos más importantes e indispensables en el desarrollo de nuestro proyecto que fueron analizados y estudiados pre - ejecución del mismo son el robot

Rosmaster X3 que previo se debió ensamblar la tarjeta de inicio y de expansión, el lidar laser y la cámara Depth, tarjeta Jetson Nano de 4 Gb y una computadora o laptop con un procesador mínimo de 6 núcleos.

### 3.2.1.1 Ensamblaje del robot Rosmaster X3

El robot Rosmaster X3 desarrollado por la compañía ROS, viene separado en varias partes para el respectivo armado del mismo, cualquier tipo de información básica en cuanto al ensamble del mismo se proporcionará la respectiva información del manual al final del documento. Figura 22 muestra el ensamble del ROSMASTER X3.

**Figura 22**

**Ensamble de piezas para el ROSMASTER X3.**



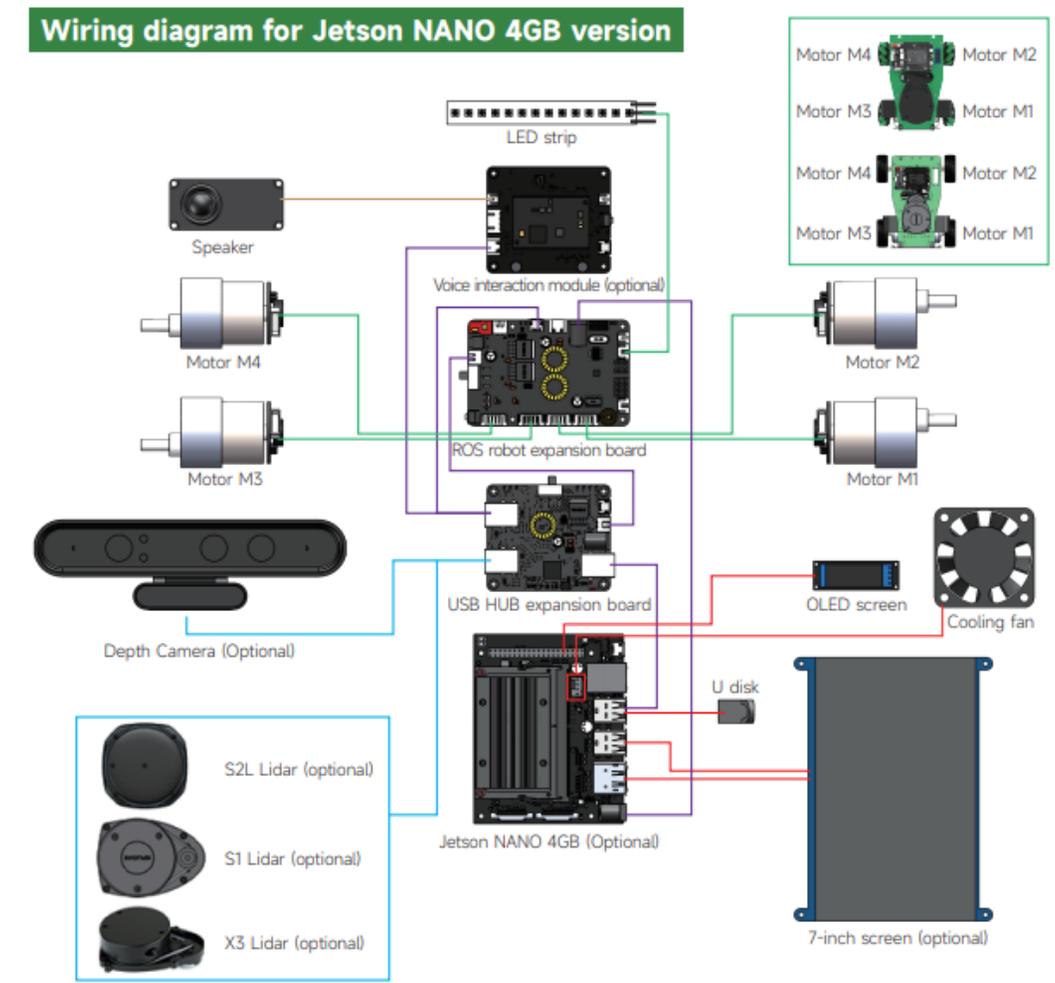
Los elementos que vamos a detallar para tener en cuenta es las respectivas conexiones de

las tarjetas, el montaje de la cámara y del lidar laser.

Para las conexiones de las tarjetas deben ser montadas tal y cual indica el manual de instrucciones proporcionadas por la compañía. A continuación, se muestra una imagen de cómo deben ser las conexiones de la tarjeta de inicio, de expansión y la Jetson Nano de 4 Gb. En la figura 23 muestra un diagrama completo de la conexión de las tarjetas del ROSMASTER X3 con todas sus entradas y periféricos.

**Figura 23**

**Diagrama de conexión de tarjetas del ROSMASTER X3.**

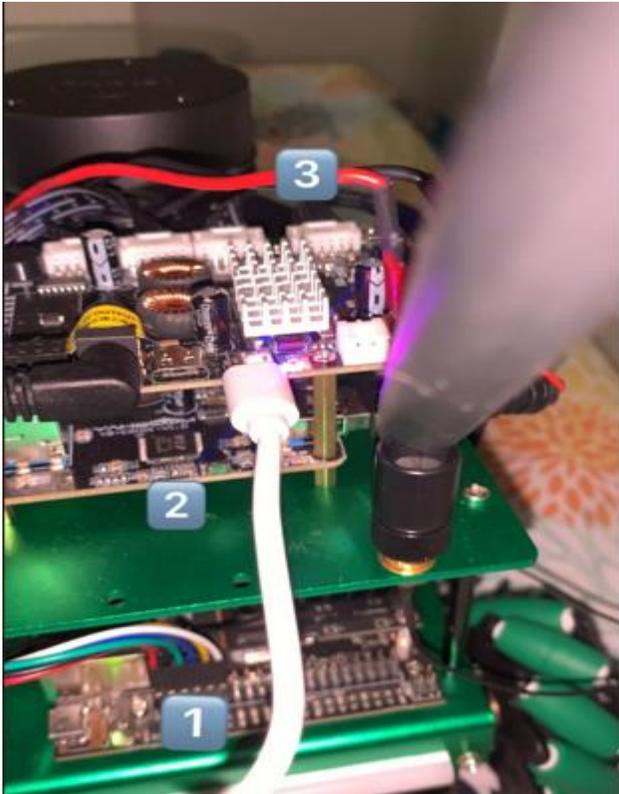


La imagen mostrada sería el diagrama a seguir para el respectivo ensamblaje de las tarjetas, ahora se mostrará una imagen de referencia una vez armado el robot. Los números en la

siguiente imagen son las tarjetas siendo el 1 la Jetson Nano, la 2 la tarjeta de expansión USB y la 3 la tarjeta de expansión del Robot Ros. En la figura 24 muestra una descripción enumerado de las tarjetas ensambladas en el ROSMASTER X3.

**Figura 24**

**Tarjetas ensambladas en el ROSMASTER X3.**



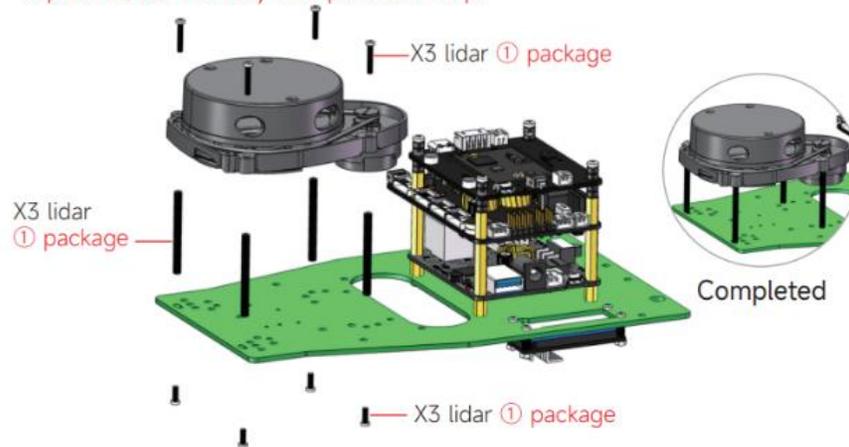
A continuación, se muestra una imagen guía y otra imagen tomada después de montar el lidar laser.

En la figura 25 se tiene una imagen de referencia de la ubicación del Lidar Laser en ROSMASTER X3 y en la figura 26 una imagen real del Lidar Laser en el ROSMASTER X3.

**Figura 25**

**Imagen de referencia del LIDAR laser en el ROSMASTER X3.**

## 27. Installation of X3 Lidar (Optional, do not buy this part can skip)



**Figura 26**

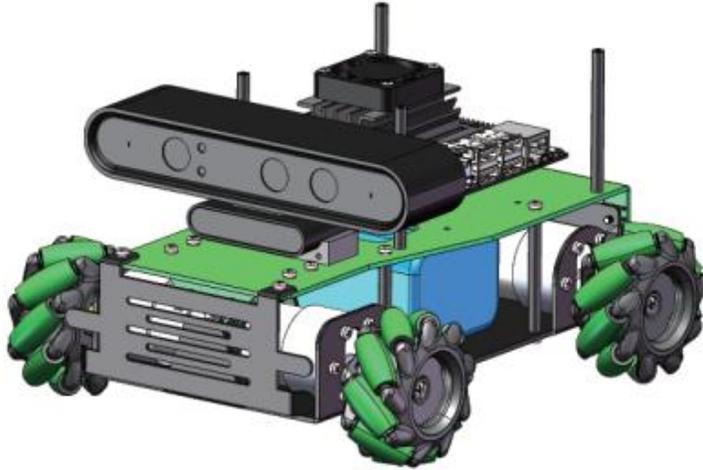
**Imagen de puesta final del LIDAR en el ROSMASTER X3.**



La cámara de nuestro Rosmaster X3 no será problema en instalar debido a que el manual proporciona la información necesaria para la instalación de la misma, una vez montada quedaría de la siguiente manera. En la figura 27 se tiene una imagen de referencia de la cámara Astra Pro montada en el ROSMATER X3 y en la figura 28 una imagen real ensamblada del ROSMASTER X3.

**Figura 27**

**Imagen de referencia Depth cámara en el ROSMASTER X3.**



**Figura 28**

**Depth cámara montada en el ROSMASTER X3.**

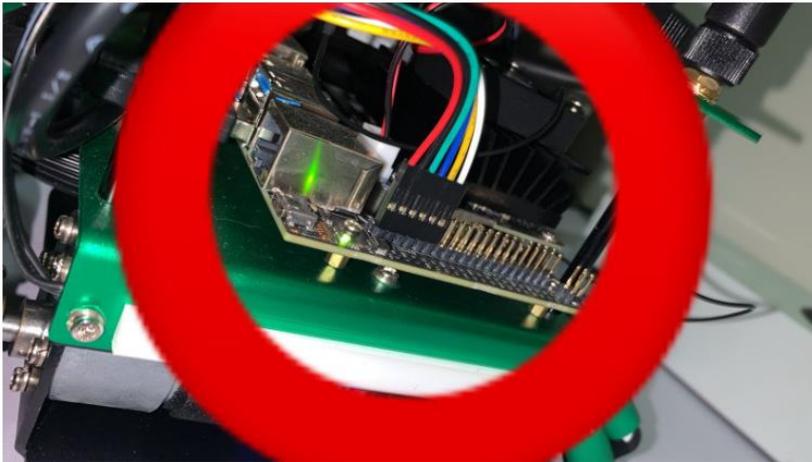


Finalmente, para la tarjeta Jetson Nano de 4 Gb que está montada en el robot Rosmaster X3 al inicio no hace falta que se configure nada debido a que la tarjeta fue adquirida con el robot ya viene con documentos y configurada para el uso del Robot Rosmaster X3 de forma básica y preliminar, lo que se necesitará para verificar el entorno de la Jetson Nano

será conectarla a un monitor o desde una laptop usando el software VNC. En la siguiente fase se detallará más a fondo que archivos, librerías y redes neuronales se deben descargar a la tarjeta Jetson Nano. En la figura 29 se muestra la tarjeta Jetson Nano de 4GB ensamblada y montada en ROSMASTER X3 y en la figura 30 en entorno de inicio de la Jetson Nano.

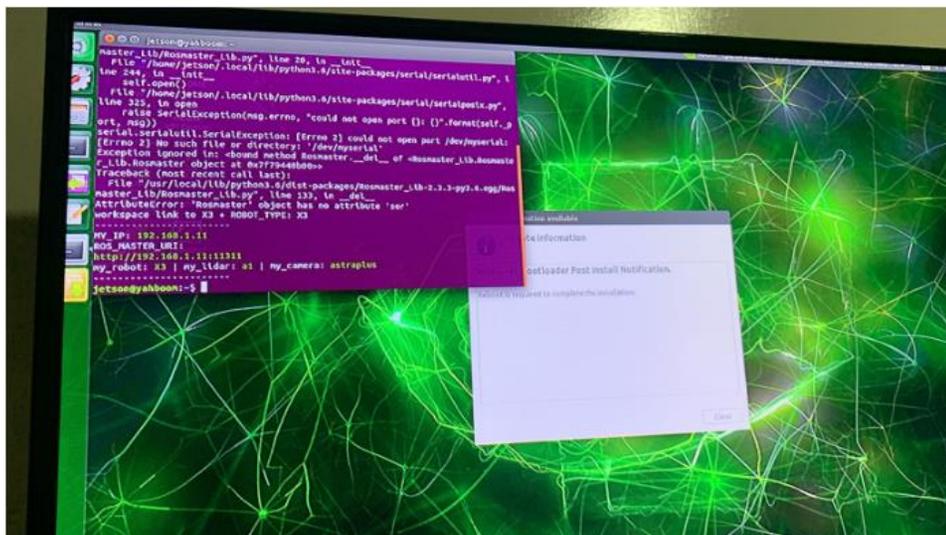
**Figura 29**

**Tarjeta Jetson Nano de 4Gb montada en el ROSMASTER X3.**



**Figura 30**

**Entorno de inicio de Nvidia para la Jetson Nano.**



Terminada la primera fase tendremos listo nuestro robot Rosmaster X3 para empezar con las siguientes fases en cuanto al mapeo de entornos y la codificación y entrenamiento de redes neuronales para el reconocimiento de objetos en tiempo real.

## **3.3 FASE II**

### **3.3.1 Configuración Jetson Nano**

El primer paso de la segunda fase será la instalación de las librerías y archivos necesarios para empezar el entrenamiento de nuestra red neuronal para el reconocimiento de objetos en tiempo real. En primer lugar, necesitamos descargar el software Real VNC viewer para poder conectarnos directamente desde la red del robot, cabe recalcar que este software es necesario si se va a realizar toda la programación desde una laptop. Una vez conectada a la red del ROSMASTER X3 cuya dirección IP es 192.168.1.1 y la contraseña para acceder a la red es 123456789. A continuación, en las siguientes imágenes se mostrará como conectar a la red del robot desde una laptop y la interfaz de inicio del software VNC. En la figura 31 muestra el nombre de la red del ROSMASTER X3 a la que se debe conectar.

**Figura 31**

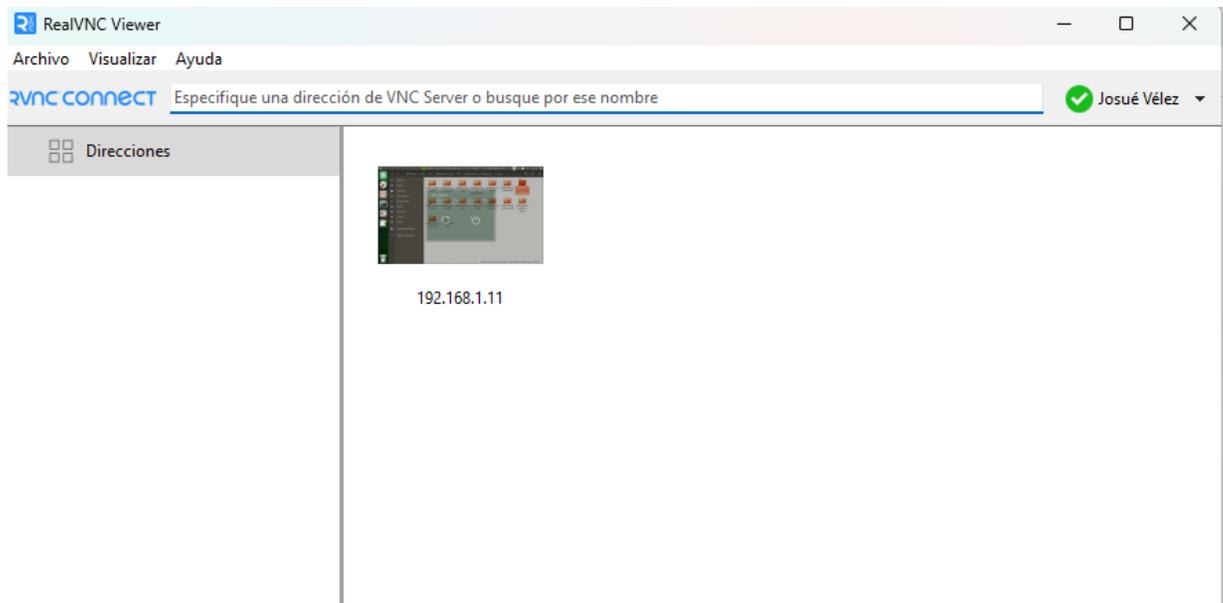
**Nombre de red del ROSMASTER X3.**



Una vez conectada a la red del ROSMASTER X3, se debe iniciar el software RealVNC para acceder a la interfaz de inicio y programación de la Jetson Nano. En la figura 32 se observa la interfaz de inicio del software RealVNC para la conexión remota con el ROSMASTER X3.

**Figura 32**

**Interfaz de inicio de RealVNC.**



### 3.3.2 Instalación de Jetson Inference

Se debe abrir una terminal para la introducción de los siguientes comandos para la descarga de las primeras librerías de la Jetson Inference en nuestra tarjeta las cuales son el soporte para la instalación de las librerías y comandos. La figura 33 muestra los comandos para la descarga del soporte Jetson Inference que deben ser ingresados en una ventana de comandos de la Jetson Nano para su respectiva ejecución.

En la figura 33 muestra los códigos de descarga para el soporte de Jetson Inference. Y en la figura 34 los códigos para la red neuronal.

**Figura 33**

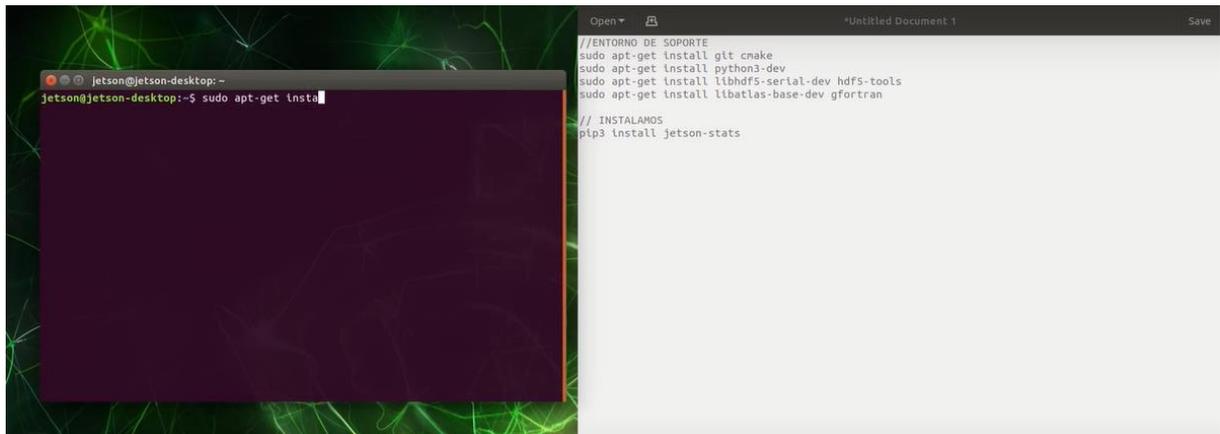
**Códigos de descarga para el soporte Jetson Inference.**

```
Open [icon] *Untitled Document 1
//ENTORNO DE SOPORTE
sudo apt-get install git cmake
sudo apt-get install python3-dev
sudo apt-get install libhdf5-serial-dev hdf5-tools
sudo apt-get install libatlas-base-dev gfortran

// INSTALAMOS
pip3 install jetson-stats
```

**Figura 34**

**Códigos de actualización para la red neuronal.**



A continuación, se adjunta captura de los comandos a colocar en las terminales.

**Figura 35**

**Actualización de paquetes.**

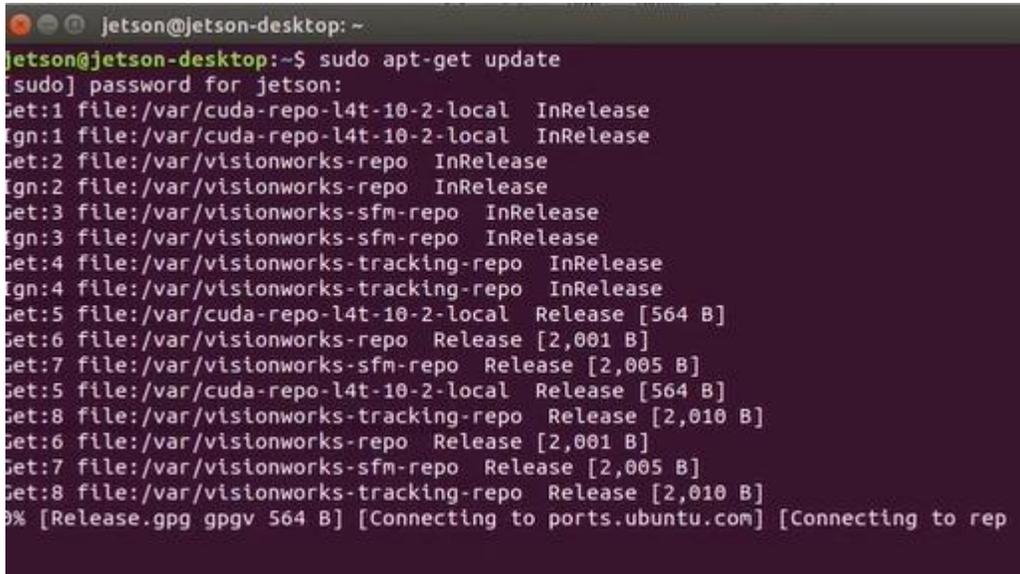
En la figura 35 muestra los comandos para la actualización de la red neuronal.

```
$ sudo apt-get update
$ sudo apt-get install git cmake
```

**Figura 36**

**Descarga en tiempo real de las actualizaciones.**

En la figura 36 muestra la ejecución de la descarga de las actualizaciones para la red neuronal.



```
jetson@jetson-desktop: ~  
jetson@jetson-desktop:~$ sudo apt-get update  
[sudo] password for jetson:  
Get:1 file:/var/cuda-repo-l4t-10-2-local InRelease  
Ign:1 file:/var/cuda-repo-l4t-10-2-local InRelease  
Get:2 file:/var/visionworks-repo InRelease  
Ign:2 file:/var/visionworks-repo InRelease  
Get:3 file:/var/visionworks-sfm-repo InRelease  
Ign:3 file:/var/visionworks-sfm-repo InRelease  
Get:4 file:/var/visionworks-tracking-repo InRelease  
Ign:4 file:/var/visionworks-tracking-repo InRelease  
Get:5 file:/var/cuda-repo-l4t-10-2-local Release [564 B]  
Get:6 file:/var/visionworks-repo Release [2,001 B]  
Get:7 file:/var/visionworks-sfm-repo Release [2,005 B]  
Get:5 file:/var/cuda-repo-l4t-10-2-local Release [564 B]  
Get:8 file:/var/visionworks-tracking-repo Release [2,010 B]  
Get:6 file:/var/visionworks-repo Release [2,001 B]  
Get:7 file:/var/visionworks-sfm-repo Release [2,005 B]  
Get:8 file:/var/visionworks-tracking-repo Release [2,010 B]  
9% [Release.gpg gpgv 564 B] [Connecting to ports.ubuntu.com] [Connecting to rep
```

**Figura 37**

**Finalización de librerías y códigos de Jetson Inference.**



```
Then clone the jetson-inference project:  
$ git clone https://github.com/dusty-nv/jetson-inference  
$ cd jetson-inference  
$ git submodule update --init  
qml-module-gtquick2 rdate tasksel tasksel-data  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 196 not upgraded.  
jetson@jetson-desktop:~$ git clone https://github.com/dusty-nv/jetson-inference
```

En la figura 37 muestra la descarga de todos los comandos necesarios en GIB de Jetson Inference, una vez finalizado tendremos listas las carpetas, librerías y códigos para empezar el entrenamiento de nuestra red neuronal.

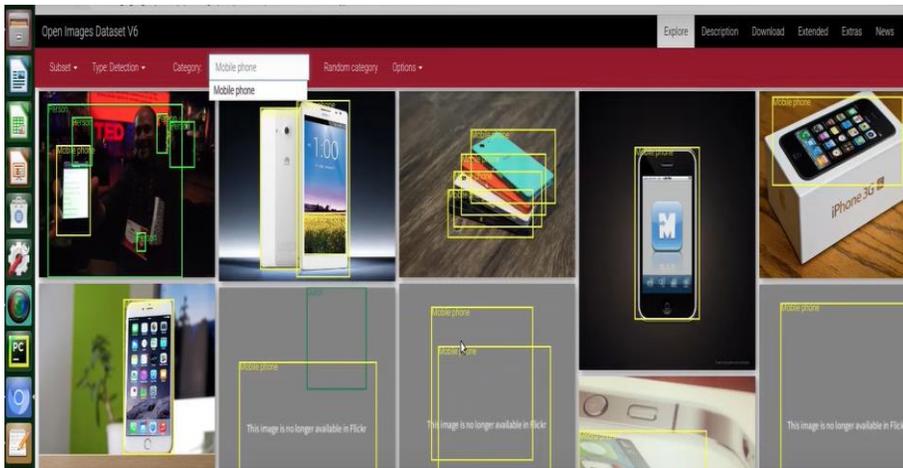
**3.4 Instalación de red neuronal – reconocimiento de objetos**

En primer lugar, debemos instalar un paquete de imágenes que será nuestro banco de imágenes para el reconocimiento de objetos en tiempo real referente a las telecomunicaciones, se puede hacer desde cualquier página web, también la opción de

realizarlo subiendo nuestras propias fotos tomada de los objetos del laboratorio y decanato de la facultad. En este caso vamos a descargar un banco de imágenes ya establecidos en los que se encuentran routers, switch, antenas, entre otros de todo tipo para el correcto reconocimiento de los objetos.

Para eso en el navegador de la tarjeta Jetson Nano nos vamos a dirigir al siguiente link <https://storage.googleapis.com/openimages/web/visualizer/index.html?type=detection&set=train&c=%2Fm%2F0dftk> para realizar la descarga del paquete de imágenes. En la figura 38 muestra la interfaz de la página web donde se descargará todo el banco de imágenes para la detección de objetos.

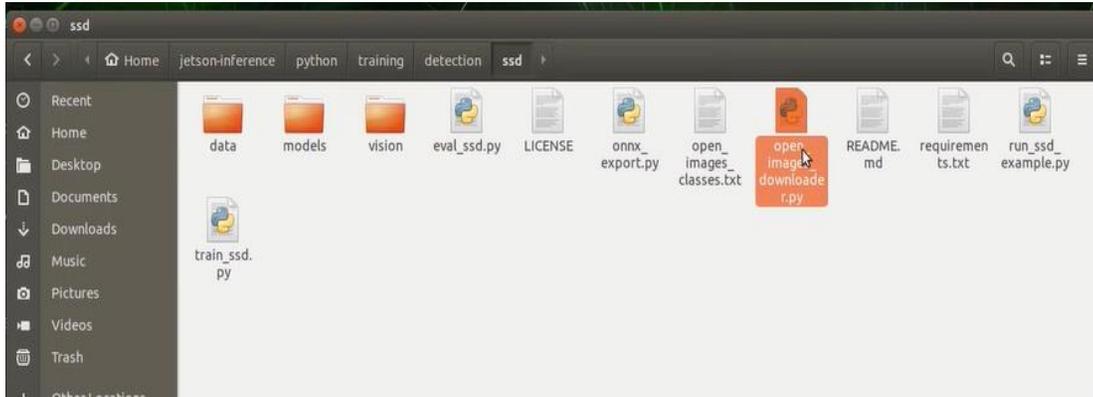
**Figura 38**  
**Interfaz del banco de imágenes para descargar.**



Como siguiente paso, es verificar que el banco de imágenes se haya descargado de manera correcta, para esto buscamos en la carpeta de Jetson Inference y buscamos una carpeta SSD que tendrá el archivo de las imágenes en .py

**Figura 39**

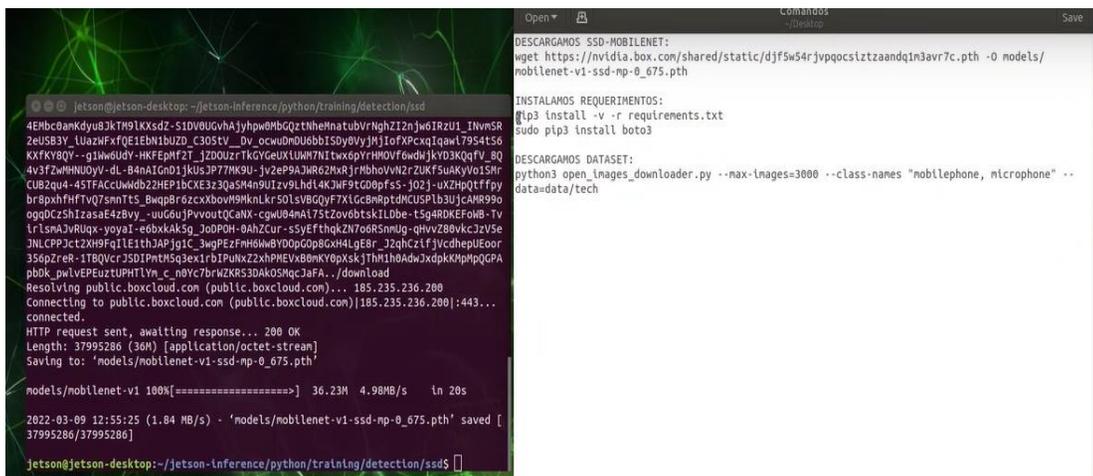
**Verificación del archivo descargado del banco de imágenes.**



Los siguientes códigos que debemos ingresar en nuestra terminal serán de los más importantes porque es nuestra red neuronal previamente entrenada para el reconocimiento de objetos en tiempo real, los dataset y los requerimientos necesarios de la misma.

**Figura 40**

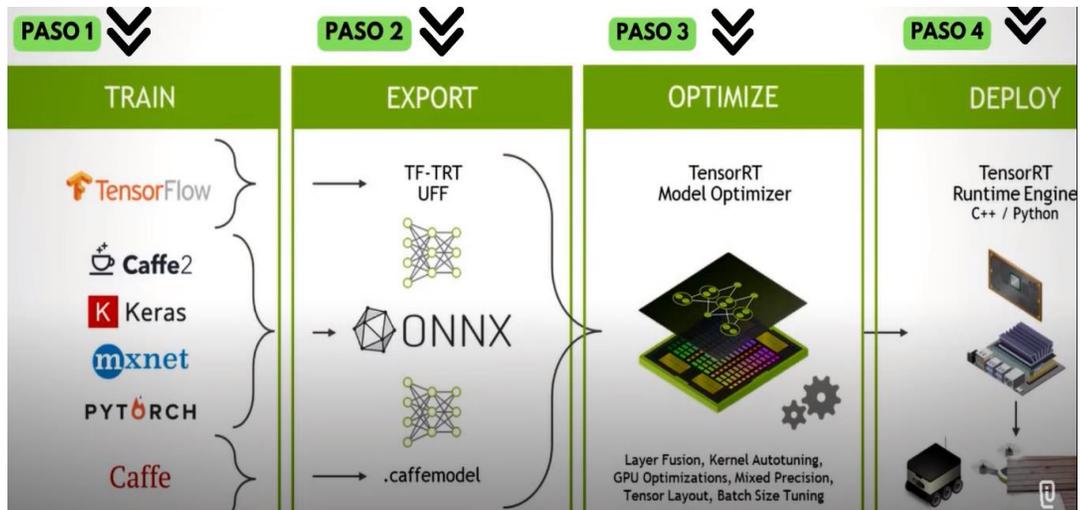
**Datasheet de la red neuronal previamente entrenada.**



La figura 40 muestra finalizada descargado los archivos y permisos necesarios está lista para el uso de la red neuronal, sin embargo, en este proyecto se le va a dar el plus de optimizar esta red neuronal convirtiendo el archivo en una extensión. Onnx.

**Figura 41**

**Guía de optimización del código de la red neuronal.**



La figura 41 es el procedimiento para la optimización de la red neuronal. Para este siguiente paso debemos colocar las extensiones en sus respectivas carpetas de nuestra red neuronal, permisos, el banco de imágenes. En la siguiente captura muestra el código a implementar para su respectiva ejecución, una vez finalizada la inserción del código se oprime enter para la ejecución y con esto quedará lista nuestra red neuronal para la programación final.

**Figura 42**

**Finalización de la ejecución de comandos para la red neuronal.**

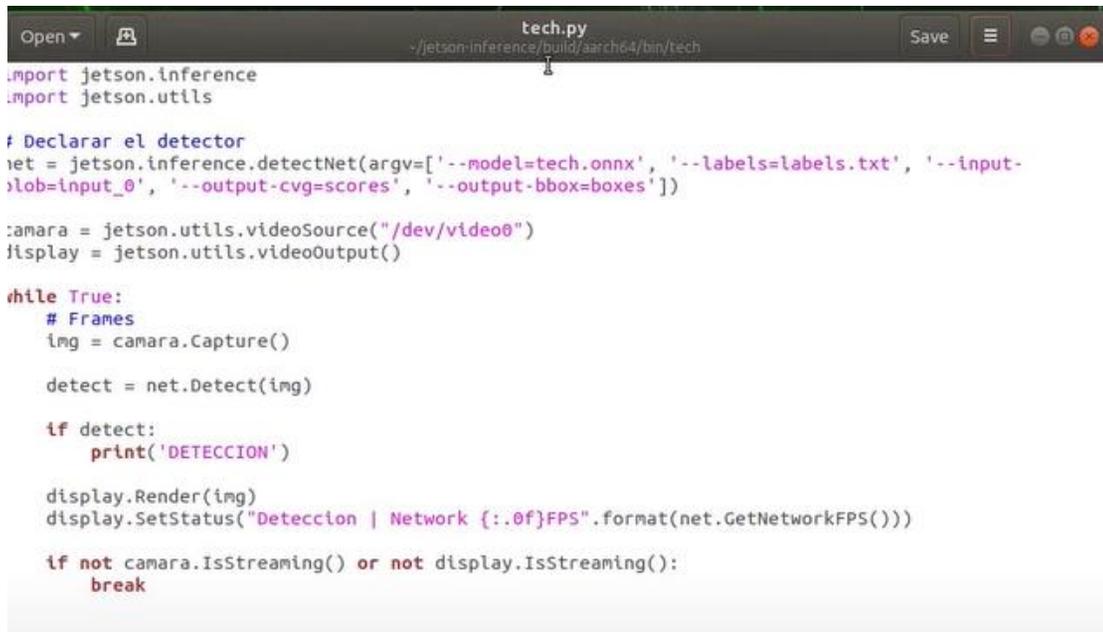
```
2022-03-09 13:07:49 - Task Done.  
jetson@jetson-desktop:~/jetson-inference/python/training/detection/ssd$ python3  
train_ssd.py --data=data/tech --model-dir=models/tech --batch-size=4 --workers=1  
--epochs=30
```

En la figura 42 muestra los comandos para la ejecución de la red neuronal. El código final para que cumpla con el objetivo de detección de objetos en tiempo real está dado por el

siguiente código que funcionará con la cámara Depth del Rosmaster X3 para el reconocimiento.

### Figura 43

#### Código de ejecución final para la red neuronal.



```
tech.py
~/jetson-inference/build/aarch64/bin/tech

import jetson.inference
import jetson.utils

# Declarar el detector
net = jetson.inference.detectNet(argv=['--model=tech.onnx', '--labels=labels.txt', '--input-
lob=input_0', '--output-cvg=scores', '--output-bbox=boxes'])

camara = jetson.utils.videoSource("/dev/video0")
display = jetson.utils.videoOutput()

while True:
    # Frames
    img = camara.Capture()

    detect = net.Detect(img)

    if detect:
        print('DETECCION')

    display.Render(img)
    display.SetStatus("Deteccion | Network {:.0f}FPS".format(net.GetNetworkFPS()))

    if not camara.IsStreaming() or not display.IsStreaming():
        break
```

En la figura 43 se muestra el código final para la ejecución de la red neuronal. Para la ejecución del código en una terminal ingresaremos la siguiente dirección detallada en la captura. En pantalla se deberá abrir la cámara y ver el entorno, en el código expuesto anteriormente también cuenta con reconocimiento humano.

### Figura 44

#### Llamamiento de librerías para la ejecución.

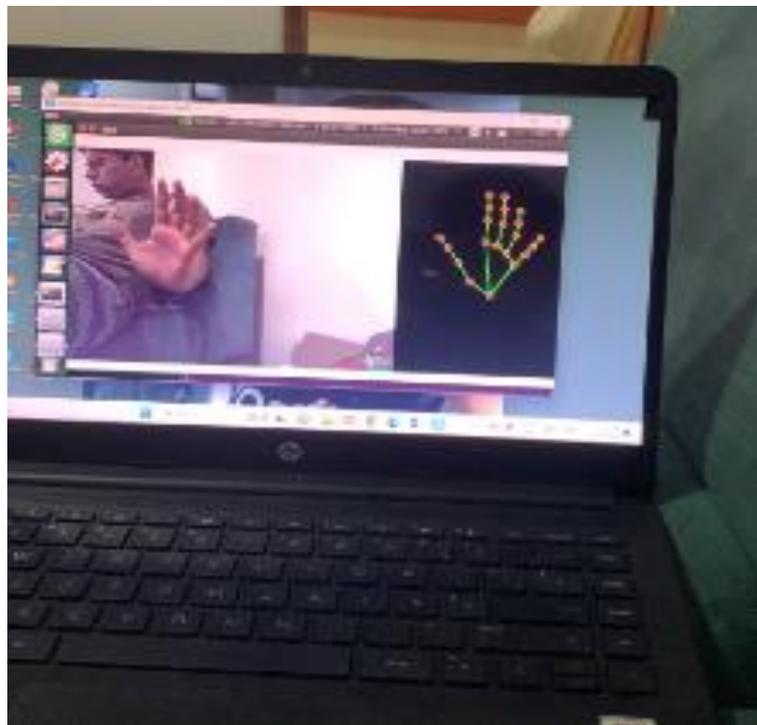
En la figura 44 se muestra la entrada de comandos para la activación de librerías.

```
Jetson@jetson-desktop: ~/jetson-inference/build/aarch64/bin/tech
File Edit View Search Terminal Help
Jetson@jetson-desktop:~$ cd jetson-inference/build/aarch64/bin/tech/
Jetson@jetson-desktop:~/jetson-inference/build/aarch64/bin/tech$ ls
labels.txt  tech.onnx  tech.onnx.1.1.8001.GPU.FP16.engine  tech.py
Jetson@jetson-desktop:~/jetson-inference/build/aarch64/bin/tech$ python3 tech.py
```

**Figura 45**

**Detección de prueba con un código sencillo de reconocimiento de manos.**

En la figura 45 se muestra un ejemplo de detección en tiempo real.



## 3.5FASE III

### 3.5.1 Mapeo en 2D en entornos controlados

La función "depthimage\_to\_laserscan" toma una imagen de profundidad (en metros codificada en punto flotante o preferiblemente en milímetros codificada en uint16 para dispositivos Open NI) y genera un escaneo láser 2D según los parámetros proporcionados. "depthimage\_to\_laserscan" utiliza suscripciones diferidas y no se suscribe a imágenes o información de la cámara hasta que un usuario realiza un escaneo.

El paquete de funciones de la función "depthimage\_to\_laserscan" convierte imágenes de profundidad en datos lidar, y sus funciones de mapeo y navegación son similares a las del lidar. Nota: El rango de escaneo de la cámara de profundidad no es de 360°.

Para el mapeo de cualquier entorno en una terminal debemos ingresar los siguientes comandos para identificar el tipo de robot, el tipo de cámara y el lidar con el que se está trabajando. En la figura 46 muestra los comandos que se deben introducir en una ventana de comandos para la activación e identificación del robot, cámara y lidar laser.

#### Figura 46

#### Llamamiento para la identificación del robot, cámara y lidar.

Open the [.bashrc] file

```
sudo vim .bashrc
```

Find the [ROBOT\_TYPE] parameter and modify the corresponding model

```
export ROBOT_TYPE=X3 # ROBOT_TYPE: X1 X3 X3plus R2 X7
```

Para poder acceder a la cámara desde la computadora o laptop donde se está trabajando se deben ingresar en una nueva terminal los siguientes comandos. En la figura 47 se muestra los comandos para el acceso a la cámara y lidar.

## Figura 47

Acceso a la cámara y lidar mediante los siguientes códigos.

Start command(robot side)

```
roslaunch yahboomcar_nav astrapro_bringup.launch
```

Mapping command(robot side)

```
roslaunch yahboomcar_nav yahboomcar_map.launch use_rviz:=false  
map_type:=gmapping
```

- [use\_rviz] parameter: whether to enable rviz visualization.
- [map\_type] parameter: set the mapping algorithm [gmapping].

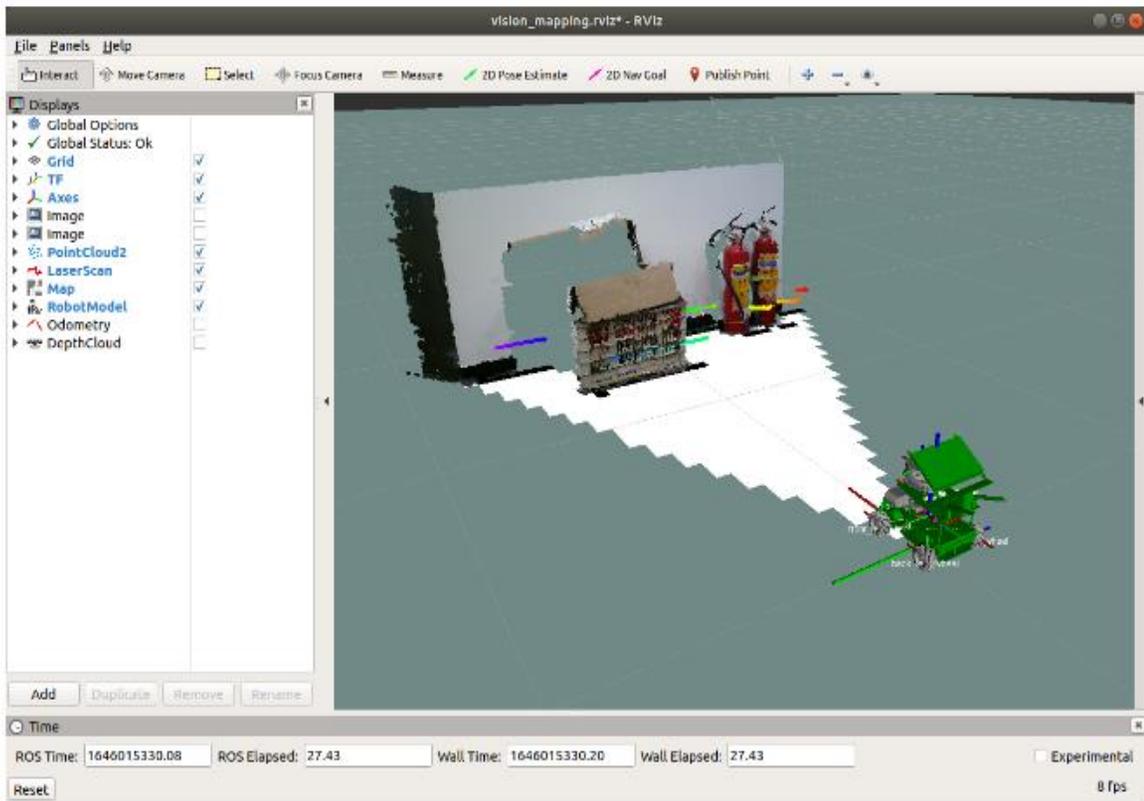
Open the visual interface(virtual machine side)

```
roslaunch yahboomcar_nav view_vision_mapping.launch
```

A continuación, una vez ingresado los comandos se nos abrirá una nueva ventana la cual nos mostrará lo que está frente a la cámara del robot, tal como se muestra en la siguiente captura tomada en tiempo real. En la figura 48 muestra la ejecución del mapeo del ROSMASTER X3 en 2D.

## Figura 48

Mapeo en 2D del ROSMASTER X3.



Una vez que el robot muestra en la pantalla del entorno de la Jetson Nano está listo para ser dirigido para el respectivo mapeo del área, esto lo podemos realizar de dos maneras. Una directamente desde una app en nuestro celular y la otra controlado desde un dualshock. Finalizado el mapeo de la zona vamos a que tener guardar el mapa. Para esto se debe ingresar el siguiente código. El mapa quedará guardado en alguna carpeta del entorno de Nvidia de la Jetson Nano, para poder llevar hasta nuestro computador simplemente con una USB lo podríamos realizar. La figura 49 son los comandos para poder guardar los mapas y planos realizados al momento del mapeo del ROSMASTER X3.

**Figura 49**

**Comandos para guardar de manera directa el mapeo.**

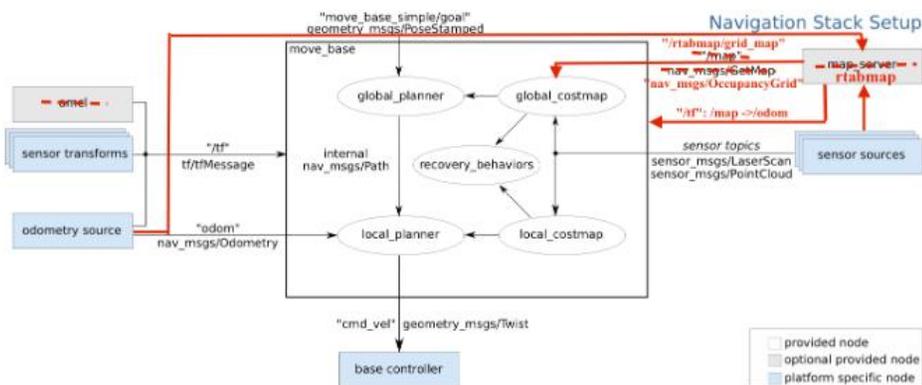
```
# The first way
roslaunch map_server map_saver -f
~/yahboomcar_ws/src/yahboomcar_nav/maps/my_map
# second way
bash ~/yahboomcar_ws/src/yahboomcar_nav/maps/map.sh
```

### 3.5.2 Mapeo en 3D en entornos controlados

Este paquete es un paquete funcional de ROS para RTAB Map, un método RGB-D SLAM basado en un detector global de cierres de bucle con restricciones en tiempo real. Este paquete puede utilizarse para generar nubes de puntos tridimensionales del entorno y crear mapas de ocupación bidimensionales para la navegación. En la figura 50 muestra un diagrama completo del lidar laser.

**Figura 50**

**Diagrama de funcionamiento del SLAM Lidar en 3D.**



Como se puede observar en la figura anterior, no es necesario utilizar el posicionamiento

de Monte Carlo (amcl), ya que RTAB Map tiene su propia función de posicionamiento. Si se utiliza amcl, esto ocasionará posicionamientos repetidos y fallos en el posicionamiento. Al emplear RTAB Map para la navegación en el marco central, el mapa inicializado es proporcionado por RTAB Map, no por map\_server.

Para realizar el mapeo en 3D es muy parecido a lo que se hizo de manera previa en el mapeo de 2D. Primero ingresar el código para la detección de la cámara, lidar y robot y en segundo lugar escoger como se lo quiero controlar, si de forma manual o de forma automática (app del celular o dualshock). Se ingresa la siguiente línea de comandos para que puede aparecer la pantalla del entorno de Nvidia el cual nos muestre como el robot empezará hacer el mapeo del área. En la figura 51 muestra los comandos a ejecutar para la activación de la cámara y el lidar para el mapeo en 3D.

### Figura 51

#### Comandos para activación de cámara y lidar en mapeo 3D.

```
roslaunch yahboomcar_nav laser_astrapro_bringup.launch
```

Command to start mapping or navigation(robot side)

```
roslaunch yahboomcar_nav yahboomcar_rtabmap.launch use_rviz:=False
```

- use\_rviz parameter: whether to open rviz.

Start Visualization(Virtual Machine)

```
roslaunch yahboomcar_nav view_rtabmap.launch
```

Keyboard control node(virtual machine)

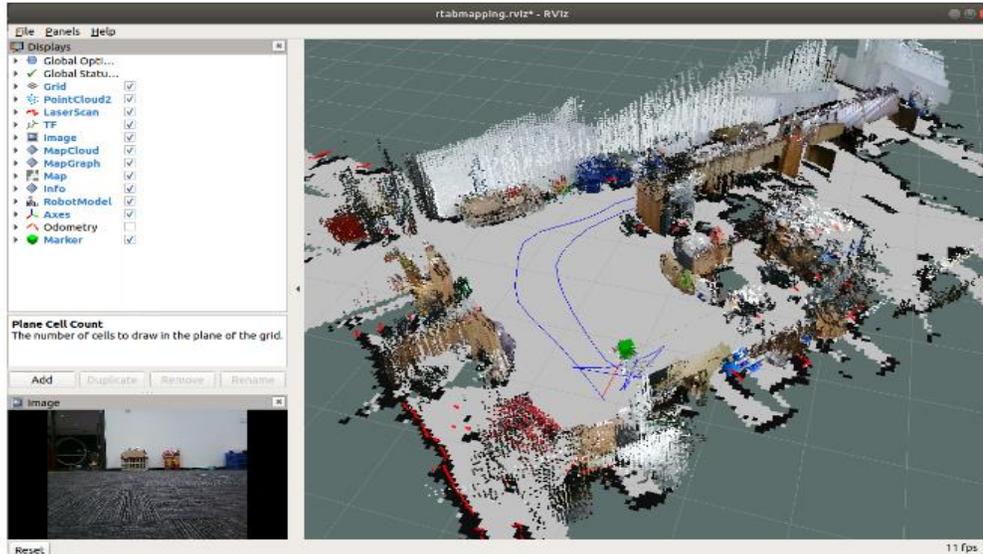
```
roslaunch yahboomcar_nav view_rtabmap.launch
rosrun teleop_twist_keyboard teleop_twist_keyboard.py # system
integration
roslaunch yahboomcar_ctrl yahboom_keyboard.launch # yahboomcar
custom
```

Una vez ingresado los códigos en una nueva terminal, se abrirá una ventana donde muestre

lo que está frente al robot, como la captura a continuación. Para poder guardar el mapa que se ha generado se debe repetir el mismo paso que se hizo en el mapeo en 2D. En la figura 52 muestra el mapeo en 3D del ROSMASTER X3.

**Figura 52**

**Mapeo en 3D del ROSMASTER.**



**3.6 WiFi Heatmap**

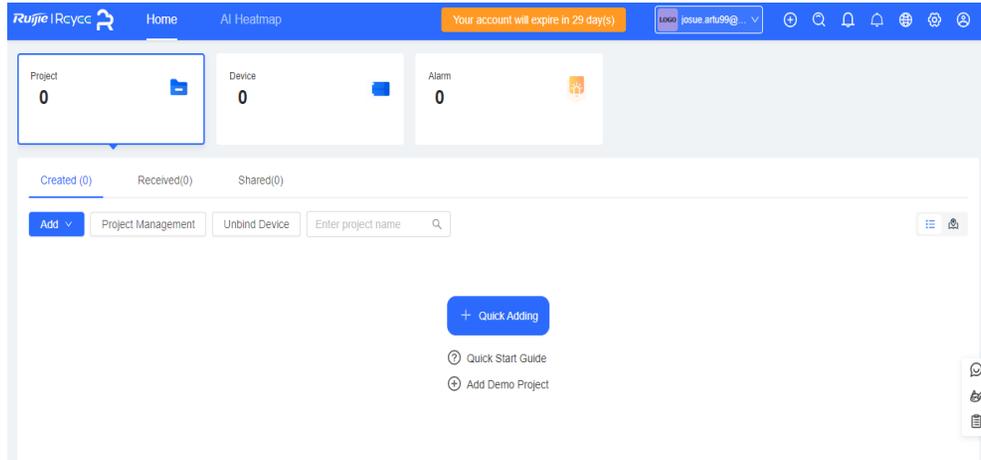
Finalizada la tarea de realizar el mapeo de las respectivas áreas y guardado de manera correcto el plano del mapeo lo que se procede a continuación gracias al software en línea WiFi Heatmap es adherir los equipos que se identificaron previamente en la fase II, y que este de manera automática nos dé un bosquejo de donde ubicar los equipos que seleccionamos en la interfaz para evitar bajas latencias, zonas muertas, interferencia entre equipos y ese tipo de problemas que las redes sufren y que bajan la calidad del servicio de las mismas.

Para realizar esta parte de la fase III se debe ingresar al siguiente link [https://cloud-la.ruijienetworks.com/macc5/adminIntl/#/monitor\\_overview\\_global\\_menu](https://cloud-la.ruijienetworks.com/macc5/adminIntl/#/monitor_overview_global_menu) y crear una cuenta para poder utilizar el software. Una vez dentro tendremos la interfaz.

## Figura 53

### Interfaz de inicio WiFi Heatmap.

En la figura 53 muestra la interfaz de inicio de WiFi Heatmap.

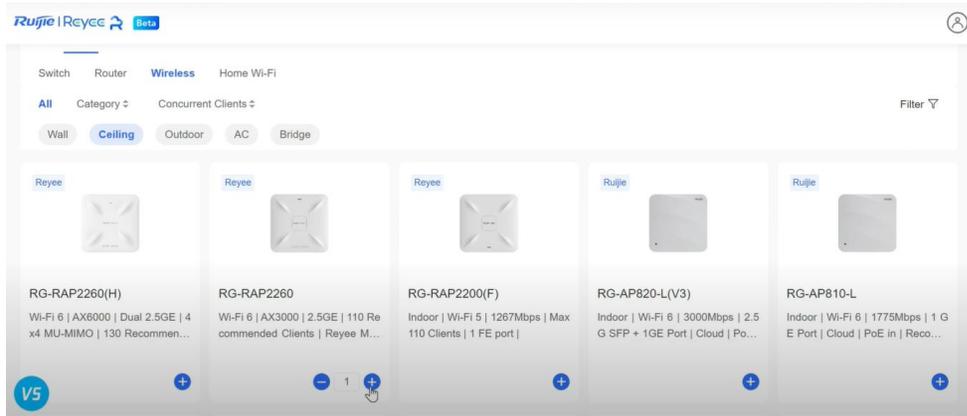


Damos clic en proyecto para realizar un nuevo proyecto y poder subir nuestros mapas y seleccionar los equipos que el robot Rosmaster X3 identifico fases previas. Para este caso se tomaron los planos de un departamento en donde modificamos los elementos de redes existentes tales como la ubicación de la caja de la fibra que daba la posibilidad moverlas junto al router, otro router que sirve como punto de acceso en una de las habitaciones y una cámara IP. Con el propósito de superar desafíos como zonas sin cobertura e interferencias en el entorno, se pretende aplicar técnicas avanzadas utilizando MediaPipe Development y el robot Rosmaster X3. La idea es emplear el software WiFi Heatmap para analizar de manera automatizada el entorno, identificando problemas como áreas con señal débil o interferencias. Posteriormente, mediante la integración de estos recursos tecnológicos, se buscará obtener de forma automatizada ubicaciones óptimas para la colocación de equipos, con el fin de optimizar la cobertura y el rendimiento de la red Wi-Fi. Este enfoque integral aspira a mejorar tanto la planificación como el despliegue de redes inalámbricas, abordando eficientemente problemas y proporcionando soluciones adaptativas y dinámicas. En la figura 54 se muestra los equipos a escoger según lo identificado y mapeado en el software WiFi Heatmap.

## Figura 54

### Selección de los equipos a utilizar en nuestra práctica.

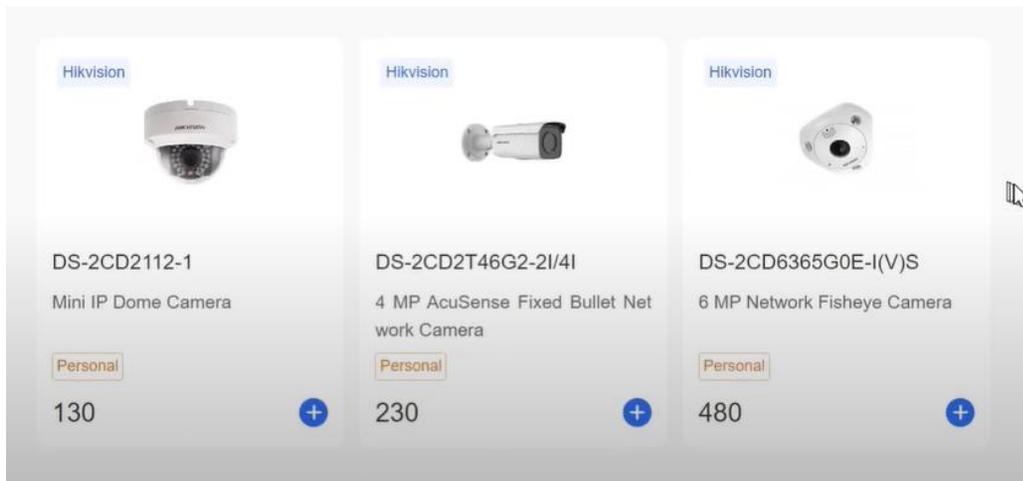
En la figura 54 muestra la selección de todos los equipos que se usaron en la práctica.



## Figura 55

### Selección de cámara IP.

En la figura 55 muestra la selección de equipos a utilizar según nuestro entorno.



## Figura 56

### Plano de casa donde se realizó la práctica.

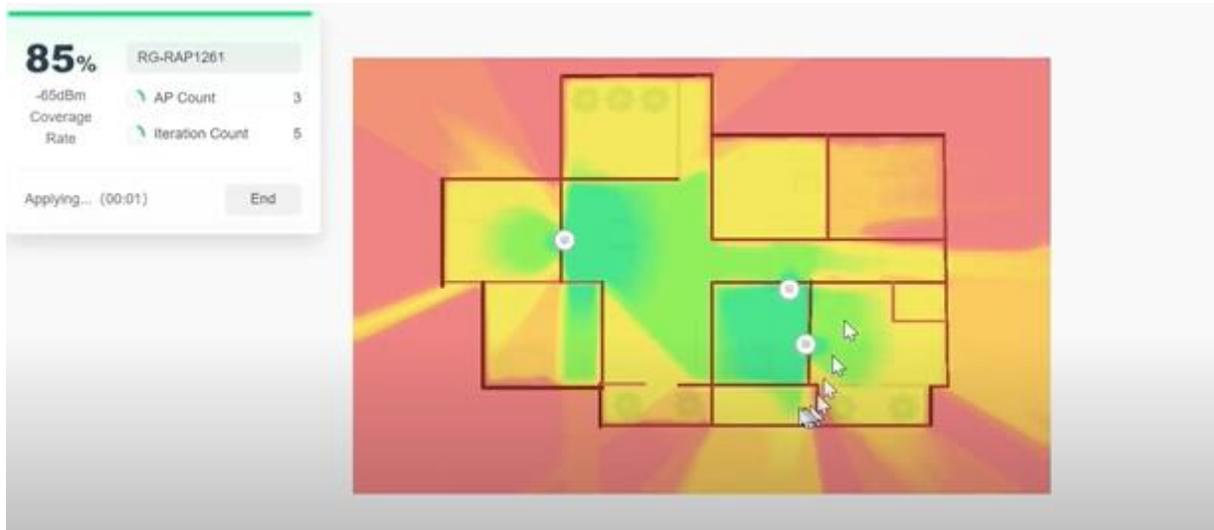
En la figura 56 se muestra la carga del plano en el software WiFi HeatMap mapeado con el Rosmaster X3.



Se colocan los equipos en ciertas partes del entorno donde habíamos hecho mapeo en esta misma fase. A un costado del software podemos ver qué tipo de materiales son de los que están hecho las puertas, paredes, etc. Todo esto para que sea lo más realista posible. En la figura 57 muestra una imagen de referencia con los equipos colocados a propósito de manera incorrecta.

**Figura 57**

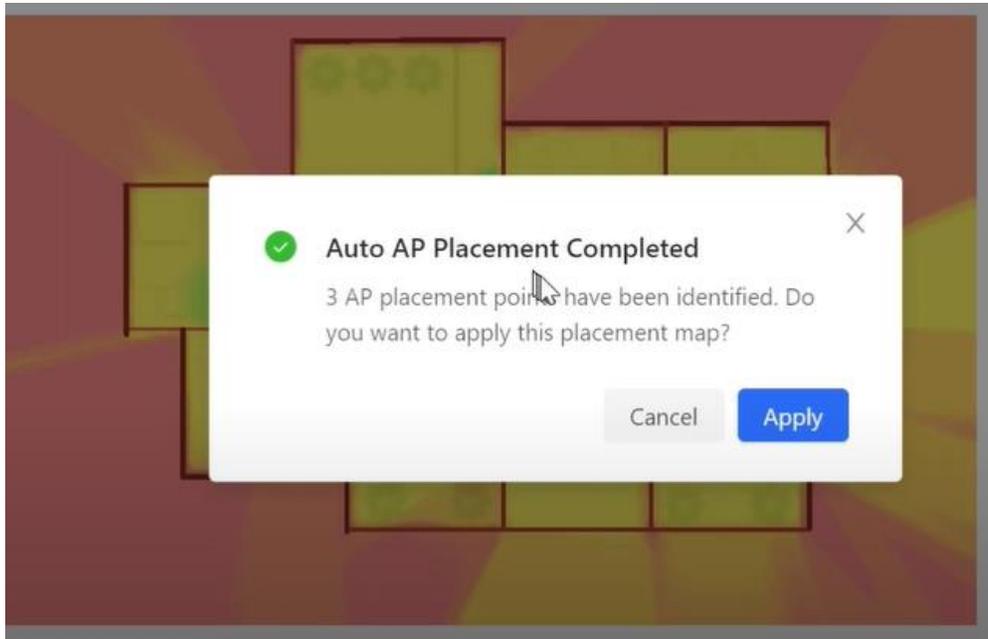
**Zona de cobertura de equipos “mal posicionados”.**



En este punto por la ubicación de los elementos se puede observar que hay zonas amarillas, que no son del todo malas, sin embargo, no son zonas de muy buena recepción de la señal inalámbrica. A un costado en la parte superior se puede notar el 85% de zona cubierta de nuestra red. Para que el software de manera automática nos dé la opción de colocar los equipos en las zonas más óptimas. Cabe recalcar que se lo puede también de manera manual. En la figura 58 muestra la optimización recomendada para la ubicación de equipos por el software.

**Figura 58**

**Optimización y ubicación adecuada de los equipos para mejor cobertura y rendimiento.**



Una vez que se le da la orden al programa de efectuar de manera automática la imagen del plano aparecerá diferente y las zonas verdes que son las de mayor intensidad de la señal se verán mucho más que en la anterior imagen. En la figura 59 muestra el mapa de cobertura de los equipos.

**Figura 59**

**Mapa final de la cobertura de red inalámbrica mejorada.**



### 3.7 Costos de la implementación de la propuesta

En la tabla mostrada a continuación, se podrán observar de manera detallada los costos que implicó la implementación total del proyecto. En la tabla los equipos necesarios son muchos, debido a que el robot Rosmaster X3 es que tiene ensamblada la mayor parte de los elementos. La tabla 4 describe el costo de los equipos que se utilizaron en el proyecto. *Nota: el costo de los equipos no incluye valor de envío, puesto que la mayoría de equipos fueron pedidos en China hasta EE. UU, posterior la llegada a Ecuador.*

**Tabla 4**

**Tabla de costos de los elementos del proyecto.**

<b>DISPOSITIVO</b>	<b>CANTIDAD</b>	<b>VALOR</b>
ROSMASER X3	1	\$739.00
JETSON NANO 4GB	1	\$209.99
LAPTOP HP I7	1	\$874.46
ROUTER ZTE	2	\$80
CAMARA IP	1	\$30
	<b>Total</b>	<b>\$1.933.45</b>

## CAPITULO IV

### 1. RESULTADOS ESPERADOS

#### 4.1 Análisis resultados FASE I

En la primera fase del proyecto abarcó todo lo relacionado a componentes y equipos físicos. Los resultados de esta fase fueron de manera satisfactoria, cabe destacar que esta fase es muy esencial e importante porque todos los componentes físicos y periféricos debían estar conectado de manera adecuada para la ejecución correcta del Robot Rosmaster X. Con la guía que viene incluida en la caja del Robot y siguiendo el paso a paso el robot se debería armar sin ningún problema. En la figura 60 muestra la guía otorgada por el ROSMASTER X3 y en la figura 61 muestra el primer encendido del ROSMASTER X3.

#### **Figura 60**

Guía de los elementos del ROSMASTER X3.



**Figura 61**

Primer arranque del ROSMASTER X3.







Finalizado el ensamblaje del robot, se procede a encenderlo verificando mediante la app del robot MakerControl (la descripción de la descarga se encuentra en el folleto disponible para IOS y Android). Mediante la app se verificarán cosas básicas tales como la encendida de la barra de luces, verificar que se mueva el robot con las ordenes indicadas, el funcionamiento correcto de LIAR y la cámara Depth.

Se mostrará cual es procedimiento a seguir para la conexión del robot a la app del celular. En primer lugar, observamos en la pantalla pequeña LED que nos indica la dirección IP del robot que es a la que debemos conectarnos desde nuestro celular siendo 192.168.1.1 y la contraseña 123456789. La configuración de parámetros de los puertos de audio y video se los coloca tal y como aparecen por defecto, una vez insertado los parámetros se podrá iniciar la interfaz de usuario.

**Figura 64**

En la figura 64 muestra la Interfaz de inicio en la APP del ROSMASTER X3.



**Figura 65**

En la figura 65 se muestra los datos a insertar: la IP correcta y los puertos de audio y video.



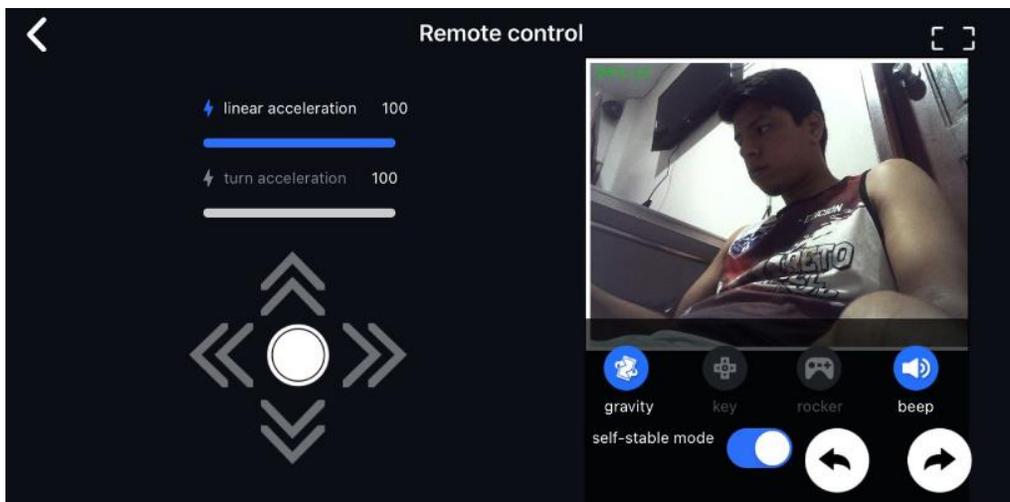
**Figura 66**

Menú del Rosmaster X3.



**Figura 67**

Verificación del correcto funcionamiento de la APP para el manejo del ROSMASTER X3.



Con esto queda lista de manera satisfactoria la primera fase del proyecto dejando demostrado que el robot está listo para su funcionamiento y posterior programación de la tarjeta Jetson Nano para la instalación de la red neuronal.

## 4.2 Análisis resultados FASE II.

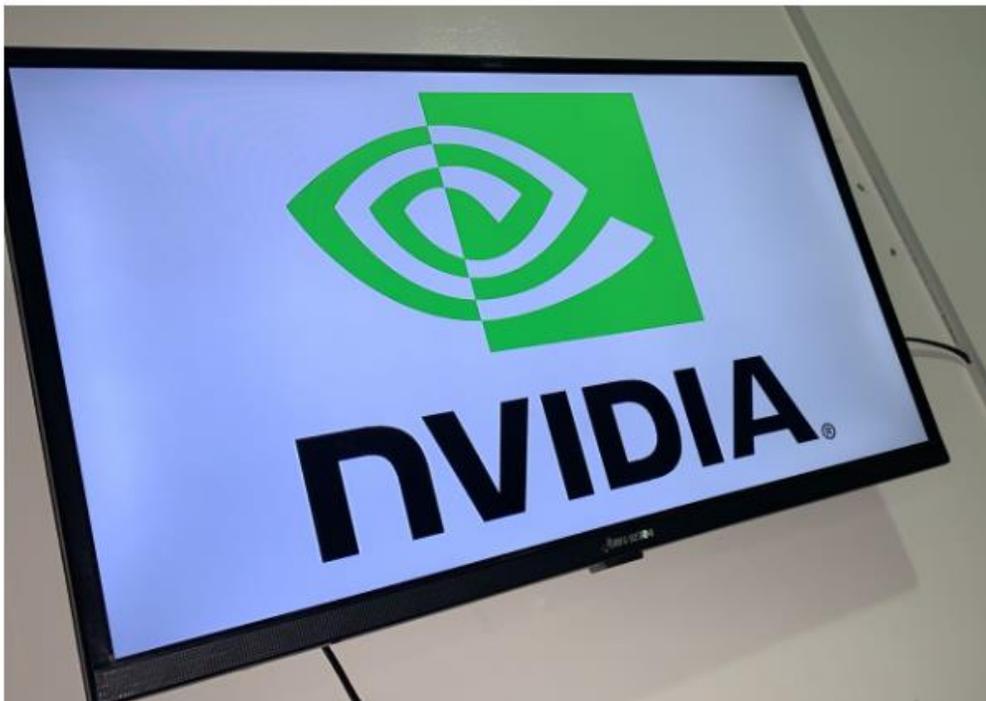
En la siguiente fase del proyecto los resultados obtenidos fueron muy satisfactorios, puesto que se alcanzó a realizar lo que se tenía esperado. Tener en cuenta que esta fase era la más complicada del proyecto debido a que se tenía que programar de manera correcta las

librerías y códigos de la red neuronal entrenada previamente para la detección de objetos en tiempo real.

Como primer punto se entró al entorno de la tarjeta Jetson Nano y familiarizarse con la misma. En una terminal se realizó la configuración y descarga de actualizaciones de la red neuronal. Este paso ya fue explicado en el capítulo anterior. Lo importante de este primer punto era la ejecución de la carpeta de Jetson Inference que es donde se van a guardar todos los archivos, librerías, códigos y banco de imágenes de la red neuronal.

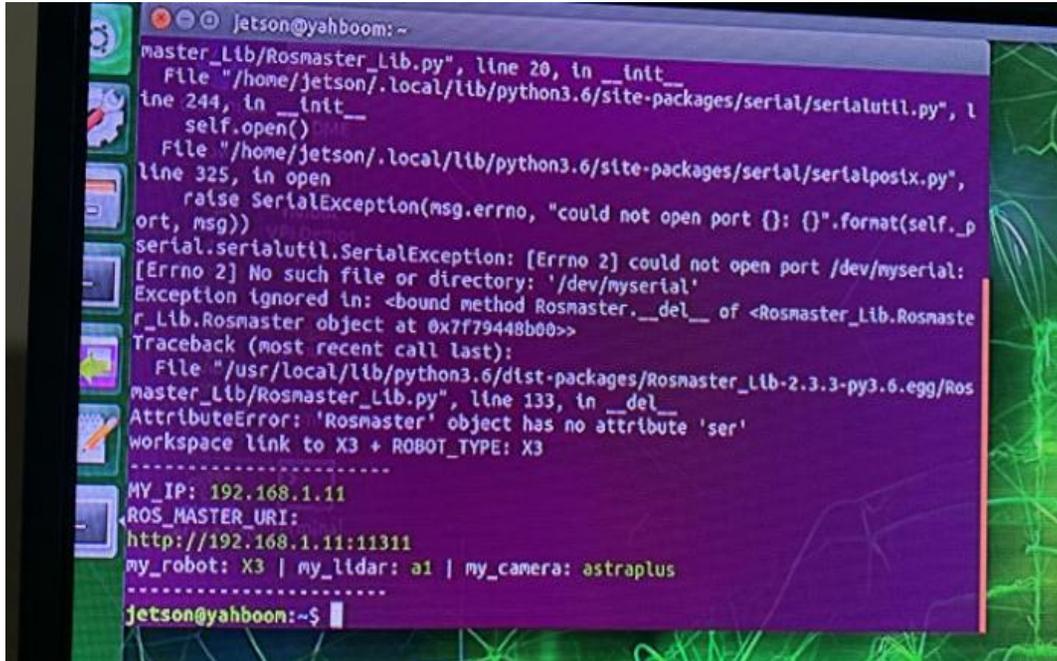
### **Figura 68**

Inicio del entorno Nvidia de la Jetson Nano.



**Figura 69**

Inicio de la Jetson Nano.



```
jetson@yahboom: ~  
master_Lib/Rosmaster_Lib.py", line 20, in __init__  
File "/home/jetson/.local/lib/python3.6/site-packages/serial/serialutil.py", l  
line 244, in __init__  
self.open()  
File "/home/jetson/.local/lib/python3.6/site-packages/serial/serialposix.py",  
line 325, in open  
raise SerialException(msg.errno, "could not open port {}: {}".format(self.p  
ort, msg))  
serial.serialutil.SerialException: [Errno 2] could not open port /dev/myserial:  
[Errno 2] No such file or directory: '/dev/myserial'  
Exception ignored in: <bound method Rosmaster.__del__ of <Rosmaster_Lib.Rosmaste  
r_Lib.Rosmaster object at 0x7f79448b00>>  
Traceback (most recent call last):  
File "/usr/local/lib/python3.6/dist-packages/Rosmaster_Lib-2.3.3-py3.6.egg/Ros  
master_Lib/Rosmaster_Lib.py", line 133, in __del__  
AttributeError: 'Rosmaster' object has no attribute 'ser'  
workspace link to X3 + ROBOT_TYPE: X3  
-----  
MY_IP: 192.168.1.11  
ROS_MASTER_URI:  
http://192.168.1.11:11311  
my_robot: X3 | my_lidar: a1 | my_camera: astraplus  
-----  
jetson@yahboom:~$
```

**Figura 70**

En la figura 70 muestra el código para la detección de objetos.



```
tech.py  
~/jetson-inference/build/aarch64/bin/tech  
import jetson.inference  
import jetson.utils  
  
# Declarar el detector  
net = jetson.inference.detectNet(argv=['--model=tech.onnx', '--labels=labels.txt', '--input-  
>lob=input_0', '--output-cvg=scores', '--output-bbox=boxes'])  
  
:amara = jetson.utils.videoSource("/dev/video0")  
:isplay = jetson.utils.videoOutput()  
  
while True:  
    # Frames  
    img = camara.Capture()  
  
    detect = net.Detect(img)  
  
    if detect:  
        print('DETECCION')  
  
    display.Render(img)  
    display.SetStatus("Deteccion | Network {:.0f}FPS".format(net.GetNetworkFPS()))  
  
    if not camara.IsStreaming() or not display.IsStreaming():  
        break
```

Las instalaciones una vez finalizadas de manera satisfactoria en el robot Rosmaster X3, se debería realizar de manera automática la ejecución de la misma para la detección de objetos en tiempo real. Esto es una innovación puesto que la detección de objetos se realizaba, pero no en tiempo real, previamente se debían tener imágenes o captura de los objetos para el reconocimiento de las mismas. Gracias a las redes neuronales previamente entrenadas se puede hacer reconocimiento de cualquier tipo de objeto en tiempo real. MediaPipe fue la base para la aplicación en PYTHON3 de la misma, se investigó de manera profunda si MediaPipe Development era la base de desarrollo necesario debido a que más se enfoca en reconocimiento humano, sin embargo, se pudo lograr ejecutar el código de reconocimiento de objetos en tiempo real gracias a la codificación de MediaPipe Development dando resultados exitosos en la fase II del proyecto. En la figura 71 muestra los comandos para la activación de la red neuronal.

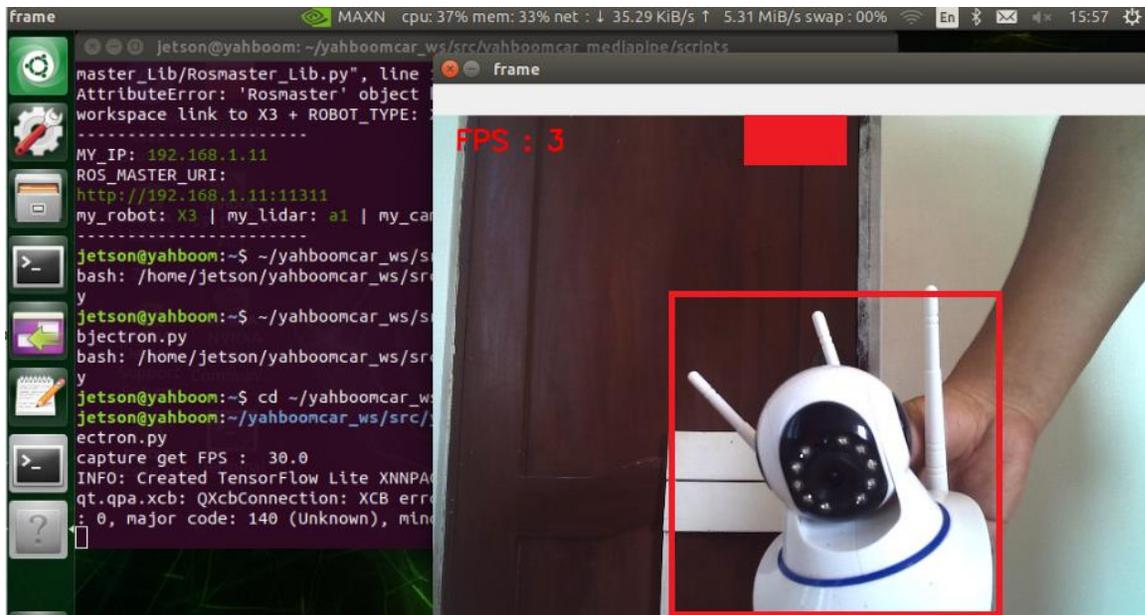
### Figura 71

Llamamiento de comandos para la ejecución final de la red neuronal.

```
roslaunch yahboomcar_mediapipe 03_Holistic.launch # Overall  
detection  
python3 08_Objectron.py # 3D object recognition  
python3 09_VirtualPaint.py # brushes
```

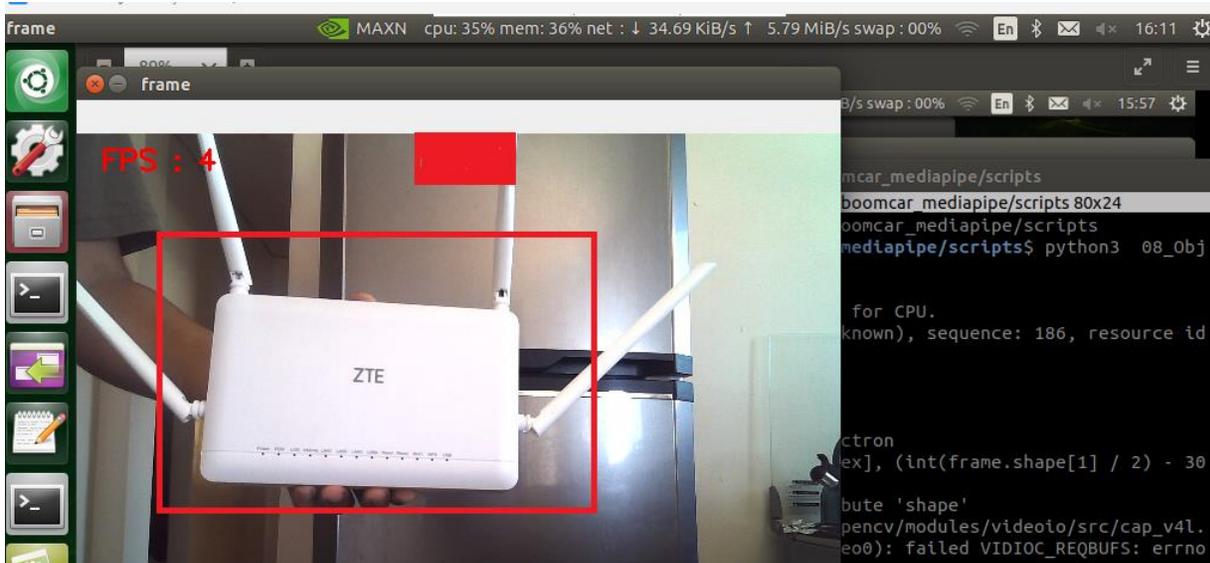
**Figura 72**

Reconocimiento de equipos de telecomunicaciones en tiempo real.



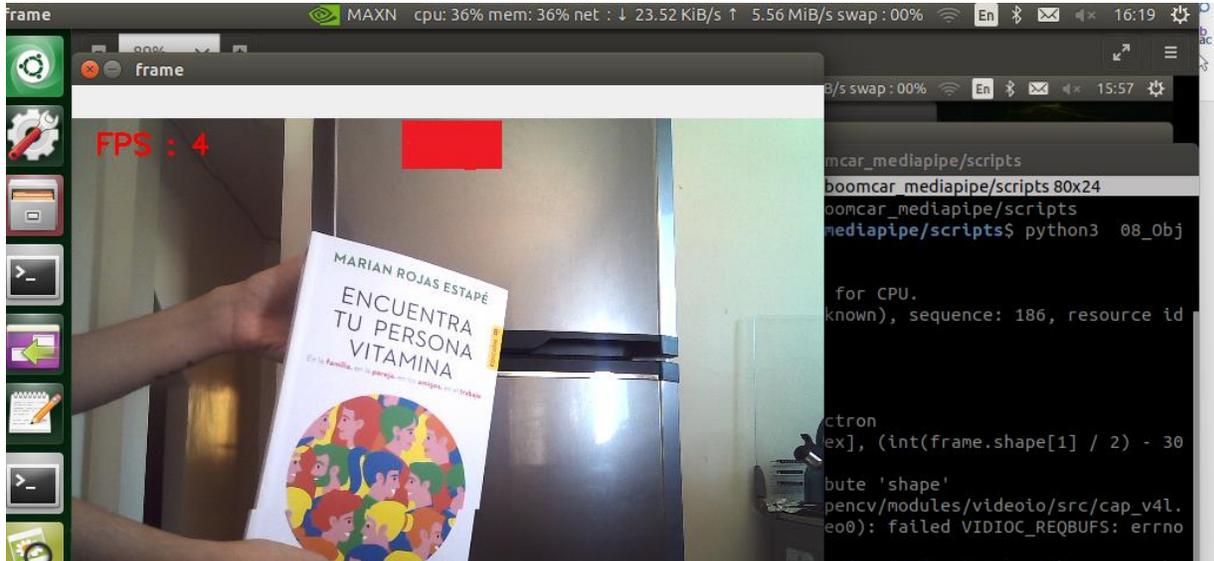
**Figura 73**

Reconocimiento de equipos de telecomunicaciones en tiempo real II.



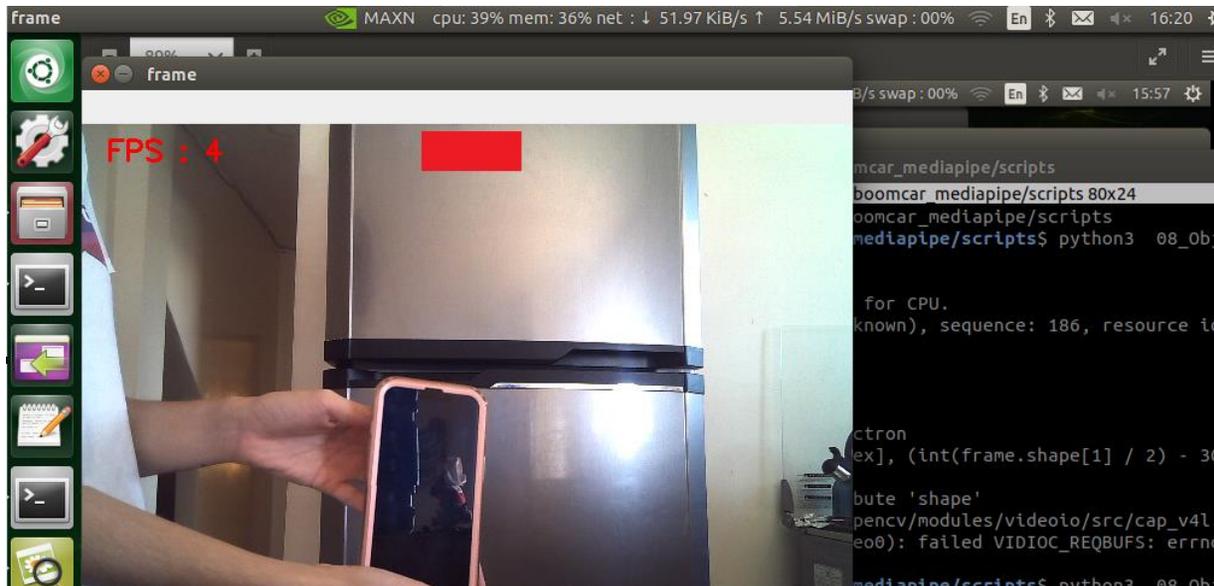
**Figura 74**

No identifica objetos que no hayan sido programados previo a la ejecución de la red neuronal.



**Figura 75**

No identifica objetos que no hayan sido programados en la red neuronal II.



Las capturas muestran el reconocimiento de una cámara y un router en tiempo real que fueron los equipos que se utilizaron en la demostración del proyecto. Adicional otras dos capturas donde

muestra un celular y un libro, estos no fueron reconocidos debido a que no se encontraban codificados en Jetson Inference, ni tampoco había una referencia en el banco de imágenes descargado previamente en los archivos de Jetson Inference. Esta segunda fase fue exitosa con los resultados esperados

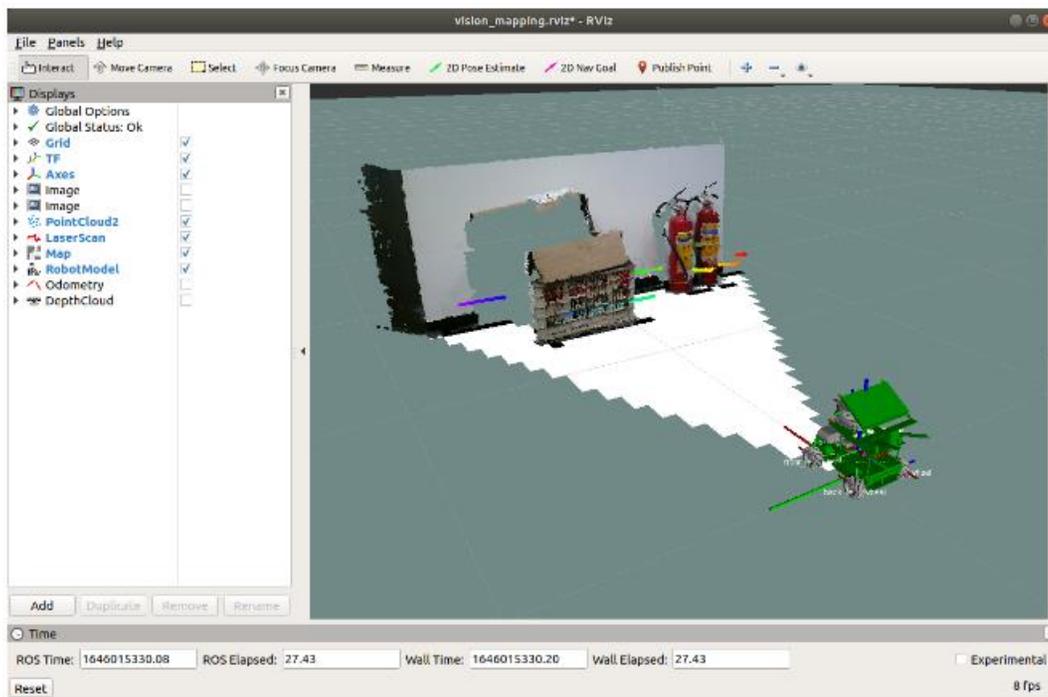
### 4.3 Análisis de resultados FASE III

La fase final de nuestro proyecto constaba en el mapeo de áreas mediante el uso del LIDAR laser y la cámara Depth del robot Rosmaster X3, esta fase consta la codificación para acceder a la cámara y a los diferentes tipos de mapeos en tiempo real, estos códigos se encuentran en el repositorio del Rosmaster X3 (adjunto link al final del documento).

Para la fase III, seleccionamos un tipo de mapeo en 2D sencillo para tener una referencia del entorno y verificar que las herramientas tanto como el LIDAR laser y la cámara Depth estén funcionando de manera correcta. En la figura 76 muestra que los resultados fueron muy satisfactorios en cuanto al mapeo en 2D de nuestro entorno.

**Figura 76**

Mapeo en 2D.

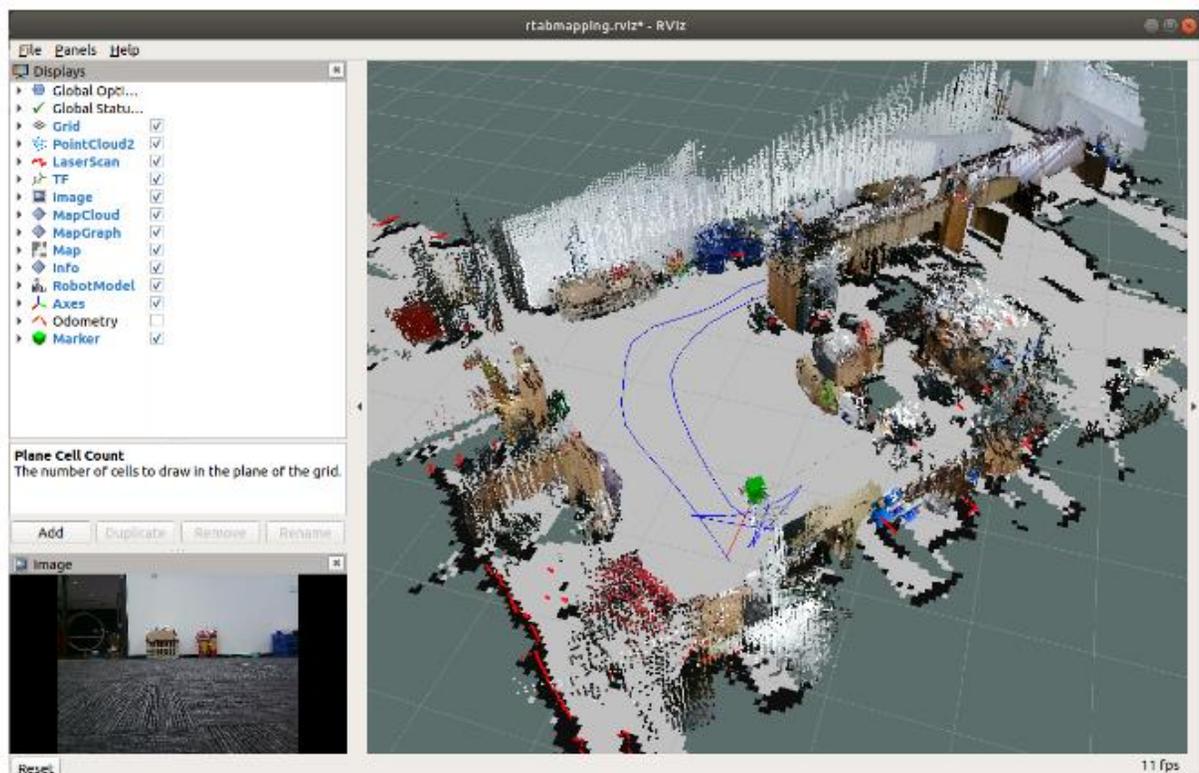


El mapeo se realizó con éxito y quedó guardado el plano en los archivos de la tarjeta Jetson Nano, para su exportación a nuestra laptop o computadora y usarlo en software online WiFi Heatmap el cual se describirán los resultados más adelante.

Se procedió a realizar el mapeo del entorno, pero en 3D para mejores y óptimos resultados, aquí en esta sección si es muy importante las entradas y comandos correctos del LIDAR laser ya que este es el que nos da una percepción más precisa del entorno en donde se está realizando el mapeo.

### Figura 77

Mapeo en 3D.



De la misma manera una vez finalizada de manera correcta el mapeo en 3D de forma automática o por medio de comandos se guardará el mapa que se registró todo esto para la exportación final a nuestra laptop o PC para llevarlo al software WiFi Heatmap.

#### 4.3.1 Comparación de resultados pre y post uso del software WiFi Heatmap.

De manera previa se habían ubicados los equipos en zonas alejadas entre si para evitar que abarquen tanta cobertura y mostrar la comparación de resultados finales al momento de usar el software y optimizar el uso de los equipos y su ubicación. Los resultados de velocidad previo a la ubicación correcta de los equipos fueron los siguientes. El paquete de internet en donde se llevaron a cabo las pruebas era de 150 Mbps, siendo esta una buena velocidad con un buen ancho de banda, sin embargo, al colocar de manera arbitraria los equipos para empeorar el rendimiento de la red los resultados fueron los siguientes. En la tabla 5 muestra los datos de rendimiento de la red con los equipos colocados en mala ubicación. En la tabla 6 muestra los datos del rendimiento de la red con los equipos ubicados en la ubicación mas óptima.

**Tabla 5**

**Tabla de valores del rendimiento de la red primera prueba.**

Velocidad de internet	46 Mbps
Ping – latencia	112 ms
Rango de cobertura	Media – baja (80%)
Perdida de paquetes	12%

Como se pueden observar son datos malos para la cantidad de megas que tiene el paquete que da el proveedor. Por esta razón se llevó el plano al software WiFi Heatmap y este de manera automática nos da la mejor ubicación para nuestros equipos como lo son el router principal, el punto de acceso y una cámara IP. Los resultados obtenidos posterior al cambiar de posición los equipos fueron los siguientes.

**Tabla 6**

**Tabla de valores del rendimiento de la red con la ubicación de equipos optimizados.**

Velocidad de internet	98 Mbps
Ping – latencia	15 ms
Rango de cobertura	ALTA (95%)
Perdida de paquetes	0%

## Figura 78

Zona de cobertura optimizada.



### 4.4 Resultados finales

Los resultados finales se pudieron cumplir referente a los objetivos secundarios y los resultados esperados, las tres fases fueron realizadas de manera exitosa. En la fase I se ensambló el robot de manera correcta para su correcto funcionamiento, se configuró el entorno en el que se iba a trabajar y la app para el funcionamiento del mismo en conjunto con las tres tarjetas principales como lo son la de expansión principal, la de puertos USB y la tarjeta Jetson Nano de 4 Gb, el uso correcto de la cámara Depth y el LIDAR laser que en conjunto a las siguientes fases serían de vital importancia para el funcionamiento del mismo. en la fase II se codificó de manera exitosa Jetson Inference para el soporte de la red neuronal previamente entrenada dando los resultados esperados en la identificación de objetos de telecomunicaciones. Y por último la fase III era el mapeo de entornos controlados gracias a la cámara Depth y el LIDAR laser que finalmente los planos fueron exportados al software WiFi Heatmap donde de manera automatizada nos arrojaba la mejor ubicación de los equipos con los que se habían identificado previamente en la fase II para abarcar mejor cobertura de rango, evitar interferencias y mejorar la calidad del usuario en cuanto a navegación por la red.

Finalmente, demostrada cada una de las fases está en llevar a cabo la fase II y la fase III a la facultad de sistemas y telecomunicaciones para la mejorar la percepción visual de las herramientas y recursos en el área de telecomunicaciones y mejor la calidad de la red que se le da al usuario final que serían los estudiantes de la universidad, de la misma manera con el reconocimiento de objetos en tiempo real se puede llevar un mejor inventario de los equipos que sobre todo están en los laboratorios de telecomunicaciones. Se plantearía realizar un mapeo general en cada uno de los edificios de la facultad tales como los laboratorios, sala de profesores y el decanato de la facultad y verificar el uso correcto de los equipos para evitar zonas de baja cobertura, uso de equipos muy cercanos que merman la calidad y rendimiento de los mismo al existir interferencia.

Esta investigación y aplicación de la misma queda para futuras prácticas en la que se la puede llevar a cabo de manera general en toda la universidad, es decir en cada una de las facultades y al desarrollo de mejores y nuevos códigos que ayudaran a la comunidad universitaria en un futuro.

## **CAPITULO V**

### **CONFIGURACIÓN INICIAL ROSMASTER X3 Y JETSON NANO 4GB**

#### **5.1 FICHA TECNICA ROSMASTER X3**

Este capítulo proporciona una ficha técnica exhaustiva que abarca tanto el hardware como el software, así como las características fundamentales que hacen del ROSMaster X3 una herramienta versátil para diversas aplicaciones en investigación, educación y desarrollo de soluciones robóticas.

##### **5.1.1 Información general**

Nombre del robot: ROSMASTER X3

Modelo: ROSMASTER X3 ROS 2

Fabricante: Laboratorio de inteligencia artificial de Stanford

### 5.1.2 Descripción general

ROSMaster X3 se presenta como un robot educativo que utiliza el sistema operativo robótico y está equipado con ruedas Mecanum. Es compatible con dispositivos como Jetson NANO/Orin NX/Orin NANO y Raspberry Pi 4B. Además, cuenta con características avanzadas, como lidar, cámara de profundidad y un módulo de interacción por voz, entre otros componentes de hardware de alto rendimiento. A través de la programación en Python, ROSMaster X3 es capaz de realizar diversas funciones, que incluyen mapeo y navegación, seguimiento o evasión de obstáculos, autopilotaje y la detección de posturas del cuerpo humano.

### 5.1.3 Hardware

Procesador: NVIDIA JETSON

Sensores incorporados: Laser Lidar, cámara Astra pro Depth, tarjeta de expansión

Motores y servos: Ruedas Mecanum, servo motores incorporados en las llantas conectados a la tarjeta.

Batería: Batería de litio recargable de 12v.

### 5.1.4 Software

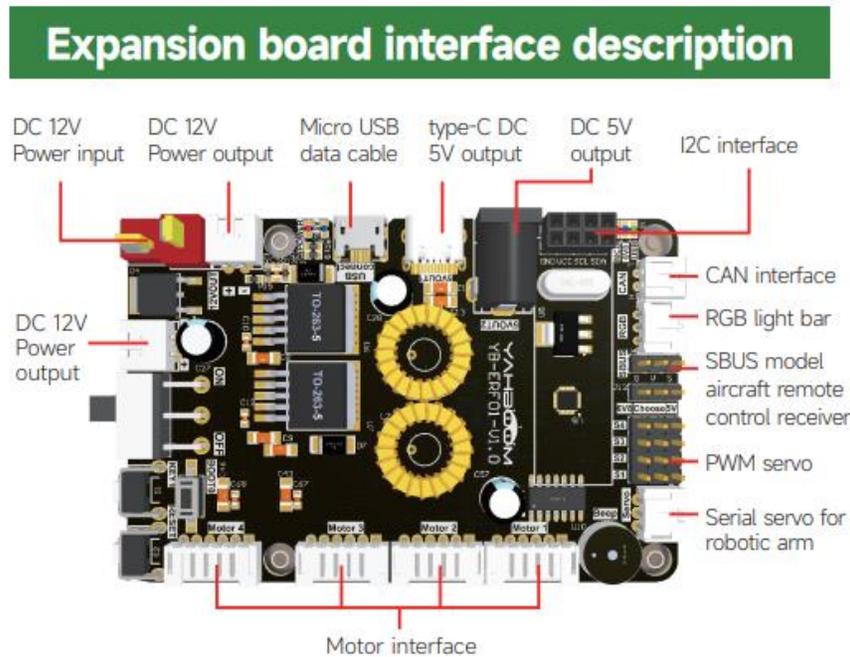
Versiones ROS: Compatible con ROS2.

Sistema operativo: Ubuntu o Linux.

## 5.2 Tarjeta de expansión

La tarjeta de expansión que viene dentro del paquete del ROSMASTER X3 es aquella que nos va ayudar a conectar todos los periféricos correspondientes necesarios del ROSMASTER X3. A continuación una imagen de referencia de la tarjeta de expansión con todos sus componentes y entradas. En la figura 79 muestra la tarjeta de expansión con todas sus entradas

**Figura 79**  
**Tarjeta de expansión.**



### 5.3 PID del movimiento del robot ROSMASTER X3

En el contexto del movimiento de objetos, el controlador PID (Proporcional, Integral y Derivativo) se utiliza para regular y ajustar la posición de un objeto en movimiento. Este controlador es esencialmente un algoritmo de retroalimentación que busca minimizar el error entre la posición deseada y la posición real del objeto.

**Proporcional (P):** Controla la magnitud de la respuesta del sistema en proporción al error actual. A medida que el objeto se desplaza lejos de la posición deseada, el control proporcional ajusta la velocidad para reducir este error.

**Integral (I):** Maneja la acumulación de errores a lo largo del tiempo. Ayuda a corregir desviaciones persistentes entre la posición real y la deseada. Esto es crucial para superar problemas como el deslizamiento o la fricción que pueden afectar el movimiento a lo largo del tiempo.

**Derivativo (D):** Anticipa la tendencia del error, ayudando a suavizar la respuesta del sistema y evitar oscilaciones indeseadas o inestabilidades en el movimiento del objeto.

En el movimiento de objetos, el controlador PID se implementa en sistemas de control automático, como robots móviles o brazos robóticos, para lograr trayectorias precisas y suavizar la respuesta del movimiento. Ajustando los parámetros P, I y D, se busca optimizar la eficiencia del control para adaptarse a las características específicas del sistema y del entorno en el que se encuentra el objeto en movimiento.

PID se utiliza para realizar la proporción integral derivada en la desviación de entrada, y los resultados de la superposición de la operación se utilizan para controlar el actuador. La fórmula es la siguiente:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

Donde:

- $u(t)$  es la salida del controlador en función del tiempo  $t$ .
- $K_p$  es la ganancia proporcional.
- $e(t)$  es el error en función del tiempo  $t$ .
- $T_i$  es el tiempo integral.
- $T_d$  es el tiempo derivativo.

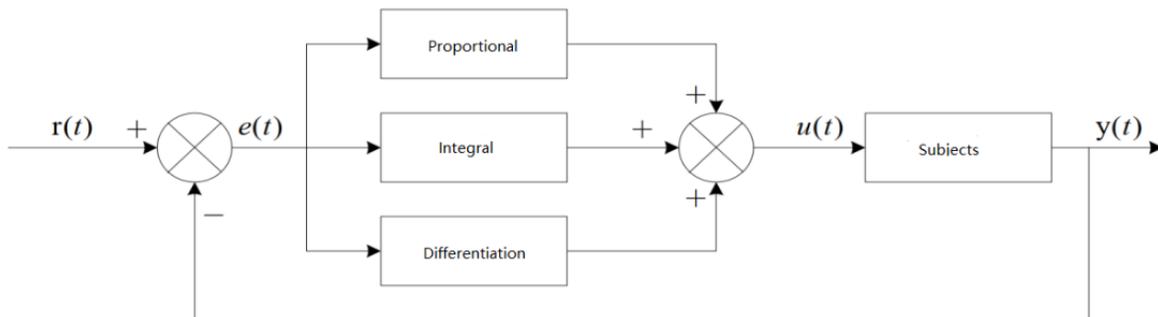
La fórmula describe cómo se calcula la señal de control  $u(t)$  en un controlador PID en función del error  $e(t)$ , la integral del error a lo largo del tiempo y la derivada del error con respecto al tiempo. Este tipo de controladores PID se utiliza comúnmente en sistemas de control para ajustar y mantener la respuesta del sistema a un valor deseado.

En la siguiente figura de diagrama de bloques describe el diagrama de bloques de un controlador PID en robots de ROS muestra cómo el sistema utiliza la retroalimentación de posición, la referencia deseada y los componentes proporcional, integral y derivativo para generar un comando de control que ajusta la posición del robot según lo deseado. Este enfoque es fundamental para

lograr movimientos precisos y estables en entornos robóticos. En la figura 80 muestra el diagrama de bloques completo de cómo funciona el PID de movimiento del ROSMASTER X3.

**Figura 80**

*Diagrama de bloques PID.*

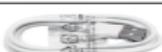


#### 5.4 Ensamblaje del ROSMASTER X3.

El ROSMASTER X3 cuenta con su propio manual para la instalación de sus piezas de forma específica. A continuación, se proporcionarán los elementos que viene dentro del paquete del ROSMASTER X3 y el paso a paso para armar de manera correcta el ROSMASTER X3.

- Detallado de las piezas que conforman el ROSMASTER X3.

**Figura 81**

	Pendular suspension bracket-1		Anti-collision beam
	Motor base plate		Main controller fixed plate
	Pendulum suspension bracket-2		RGB strip fixed bracket
	Lidar fixed plate		Robot expansion board
	USB HUB expansion board		Motor*4
	OLED		Coupling*4
	LED strip		Several cables
	Data line		Screwdriver
	Handle+ AAA battery		Battery
	USB 3.0		Charger
	Parts kit		Handle mobile phone holder
	Plastic tire*4 <small>(DX1 version)</small>		Mecanum wheel*4 <small>(DX3 version)</small>

- Elementos del Laser Lidar  
**Figura 81**

	SLAM A1 Lidar		Micro USB data cable (right bend)
	No.6 screw pack		

- Elementos de la cámara ASTRA PRO DEPTH.  
**Figura 82**

	Astra Pro depth camera		Fixed bracket
	No.5 Screw pack		

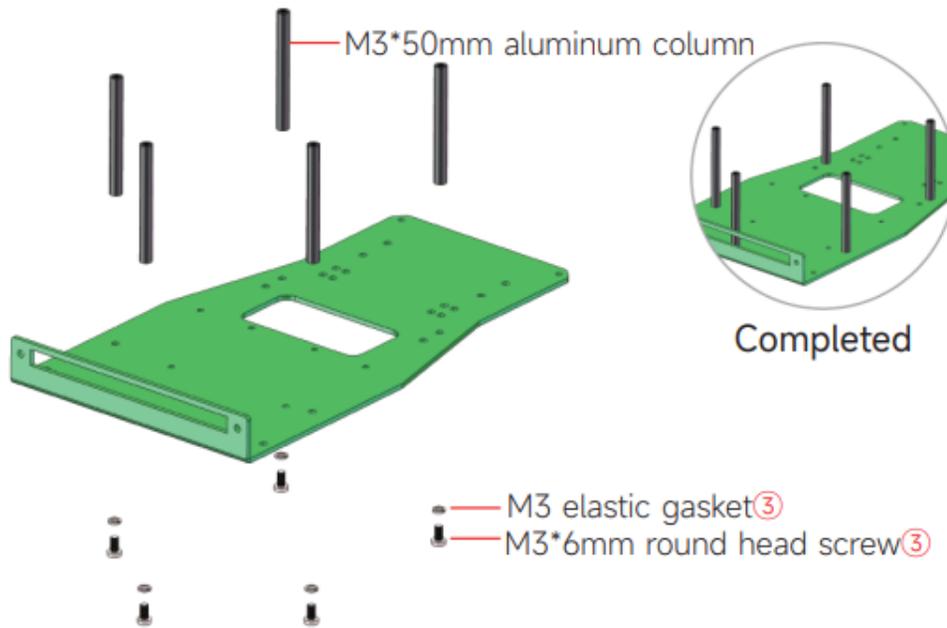
- Elementos de la tarjeta Jetson Nano.

**Figura 83**

	Jetson NANO 4GB (optional)		U Disk
	(NANO 4GB) parts package		4010 cooling fan
	M.2 Antenna		DC Power cable

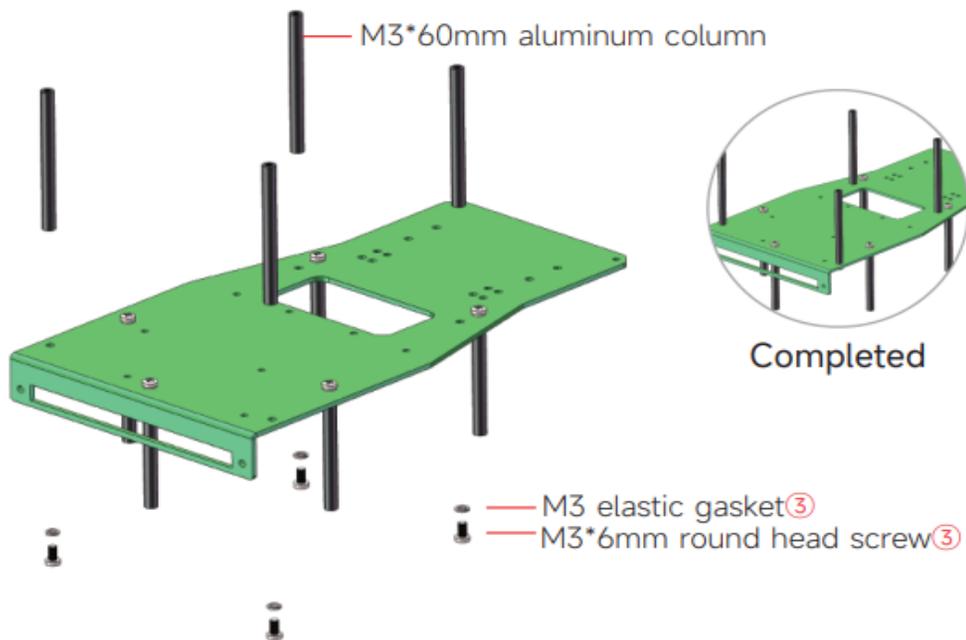
1) Montaje de las columnas de aluminio de 50 mm M3.

**Figura 84**



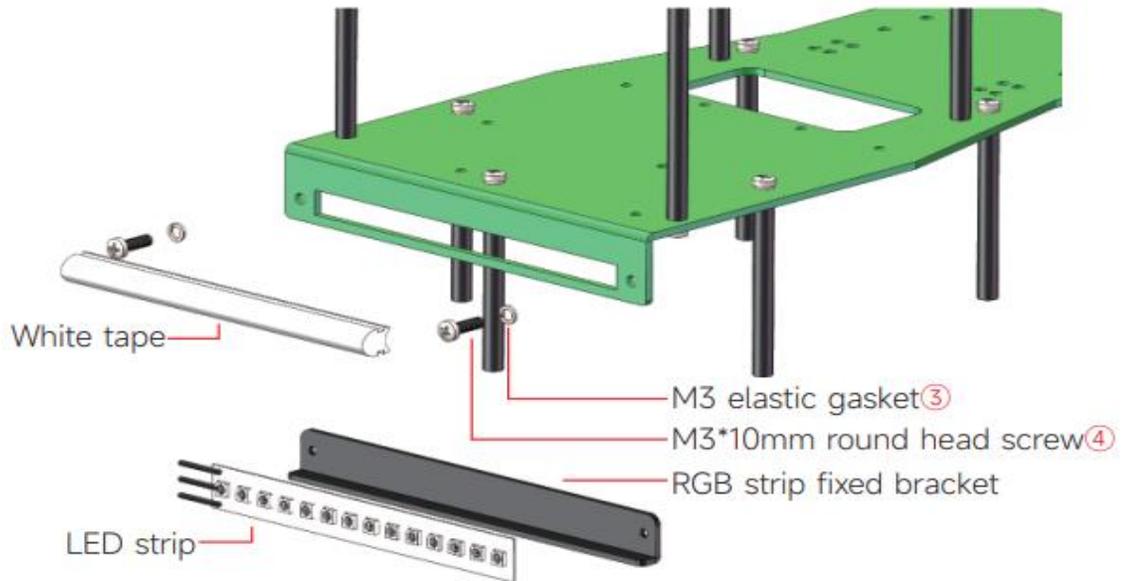
2) Se procede a instalar las columnas de aluminio de 60 mm.

**Figura 85**



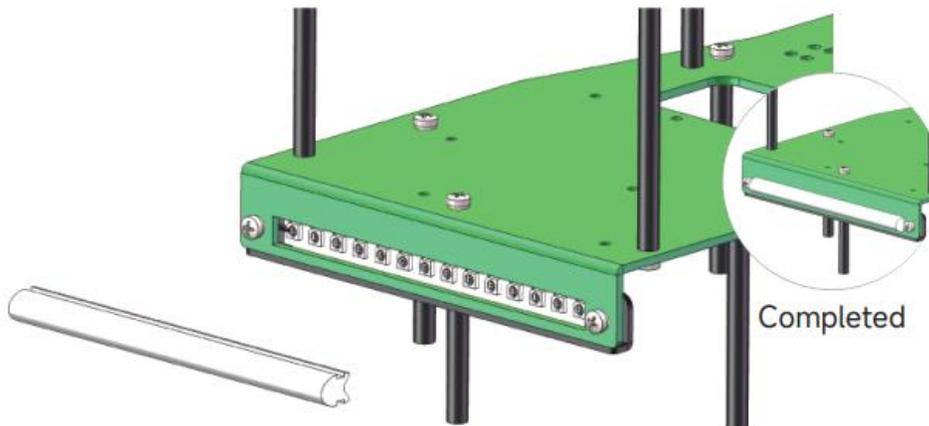
3) Instalación de la barra de luz led.

**Figura 86**



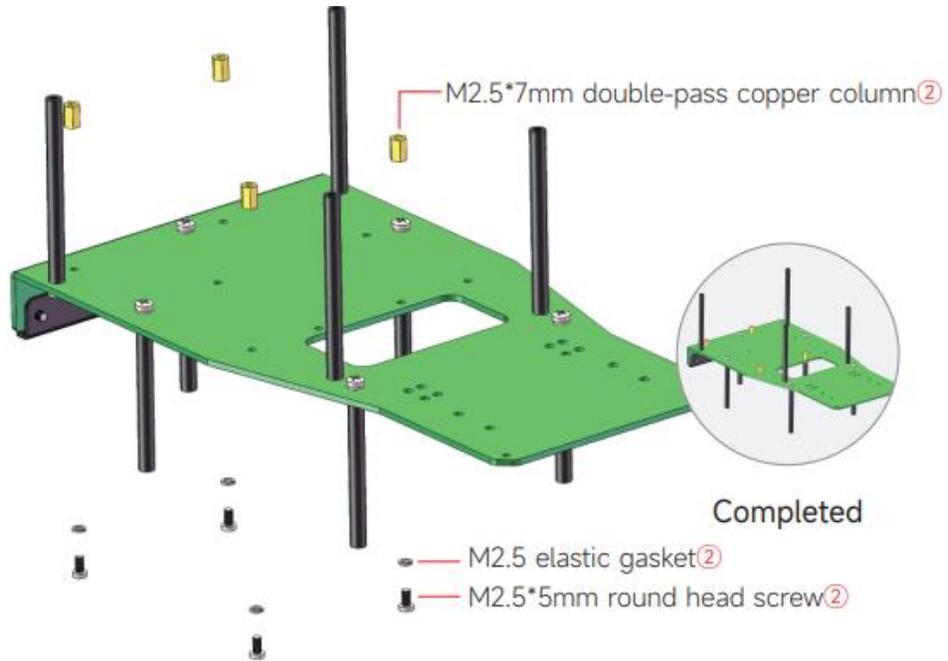
- 4) Finalizada la instalación del soporte y la barra de luz led debería quedar de la siguiente manera.

**Figura 87**



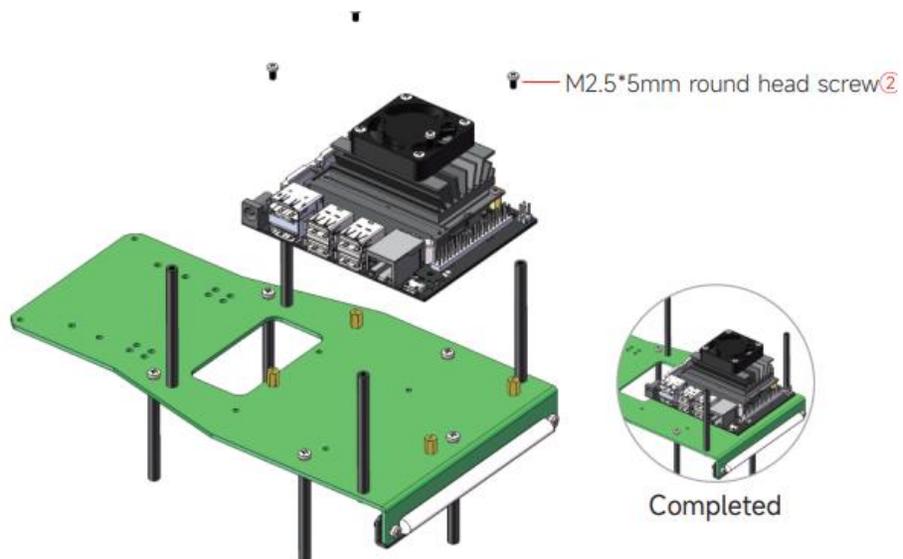
- 5) En la base previamente armada del paso número 2 se procederá a colocar la tarjeta Jetson Nano, utilizando los pernos de 5.5 mm y colocando los soportes de cobre en las barras de aluminio para el respectivo soporte de la tarjeta.

**Figura 88**



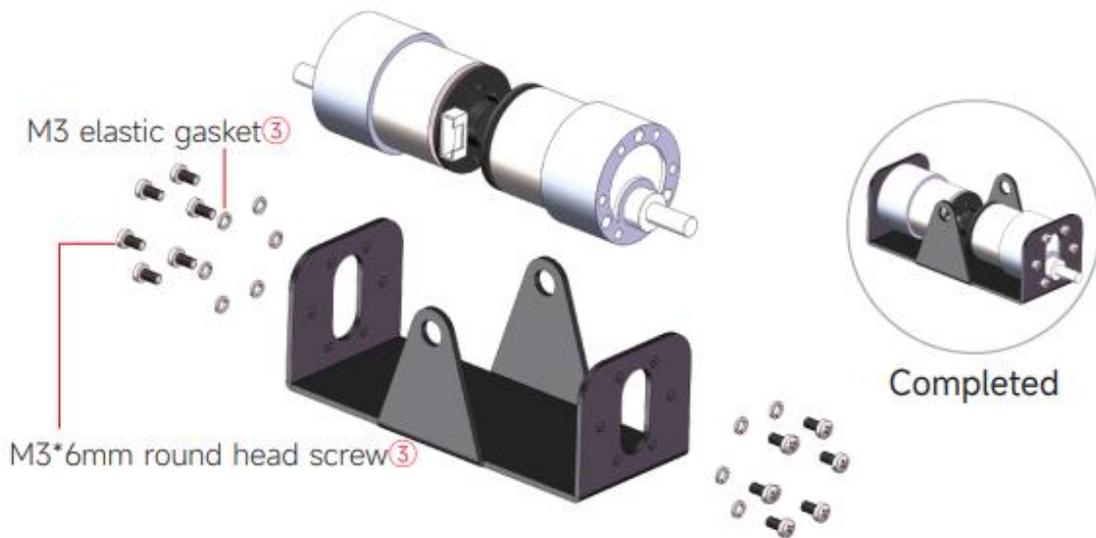
- 6) Una vez instaladas los pernos y las barras de cobre en las columnas de aluminio el montaje de la tarjeta debería quedar de la siguiente manera.

**Figura 89**



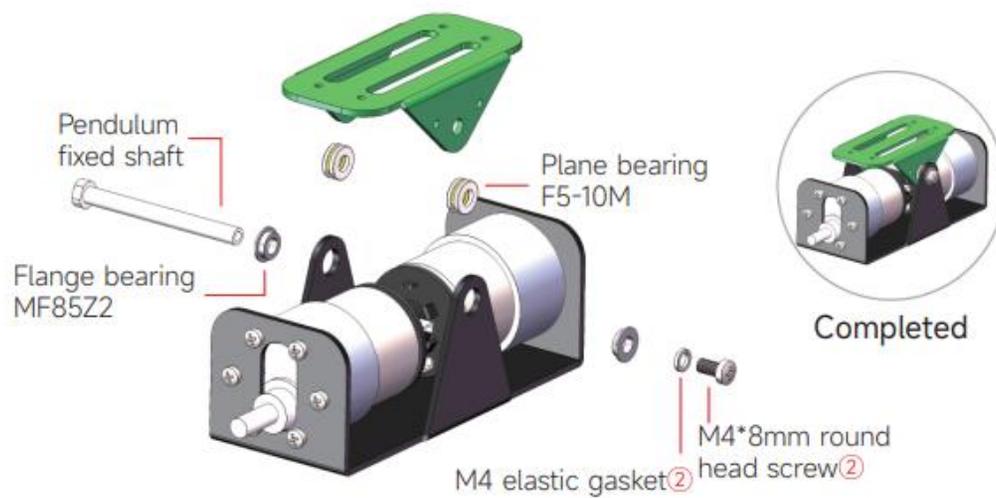
7) Instalación de los péndulos de suspensión bracket – 1 y motores

**Figura 90**



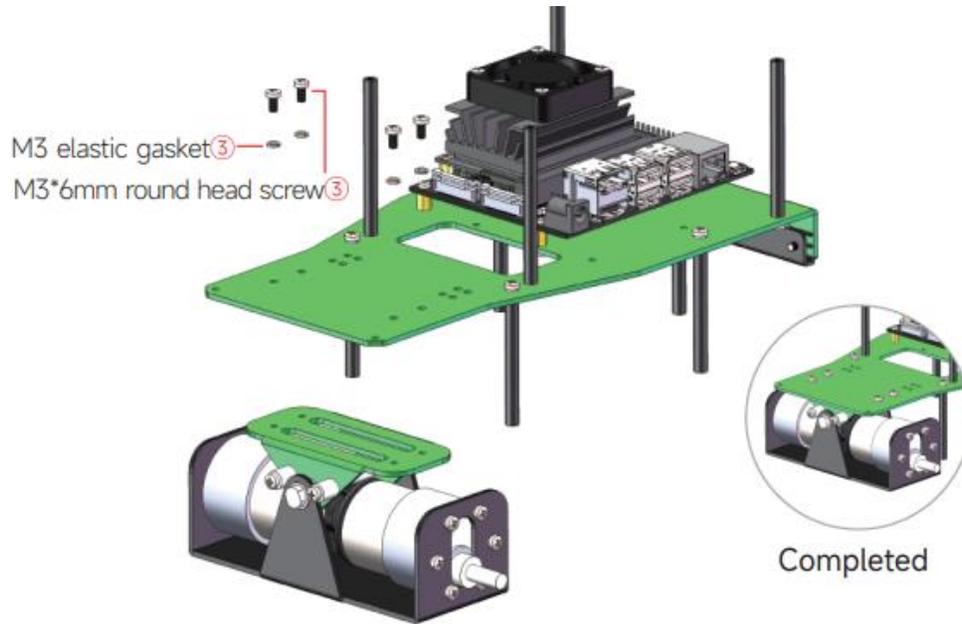
8) Se procede a realizar la instalación del bracket – 2.

**Figura 91**



9) Una vez armados los péndulos de suspensión se proceden a colocar en la placa principal de la siguiente manera.

**Figura 92**



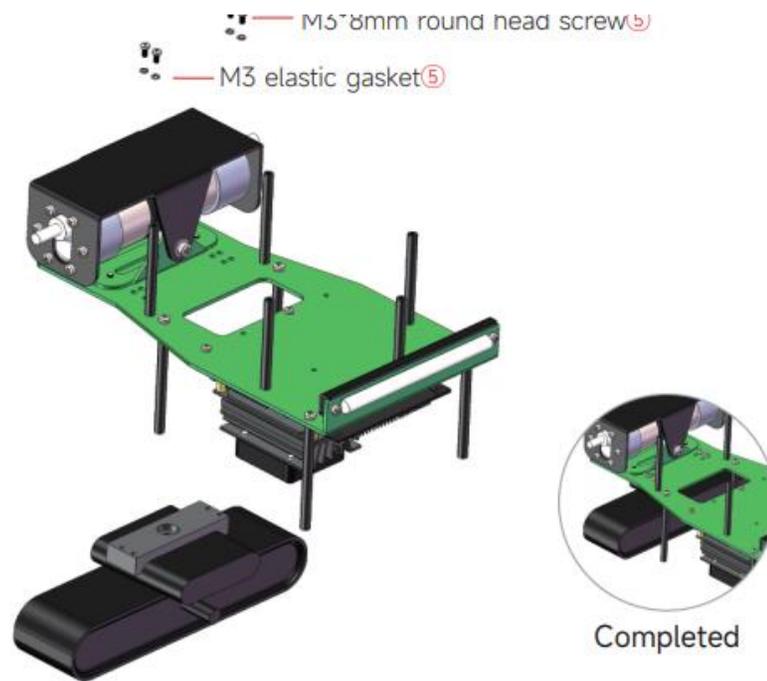
10) Instalación de la cámara Astra Pro, procedemos a entornillar la base de la cámara que se debe colocar en la placa principal del ROSMASTER X3.

**Figura 93**



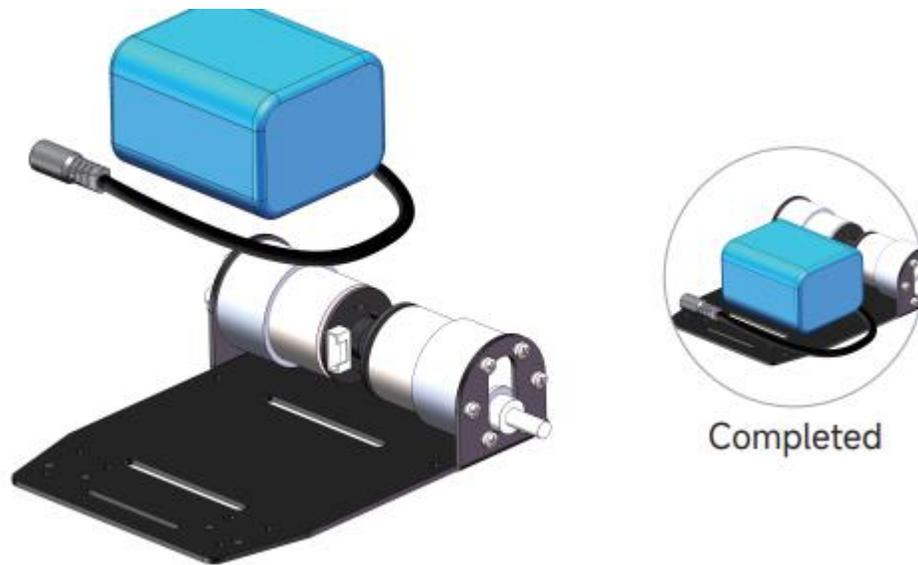
11) Una vez terminada de armar la cámara, se debe voltear la placa para entornillar y postrar la cámara en la placa de la siguiente manera.

**Figura 94**



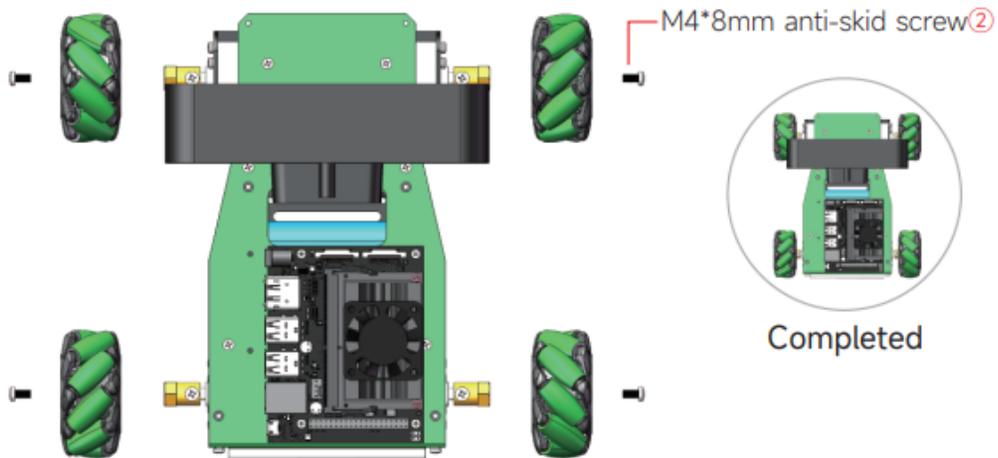
12) Procedemos a insertar la batería en la parte de debajo de la placa del ROSMASTER X3.

**Figura 95**



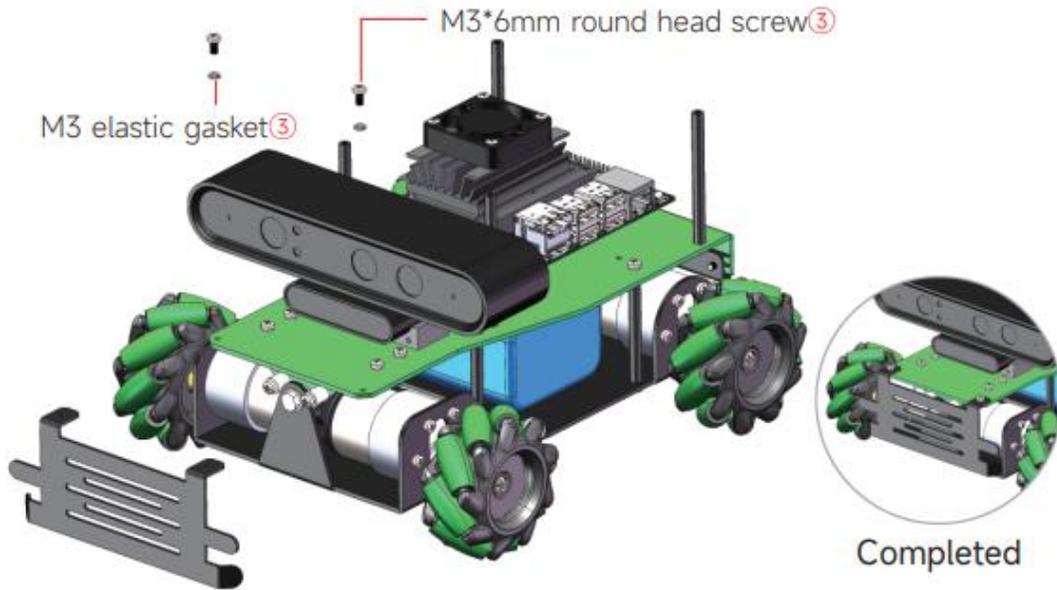
13) El siguiente paso ya será montar las llantas mecánicas que anteriormente fueron ensamblados con los péndulos de suspensión.

**Figura 96**



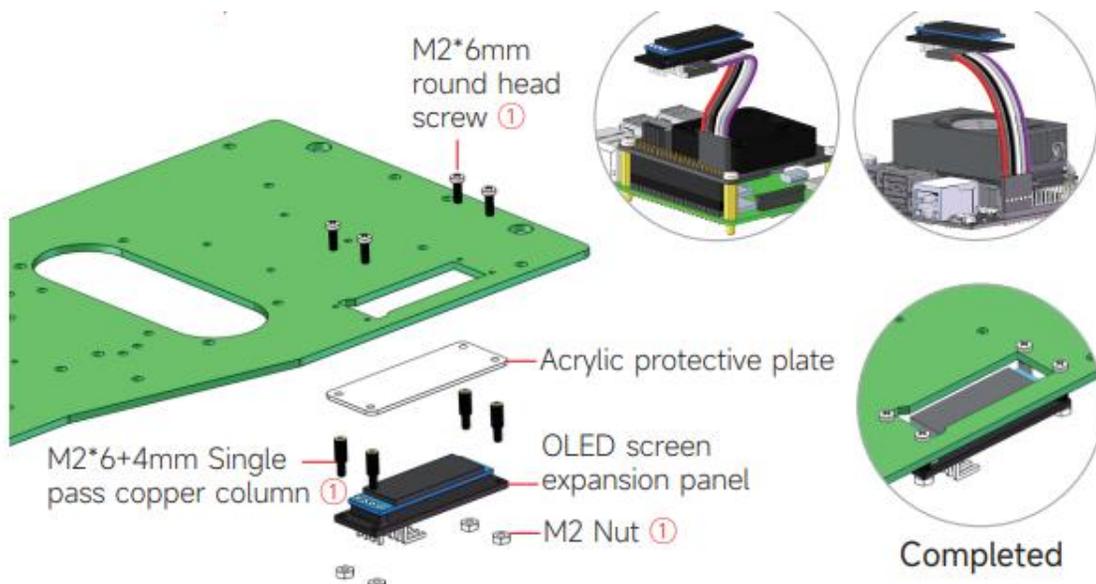
14) Colocación de la parrilla antichoque.

**Figura 97**



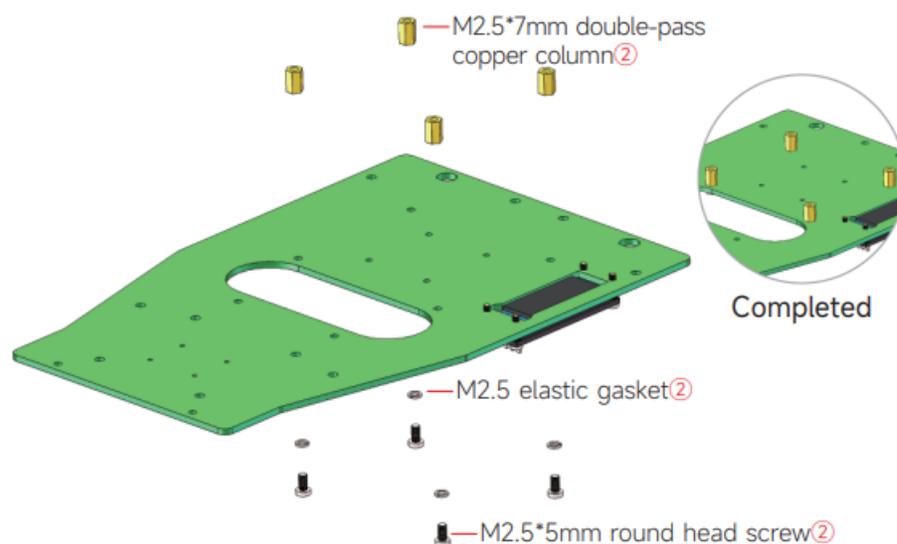
15) En el siguiente está la instalación de la pantalla led, la cual es la que refleja el direccionamiento IP del robot y los datos generales en cuanto a memoria RAM, GPU y batería del mismo.

**Figura 98**



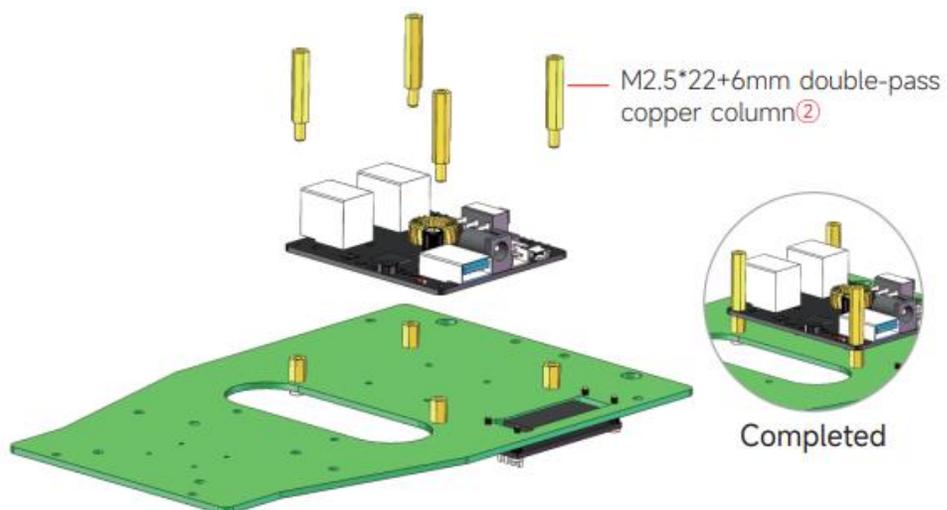
16) Con la instalación de la pantalla LED se instaló todas las piezas y accesorios de la parte inferior del robot. A continuación, se deben colocar los pernos de cobre para el ensamblaje de la placa superior que contendrá a las tarjetas de expansión y lidar laser.

**Figura 100**



17) Se procede a colocar la tarjeta de expansión USB en los espacios de la placa.

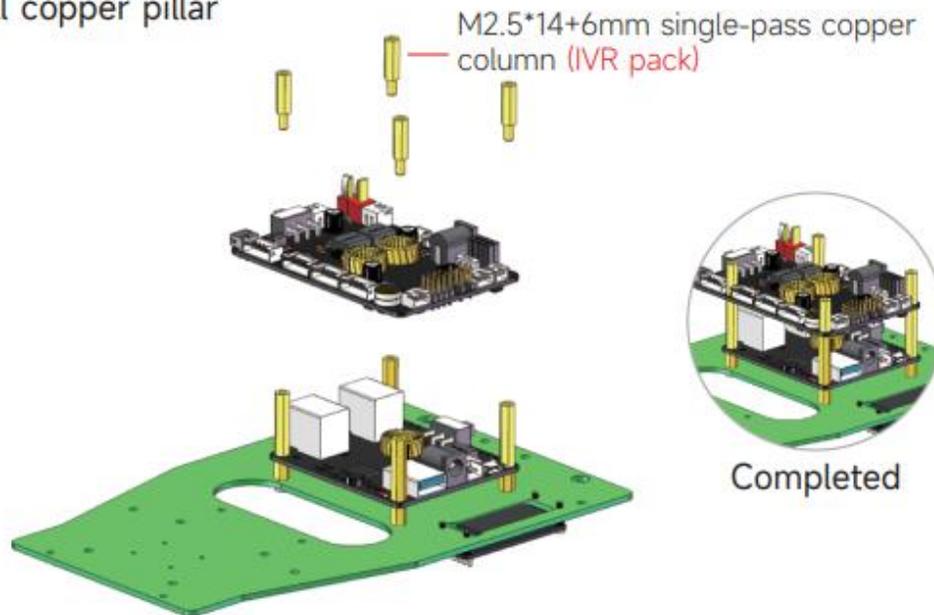
**Figura 101**



18) Finalizada la colocación de la tarjeta de expansión USB, se procede a colocar encima con las bases de cobre la placa de expansión.

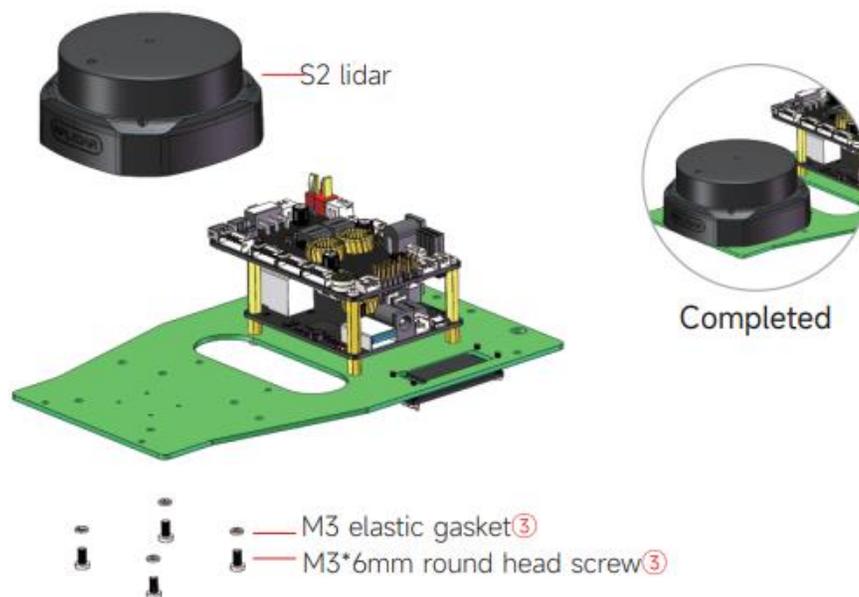
**Figura 102**

① Install copper pillar



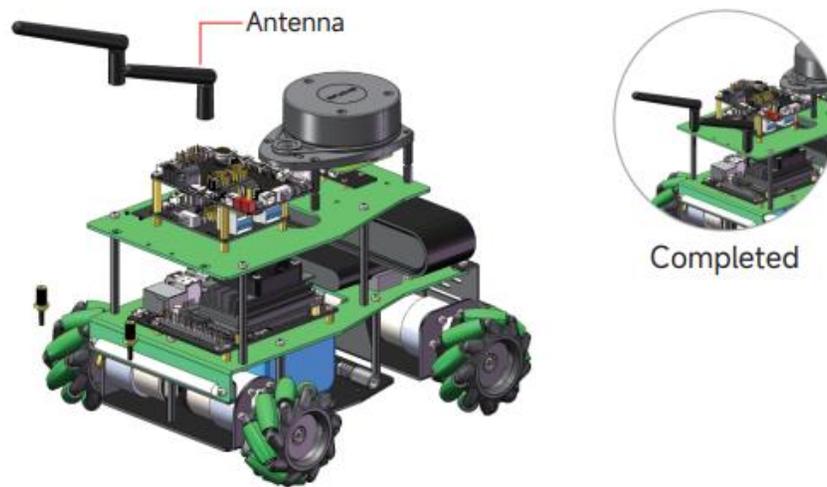
19) En este punto ya casi se tiene listo el ensamblaje completo del ROSMASTER X3, a continuación, se procede a colocar un elemento importante como lo es el LIDAR LASER.

**Figura 103**



20) Como último paso en el armado del robot se colocan las antenas receptoras de la siguiente manera.

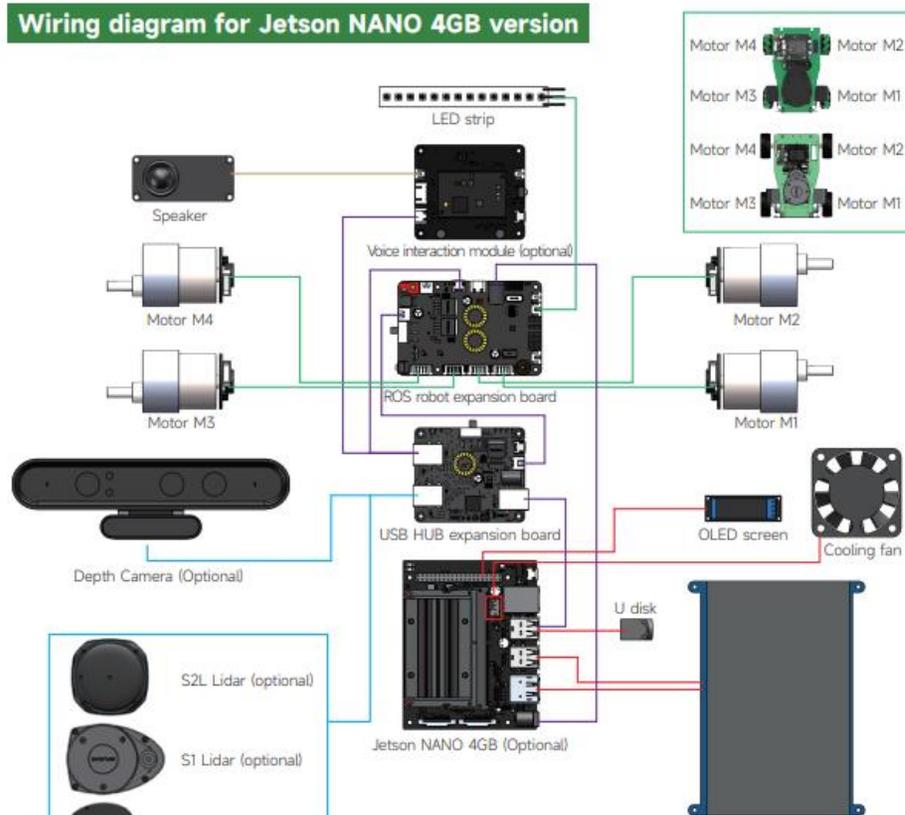
**Figura 104**



## 5.5 Conexiones de tarjetas

En la siguiente imagen de referencia, se mostrará cómo debería ser las conexiones correctas de todas las tarjetas del ROSMASTER X3. Se tiene a la tarjeta de expansión, a la tarjeta de entradas USB y la tarjeta principal, la Jetson Nano de 4 Gb.

**Figura 106**

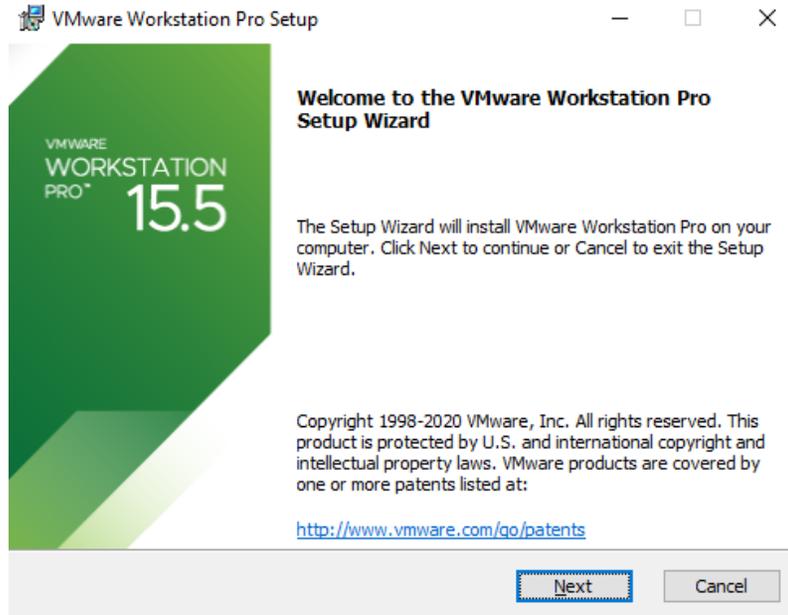


## 5.6 Instalación de VMware (máquina virtual)

Nos dirigimos a la página oficial de VMware Workstation Pro, para la descarga del software de forma gratuita. El software de la máquina virtual es software de paga, sin embargo, el programa regala 30 días de prueba gratis al descargar por primera vez. Se detallarán los pasos necesarios para la instalación de la máquina virtual.

- 1) Luego de haber descargado el software, se debe ejecutar el archivo descargado para realizar la correcta instalación en nuestro ordenador o laptop.

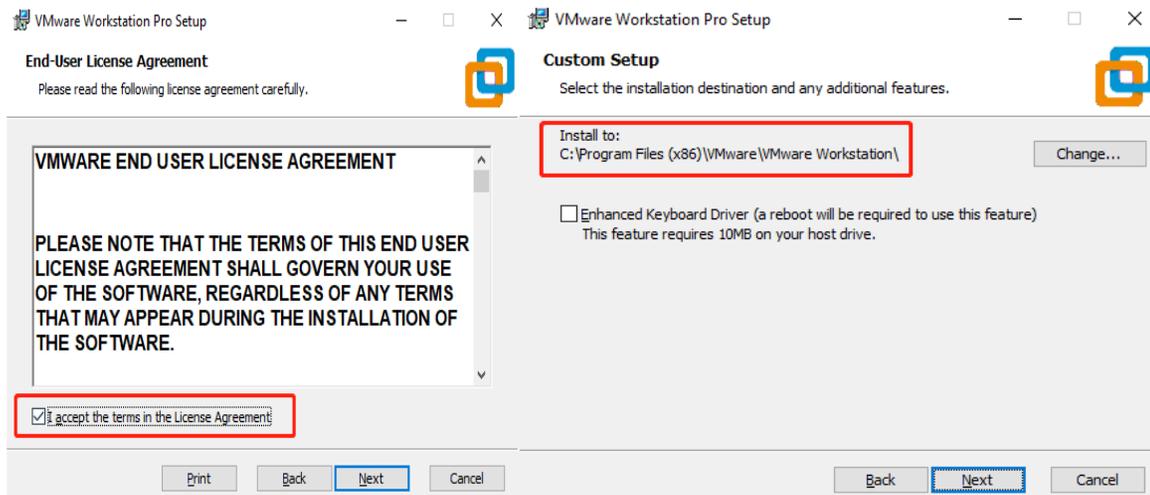
**Figura 107**



2) Aceptar todo los términos y condiciones y elegir la ruta donde se va a guardar nuestro software.

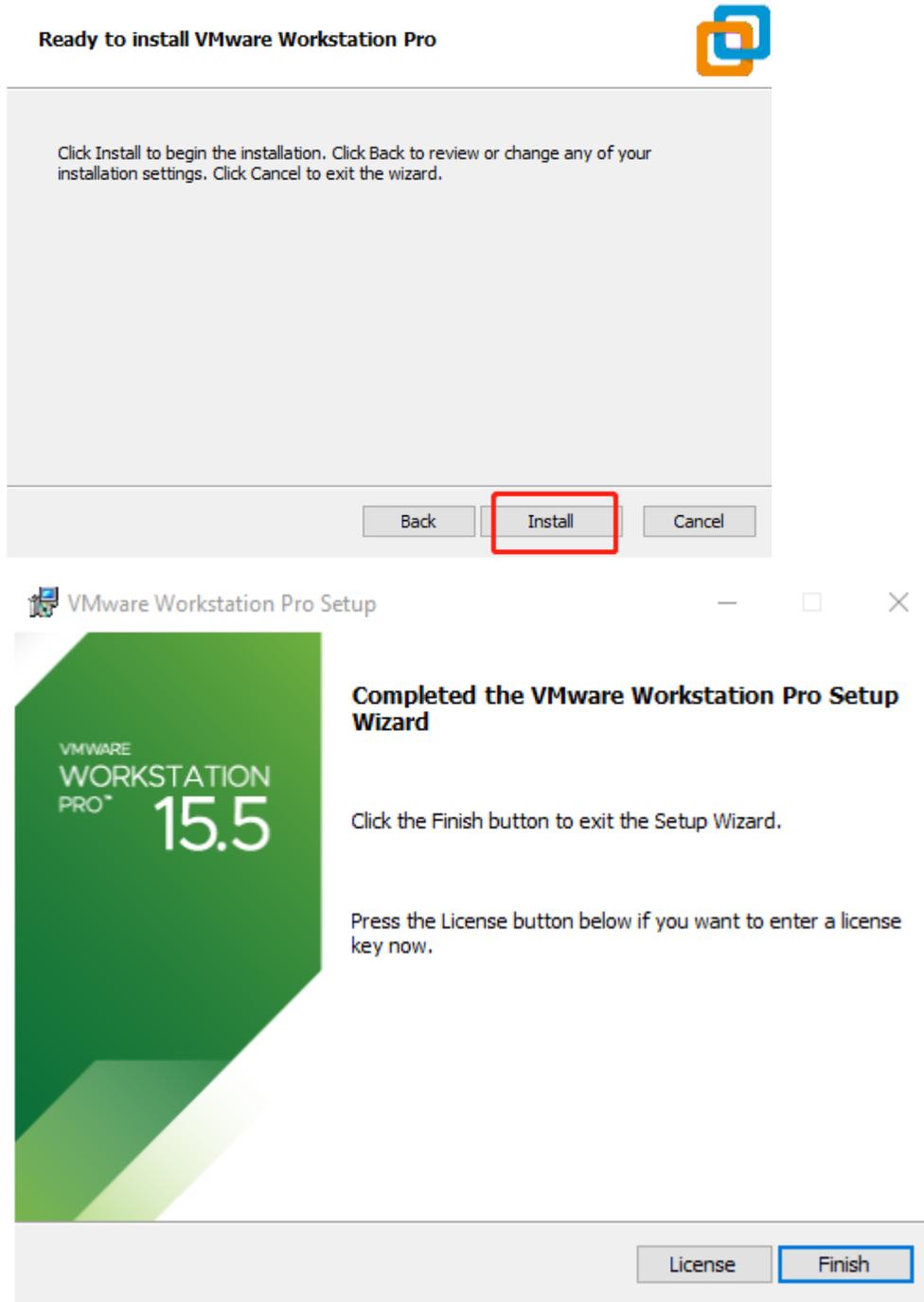
**Figura 108**

**Figura 109**



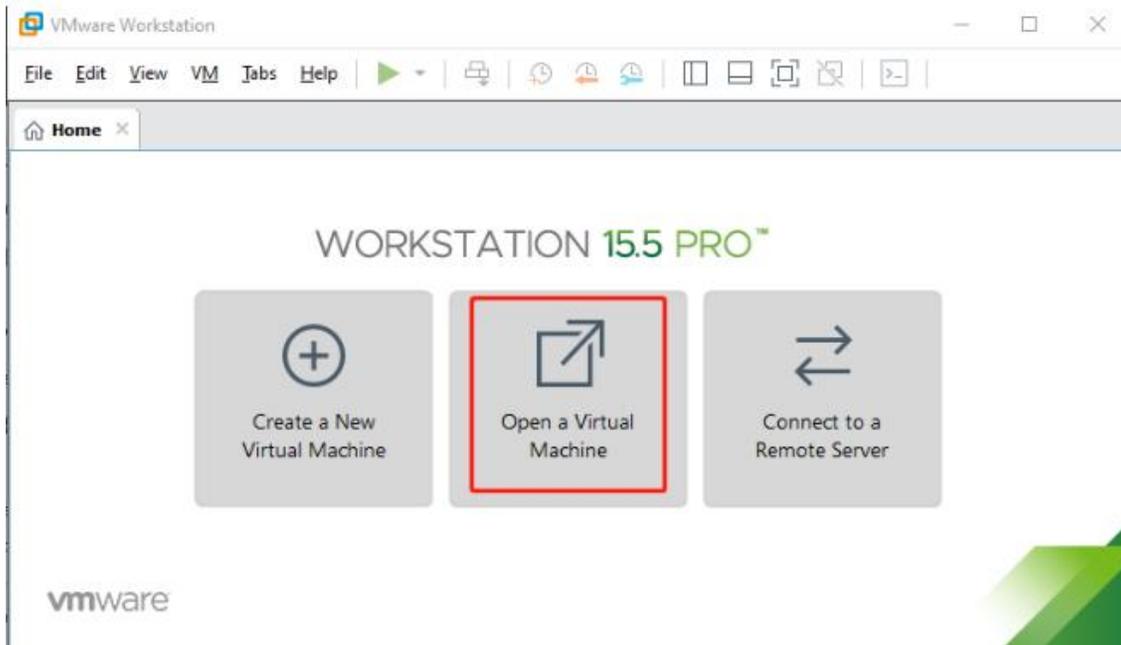
- 3) Con esto está listo para ser instalado y ejecutado en nuestro ordenador. Se debe esperar un momento mientras termina la instalación para finalmente hacer clic en finalizar y se debe reiniciar el computador.

**Figura 110**



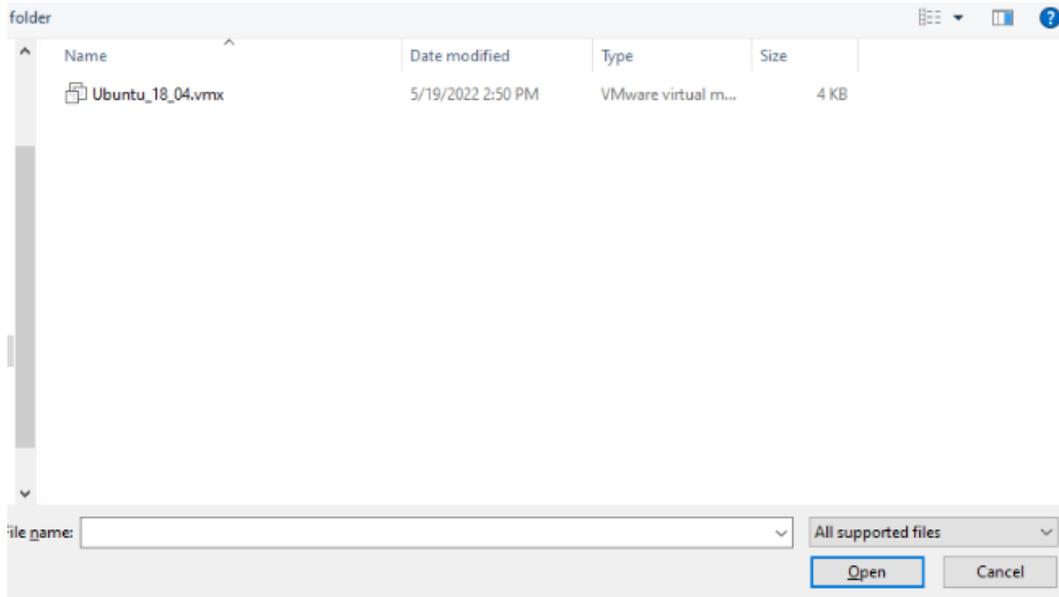
- 4) Una vez ejecutado el software de manera normal se debe presentar la interfaz de inicio donde debemos hacer clic en abrir una máquina virtual. Se va a utilizar una máquina virtual proporcionada por Ubuntu. Para acceder al archivo de la máquina virtual proporcionada por Ubuntu vamos a irnos al siguiente link y buscar lo indicado según las capturas.

**Figura 111**



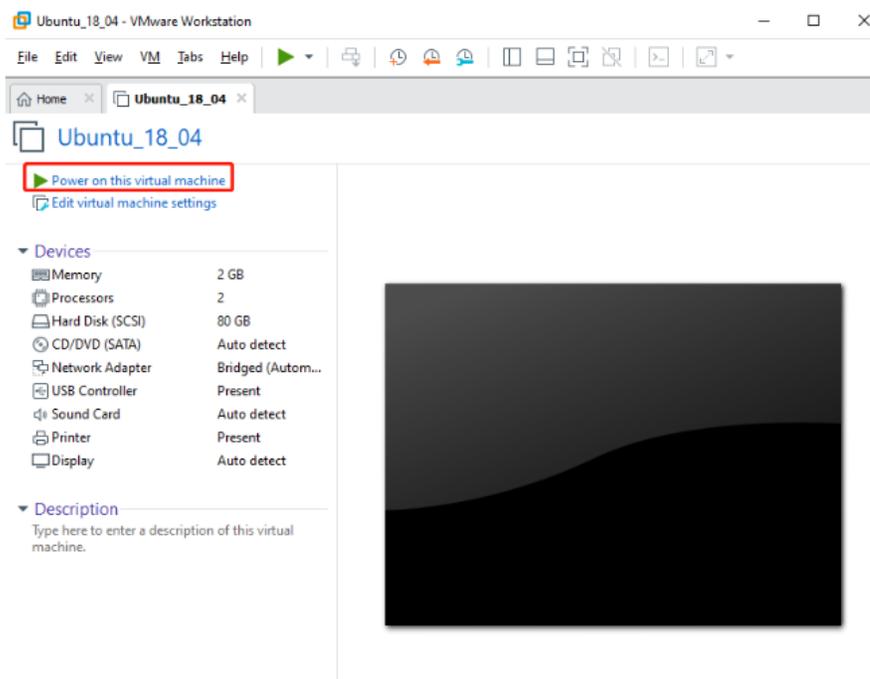
- Link de descarga del archivo de la maquina virtual:  
[https://drive.google.com/drive/folders/1BaD670BNv-rmhZ8y\\_mY8T-0pxXyB53Ax](https://drive.google.com/drive/folders/1BaD670BNv-rmhZ8y_mY8T-0pxXyB53Ax) (se debe elegir la carpeta para X3\_RO2)
- 
- 5) Al hacer clic en abrir una máquina virtual de manera automática va a buscar la carpeta previamente descargada de la máquina del link anterior. Se verá de la siguiente manera.

**Figura 112**



- 6) Encendemos nuestra máquina virtual y esperamos a que se ejecute. Como recomendación se debe tener una computadora con un procesador I5 en adelante o AMD Ryzen 5 para mejores tiempos de espera.

**Figura 113**

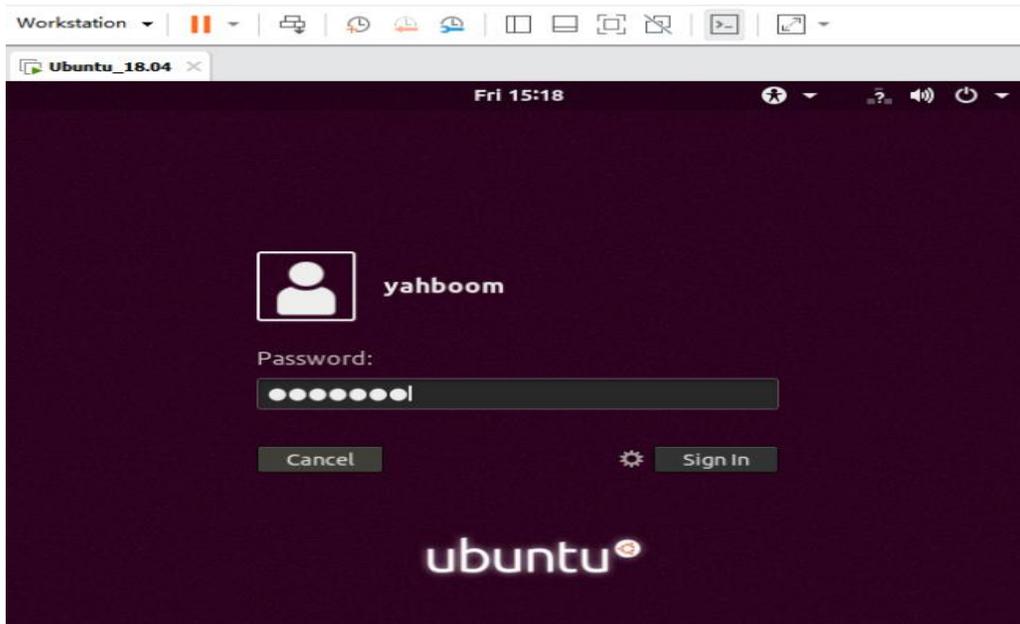


- 7) Luego de la espera para el arranque de la máquina virtual, nos mostrará la interfaz de Yahboom que es el desarrollador de nuestro ROSMASTER X3. Al momento de hacer la ejecución nos va a pedir credenciales de inicio.

Usuario: yahboom

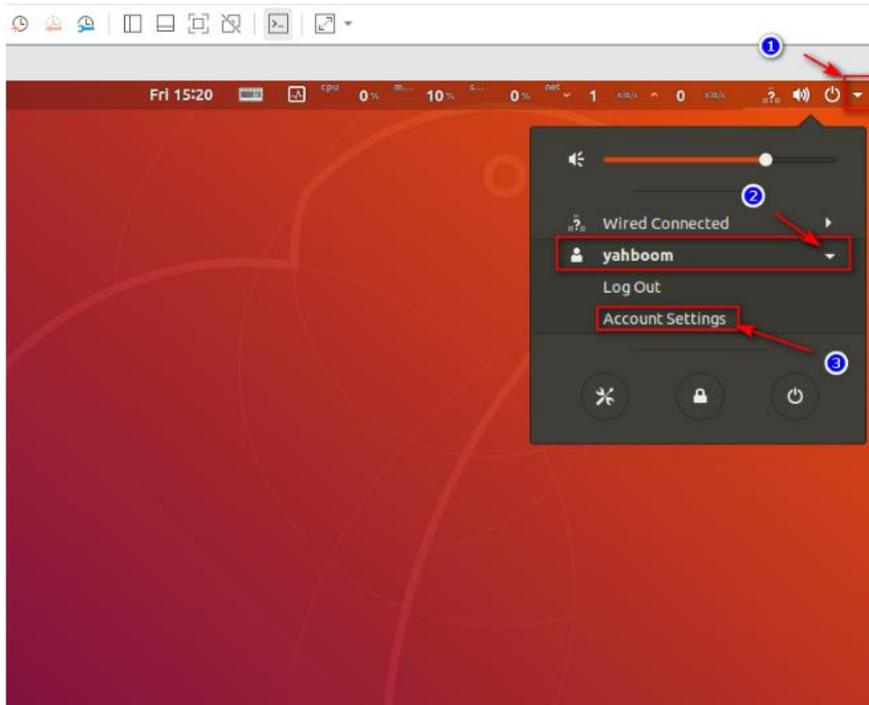
Contraseña: yahboom

**Figura 114**



- 8) Con esto ya tendremos listo el software VMware para la ejecución de la máquina virtual que es la encargada de ejecutar el mapeo de nuestro ROSMASTER X3.

**Figura 115**



## 5.6 Configuración de la Jetson Nano.

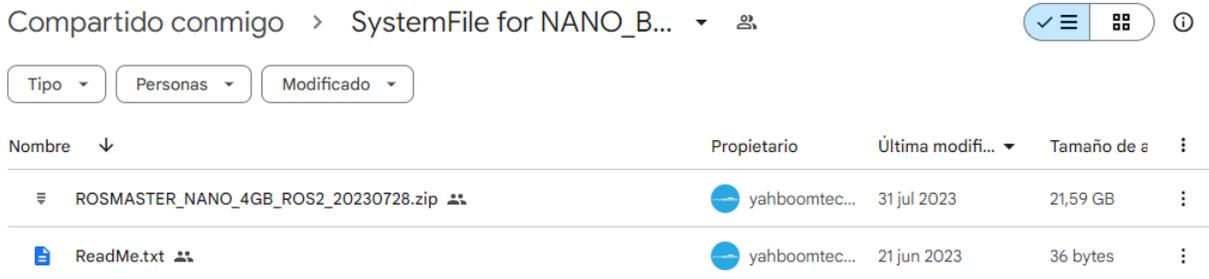
Jetson Nano tiene varias versiones, Jetson Nano 4G TF versión (versión pública) y Jetson Nano 4G SUB versión, ambas versiones se pueden utilizar con el robot, este producto se suministra de fábrica como Jetson Nano 4G SUB versión, antes de salir de fábrica el sistema de espejo del robot ha sido configurado. Generalmente, los usuarios no necesitan seguir este tutorial para grabar el sistema de espejo.

Las diferentes versiones tienen diferentes métodos de grabación. Entre ellas, la versión Jetson Nano 4G TF (versión pública) utiliza el método de grabación de la tarjeta TF y la versión Jetson Nano 4G SUB necesita grabar la imagen en la U disk + método de arranque de EMMC.

El método de grabación del sistema en la U disk es el mismo que el proceso de grabación del sistema en la tarjeta TF. El contenido siguiente toma la grabación de la imagen en la U disk como ejemplo.

Para tener los archivos necesarios para la configuración inicial de la Jetson Nano debemos descargar una carpeta que se encuentra en el repositorio de ROSMASTER X3. El link de descarga es el siguiente: <https://drive.google.com/drive/folders/1nyf-BhgrBftryZCUAIYJwh2Tsl45R1Ju>

**Figura 116**



Nombre	Propietario	Última modifi...	Tamaño de a
ROSMaster_NANO_4GB_ROS2_20230728.zip	yahboomtec...	31 jul 2023	21,59 GB
ReadMe.txt	yahboomtec...	21 jun 2023	36 bytes

- 1) Prepara un ordenador con Windows 10 y una unidad flash USB (se recomienda 32 GB o más). Este paso de grabar la unidad flash USB no requiere la participación de Jetson Nano.

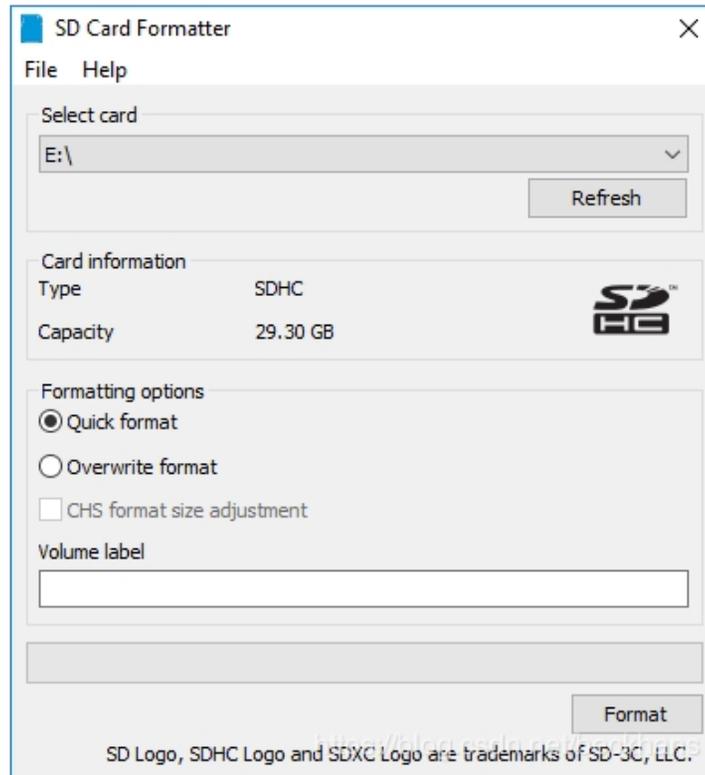
### Descarga la imagen

Encuentra la imagen de fábrica en los datos y descarga el archivo comprimido correspondiente según las diferentes placas base.

Nombre de usuario por defecto del sistema: jetson, contraseña: yahboom

### Formatea la tarjeta SD

Utiliza SDFormatter para formatear la U disk, y ten cuidado de no seleccionar la unidad equivocada aquí, de lo contrario, causará problemas innecesarios. Si la U disk ya ha programado el sistema, puede producirse un error en el primer formato; simplemente, ejecútalo de nuevo.

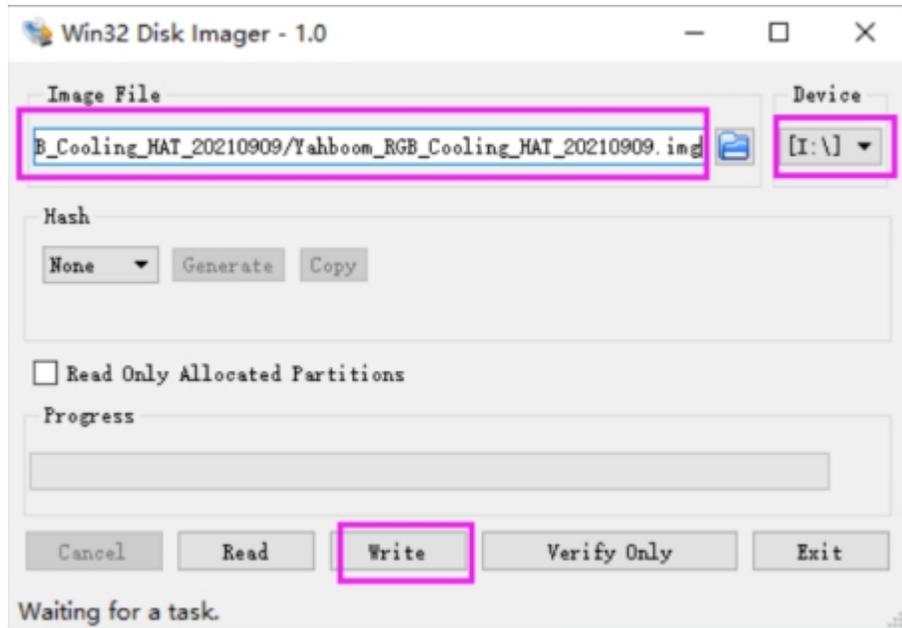


- 2) Descomprime el archivo comprimido del sistema descargado para obtener el archivo de imagen img.

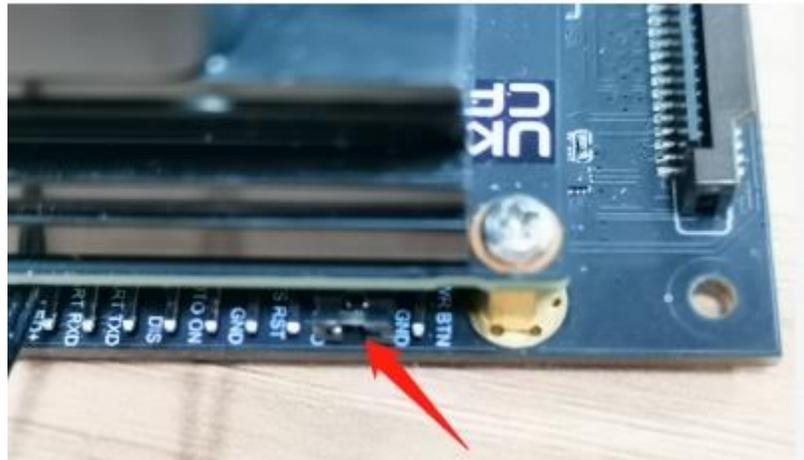
Inserta la U disk en el puerto USB de la computadora.

Descomprime y ejecuta la herramienta Win32DiskImager.

Selecciona el archivo img (imagen) en el software, elige la letra de unidad de la U disk en "Device", luego selecciona "Write" y comienza a grabar el sistema. La velocidad del proceso de grabación dependerá de la velocidad de tu U disk, por lo que puede ser rápido o lento.



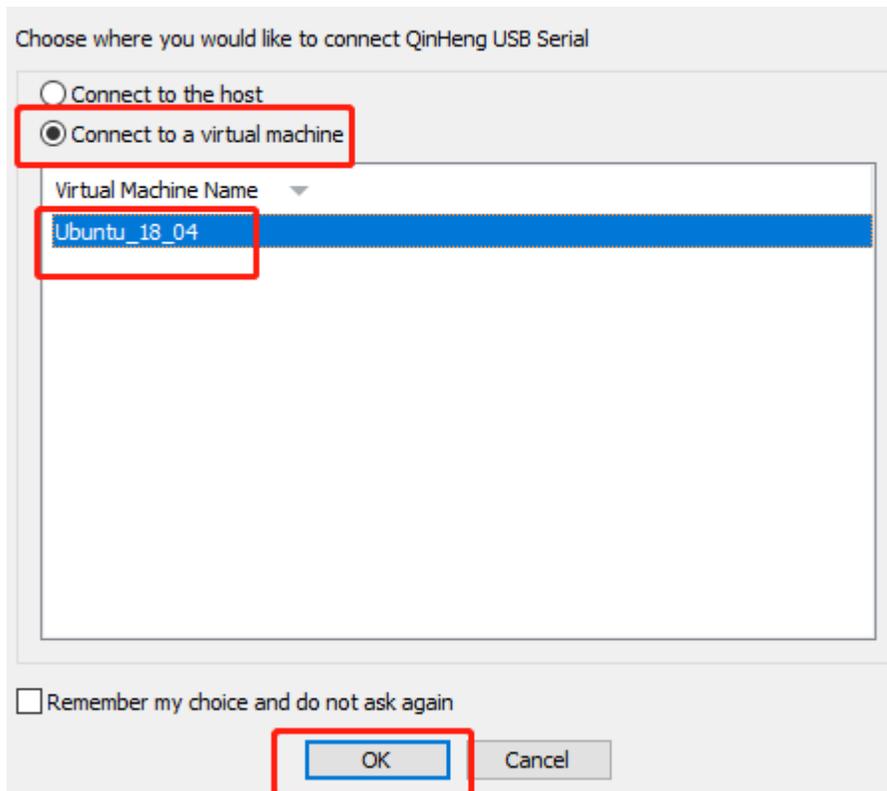
- 3) Este paso que es grabar el arranque se debe preparar la placa base Jetson Nano, tapones de puente, pantalla, ratón y teclado, entre otros. Permite que Jetson Nano entre en el modo de destello REC del sistema. Conecta los tapones de puente a los pines FC REC y GND, es decir, al segundo y tercer pin de la placa base debajo de la placa principal, como se muestra en la imagen a continuación:



- 4) Conecta los cables, conecta la pantalla HDMI, el ratón y el teclado al Jetson Nano, luego conecta la fuente de alimentación y, finalmente, inserta el cable de datos microUSB. Dado que el tapón de puente ya ha sido conectado a los pines FC REC y GND en el paso anterior, entrará automáticamente en el modo de destello REC después de encenderlo.



- 5) Bajo circunstancias normales, la siguiente ventana se abrirá después de insertar el cable de datos micro USB. Ten en cuenta que necesitas configurar el dispositivo para que se conecte a la máquina virtual si estás utilizando una.



- 6) Por favor, transfiera el archivo Jetson\_Boot\_USB.tar.gz que está en la carpeta de datos al sistema Ubuntu 18.04 y abra la terminal para ejecutar el comando de descompresión.

```
tar xzvf Jetson_Boot_USB.tar.gz
```

```
yahboom@YAB:~$ tar xzvf Jetson_Boot_USB.tar.gz
./Jetson_Boot_USB/
./Jetson_Boot_USB/jetson-nano-devkit.conf
./Jetson_Boot_USB/jetson-nano-devkit-emmc.conf
./Jetson_Boot_USB/jetson-tx1-devkit.conf
./Jetson_Boot_USB/nv_tools/
./Jetson_Boot_USB/nv_tools/scripts/
```

- 7) Después de la descompresión se deberá llamar la interfaz de la Jetson nano con el siguiente comando.

```
cd Jetson_Boot_USB/
ls
```

```
yahboom@YAB:~$ cd Jetson_Boot_USB/
yahboom@YAB:~/Jetson_Boot_USB$ ls
apply_binaries.sh          p2597-0000+p2180-1000-24x7.conf
bootloader                 p2597-0000+p2180-1000.conf
build_l4t_bup.sh           p3448-0000.conf.common
flash.sh                   p3448-0000-max-spi.conf
hybrid-qspt.conf           p3448-0000-max-spi-sd.conf
jetson-nano-2gb-devkit.conf p3449-0000+p3448-0000-qspt.conf
jetson-nano-devkit.conf    p3449-0000+p3448-0000-qspt-sd.conf
jetson-nano-devkit-emmc.conf p3449-0000+p3448-0002.conf
jetson-nano-emmc.conf      p3450.conf
jetson-nano-qspt.conf      p3542-0000+p3448-0003-qspt.conf
jetson-nano-qspt-sd.conf   p3542-0000+p3448-0003-qspt-sd.conf
jetson-tx1.conf            p3542.conf
jetson-tx1-devkit.conf     README_Autoflash.txt
kernel                     README_Massflash.txt
l4t_generate_soc_bup.sh    rootfs
nvautoflash.sh             source
nvmasflashgen.sh           source_sync.sh
nv_tegra                   tools
nv_tools                   TX1_boot-firmware-redundancy.txt
p2371.conf
```

- 8) Finalmente, para poder ejecutar el comando para grabar la carpeta de inicio EMMC insertamos.

```
sudo ./flash.sh -r jetson-nano-devkit-emmc mmcblk0p1
```

```
yahboom@YAB:~/Jetson_Boot_USB$ sudo ./flash.sh -r jetson-nano-devkit-emmc mmcblk0p1
[sudo] password for yahboom:
#####
# L4T BSP Information:
# R32 , REVISION: 5.2
#####
# Target Board Information:
# Name: jetson-nano-devkit-emmc, Board Family: t210ref, SoC: Tegra 210,
# OpMode: production, Boot Authentication: ,
# Disk encryption: disabled ,
```

- 9) Como muestra final que nuestro sistema ha sido grabado de manera correcta al final de la ejecución se deberá mostrar un mensaje en ingles en el cual indica que debemos reiniciar la tarjeta.

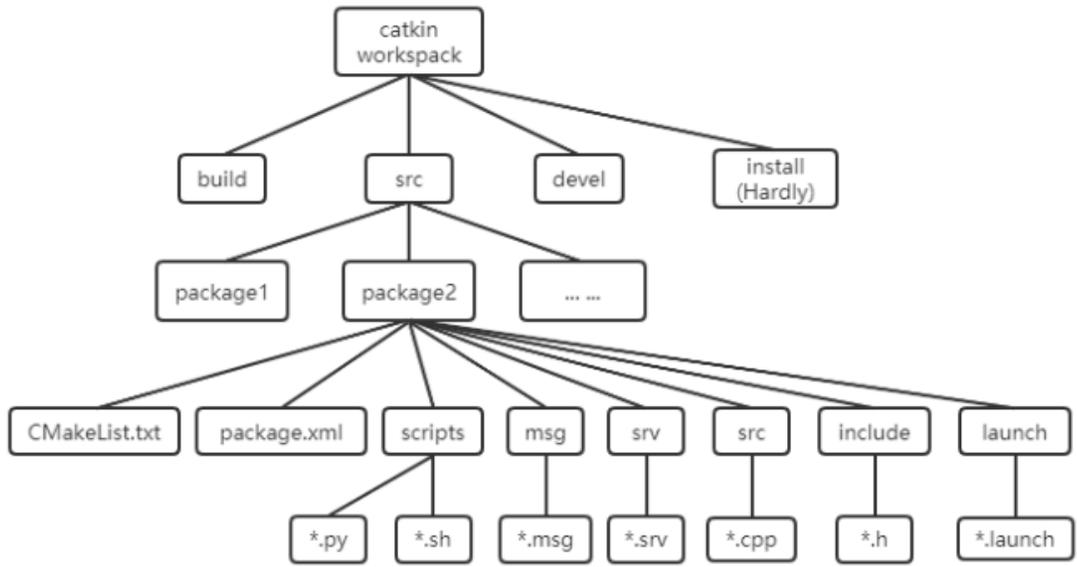
```
[ 8.3892 ]
*** The target t210ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

## 5.7 Estructura de ejecución del ROSMASTER X3

En cuanto a la ejecución de los archivos y programas del ROSMASTER X3 todo esto es apoyado con la tarjeta de NVIDIA la Jetson Nano de 4 Gb que es donde se guarda toda la información, configuración, librerías y codificación del ROSMASTER X3. El siguiente gráfico muestra el orden jerárquico que se debe tener en cuenta al momento de ejecutar algún comando. Cada uno de los valores representan las carpetas dentro de la tarjeta con la información necesaria para nuestros objetivos. En la figura 117 muestra la estructura de ejecución del ROSMASTER X3.

### **Figura 117**

*Estructura de ejecución del ROSMASTER X3.*



## CONCLUSIONES

- La combinación de MediaPipe para la detección precisa de objetos y ROS para la integración y control del robot ha permitido alcanzar resultados notables en términos de precisión y velocidad de respuesta.
- Los hallazgos subrayan la importancia de la integración de herramientas de percepción visual en robots mediante la combinación de tecnologías como MediaPipe y ROS. Este enfoque no solo mejora la capacidad de los robots para interactuar y navegar en entornos dinámicos, sino que también tiene aplicaciones significativas en sectores como la logística, la manufactura y la asistencia a personas con movilidad reducida.
- Se reconoce la limitación en la variabilidad de los entornos de prueba, lo que puede afectar la generalización de los resultados a situaciones más complejas y diversas. Además, se observa que la eficiencia del reconocimiento de objetos puede verse influenciada por condiciones de iluminación adversas, lo que sugiere la necesidad de investigaciones adicionales sobre la robustez del código.
- La investigación ha demostrado que la combinación de MediaPipe y ROSMASTER X3 ofrece una plataforma efectiva para el desarrollo de técnicas de percepción visual en robots. Las implicaciones prácticas y la capacidad de superar desafíos específicos destacan la relevancia de esta investigación en el avance de la autonomía y la interacción de robots en entornos del mundo real. Este estudio proporciona una base sólida para futuras investigaciones que buscan mejorar aún más la robustez y la aplicabilidad de estos sistemas en diversas aplicaciones robóticas.

## **RECOMENDACIONES**

- Dada la sensibilidad del sistema a condiciones de iluminación variables, se recomienda investigar y desarrollar estrategias adicionales de procesamiento de imagen que mejoren la robustez del reconocimiento de objetos en situaciones de iluminación adversa. Esta recomendación se deriva directamente de los desafíos identificados durante el estudio en entornos con cambios de iluminación.
- Implementar técnicas de aumento de datos específicas para abordar la variabilidad en las condiciones de iluminación. Esto podría incluir la generación de imágenes sintéticas con diferentes niveles de iluminación para enriquecer el conjunto de datos de entrenamiento y mejorar la capacidad del modelo de percepción visual para adaptarse a entornos cambiantes.
- Dado el acceso a conjuntos de datos de entrenamiento extensos y la disponibilidad de herramientas de aumento de datos, la implementación de esta acción específica es prácticamente viable. Además, considerando la capacidad computacional actual y los recursos de entrenamiento, esta estrategia no impondría una carga significativa.
- Considerando la limitación en la variabilidad de los entornos de prueba identificada en la investigación, se sugiere realizar experimentos adicionales en entornos más diversos y complejos. Esto permitirá evaluar la generalización del sistema y su rendimiento en escenarios del mundo real con una mayor variedad de obstáculos y condiciones.
- Se sugiere explorar estrategias adicionales para mejorar la robustez del reconocimiento en condiciones de iluminación variable, como el uso de técnicas de aumento de datos. Además, investigar la adaptabilidad del sistema a entornos en constante cambio y la optimización de algoritmos para una ejecución más eficiente serían áreas fructíferas para futuras investigaciones.

## **REFERENCIAS**

- 1) Boesch, G. (2023, 9 de noviembre). MediaPipe: Google's open source framework for ML Solutions (2024 guide). viso.ai. <https://viso.ai/computer-vision/mediapipe/>
- 2) Cass, S. (2020). Nvidia makes it easy to embed AI: the Jetson nano packs a lot of machine-learning power into DIY projects-[Hands on]. IEEE Spectrum, 57(7), 14–16.
- 3) Clark, L. (2006). Manual para el Mapeo de Redes como Herramienta de Diagnóstico. CIAT.
- 4) Evans, W. (2023, 23 de marzo). What are WiFi heatmaps and how do they work? Performance Networks. <https://www.performancenetworks.co.uk/blog/post/wifi-heatmap/>
- 5) Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., & Zhu, X. (2012). VMware distributed resource management: Design, implementation, and lessons learned. VMware Technical Journal, 1(1), 45-64.
- 6) Kurniawan, A. (2021). Introduction to NVIDIA Jetson Nano. En: IoT Projects with NVIDIA Jetson Nano. Apress, Berkeley, CA. [https://doi.org/10.1007/978-1-4842-6452-2\\_1](https://doi.org/10.1007/978-1-4842-6452-2_1)
- 7) NVIDIA Jetson Nano. (s/f). NVIDIA. Recuperado el 13 de diciembre de 2023, de <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/>
- 8) Nvidia Jetson Nano Developer Kit. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- 9) Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019, June). Mediapipe: A framework for perceiving and processing reality. Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR), 2019.
- 10) Manual para el mapeo de redes como herramienta de diagnóstico. (s. f.). Google Books. [https://books.google.com.ec/books?hl=es&lr=&id=WJVGUtqi8nYC&oi=fnd&pg=PA2&dq=mapeo+de+redes&ots=jCjF46dDc8&sig=OdF1-LNyabHBo3NOTXIQ3mWcVwc&redir\\_esc=y#v=onepage&q=mapeo%20de%20redes&f=false](https://books.google.com.ec/books?hl=es&lr=&id=WJVGUtqi8nYC&oi=fnd&pg=PA2&dq=mapeo+de+redes&ots=jCjF46dDc8&sig=OdF1-LNyabHBo3NOTXIQ3mWcVwc&redir_esc=y#v=onepage&q=mapeo%20de%20redes&f=false)
- 11) Progress. (2023, 13 de marzo). ¿Qué es el Mapeo de Red y Para Qué Sirve? Progress. Recuperado 13 de diciembre de 2023, de <https://www.whatsupgold.com/es/blog/what-network-mapping-what-good-for>
- 12) RealVNC®. (2023, 11 de diciembre). RealVNC®: Software de acceso remoto para escritorio y dispositivos móviles | RealVNC. <https://www.realvnc.com/es/>
- 13) Ricardo, J. E., Vázquez, M. Y. L., Palacios, A. J. P., & Ojeda, Y. E. A. (2021). Inteligencia artificial y propiedad intelectual. Universidad y Sociedad, 13(S3), 362-368.
- 14) Rouhiainen, L. (2018). Inteligencia artificial. Madrid: Alienta Editorial, 20-21.
- 15) Sati, V., Sánchez, S. M., Shoeibi, N., Arora, A., & Corchado, J. M. (2020). Face detection and Recognition, Face emotion Recognition through NVIDIA Jetson Nano.

- 16) En Advances in intelligent systems and computing (pp. 177-185). [https://doi.org/10.1007/978-3-030-58356-9\\_18](https://doi.org/10.1007/978-3-030-58356-9_18)
- 17) Suherman, S., Suhendra, A., & Ernastuti, E. (2023). Method Development Through Landmark Point Extraction for Gesture Classification With Computer Vision and MediaPipe. TEM Journal, 12(3).
- 18) Todo lo que necesitas saber sobre Jetson Nano de NVIDIA. (2019, junio 21). Blog Bricogeek.com. <https://blog.bricogeek.com/noticias/robotica/todo-lo-que-necesitas-saber-sobre-jetson-nano-de-nvidia/>
- 19) VMware, I., & Calculator, R. O. I. (2018). VMware.
- 20) Zhang, X., Zhou, H., Cheng, H., et al. (2015). Teaching-playback of robot manipulator based on human gesture recognition and motion tracking. 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, pp 1183–1188.

