

**UNIVERSIDAD SANTO TOMÁS
SECCIONAL TUNJA**

FACULTAD DE INGENIERÍA ELECTRÓNICA



**Desarrollo e implementación de algoritmos de visión
artificial en plataforma robótica para tareas Pick and
Place**

Presentado por:

Angel Esteban Moreno Daza

Marco Antonio Lasprilla Fernandez

Para obtener el título profesional de

INGENIERO ELECTRÓNICO

Tunja Boyacá - Colombia
Abril - 2026

**UNIVERSIDAD SANTO TOMÁS
SECCIONAL TUNJA**

FACULTAD DE INGENIERÍA ELECTRÓNICA



**Desarrollo e implementación de algoritmos de visión
artificial en plataforma robótica para tareas Pick and
Place**

Presentado por

Angel Esteban Moreno Daza

Marco Antonio Lasprilla Fernandez

Director del trabajo de grado:

Ing. Edgar Andrés Gutiérrez Cáceres

Codirector del trabajo de grado:

Ing. Camilo Ernesto Pardo Beainy

Tunja Boyacá - Colombia
Abril - 2026

Exoneración

El contenido del presente trabajo de grado es responsabilidad exclusiva de sus autores. Las opiniones, resultados, conclusiones y recomendaciones aquí expresadas no comprometen a la Universidad Santo Tomás, a la Facultad de Ingeniería Electrónica, ni a los directores, codirectores o evaluadores del proyecto.

Se declara, además, que la información tomada de otras fuentes ha sido citada y referenciada de acuerdo con las normas académicas correspondientes. Las figuras, tablas y fragmentos de código ajenos incluidos en el documento se utilizan con fines académicos y se acreditan mediante la referencia respectiva. Cualquier error u omisión es responsabilidad de los autores.

Para el desarrollo del prototipo se emplearon herramientas de software y bibliotecas de terceros (incluyendo, entre otras, librerías de visión por computador, modelos de detección y utilidades asociadas). Dichas herramientas se utilizaron conforme a su documentación y condiciones de uso publicadas por sus autores o entidades responsables. La correcta instalación, configuración y uso de estas herramientas dentro del alcance del proyecto es responsabilidad de los autores.

Dedicatoria

Dedicamos este trabajo a nuestras familias, por su apoyo incondicional, su paciencia y su confianza a lo largo de todo el proceso formativo. Su acompañamiento fue el motor que nos permitió avanzar y culminar esta etapa.

A nuestros amigos, por estar presentes en los momentos de esfuerzo, por el ánimo constante y por hacer del camino universitario una experiencia más humana y significativa.

Y a la comunidad de la Facultad de Ingeniería Electrónica de la Universidad Santo Tomás – Seccional Tunja, por ser el espacio donde crecimos académica y personalmente, y por contribuir a nuestra formación con su conocimiento, dedicación y compromiso.

Agradecimientos

Expresamos nuestro más sincero agradecimiento a nuestras familias, por el apoyo constante, la paciencia y la motivación brindada a lo largo de toda la carrera. Su respaldo fue esencial para mantenernos firmes en los momentos de mayor exigencia académica y personal.

Agradecemos a los docentes que, con su orientación y rigor, contribuyeron a nuestra formación profesional y al fortalecimiento de nuestras habilidades técnicas. De igual manera, agradecemos a la Facultad de Ingeniería Electrónica de la Universidad Santo Tomás – Seccional Tunja, por proporcionarnos los recursos, espacios y acompañamiento necesarios para el desarrollo de este trabajo.

Finalmente, manifestamos un reconocimiento especial a nuestro director Ing. Edgar Andrés Gutiérrez Cáceres y codirector Ing. Camilo Ernesto Pardo Beainy, por su guía, exigencia y compromiso durante el proyecto. Su acompañamiento marcó el rumbo del trabajo y nos impulsó a elevar el nivel de cada etapa. Para nosotros, fueron ese liderazgo académico que inspira, orienta y da ejemplo.

O Captain! My Captain!.

Índice general

Exoneración	II
Dedicatoria	III
Agradecimientos	IV
Resumen	IX
Abstract	X
1 Introducción	1
1.1 Organización del documento	1
1.2 Planteamiento del Problema	2
1.3 Formulación de Preguntas	2
1.4 Justificación	3
1.5 Impactos esperados	4
1.5.1 Corto plazo	4
1.5.2 Mediano plazo	4
1.5.3 Largo plazo	4
1.6 Objetivos	4
1.6.1 Objetivo general	4
1.6.2 Objetivos específicos	5
2 Fundamentos teóricos y desarrollo metodológico	6
2.1 Estado del Arte y Análisis de Vigilancia Tecnológica	6
2.1.1 Sistemas de Visión Artificial en Robótica Actual	6
2.1.2 Deep Learning y Visión Tridimensional	6
2.1.3 Sistema Robótico Pick and Place y Visión Artificial	7
2.2 Marco Teórico	8
2.2.1 WLKATA Mirobot	8
2.2.2 Cinemática Mirobot	10
2.2.3 Pick and Place	12
2.2.4 Cámaras de profundidad	13
2.2.5 Detección y Localización de objetos	13
2.2.6 YOLO V8	14
2.2.7 RealSense D435i	15
2.3 Métricas de evaluación en detección de objetos	15
2.3.1 Precisión, recall y F1-score	16
2.3.2 mAP: precisión promedio y desempeño global	16
2.3.3 Relación de las métricas con este proyecto	17

2.4	Metodología	17
2.4.1	Materiales	18
2.4.2	Fases de desarrollo	18
3	Análisis de resultados	22
3.1	Descripción de la maqueta, conexiones y flujo de operación	22
3.2	Métricas de desempeño del modelo	24
3.2.1	Análisis de curvas de entrenamiento, curvas de confianza y matriz de confusión	24
3.3	Resultados del módulo de percepción y estimación de coordenadas	27
3.3.1	Evidencia de coordenadas registradas	28
3.4	Integración entre cámara y brazo robótico	29
3.4.1	Prueba de movimiento y comunicación con el manipulador	29
3.4.2	Asociación de coordenadas RGB-D con coordenadas espaciales del robot y ajuste del plano de trabajo	29
3.5	Resultados del criterio de selección y priorización del objetivo	30
3.5.1	Evidencia del registro de priorización (<i>rank log</i>)	30
3.5.2	Análisis del criterio de selección	31
3.5.3	Selección del punto objetivo para el Pick and Place: centroide vs punto más cercano	31
3.5.4	Evaluación del desempeño del ciclo Pick and Place	32
3.5.5	Evaluación del error en la medición de profundidad	35
4	Conclusiones	37
4.1	Objetivo específico 1	37
4.2	Objetivo específico 2	37
4.3	Objetivo específico 3	38
4.4	Objetivo específico 4	38
4.5	Limitaciones	39
4.6	Trabajo futuro	39
	Referencias Bibliográficas	42
	Anexos	44
	Anexo A: Objetos de prueba	45
	Anexo B: Código fuente y datos del proyecto	47
	Anexo C: Evidencia audiovisual del sistema	49

Índice de figuras

Figura 2.2.1	Estructura Mirobot. Tomado de [1]	9
Figura 2.2.2	Cámara de profundidad RealSense D435i. Tomado de: [2].	15
Figura 2.4.1	Flujo general del sistema propuesto en la metodología	18
Figura 3.1.1	Vista general de la maqueta y del espacio físico de trabajo (cámara, brazo robótico y plano de operación). Fuente: Elaboración propia.	23
Figura 3.1.2	Esquema general de conexión cámara–Jetson–brazo y flujo de trabajo del sistema. Fuente: Elaboración propia, adaptada de [2] y [3].	23
Figura 3.2.1	Curvas generales de entrenamiento y validación del modelo YOLOv8. Fuente: Elaboración propia.	25
Figura 3.2.2	Curva F1-Confidence del modelo YOLOv8. Fuente: Elaboración propia.	25
Figura 3.2.3	Curva Precision–Recall del modelo YOLOv8. Fuente: Elaboración propia.	26
Figura 3.2.4	Matriz de confusión normalizada del modelo YOLOv8. Fuente: Elaboración propia.	27
Figura 3.3.1	Vista del sistema estereo-activo durante la detección y estimación de coordenadas con Intel RealSense D435i. Fuente: Elaboración propia.	28
Figura 3.4.1	Referencia visual de ajuste del plano de trabajo: “X” dibujada en el frame usando los puntos <i>robot_points</i> . Fuente: Elaboración propia.	30
Figura 3.5.1	Ejemplo de solapamiento de <i>bounding boxes</i> . Fuente: Elaboración propia.	32
Figura 3.5.2	Relación entre la distancia medida por la cámara y la probabilidad de éxito del ciclo Pick and Place. Fuente: Elaboración propia.	33
Figura 3.5.3	Tasa de éxito del ciclo Pick and Place agrupada por objeto. Fuente: Elaboración propia.	34
Figura 3.5.4	Relación entre la distancia real y la distancia estimada por la cámara. Fuente: Elaboración propia.	35

Índice de tablas

Tabla 2.2.1 Especificaciones	9
Tabla 2.2.2 Parámetros de movimiento	10
Tabla 3.2.1 Métricas globales del modelo YOLOv8 al final del entrenamiento .	24
Tabla 3.3.1 Muestra de coordenadas registradas por el módulo de percepción (ejemplo representativo)	28
Tabla 3.5.1 Muestra representativa del registro de priorización del objetivo . .	31
Tabla 4.1 Objetos de prueba utilizados en el sistema (con apoyo visual) . . .	45
Tabla 4.2 Archivos principales del repositorio	48
Tabla 4.3 Registro de evidencia audiovisual del sistema	50

Resumen

El presente trabajo de tesis propone el desarrollo de un sistema robótico Pick and Place que integra visión artificial en un entorno tridimensional, utilizando el brazo robótico WLKATA Mirobot 6-axis y la cámara Intel RealSense D435i. Debido a que la automatización de tareas enfocadas en la manipulación de objetos es un factor de gran interés en la industria moderna, se busca la optimización, precisión y seguridad de estos procesos. Por lo tanto, este proyecto aborda la implementación de sistemas robóticos inteligentes capaces de identificar, localizar y manipular objetos de manera autónoma, contribuyendo al avance y desarrollo de la robótica aplicada.

El proyecto consta de la implementación de un sistema de percepción 3D en tiempo real, apoyado en un modelo de detección de objetos implementado en YOLOv8 aplicado en la cámara Intel RealSense D435i, que sirve de soporte a los movimientos y al control cinemático del brazo WLKATA Mirobot 6-axis para ejecutar operaciones Pick and Place de forma precisa y coordinada. Junto con una tarjeta NVIDIA Jetson Orin Nano como núcleo del sistema para integrar los sistemas de visión y control, permitiendo una operación eficiente en entornos tridimensionales y dinámicos.

Palabras clave: Visión Artificial, Robótica Industrial, Pick and Place, Detección de Objetos, Percepción 3D, YOLOv8.

Abstract

This thesis proposes the development of a robotic Pick and Place system that integrates computer vision in a three-dimensional environment, using the WLKATA Mirobot 6-axis robotic arm and the Intel RealSense D435i camera. Since the automation of object manipulation tasks is a major focus in modern industry, this work aims to improve the optimization, precision, and safety of such processes. Therefore, the project addresses the implementation of intelligent robotic systems capable of identifying, locating, and autonomously manipulating objects, contributing to the advancement of applied robotics.

The project includes the implementation of a real-time 3D perception system supported by an object detection model based on YOLOv8 using the Intel RealSense D435i camera. This perception layer supports motion planning and kinematic control of the WLKATA Mirobot 6-axis arm to perform Pick and Place operations accurately and in coordination. An NVIDIA Jetson Orin Nano board is used as the system core to integrate vision and control, enabling efficient operation in dynamic 3D environments.

Keywords: Computer Vision, Industrial Robotics, Pick and Place, Object Detection, 3D Perception, YOLOv8.

Capítulo 1

Introducción

En este trabajo se aborda el desarrollo de un sistema robótico orientado a tareas de Pick and Place, integrando técnicas de visión artificial y control de un manipulador en un entorno tridimensional. La motivación principal surge de la necesidad de aumentar la precisión, flexibilidad y autonomía en procesos de manipulación de objetos, ya que los enfoques tradicionales suelen depender de trayectorias rígidas y configuraciones poco adaptables ante cambios en el entorno. En respuesta a este reto, se propone una solución que combine percepción 3D y detección de objetos en tiempo real como soporte para la ejecución controlada de movimientos del robot.

Para ello, se plantea la integración de una cámara Intel RealSense D435i, un modelo de detección de objetos basado en YOLOv8 y una unidad de cómputo embebida NVIDIA Jetson Orin Nano como núcleo del sistema, coordinando el control cinemático de un brazo WLKATA Mirobot 6-axis. Esta arquitectura busca superar limitaciones actuales en localización y manipulación, y establecer una base escalable para desarrollos futuros en robótica aplicada.

1.1. Organización del documento

El presente documento se encuentra estructurado en varios capítulos, los cuales permiten desarrollar de manera progresiva los fundamentos, la implementación y la evaluación del sistema propuesto.

En el Capítulo 1 se expone el problema abordado, contextualizando la necesidad de desarrollar soluciones basadas en visión artificial y robótica para la automatización de tareas de manipulación. Asimismo, se presenta la justificación del proyecto, destacando su relevancia en el ámbito académico e industrial. De igual manera, se definen los objetivos generales y específicos que guían el desarrollo del trabajo, junto con el alcance del sistema propuesto, delimitando las condiciones bajo las cuales se lleva a cabo la implementación y evaluación de la solución.

El Capítulo 2 integra el marco teórico y la metodología, abordando los conceptos fundamentales relacionados con visión artificial, aprendizaje profundo, sensores de profundidad y manipulación robótica. Asimismo, se describe el proceso de desarrollo del sistema, incluyendo las fases de implementación, la arquitectura propuesta y la integración de los diferentes componentes.

En el Capítulo 3 se presenta el análisis de resultados, donde se evalúa el desempeño del sistema mediante pruebas experimentales, métricas de detección, análisis de la estimación de profundidad y la ejecución del ciclo Pick and Place bajo diferentes condiciones.

Finalmente, se incluyen las conclusiones del trabajo, junto con los anexos que contienen información complementaria, como código fuente, configuraciones del sistema y evidencias experimentales.

1.2. Planteamiento del Problema

La automatización de tareas de manipulación de objetos en espacios de trabajo industriales y de laboratorio sigue siendo un reto importante, ya que es necesario combinar exactitud, rapidez y capacidad de adaptación. Los robots tradicionales a menudo se basan en una programación hecha a mano y muy estricta respecto a su trayectoria y funcionamiento, lo cual limita su capacidad para operar en ambientes dinámicos y en tres dimensiones, dificultando que se adapten a cambios en la disposición de los objetos o en el espacio de trabajo.

La integración de la visión artificial con robots manipuladores de objetos ha surgido como una alternativa eficaz para aumentar la eficiencia, la flexibilidad y la autonomía de los procesos automatizados, permitiendo que estos puedan observar, interpretar y reaccionar a su entorno de manera simultánea. Sin embargo, pese a los avances en sensores 3D, algoritmos de detección de objetos y control robótico, aún existen limitaciones importantes en la ubicación precisa y en la manipulación confiable de los objetos, especialmente cuando se requiere que operen de manera autónoma y con alta exactitud en situaciones complejas. Estas limitaciones muestran la necesidad de explorar soluciones que combinen equipos avanzados de procesamiento, percepción tridimensional y control cinemático inteligente, para superar las restricciones de los sistemas actuales y mejorar la aplicabilidad de la robótica en entornos industriales y de investigación.

1.3. Formulación de Preguntas

1. ¿Cómo integrar un sistema de percepción 3D, basado en YOLOv8 con el control cinemático del WLKATA Mirobot 6-axis para ejecutar tareas Pick and Place de forma autónoma y estable?
2. ¿Qué criterio para seleccionar el punto de agarre (centroide, punto más cercano,

región estable) produce mayor tasa de éxito en el agarre cuando se usa información obtenida a través de la detección?

1.4. Justificación

Pese a los avances recientes en robótica y su integración con técnicas de deep learning y visión artificial, aún persisten limitaciones relevantes en la localización y manipulación de objetos, especialmente cuando se requiere precisión y adaptabilidad en entornos reales. En consecuencia, resulta pertinente abordar métodos y arquitecturas que permitan superar dichas limitaciones mediante la integración efectiva entre percepción, cómputo embebido y control del manipulador.

En este marco, la implementación de sistemas robóticos soportados en tarjetas de desarrollo NVIDIA Jetson (en particular la familia Orin) se ha consolidado como un estándar por su pragmatismo y escalabilidad. La combinación del procesamiento embebido en Jetson Orin Nano, la percepción estéreo de la cámara Intel RealSense D435i y el control cinemático del brazo WLKATA Mirobot 6-axis constituye una alternativa eficaz para afrontar tareas de manipulación que demandan precisión y adaptabilidad. De esta manera, el proyecto no solo contribuye al fortalecimiento de capacidades técnicas en robótica y visión artificial, sino que también aporta a la competitividad y la innovación tecnológica en contextos productivos y académicos.

En el contexto específico de un sistema Pick and Place guiado por visión, surge la necesidad de transformar la información visual en una referencia espacial útil para la manipulación. No basta con detectar el objeto: es indispensable estimar su ubicación en el espacio tridimensional y asociarla con el sistema de coordenadas del manipulador para ejecutar aproximación, agarre y colocación de manera consistente. En este sentido, la integración de una cámara RGB-D con un modelo de detección permite construir un flujo percepción-control que reduce la dependencia de configuraciones rígidas y habilita la operación sobre escenas donde la posición de los objetos puede variar.

Adicionalmente, este proyecto se plantea como una propuesta experimental y replicable para validar la integración de visión artificial, percepción tridimensional y control cinemático en una plataforma real. La implementación sobre hardware embebido, mediante la NVIDIA Jetson Orin Nano, junto con la cámara Intel RealSense D435i y el manipulador WLKATA Mirobot, permite evaluar el desempeño del sistema bajo restricciones prácticas de tiempo real y recursos de cómputo. El aporte principal del trabajo se centra en articular detección de objetos, estimación espacial, priorización de objetivos y selección del punto de referencia mediante el centroide del *bounding box* dentro de un flujo completo percepción-control. Desde la perspectiva de la Ingeniería Electrónica, esta integración relaciona sistemas embebidos, procesamiento de imágenes, instrumentación, automatización y control robótico, evidenciando la aplicación práctica de competencias propias del programa en una solución funcional de robótica aplicada.

1.5. Impactos esperados

1.5.1. Corto plazo

A corto plazo, el proyecto permitirá validar técnicamente un sistema robótico Pick and Place que integra visión artificial 3D mediante la cámara Intel RealSense D435i, detección de objetos con YOLOv8 y control embebido en la tarjeta NVIDIA Jetson Orin Nano, coordinando los movimientos del brazo WLKATA Mirobot 6-axis. Se espera obtener resultados asociados a la precisión del sistema, mediante un análisis de desempeño y la validación académica por parte de jurados y pares.

1.5.2. Mediano plazo

A mediano plazo, el sistema desarrollado podrá emplearse como base para nuevos proyectos de investigación en robótica, visión artificial y automatización, principalmente en el ámbito educativo y experimental. Se contempla como resultado la publicación de al menos un artículo científico o ponencia en eventos académicos, así como la propuesta de incorporación del prototipo en laboratorios de la Universidad Santo Tomás como material de enseñanza en cursos de robótica y automatización. De manera cualitativa, se busca que el sistema sea reconocido como un ejemplo de integración entre hardware embebido, percepción tridimensional y manipulación robótica de bajo costo.

1.5.3. Largo plazo

A largo plazo, el conocimiento adquirido podrá transferirse hacia aplicaciones industriales y de investigación aplicada, favoreciendo la implementación de soluciones de automatización accesibles y escalables. Se proyecta que los resultados puedan derivar en el desarrollo de prototipos con enfoque industrial o en iniciativas de innovación tecnológica.

1.6. Objetivos

1.6.1. Objetivo general

Desarrollar un sistema robótico Pick and Place, usando como base un brazo WLKATA Mirobot 6-axis y una cámara Intel RealSense, integrando visión artificial para la detección, localización y manipulación de objetos en una perspectiva tridimensional, mediante una tarjeta embebida NVIDIA Jetson Orin Nano.

1.6.2. Objetivos específicos

1. Coordinar en tiempo real mediante una tarjeta NVIDIA Jetson Orin Nano, como núcleo del sistema, y la cámara Intel RealSense D435i para la ejecución de movimientos del brazo robot WLKATA Mirobot.
2. Diseñar un sistema de percepción tridimensional utilizando la cámara Intel RealSense D435i, asociado a un modelo de detección de objetos basado en YOLOv8, para obtener información precisa de la ubicación espacial de los objetos.
3. Desarrollar algoritmos para la planificación de trayectorias y movimientos del brazo WLKATA Mirobot 6-axis para operaciones de Pick and Place.
4. Analizar los resultados obtenidos para identificar posibles mejoras en el sistema frente a su aplicabilidad en entornos industriales o de investigación.

Capítulo 2

Fundamentos teóricos y desarrollo metodológico

2.1. Estado del Arte y Análisis de Vigilancia Tecnológica

2.1.1. Sistemas de Visión Artificial en Robótica Actual

La robótica integrada con visión artificial ha experimentado avances significativos en los años recientes, principalmente impulsada por el desarrollo de algoritmos de aprendizaje profundo, la mejora en sensores 3D y el aumento en la capacidad de procesamiento de sistemas embebidos. Esto marca una tendencia clara hacia la innovación y el desarrollo de procesos de automatización con poca o casi nula intervención humana. Un estudio publicado en abril de 2024 por Deloitte y The Manufacturing Institute sugiere que más de la mitad de los 1.9 millones de empleos manufactureros que se espera cubrir para 2033 podrían quedar vacantes [4]. Esta escasez de mano de obra impulsa la adopción de sistemas automatizados más flexibles y adaptables, como la robótica guiada por visión (VGR, *Vision Guided Robotics*) y las herramientas de aprendizaje profundo, estableciendo una transición de sistemas programados manualmente hacia sistemas adaptativos que pueden aprender de datos de entrenamiento.

2.1.2. Deep Learning y Visión Tridimensional

La integración entre sistemas de visión estéreo y algoritmos de detección de objetos basados en redes neuronales convolucionales, como YOLO, representa una tendencia consolidada en la investigación actual. Esto permite no solo identificar objetos en la escena, sino también determinar su posición tridimensional precisa, lo cual es fundamental para aplicaciones de manipulación robótica. La combinación de datos RGB-D (color y profundidad) procedentes de cámaras estéreo con modelos de detección

entrenados proporciona la base para sistemas robóticos autónomos capaces de operar en entornos dinámicos y no estructurados.

Dentro de este ámbito, las plataformas de cómputo embebido, como la línea NVIDIA Jetson Orin Nano, se han consolidado como un estándar de facto para visión e inteligencia artificial por su relación rendimiento-consumo. Las especificaciones y documentación oficiales reportan hasta 67 TOPS en los módulos más recientes y opciones de potencia configurables, habilitando la inferencia de modelos como YOLOv8 con latencias inferiores a 100 ms cuando se optimiza con TensorRT [3].

Adicionalmente, se han realizado evaluaciones en el desempeño de algoritmos de detección de objetos de la familia YOLO en plataformas embebidas, evidenciando que la relación entre precisión y latencia depende no solo del modelo empleado, sino también del pipeline de implementación y del hardware utilizado. En este sentido, la ejecución sobre plataformas NVIDIA Jetson ha mostrado ser una alternativa viable para tareas de percepción en tiempo real, siempre que se consideren las limitaciones de cómputo y optimización del sistema [5, 6].

2.1.3. Sistema Robótico Pick and Place y Visión Artificial

En este contexto, los brazos robóticos ofrecen flexibilidad cinemática para adaptarse a diferentes tareas de manipulación, permitiendo alcanzar posiciones y orientaciones diversas en entornos tridimensionales, lo cual es fundamental para operaciones de Pick and Place, especialmente al ser combinadas con visión artificial. Un ejemplo concreto es presentado por Sroymuk et al. [7], quienes integraron el procesamiento de imágenes con un brazo robótico 6-DoF donde se evidenció la utilidad de la percepción visual para la manipulación precisa. De forma complementaria Li, Chen, Wu y Liu [8] desarrollaron un método de detección de agarre 6-DoF que utiliza imágenes RGB-D para predecir posturas de agarre con una alta tasa de éxito, evidenciando con estos trabajos que unir visión artificial con control robótico da pie a sistemas de automatización más precisos, adaptables y capaces de operar con menor intervención humana. Desde una perspectiva más amplia de automatización inteligente, la integración entre visión artificial, brazo robótico, sensores y sistemas de supervisión también ha sido abordada en arquitecturas de fábrica inteligente, donde los procesos pick and place sirven como caso de prueba para validar flexibilidad, adaptabilidad e interconexión entre módulos físicos y digitales [9].

Dentro del ámbito académico el uso de robots educativos en el diseño de sistemas logísticos ha permitido el desarrollo de metodologías basadas en modelos físicos a pequeña escala, los cuales representan procesos industriales reales. Estos modelos integran componentes como brazos robóticos, bandas transportadoras y sistemas de control, facilitando la simulación de flujos de materiales y la optimización de procesos [10].

Este enfoque permite validar configuraciones antes de su implementación en entornos reales, reduciendo costos y riesgos operativos. Además, favorece el desarrollo de competencias en automatización, control y diseño de sistemas ciberfísicos, alineándose

con los principios de la Industria 4.0 [10].

2.2. Marco Teórico

2.2.1. WLKATA Mirobot

Mirobot como se describe en el manual del fabricante [11], es un brazo robótico de escritorio de seis grados de libertad desarrollado por WLKATA Robotics, diseñado para prototipado y aprendizaje, una opción destacada en el sector educativo y entre creadores, sobresaliendo por su diseño compacto y su compatibilidad con múltiples lenguajes de programación incluyendo Python, C, C++, y Java, facilitando así la experimentación con robótica y programación en proyectos reales. La estructura del Mirobot como se observa en la Figura 2.2.1 [1], está compuesta por 5 elementos principales los cuales son su base, el brazo, antebrazo, muñeca y el efector terminal, siendo este último el que permite al robot interactuar físicamente con el entorno. Este, como se mencionó anteriormente cuenta con 6 articulaciones J1, J2, J3, J4, J5 Y J6, cada una de estas es controlada por un motor paso a paso.

Además de la documentación del fabricante, el WLKATA Mirobot ha sido reportado dentro de la literatura como una plataforma útil para actividades de simulación, formación e investigación en laboratorio, resaltando su portabilidad, precisión y compatibilidad con entornos de programación y control. Estas características lo convierten en una alternativa adecuada para prototipos académicos orientados a automatización y robótica aplicada [12].

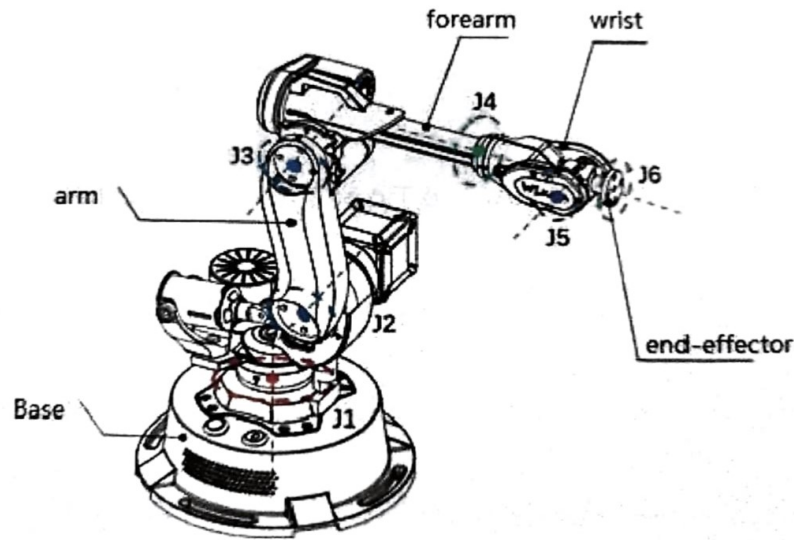


Figura 2.2.1: Estructura Mirobot. Tomado de [1]

En las tablas 2.2.1[1] y 2.2.2[1] se puede observar respectivamente, las especificaciones técnicas y los parámetros de movimiento de los ejes del brazo robótico Mirobot, en la primera se detallan las características generales del Mirobot incluyendo el número de ejes, carga útil, precisión del posicionamiento, interfaces de comunicación, alimentación requerida entre otros datos. En la segunda se detallan los rangos de movimiento y velocidades máximas de cada articulación.

Parameter	Value
Axis	6+1
Payload	150 g
Positioning	0.2 mm
Interface	USB / WiFi / Bluetooth
Voltage	100–240 V, 50/60 Hz
Input supply	12 V / 4 A DC
Power	50 W (Max)
Environment	−10°C – 60°C

Tabla 2.2.1: Especificaciones

Axis	Range	Max speed
1	-100° to $+100^\circ$	$31^\circ/\text{s}$
2	-60° to $+90^\circ$	$65^\circ/\text{s}$
3	-180° to $+50^\circ$	$28^\circ/\text{s}$
4	-180° to $+180^\circ$	$110^\circ/\text{s}$
5	-180° to $+40^\circ$	$33^\circ/\text{s}$

Tabla 2.2.2: Parámetros de movimiento

La plataforma WLKATA Mirobot ha demostrado ser una herramienta efectiva para la simulación y validación de procesos industriales en entornos controlados, permitiendo la implementación de sistemas de clasificación, manipulación y automatización de tareas [13].

Estos sistemas permiten la construcción de mini-líneas de producción donde múltiples robots, sensores y sistemas de visión trabajan de manera coordinada. En este tipo de configuraciones, un robot puede encargarse de la manipulación inicial de objetos, mientras que otro sistema, apoyado en visión artificial, realiza tareas de clasificación o inspección, replicando escenarios reales de manufactura inteligente [13].

2.2.2. Cinemática Mirobot

En el campo de la robótica, la cinemática se encarga del estudio del movimiento del robot ignorando las fuerzas necesarias para el mismo, en el caso de un brazo robot se enfoca en la relación existente entre las articulaciones del robot y la posición del efector final. En robótica se analizan principalmente dos tipos, la cinemática directa en la cual la posición del efector final se calcula teniendo en cuenta los ángulos de las articulaciones del robot. Por su parte en la cinemática inversa se calculan los ángulos que deben tener las articulaciones para llevar al efector final a una posición determinada.

Cinemática Inversa Mirobot

Como se mencionó anteriormente en la cinemática inversa, se conoce la posición del efector final, a partir de la posición deseada del efector final se hallan las variables de los ejes. Para el caso del Mirobot según el manual la función de la solución es muy relevante debido a que con un único chip de rendimiento limitado como unidad aritmética se vuelve necesario que el cálculo algorítmico no sea extenso, en el caso del Mirobot sus últimos tres ejes son perpendiculares entre sí y, además, intersecan en un mismo punto, con lo cual se cumple el criterio de Pieper lo cual implica que la solución es cerrada.

Con la anatomía del Mirobot en cuenta, conociendo que cumple el criterio de Pieper se pueden separar los ejes, siendo los ejes J1, J2 y J3 los que determinarán la posición

mientras que los ejes J4, J5 y J6 determinarán la orientación. Para lo primero tres ejes se usa el método geométrico, por su parte los últimos se resuelven mediante la transformación inversa de ángulos de Euler.

La posición del efector final es dada por el ángulo RPY (Roll-Pitch-Yaw) el cual permite expresar la posición del sistema como una secuencia de rotaciones alrededor de los ejes coordenados cartesianos, en este caso es necesario expresarlo en la forma representada por la matriz de rotación R . La rotación final se compone por la rotación del eje X en un ángulo α , continuando con la rotación en el eje Y por un ángulo de β y finalmente una rotación en el eje Z en un ángulo γ . Matemáticamente dicha relación se expresa como el producto de las matrices de rotación:

$$R = R_z(\gamma) R_y(\beta) R_x(\alpha) \quad (2.2.1)$$

Donde R_x , R_y y R_z corresponden a las matrices de rotación asociadas a cada eje, siendo estas:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz de rotación resultante puede expresarse en forma general como:

$$R = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$$

Los últimos tres ejes se resuelven mediante un método analítico para así determinar la orientación, la orientación se determina mediante la matriz de rotación R_0^6 , la cual es posible descomponer como el producto de dos matrices la rotación desde la base hasta el sistema de coordenadas 3, R_0^3 , y la rotación desde el sistema 3 hasta el efector final, R_3^6 relación que se expresa como:

$$R_0^6 = R_0^3 \cdot R_3^6 \quad (2.2.2)$$

La matriz R_3^6 , que contiene la información de la orientación de la muñeca, se puede obtener mediante:

$$R_3^6 = (R_0^3)^T \cdot R_0^6 \quad (2.2.3)$$

La matriz obtenida se representa de la siguiente manera:

$$R_3^6 = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}$$

En base a esta matriz se determinan los ángulos de los ejes J4, J5 y J6 utilizando la descomposición de ángulos de Euler del tipo ZYZ, con lo cual se pueden obtener de la siguiente forma:

$$\theta_5 = \text{atan2} \left(\sqrt{g_{31}^2 + g_{32}^2}, g_{33} \right) \quad (2.2.4)$$

$$\theta_4 = -\text{atan2} \left(\frac{g_{23}}{\sin \theta_5}, \frac{g_{13}}{\sin \theta_5} \right) \quad (2.2.5)$$

$$\theta_6 = \text{atan2} \left(\frac{g_{32}}{\sin \theta_5}, -\frac{g_{31}}{\sin \theta_5} \right) \quad (2.2.6)$$

Se debe considerar que existen ocho conjuntos de soluciones para los primeros tres ejes y dos conjuntos de soluciones para los últimos tres ejes, con esto en cuenta se debe limitar la posición de acuerdo al rango de cada eje y las restricciones físicas del robot para seleccionar la más adecuada.

2.2.3. Pick and Place

Un sistema pick and place hace referencia a una de las aplicaciones de la robótica, la cual se enfoca en automatizar la manipulación de objetos, como su nombre lo indica su función principal recoger (pick) un objeto determinado y colocar (place) dicho objeto en una ubicación determinada, logrando así realizar con precisión y eficiencia tareas repetitivas, debido a esto es uno de los usos más comunes de la robótica en el sector industrial[14].

Estos sistemas están compuestos por diferentes etapas comenzando por la detección, en esta etapa se identifica la presencia del objeto generalmente mediante sistemas de visión o utilizando sensores tras esto se procede a la localización, determinando la posición del objeto dentro del espacio de trabajo, tras esto se realiza el agarre, en esta etapa el robot a través del efector terminal recoge el objeto y por último en la etapa de colocación, traslada el objeto y lo ubica en la locación deseada siguiendo una trayectoria predefinida.

Las aplicaciones de los sistemas pick and place se encuentran en varios sectores, entre estos en manufactura [15] permiten automatizar procesos de ensamblaje[16], clasificación y empaquetado de productos. En logística se emplean para manipulación de mercancías, organización de inventarios y la automatización de bodegas, incluso en el

ámbito educativo sirven como herramientas para comprender conceptos de control, cinemática entre otros permitiendo experimentar procesos de automatización con escenarios reales a los estudiantes. El uso de capacidades de aprendizaje autónomo para el control de brazos robóticos de seis ejes en tareas de manipulación. Estos enfoques permiten aprender trayectorias de movimiento y ajustar acciones de agarre de manera progresiva, mostrando que la integración entre percepción, trayectoria y acción constituye un elemento central para mejorar el desempeño del sistema [17].

2.2.4. Cámaras de profundidad

Una de las principales tecnologías de percepción en sistemas robóticos son las cámaras de profundidad. A diferencia de las cámaras convencionales, que capturan únicamente información bidimensional, estas permiten obtener información tridimensional del entorno, proporcionando datos sobre la distancia entre el sensor y los objetos presentes en la escena.

Los datos de profundidad pueden obtenerse mediante distintas tecnologías. Una de ellas es la visión estéreo, basada en el uso de dos o más cámaras separadas por una distancia conocida, a partir de la cual se estima la profundidad mediante la disparidad entre imágenes. Otro método es la luz estructurada, en la cual se proyectan patrones luminosos sobre la escena y, a partir de su deformación, se infiere la geometría del entorno. También existe la tecnología *Time of Flight* (ToF), que calcula la distancia midiendo el tiempo que tarda un pulso de luz en ir y regresar al sensor.

En el caso de la familia Intel RealSense, la estimación de profundidad se apoya en visión estéreo activa, combinando cámaras laterales y un proyector infrarrojo para mejorar la correspondencia en escenas con baja textura. Esta configuración permite construir mapas de profundidad útiles para tareas de percepción robótica y localización de objetos [18]. En aplicaciones robóticas, estos mapas constituyen una base importante para la detección de obstáculos, la estimación espacial y la interacción con el entorno. Los sistemas de visión estéreo activos combinan el uso de múltiples cámaras con técnicas de proyección y análisis de flujo óptico para mejorar la percepción tridimensional del entorno. Este enfoque permite ampliar el campo de visión y mejorar la estimación de profundidad en escenarios complejos, siendo una base teórica relevante para sistemas modernos de percepción como las cámaras RGB-D utilizadas en robótica [19].

2.2.5. Detección y Localización de objetos

Constituye un componente esencial en sistemas de percepción robótica, ya que permite identificar la presencia de elementos en la escena y estimar su posición en el espacio. La detección se refiere al reconocimiento de un objeto dentro de los datos sensoriales, mientras que la localización implica determinar su ubicación espacial. En robótica manipuladora, estos procesos son fundamentales para la ejecución de tareas de agarre, ensamblaje y clasificación automatizada.

La forma en que un sistema robótico “percibe” los objetos se basa en el procesamiento de información proveniente de sensores visuales, como cámaras RGB o de profundidad. Mediante técnicas de visión artificial, el sistema analiza características visuales relevantes para diferenciar los objetos del entorno y segmentar regiones de interés. Este procesamiento permite estimar la posición relativa de los objetos dentro del campo de visión del sensor.

Un aspecto clave en la integración de visión y manipulación robótica es la relación entre coordenadas de imagen y coordenadas del robot. Las coordenadas de imagen se expresan en píxeles dentro del plano de la cámara, mientras que las coordenadas del robot se definen en un sistema de referencia tridimensional asociado a su base. Para que el manipulador interactúe con los objetos detectados, es necesario aplicar procesos de calibración y transformaciones geométricas que conviertan la información visual en posiciones métricas utilizables por el robot. Esta correspondencia espacial resulta determinante para la precisión en tareas de manipulación. Los sistemas modernos de visión por computador para detección y localización de objetos se basan en modelos de aprendizaje profundo que permiten identificar regiones de interés en imágenes y video en tiempo real. Entre los enfoques más utilizados se encuentran Faster R-CNN, Single Shot Detector (SSD) y You Only Look Once (YOLO), los cuales presentan diferentes compromisos entre precisión y velocidad [20].

En particular, los modelos tipo YOLO destacan por su capacidad de detección en tiempo real, mientras que arquitecturas como SSD combinadas con MobileNet permiten implementaciones eficientes en entornos embebidos, reduciendo significativamente la complejidad computacional sin comprometer el rendimiento [20].

Adicionalmente, técnicas complementarias como el flujo óptico, la sustracción de fondo y el análisis de secuencias de video permiten mejorar la localización y seguimiento de objetos en escenarios dinámicos, lo cual resulta fundamental en aplicaciones robóticas donde es necesario no solo detectar sino también estimar la trayectoria de los objetos en el espacio [20].

2.2.6. YOLO V8

YOLOv8.0 (You Only Look Once) es un modelo de detección de objetos basado en aprendizaje profundo que pertenece a la familia de detectores de una sola etapa (single-stage detectors). Su arquitectura se fundamenta en redes neuronales convolucionales capaces de predecir simultáneamente la clase de los objetos y las coordenadas de sus cajas delimitadoras (bounding boxes). Este enfoque permite realizar detección en tiempo real con un balance favorable entre precisión y velocidad de inferencia.

En aplicaciones robóticas, YOLOv8.0 resulta especialmente útil debido a su capacidad para operar en tiempo real y adaptarse a diferentes escenarios mediante entrenamiento con conjuntos de datos específicos. Su implementación en sistemas de manipulación robótica facilita la identificación rápida de objetos y su posterior localización, contribuyendo a

mejorar la eficiencia de tareas automatizadas como los sistemas pick and place.

2.2.7. RealSense D435i

La RealSense D435i, presentada en la Figura 2.2.2 de acuerdo al manual de usuario [21] es una cámara de profundidad desarrollada por Intel, ampliamente utilizada en aplicaciones de robótica, visión artificial y percepción tridimensional. Esta integra sensores estéreo infrarrojos para estimación de profundidad y una cámara RGB para captura de color, permitiendo obtener simultáneamente información visual y espacial del entorno. Una característica distintiva de la D435i es su unidad de medición de inercia (IMU) combinando acelerómetro y giroscopio, gracias al cual se obtienen datos de movimiento y orientación refinando la percepción de profundidad en situaciones en las cuales la cámara entre en movimiento.



Figura 2.2.2: Cámara de profundidad RealSense D435i. Tomado de: [2].

Desde la perspectiva de la visión robótica, la familia RealSense ha sido empleada en aplicaciones de generación de imágenes de profundidad y detección de objetos, reportándose que la calidad de la medición depende del modelo de cámara y de las condiciones de captura. En particular, se ha señalado que la D415 ofrece mayor precisión en profundidad que la D435 en ciertos escenarios cercanos, mientras que la D435 privilegia un campo de visión más amplio y un mejor comportamiento ante movimiento rápido [18], esta proyecta un patrón infrarrojo ayudando a mejorar la correspondencia entre imágenes permitiendo estimar con precisión distancias a diferentes rangos facilitando la detección de objetos. Es también valorada en entornos educativos y de investigación debido a su compatibilidad con bibliotecas de desarrollo. Su uso contribuye al desarrollo de sistemas capaces de interpretar la geometría del entorno y apoyar la toma de decisiones en aplicaciones de manipulación automatizada.

2.3. Métricas de evaluación en detección de objetos

La evaluación de modelos de detección de objetos requiere métricas que consideren, de forma simultánea, la capacidad del modelo para *clasificar* correctamente los objetos y

localizarlos mediante cuadros delimitadores (*bounding boxes*). En este trabajo se emplean métricas ampliamente aceptadas en la literatura y en herramientas de entrenamiento modernas: precisión (Precision), exhaustividad (Recall), F1-score y la media de precisión promedio (mAP). Estas métricas permiten contrastar el desempeño del detector tanto en escenarios de validación como durante el ajuste del modelo, y son particularmente útiles para sistemas robóticos guiados por visión, donde errores de detección o localización afectan etapas posteriores como la estimación espacial y el control del manipulador [22].

2.3.1. Precisión, recall y F1-score

La **precisión** (Precision) cuantifica la proporción de detecciones realizadas por el modelo que corresponden efectivamente a objetos correctos, mientras que el **recall** mide la proporción de objetos presentes que fueron detectados. En detección de objetos, ambas métricas dependen de un umbral de confianza para aceptar predicciones y de un criterio geométrico que determina si un *bounding box* predicho coincide con el real. Por su parte, el **F1-score** se utiliza como una medida de equilibrio entre precisión y recall, ya que corresponde a su media armónica. En la práctica, la inspección de curvas asociadas a distintos umbrales de confianza (por ejemplo, F1-Confidence) permite seleccionar valores operativos que balanceen falsos positivos y falsos negativos, de acuerdo con las necesidades de la aplicación [23].

2.3.2. mAP: precisión promedio y desempeño global

La métrica **AP** (*Average Precision*) se obtiene a partir de la curva Precision–Recall, y resume el desempeño del detector en una clase específica mediante el área bajo dicha curva. La **mAP** (*mean Average Precision*) corresponde al promedio de AP sobre todas las clases, y se utiliza como un indicador global del detector [22]. En términos operativos, la mAP permite comparar modelos o configuraciones de entrenamiento de manera consistente, considerando el compromiso entre precisión y recall a través de diferentes umbrales de decisión.

En implementaciones modernas de YOLO se reportan variantes de mAP que difieren en el criterio de coincidencia geométrica entre cuadros predichos y reales, expresado mediante el umbral de IoU (*Intersection over Union*). En particular, es común reportar **mAP@0.5** (**mAP50**) y **mAP@0.5:0.95** (**mAP50-95**), donde la segunda es más exigente al promediar el desempeño sobre múltiples umbrales de IoU. Estas métricas son empleadas como referencia estándar en los reportes de entrenamiento y validación de modelos YOLO, permitiendo evaluar tanto la capacidad de detección como la calidad de localización [23].

2.3.3. Relación de las métricas con este proyecto

En el contexto del presente trabajo, estas métricas permiten sustentar el uso de YOLOv8 como módulo de percepción para tareas Pick and Place. Valores altos de Precision y Recall indican que el modelo detecta correctamente las clases definidas y cubre la mayoría de los objetos presentes. Asimismo, valores elevados de mAP sugieren que los *bounding boxes* se ajustan de manera adecuada a los objetos, lo cual es relevante debido a que el centroide del cuadro se emplea como referencia para consultar profundidad y estimar coordenadas espaciales. Por tanto, el análisis conjunto de Precision, Recall, F1-score y mAP constituye la base para interpretar el desempeño del detector y su idoneidad dentro del flujo percepción-control del sistema robótico [22, 23].

Un importante factor a considerar en el desempeño de estos sistemas no depende únicamente del algoritmo de detección, sino también de la calidad de los datos de entrada y de las condiciones del entorno. Factores como la iluminación, la textura de los objetos y la presencia de ruido en la señal pueden afectar significativamente la precisión del sistema. Por esta razón, en aplicaciones de robótica guiada por visión, se hace necesario complementar los modelos de detección con estrategias de procesamiento adicional que permitan mejorar la estabilidad y confiabilidad de las mediciones obtenidas.

2.4. Metodología

El proyecto se basa en el desarrollo de un prototipo Pick and Place guiado por visión, construido a partir de la integración progresiva de tres componentes principales: un módulo de percepción tridimensional proveniente de la cámara Intel RealSense D435i, un módulo de detección de objetos sustentado en YOLOv8, y un módulo de control y ejecución de movimiento para el brazo WLKATA Mirobot 6-axis. Esta integración se realiza sobre una plataforma embebida NVIDIA Jetson Orin Nano, que actúa como núcleo de procesamiento para ejecutar la inferencia y coordinar el flujo de información entre percepción y control.

El trabajo se desarrolla por fases que inician con la preparación del entorno y los recursos, continúan con la construcción del conjunto de prueba y la puesta a punto del modelo de detección, y posteriormente integran la estimación espacial del objeto con la planificación de trayectorias del manipulador para completar el ciclo de agarre y colocación. Finalmente, el sistema se valida mediante pruebas experimentales con distintos objetos, registrando métricas como tasa de éxito y precisión de colocación, de modo que los resultados permitan ajustar parámetros y refinar el desempeño del prototipo como se observa en la figura 2.4.1. Cabe resaltar que, aunque el sistema se describe como un flujo continuo, su desarrollo se llevó a cabo de manera progresiva por fases, lo cual permitió validar cada componente antes de su integración final.

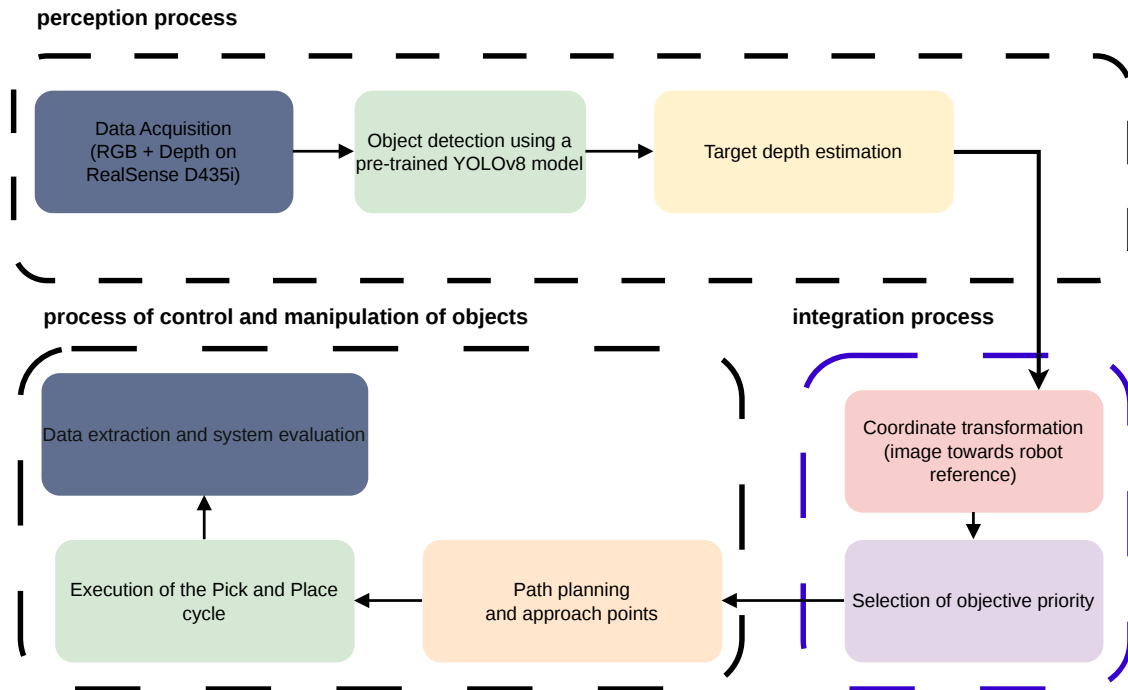


Figura 2.4.1: Flujo general del sistema propuesto en la metodología
Fuente: Elaboración propia.

2.4.1. Materiales

Los recursos implementados en el desarrollo del sistema son:

- Brazo robótico WLKATA Mirobot 6-axis.
- Cámara Intel RealSense D435i.
- Tarjeta embebida NVIDIA Jetson Orin Nano.
- Cuatro objetos de prueba con geometría, tamaño y color diferentes.
- Software y librerías especializadas para visión artificial y control.

2.4.2. Fases de desarrollo

Fase 1: Inicio y planificación

En esta etapa se definieron el alcance técnico del prototipo, los requerimientos de operación y los criterios de validación. Se estableció el flujo general de trabajo para un ciclo Pick and Place (detección, estimación 3D, ejecución de trayectoria y verificación),

así como los recursos necesarios para su implementación. También se definieron los criterios mínimos de verificación para poder avanzar a las fases siguientes. En particular, se precisó la secuencia general del ciclo Pick and Place (detección, estimación de coordenadas, selección de objetivo y ejecución del movimiento) y se determinaron las condiciones del entorno de prueba (disposición de la maqueta, ubicación de la cámara y zona de operación del brazo). Como resultado, se planteó un flujo de trabajo coherente y verificable, junto con la identificación de las entradas y salidas esperadas de cada módulo, de modo que la integración posterior no dependiera de ajustes improvisados.

Asimismo, se seleccionó el conjunto de objetos de prueba bajo el criterio de que estos debían presentar diferencias claras en altura, forma y tamaño, con el fin de evaluar el sistema de manera más efectiva.

Fase 2: Configuración del núcleo embebido e integración base

Durante esta fase se configuró la NVIDIA Jetson Orin Nano como unidad central de procesamiento para ejecutar los módulos de visión y coordinación del sistema. Se realizaron pruebas de comunicación e intercambio de datos entre la cámara Intel RealSense D435i y el módulo de control del brazo robótico, comprobando estabilidad operativa bajo ejecución continua.

Dicha estabilidad se verificó mediante la visualización simultánea de color y profundidad en la cámara (y su alineación entre *frames*), así como la ejecución repetible de comandos básicos del manipulador hacia posiciones predefinidas y el retorno al *waypoint*. De esta forma, la salida de la fase corresponde a un entorno listo para pruebas: adquisición RGB-D estable y control del brazo funcional, condiciones necesarias para integrar detección y estimación espacial sin introducir fallos atribuibles a problemas de comunicación o configuración.

Fase 3: Percepción tridimensional y detección de objetos

En esta etapa se desarrolló el componente de percepción 3D, utilizando la información RGB y profundidad de la cámara RealSense D435i para estimar la ubicación espacial de los objetos. Paralelamente, se implementó un modelo de detección de objetos basado en YOLOv8, el cual permite identificar el objeto objetivo en la escena. A partir de estas salidas, se definió un algoritmo de extracción de datos, para obtener coordenadas XYZ en la zona de trabajo.

Para asegurar consistencia, se adoptó el centroide del *bounding box* como referencia y se aplicó una estrategia robusta para la profundidad basada en la mediana de una ventana local, reduciendo el impacto de ruido y valores atípicos del mapa *depth*. El resultado esperado de esta fase es, por tanto, un modelo con métricas de desempeño que respalden su uso y un algoritmo para convertir detección 2D en una estimación espacial en el control del manipulador.

Fase 4: Integración percepción–control y ejecución del ciclo Pick and Place

En esta fase se integró la salida del sistema de visión con el módulo de control del manipulador, de manera que la detección del objeto y su ubicación espacial se conviertan en una referencia de movimiento para el brazo robótico. A partir de las coordenadas estimadas del objetivo, se definió el procedimiento para calcular el punto de aproximación, el punto de agarre y definir el punto de colocación, asegurando una secuencia coherente del ciclo Pick and Place. Posteriormente, se implementaron rutinas de movimiento para ejecutar cada etapa (aproximación, agarre, traslado y posicionamiento), ajustando velocidades y movimientos intermedios con el fin de mejorar la estabilidad, reducir desviaciones y minimizar fallos de ejecución, añadiendo parámetros de seguridad como alturas mínimas y retorno a un *waypoint* predefinido. Asimismo, cuando se presentan múltiples detecciones, se incorporó un criterio de selección del objetivo para evitar cambios abruptos entre iteraciones y mantener estabilidad durante la ejecución. En consecuencia, la salida de esta fase es un flujo integrado que permite iniciar el ciclo Pick and Place de manera consistente a partir de la información sensorial.

Fase 5: Evaluación experimental y análisis de desempeño

Con el sistema integrado, se realizaron pruebas experimentales utilizando los objetos definidos en la fase de preparación, con el objetivo de supervisar el comportamiento del prototipo en condiciones controladas. La evaluación se centró en determinar la eficacia en la localización y colocación, la repetibilidad del ciclo, y la tasa de éxito del agarre y posicionamiento. Además, se documentaron fallos típicos (detección insuficiente, errores de estimación espacial o desalineación durante el agarre) para identificar causas y oportunidades de mejora. Los resultados obtenidos en esta fase sirven como base para el ajuste de parámetros y la optimización del sistema.

La validación se orientó a recolectar evidencia que permita sustentar el comportamiento del sistema más allá de una demostración puntual. Para ello se registraron salidas del detector, coordenadas estimadas y el comportamiento del criterio de priorización cuando existen múltiples objetos en escena.

Fase 6: Ajustes y mejora iterativa

Finalmente, con base en los resultados de validación, se realizaron ajustes sobre la rutina de detección y el refinamiento de la estrategia de trayectoria, con el fin de mejorar y optimizar el sistema. Se propusieron recomendaciones para futuras extensiones del sistema o versiones futuras del mismo.

En esta fase los ajustes se realizaron con base en la evidencia recopilada, priorizando mejoras que impactan directamente la estabilidad del flujo percepción–control. Entre estos ajustes se incluyen el ajuste del umbral de confianza del detector a partir de las

curvas de desempeño y pruebas de detección de los objetos sobre el escenario. La selección del punto objetivo para consultar profundidad, buscando reducir falsos positivos y evitar estimaciones erróneas en situaciones de solapamiento. De igual manera, se refinaron parámetros de ejecución del brazo (alturas de seguridad y secuencia de movimiento) con el objetivo de aumentar la repetibilidad en el entorno de pruebas.

Capítulo 3

Análisis de resultados

3.1. Descripción de la maqueta, conexiones y flujo de operación

La maqueta utilizada en este trabajo corresponde a un entorno físico controlado diseñado para validar el funcionamiento del sistema Pick and Place guiado por visión. El montaje se realizó sobre una base fija que integra el plano de trabajo, la ubicación del brazo robótico WLKATA Mirobot 6-axis y el soporte de la cámara Intel RealSense D435i. En la Figura 3.1.1 se observa la disposición general: el manipulador se posiciona de forma que cubra el área de operación sobre el plano, mientras que la cámara se ubica en un soporte vertical orientado hacia la zona de trabajo para capturar información RGB-D. Esta configuración permite realizar pruebas repetibles, controlar la posición relativa entre sensores y actuadores, y delimitar de manera clara la zona donde se ubican los objetos de prueba durante la ejecución.

Una vez establecido el espacio físico, el sistema se integra a nivel de conexiones entre módulos. La cámara Intel RealSense D435i transmite a la NVIDIA Jetson Orin Nano los datos de color y profundidad, permitiendo la adquisición de información RGB-D de manera sincronizada. Sobre la Jetson se ejecuta el módulo de visión y el pipeline de percepción, encargado de procesar la escena, identificar el objeto, estimar su ubicación y ejecutar la secuencia de manipulación. La Figura 3.1.2 resume estas conexiones y el intercambio de información entre los componentes principales.

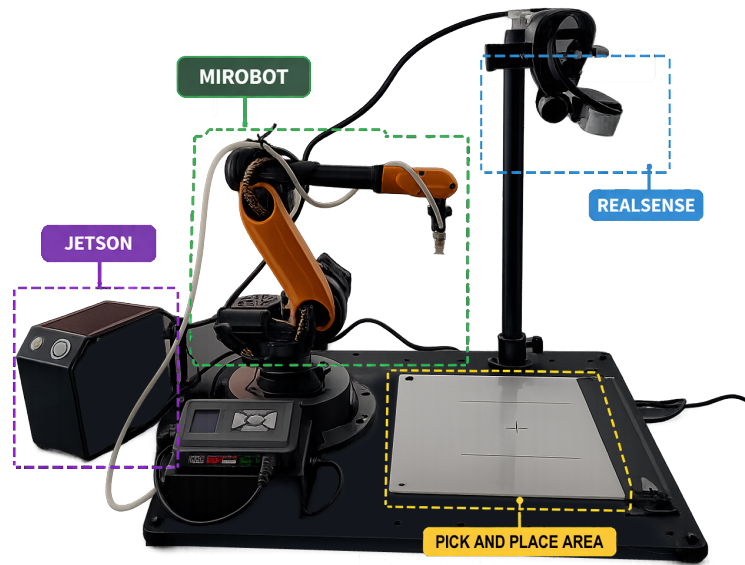


Figura 3.1.1: Vista general de la maqueta y del espacio físico de trabajo (cámara, brazo robótico y plano de operación). Fuente: Elaboración propia.

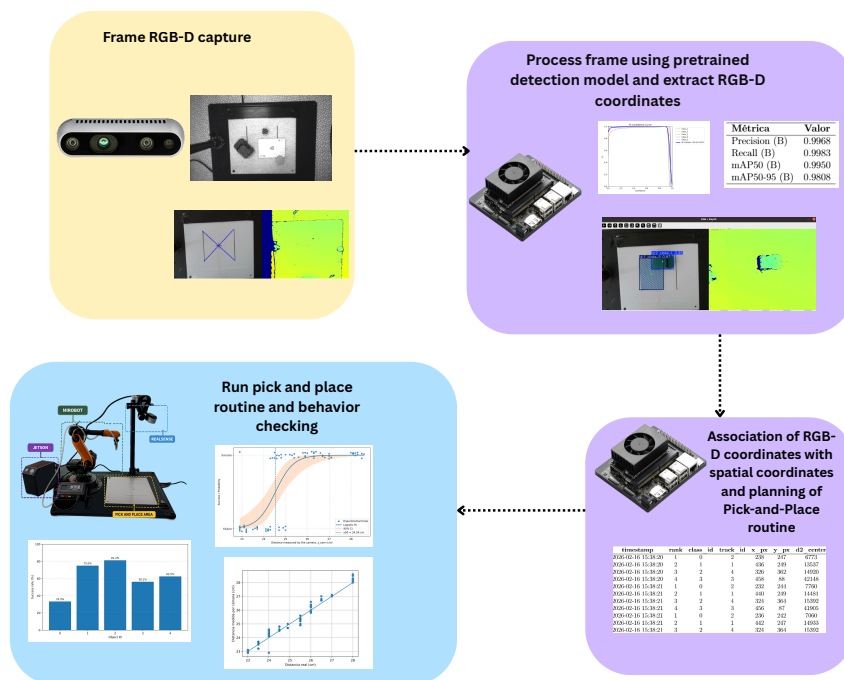


Figura 3.1.2: Esquema general de conexión cámara-Jetson-brazo y flujo de trabajo del sistema. Fuente: Elaboración propia, adaptada de [2] y [3].

En términos funcionales, el flujo de operación del sistema en la maqueta sigue una secuencia definida: (1) adquisición RGB-D, (2) detección y clasificación del objeto por el modelo YOLOv8, (3) estimación espacial a partir del centroide del *bounding box* y la profundidad alineada al frame de color, (4) selección del objetivo cuando existen

múltiples detecciones en escena, y (5) ejecución de la rutina del manipulador (aproximación, agarre, traslado y colocación). Además, el sistema genera evidencia del proceso mediante registros (por ejemplo, archivos CSV con coordenadas y priorización del objetivo) e imágenes de seguimiento, lo que permite sustentar el análisis presentado en el Capítulo de Resultados.

3.2. Métricas de desempeño del modelo

El desempeño del modelo entrenado se evaluó mediante métricas estándar de detección: Precision, Recall, mAP50 y mAP50-95. Estas métricas permiten verificar si el detector es suficientemente confiable para ser utilizado como entrada del sistema de Pick and Place, ya que una detección incorrecta o inestable afecta directamente la estimación de coordenadas y la selección del objetivo.

La Tabla 3.2.1 presenta los resultados globales obtenidos al finalizar el entrenamiento. En general, los valores altos de Precision y Recall indican que el modelo identifica correctamente las clases definidas y que, en la mayoría de los casos, detecta los objetos presentes en escena. Por su parte, los valores elevados de mAP50 y mAP50-95 evidencian un buen desempeño en la localización de los objetos mediante *bounding boxes*, lo cual es especialmente relevante para este proyecto debido a que el centroide de la caja se utiliza como referencia para consultar la profundidad y estimar coordenadas espaciales.

Estos resultados respaldan el uso de YOLOv8 como módulo de detección para el prototipo en el entorno de pruebas utilizado, permitiendo continuar con la integración de percepción RGB-D y control del manipulador.

Tabla 3.2.1: Métricas globales del modelo YOLOv8 al final del entrenamiento

Métrica	Valor
Precision (B)	0.9968
Recall (B)	0.9983
mAP50 (B)	0.9950
mAP50-95 (B)	0.9808

3.2.1. Análisis de curvas de entrenamiento, curvas de confianza y matriz de confusión

Curvas generales de entrenamiento. La Figura 3.2.1 resume el comportamiento del entrenamiento y la validación a lo largo de las épocas. En ella se observan, típicamente, las pérdidas asociadas a localización y clasificación (tendencia decreciente) y el aumento progresivo de las métricas de validación hasta estabilizarse. En este proyecto, al tratarse de cinco clases con diferencias marcadas (Objeto 1–5), es esperable que la convergencia se alcance con métricas elevadas.

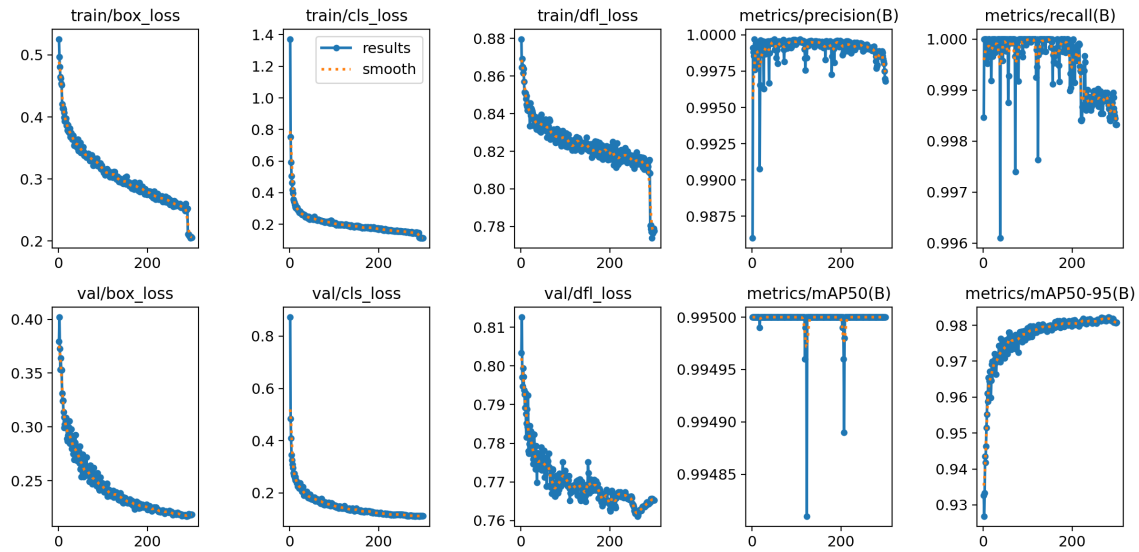


Figura 3.2.1: Curvas generales de entrenamiento y validación del modelo YOLOv8. Fuente: Elaboración propia.

Curva F1-Confidence. La Figura 3.2.2 muestra el comportamiento del F1-score en función del umbral de confianza. El **F1-score** es la media armónica entre precisión y recall, por lo que resume el balance entre falsos positivos y falsos negativos. Esta curva es clave para identificar un umbral de operación que mantenga un compromiso adecuado. En este proyecto, el umbral operativo se definió teniendo en cuenta este balance, fijando un **umbral de confianza del 87 %** ($\text{conf} \geq 0,87$) para aceptar detecciones.

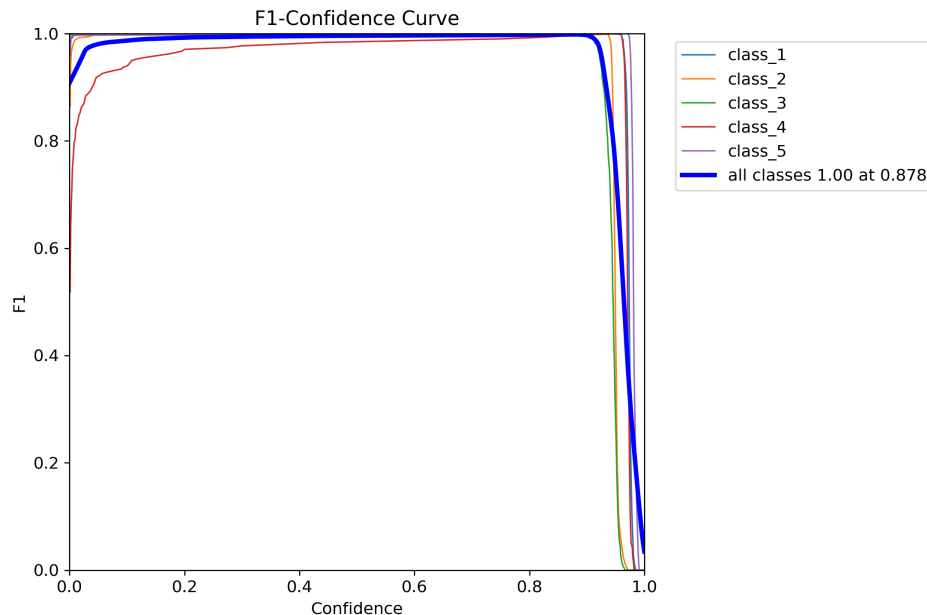


Figura 3.2.2: Curva F1-Confidence del modelo YOLOv8. Fuente: Elaboración propia.

Curva Precision-Recall. La Figura 3.2.3 presenta la relación entre precisión y recall

para distintos umbrales de confianza. Esta curva muestra el compromiso natural del detector: al exigir mayor confianza, normalmente aumenta la precisión, pero disminuye el recall. Un comportamiento favorable se refleja en una curva que mantiene altos valores de precisión para rangos amplios de recall (curva cercana a la esquina superior derecha). En el modelo entrenado, la curva Precision–Recall sugiere que el detector mantiene un equilibrio fuerte entre detecciones correctas y cobertura de objetos presentes.

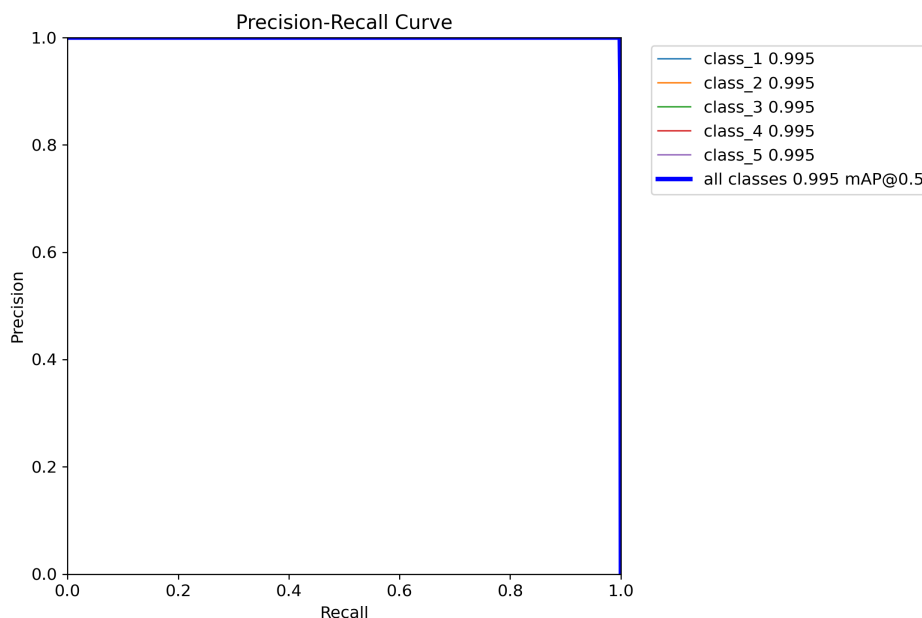


Figura 3.2.3: Curva Precision–Recall del modelo YOLOv8. Fuente: Elaboración propia.

Matriz de confusión. La Figura 3.2.4 muestra la matriz de confusión normalizada. Esta matriz permite observar cómo se distribuyen las predicciones del modelo: idealmente, la mayor proporción debe concentrarse en la diagonal (predicciones correctas por clase), y valores fuera de la diagonal representan confusiones entre clases.

A primera vista, una matriz casi perfecta y métricas cercanas a 1 pueden interpretarse como indicio de sobreentrenamiento. Sin embargo, en este caso existen factores que hacen razonable dicho comportamiento: (i) las clases están compuestas por objetos con diferencias visuales notorias (geometría/tamaño/color), lo cual reduce la probabilidad de confusión; (ii) el escenario de captura es controlado, por lo que la variabilidad del mismo es limitada; y (iii) el objetivo del proyecto en esta etapa es validar funcionalmente la integración detección–percepción–control, antes de escalar a condiciones más exigentes. Por ello, un rendimiento alto es aceptable como resultado del entrenamiento.

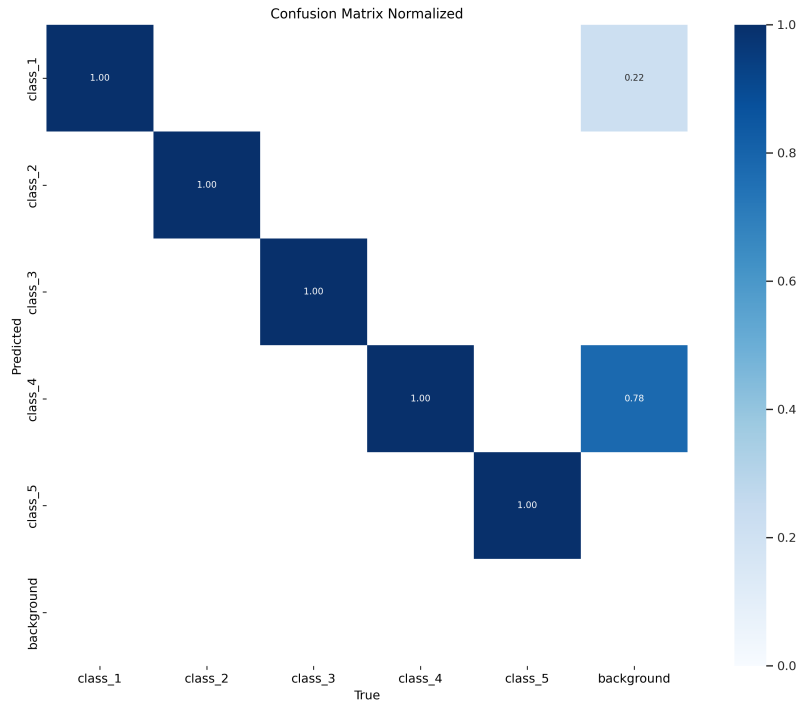


Figura 3.2.4: Matriz de confusión normalizada del modelo YOLOv8. Fuente: Elaboración propia.

3.3. Resultados del módulo de percepción y estimación de coordenadas

Una vez el modelo YOLOv8 identifica el objeto objetivo (Objeto 1–5), el sistema de percepción estima su ubicación espacial combinando el centroide del *bounding box* en la imagen RGB con la información de profundidad provista por la cámara Intel RealSense D435i. En la Figura 3.3.1 puede apreciarse el comportamiento de la percepción estereoscópica de la cámara y cómo esta interactúa con la imagen RGB. Esta etapa es crítica, ya que la salida del módulo de percepción se utiliza como entrada para el control del manipulador durante el ciclo Pick and Place.

En la implementación, la coordenada de profundidad se obtiene a partir del mapa *depth* alineado a la imagen de color, y se calcula de forma robusta usando la mediana de una ventana local alrededor del centroide. Posteriormente, esta profundidad se transforma a una referencia utilizable por el robot (denominada z_{robot}) y se conserva el par (x, y) en píxeles como referencia del punto detectado en el plano de imagen.

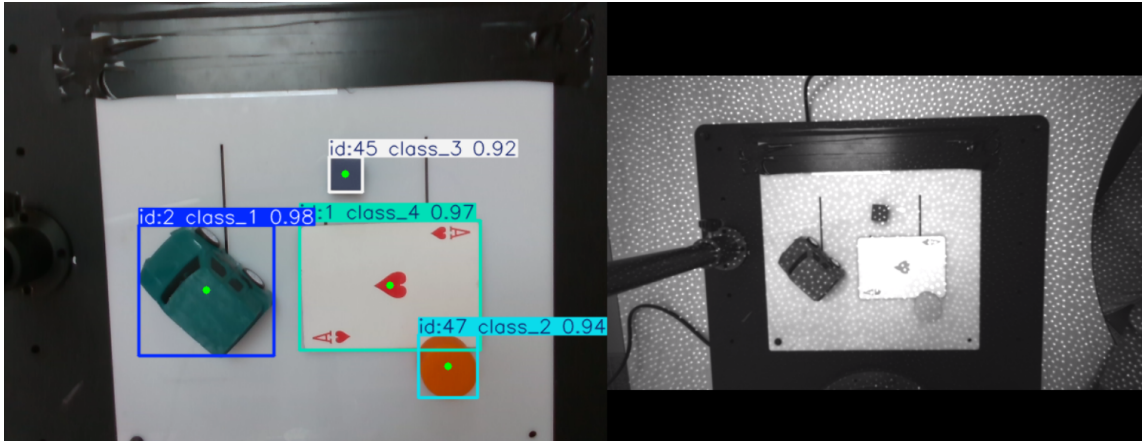


Figura 3.3.1: Vista del sistema estereo-activo durante la detección y estimación de coordenadas con Intel RealSense D435i. Fuente: Elaboración propia.

3.3.1. Evidencia de coordenadas registradas

Como evidencia del funcionamiento del módulo, se generó un registro en formato CSV donde se almacenan las coordenadas obtenidas para las detecciones realizadas. La Tabla 3.3.1 presenta una muestra representativa del archivo `camera_coords.csv`, donde se reporta el identificador de clase (*class_id*) y las coordenadas asociadas (*x-camera*, *y-camera*, *z-camera*). En este contexto, *x-camera* y *y-camera* representan la ubicación del centroide en píxeles, mientras que *z-depth* corresponde al valor de profundidad de este mismo centroide.

Estos registros evidencian que el sistema logra producir de forma continua coordenadas asociadas a las detecciones, lo que permite alimentar la etapa de selección de objetivo y la planificación de movimiento. Dado que en esta etapa del proyecto se trabaja en un entorno controlado, la consistencia de la estimación se interpreta como la capacidad de mantener valores estables de profundidad y posición para un mismo objeto bajo condiciones similares de captura.

Tabla 3.3.1: Muestra de coordenadas registradas por el módulo de percepción (ejemplo representativo)

<code>class_id</code>	<code>x-camera (px)</code>	<code>y-camera (px)</code>	<code>z-depth</code>
1	406	359	49
0	524	252	84
3	386	149	42
2	284	339	49
1	402	359	47
0	516	257	84
3	386	149	42
2	284	339	49
1	404	359	47
0	522	254	84

3.4. Integración entre cámara y brazo robótico

Previo a la ejecución del ciclo Pick and Place, se realizó la integración funcional entre la cámara Intel RealSense D435i y el brazo WLKATA Mirobot 6-axis, con el objetivo de garantizar dos condiciones básicas: (i) adquisición estable de datos RGB-D y (ii) ejecución confiable de comandos de movimiento por parte del manipulador. Esta integración es un resultado clave, ya que habilita el flujo completo de percepción–control utilizado posteriormente por el sistema.

3.4.1. Prueba de movimiento y comunicación con el manipulador

De manera complementaria, se desarrolló un script de prueba de movimiento con el fin de validar la comunicación con el manipulador y comprobar que el brazo ejecuta de forma consistente movimientos básicos requeridos para el ciclo Pick and Place. En esta prueba se verificó la ejecución de comandos hacia posiciones predefinidas, movimientos de aproximación y retorno a un punto de referencia (*waypoint*), garantizando que el manipulador responde a las instrucciones del sistema de control.

El resultado de esta validación confirma que el brazo puede operar bajo una secuencia controlada de movimientos, condición necesaria para integrar posteriormente la información espacial proveniente de la cámara y ejecutar la rutina completa de manipulación.

3.4.2. Asociación de coordenadas RGB-D con coordenadas espaciales del robot y ajuste del plano de trabajo

Como parte de la integración percepción–control, se implementó un procedimiento para asociar las coordenadas obtenidas en la imagen RGB con la información de profundidad (RGB-D) y convertirlas en una referencia espacial utilizable por el robot.

Una vez establecido el flujo base, a partir de la detección del objeto se obtiene el centroide (x, y) en píxeles, se consulta la profundidad correspondiente en el mapa *depth* alineado al color, y se calcula un valor de distancia usando una ventana local (mediana). Posteriormente, esta profundidad se transforma a la referencia de trabajo del manipulador, permitiendo que la detección en la imagen se convierta en una coordenada espacial para la ejecución del movimiento.

De forma complementaria, para ajustar la cámara respecto al entorno de trabajo y verificar visualmente la correspondencia entre el plano observado y el plano de operación del robot, se utilizó un conjunto de puntos de referencia. Dichos puntos se proyectan y se unen en el frame de la cámara, formando una marca en forma de “X” que sirve como guía visual de alineación. La Figura 3.4.1 muestra este resultado, el cual confirma que el sistema está utilizando una referencia geométrica consistente para el

ajuste del plano de trabajo previo a las pruebas de manipulación.

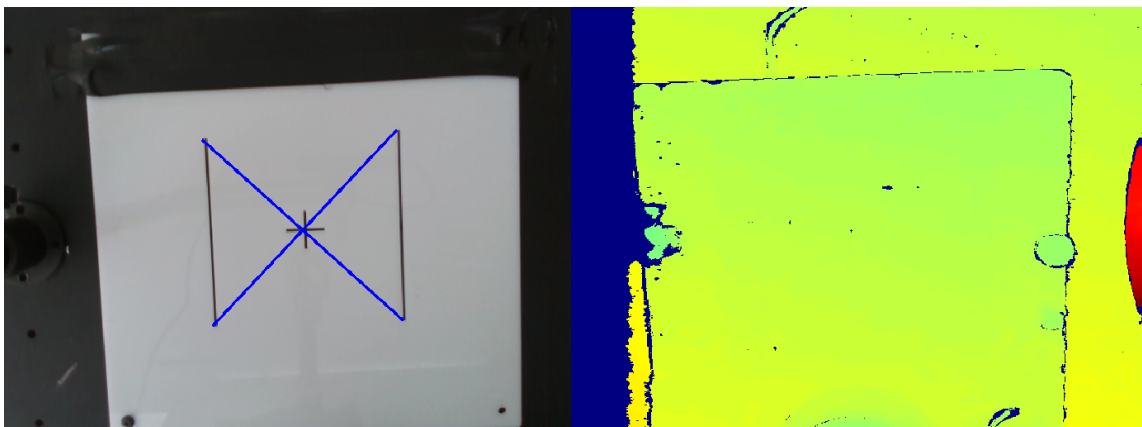


Figura 3.4.1: Referencia visual de ajuste del plano de trabajo: “X” dibujada en el frame usando los puntos *robot_points*. Fuente: Elaboración propia.

Con el fin de contextualizar el ajuste realizado, la Figura 3.4.1 muestra la referencia visual utilizada para alinear el plano de operación en la perspectiva de la cámara.

3.5. Resultados del criterio de selección y priorización del objetivo

Durante la operación del sistema es posible que se detecten múltiples objetos de forma simultánea. Para garantizar una ejecución estable del ciclo Pick and Place, se implementó un criterio de priorización que permite seleccionar un único objetivo por iteración. En este proyecto, la priorización se realiza con base en la cercanía del objeto al centro de la imagen, bajo el supuesto de que un objeto más centrado en el campo visual tiende a mantener un seguimiento más estable y una estimación espacial más consistente.

El criterio de selección se soporta en la métrica $d2_center$, que representa la distancia en el plano de la imagen entre el centroide del objeto detectado (x_{px}, y_{px}) y el centro del encuadre. A partir de esta métrica, las detecciones se ordenan y se asigna un *rank* (1, 2, 3, ...), donde $rank = 1$ corresponde al objetivo con mayor prioridad.

3.5.1. Evidencia del registro de priorización (*rank log*)

Como evidencia del funcionamiento del criterio, se generó un registro en formato CSV que almacena, por instante de tiempo, el orden asignado a cada detección. Este registro incluye el identificador de clase (*class_id*), el identificador de seguimiento (*track_id*), las coordenadas en píxeles (x_{px}, y_{px}) , una estimación de profundidad adaptada al robot (z_{robot}) y la distancia al centro ($d2_center$). La Tabla 3.5.1 presenta una muestra representativa del comportamiento del ranking.

Tabla 3.5.1: Muestra representativa del registro de priorización del objetivo

timestamp	rank	class_id	track_id	x_px	y_px	d2_center
2026-02-16 15:38:20	1	0	2	238	247	6773
2026-02-16 15:38:20	2	1	1	436	249	13537
2026-02-16 15:38:20	3	2	4	326	362	14920
2026-02-16 15:38:20	4	3	3	458	88	42148
2026-02-16 15:38:21	1	0	2	232	244	7760
2026-02-16 15:38:21	2	1	1	440	249	14481
2026-02-16 15:38:21	3	2	4	324	364	15392
2026-02-16 15:38:21	4	3	3	456	87	41905
2026-02-16 15:38:21	1	0	2	236	242	7060
2026-02-16 15:38:21	2	1	1	442	247	14933
2026-02-16 15:38:21	3	2	4	324	364	15392
2026-02-16 15:38:21	4	3	3	458	91	41245
2026-02-16 16:39:02	1	0	1	252	247	4673
2026-02-16 16:39:02	2	1	3	442	247	14933
2026-02-16 16:39:02	3	3	2	454	82	42920

3.5.2. Análisis del criterio de selección

Los registros obtenidos evidencian que el sistema mantiene una lógica consistente para seleccionar el objetivo cuando existen múltiples detecciones. En particular, la priorización basada en *d2_center* contribuye a reducir cambios abruptos del objetivo entre iteraciones, lo cual es relevante para el ciclo Pick and Place: la planificación del movimiento y el agarre requieren continuidad en la referencia del objeto durante el proceso de aproximación y manipulación.

En términos funcionales, este criterio favorece la selección del objeto más centrado en el campo visual, lo que tiende a estabilizar el seguimiento y a facilitar el flujo percepción-control para iniciar la rutina de aproximación, agarre y traslado del manipulador.

3.5.3. Selección del punto objetivo para el Pick and Place: centroide vs punto más cercano

Durante las pruebas del sistema se evaluó la forma de seleccionar el punto objetivo dentro del *bounding box* para estimar la profundidad y generar la coordenada espacial que utiliza el robot. Inicialmente se consideró emplear el punto más cercano dentro del *bounding box* (mínima profundidad), dado que, en principio, podría corresponder a la superficie frontal del objeto. Sin embargo, en escenarios con **solapamiento de objetos** (intersección parcial de *bounding boxes*) como se muestra en 3.5.1, este criterio puede inducir errores: cuando dos objetos se proyectan cercanos en la imagen, la región del *bounding box* puede incluir píxeles pertenecientes al objeto más próximo a la cámara, provocando que la profundidad estimada para el objeto objetivo sea tomada

incorrectamente desde el otro objeto. En consecuencia, el sistema puede confundir la altura/profundidad del objeto seleccionado, afectando la referencia espacial y la ejecución del movimiento. Con base en esta observación, se adoptó el **centroide del *bounding box*** como punto de referencia para la estimación de profundidad.

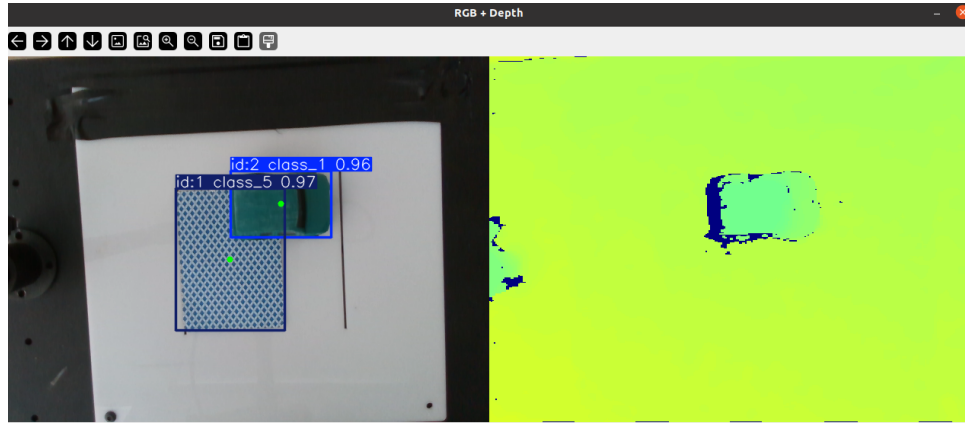


Figura 3.5.1: Ejemplo de solapamiento de *bounding boxes*. Fuente: Elaboración propia.

El uso del centroide del *bounding box* como punto objetivo para consultar profundidad y estimar coordenadas espaciales demostró ser una estrategia más confiable en presencia de solapamiento entre objetos, reduciendo la probabilidad de contaminar la medición con información de un objeto vecino.

3.5.4. Evaluación del desempeño del ciclo Pick and Place

Con el fin de evaluar el desempeño del sistema en condiciones reales de operación, se analizaron un total de 76 ejecuciones del ciclo Pick and Place. Las pruebas se realizaron bajo una configuración experimental controlada, manteniendo fija la posición relativa entre la cámara, el plano de trabajo y el brazo robótico, con el propósito de conservar condiciones repetibles durante la evaluación.

En todas las ejecuciones se siguió el mismo flujo operativo: detección del objeto, estimación de profundidad, selección del objetivo, aproximación del manipulador, agarre, traslado y colocación. Una prueba se consideró exitosa cuando el sistema completó de forma continua todas las etapas del ciclo Pick and Place sin pérdida del objeto ni interrupción de la rutina. En caso contrario, la ejecución se registró como fallida.

Durante los ensayos se registró el resultado de la tarea, indicando si fue completada exitosamente ($success = 1$) o si el objeto no fue recogido o no se completó el ciclo ($success = 0$). Asimismo, se registró la distancia medida entre el objeto y la cámara (z_{cam}), con el fin de analizar posteriormente la relación entre la profundidad estimada y la probabilidad de

La tasa global de éxito obtenida fue del 63.2 %, lo que permite establecer una línea base del comportamiento del sistema bajo las condiciones experimentales definidas. Con el fin de

modelar de manera cuantitativa la relación observada entre la distancia y el desempeño del sistema, se ajustó un modelo de regresión logística que permite estimar la probabilidad de éxito del ciclo Pick and Place en función de la distancia medida por la cámara.

El modelo obtenido se expresa como:

$$P(x) = \frac{1}{1 + e^{-(-48,99 + 2,00x)}}$$

donde x corresponde a la distancia z_cam_cm . Este tipo de modelo es adecuado para describir fenómenos con comportamiento binario (éxito/fallo), permitiendo además identificar regiones de transición entre estados de bajo y alto desempeño.

A partir del modelo, se determina un punto de inflexión ubicado aproximadamente en 24.48 cm, el cual corresponde a una probabilidad de éxito del 50 %. Este valor puede interpretarse como un umbral operativo del sistema, a partir del cual la probabilidad de completar exitosamente la tarea aumenta de manera significativa.

La Figura 3.5.2 muestra la curva logística ajustada junto con los datos experimentales. Se observa que la región comprendida entre 23 cm y 25 cm corresponde a una zona de transición, donde pequeñas variaciones en la distancia generan cambios importantes en el resultado del ciclo.

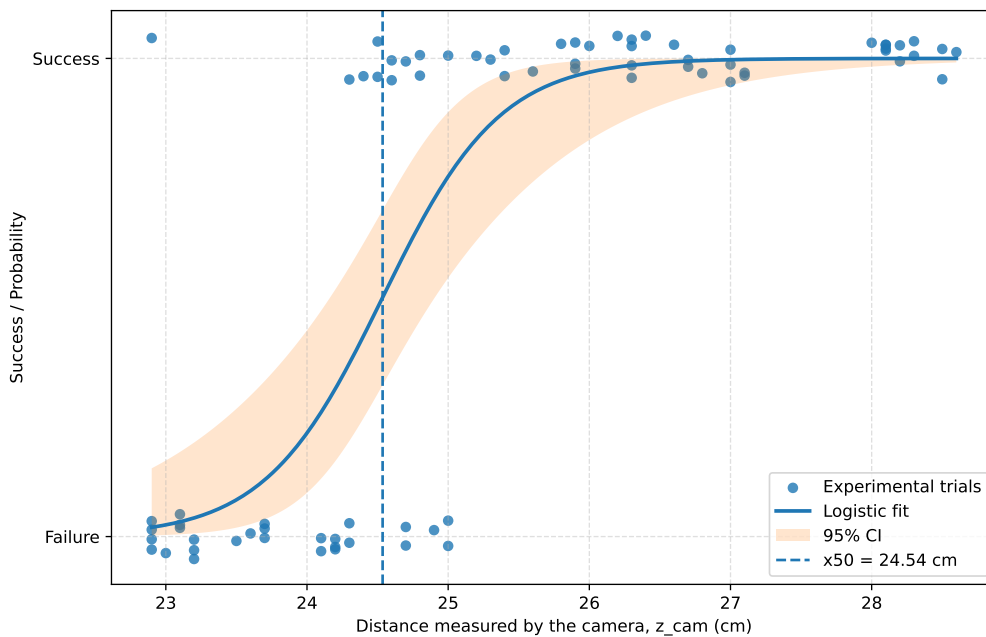


Figura 3.5.2: Relación entre la distancia medida por la cámara y la probabilidad de éxito del ciclo Pick and Place. Fuente: Elaboración propia.

Este comportamiento sugiere que la distancia de observación influye directamente en la calidad de la estimación de profundidad y, por ende, en la precisión de la transformación a coordenadas del robot. Dado que el sistema utiliza la mediana de una ventana local del

mapa de profundidad para estimar la posición del objeto, variaciones en la calidad de la medición en rangos cercanos pueden afectar la estabilidad del ciclo de manipulación.

Por otra parte, la Figura 3.5.3 presenta la tasa de éxito agrupada por identificador de objeto. Se evidencia una variabilidad en el desempeño dependiendo del objeto manipulado, lo que indica que factores como la geometría, el tamaño o la orientación influyen en la efectividad del agarre. Algunos objetos presentan tasas de éxito superiores al 80 %, mientras que otros muestran desempeños considerablemente menores, evidenciando la necesidad de ajustar estrategias de manipulación específicas por tipo de objeto.

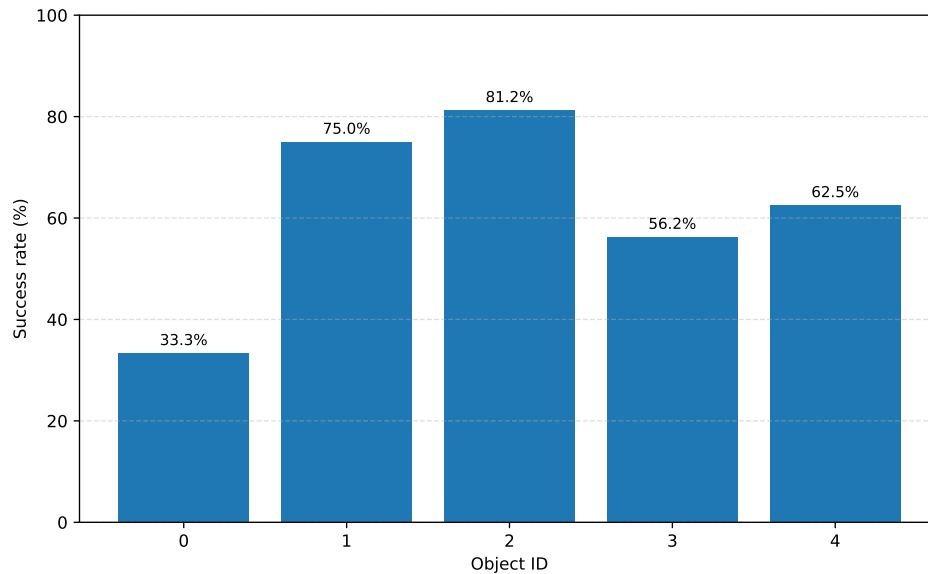


Figura 3.5.3: Tasa de éxito del ciclo Pick and Place agrupada por objeto. Fuente: Elaboración propia.

Adicionalmente, se identificó que el objeto con identificador 0 (correspondiente a un automóvil de juguete) presentó una tasa de éxito significativamente inferior en comparación con los demás objetos evaluados.

Este comportamiento no se atribuye únicamente a la estimación de la distancia, sino principalmente a limitaciones físicas en el mecanismo de agarre. En particular, la superficie del objeto no proporciona condiciones adecuadas para la adhesión de la ventosa del manipulador, lo que dificulta la sujeción efectiva durante el ciclo Pick and Place.

Aunque esta situación también se presentó en menor medida con otros objetos, fue especialmente evidente en este caso, evidenciando que el desempeño del sistema no depende exclusivamente de la percepción, sino también de la interacción física entre el efector final y las características del objeto.

En conjunto, estos resultados permiten concluir que el desempeño del sistema está condicionado tanto por la distancia entre la cámara y el objeto como por las características físicas del mismo. Estos factores deben ser considerados dentro del

proceso de calibración y ajuste del sistema, especialmente en escenarios donde se requiera alta confiabilidad en la ejecución del ciclo *Pick and Place*.

3.5.5. Evaluación del error en la medición de profundidad

Con el objetivo de validar la precisión del sistema de percepción, se realizó una comparación entre la distancia estimada por la cámara RGB-D (z_{cam_cm}) y la distancia real medida manualmente (z_{cm}).

La Figura 3.5.4 presenta la relación entre ambas magnitudes. Se incluye una línea de referencia ideal ($y = x$), que representa el caso de medición perfecta. Se observa una alta correlación entre los valores, lo que indica que el sistema de percepción proporciona estimaciones consistentes de la profundidad.

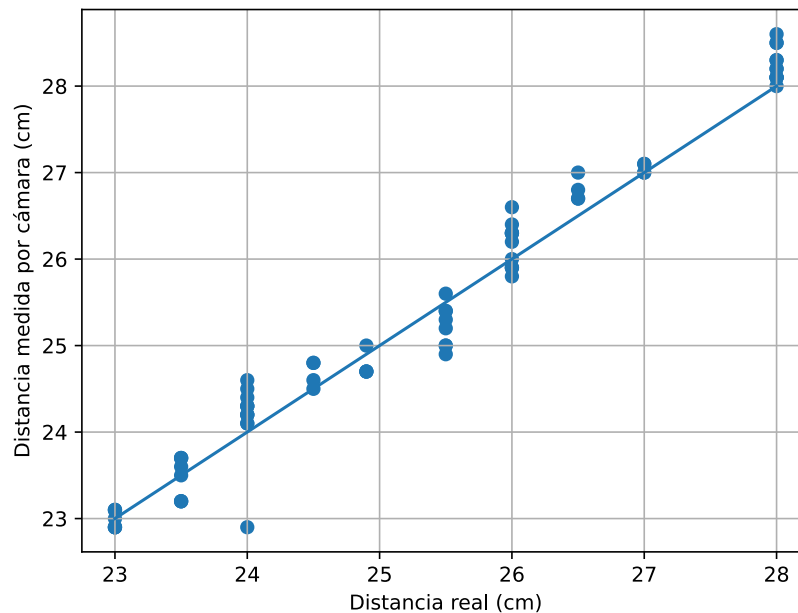


Figura 3.5.4: Relación entre la distancia real y la distancia estimada por la cámara. Fuente: Elaboración propia.

Para cuantificar el error de medición, se calcularon métricas estándar de evaluación. El error absoluto medio (MAE) obtenido fue de 0.23 cm, mientras que el error cuadrático medio (RMSE) fue de 0.30 cm. Adicionalmente, el error porcentual medio (MAPE) fue de 0.92 %.

Estos resultados indican que el sistema presenta un alto nivel de precisión en la estimación de la distancia, con errores promedio inferiores a 3 mm, lo cual es adecuado para aplicaciones de manipulación robótica en el rango de trabajo evaluado.

Sin embargo, al analizar la dispersión de los datos, se observa una ligera variabilidad en torno a la línea ideal, especialmente en distancias cercanas. Este fenómeno permite explicar parcialmente el comportamiento observado en la ejecución del ciclo Pick and Place, donde se identificó una mayor tasa de fallos en distancias cercanas. Pequeñas desviaciones en la medición de profundidad pueden traducirse en errores en la localización del objeto, afectando la precisión del agarre.

En consecuencia, se establece que, aunque el sistema de percepción presenta una precisión adecuada, su desempeño en rangos cercanos constituye un factor crítico que debe ser considerado en la planificación y ejecución del ciclo de manipulación.

Capítulo 4

Conclusiones

4.1 Objetivo específico 1

Se logró la integración funcional entre la cámara Intel RealSense D435i, la plataforma de cómputo embebido y el manipulador WLKATA Mirobot, estableciendo un flujo completo de percepción-control. Más allá de la simple comunicación entre componentes, este resultado demuestra la viabilidad de coordinar sistemas heterogéneos en tiempo real dentro de una arquitectura unificada.

El ajuste del plano de trabajo mediante una referencia geométrica visual permitió alinear el sistema de coordenadas de la cámara con el espacio físico de operación. Este aspecto es crítico, ya que constituye la base para transformar correctamente la información perceptual en acciones del robot. Sin esta correspondencia, cualquier error en la percepción se amplifica en la ejecución física.

En términos prácticos, esta integración valida que el sistema es capaz de cerrar el ciclo percepción-acción, lo cual representa un requisito fundamental en aplicaciones de manipulación robótica autónoma. Asimismo, establece una base sólida sobre la cual se construyen los módulos de detección, estimación espacial y control del manipulador.

4.2 Objetivo específico 2

Se diseñó e implementó un sistema de percepción tridimensional basado en información RGB-D, apoyado en un modelo de detección de objetos YOLOv8. Las métricas obtenidas durante el entrenamiento (Precision, Recall y mAP) evidencian un alto desempeño del modelo en la identificación de las clases definidas, lo que confirma su utilidad como componente central del sistema de visión.

Adicionalmente, se logró la estimación de coordenadas espaciales a partir de las detecciones, utilizando el centroide del bounding box junto con la profundidad alineada al

color. A partir de pruebas experimentales, se determinó que el uso de una estimación robusta basada en la mediana de una ventana local reduce significativamente el impacto del ruido y reduce significativamente la asignación incorrecta de profundidad en escenarios con solapamiento de objetos.

Este resultado no solo valida la capacidad del sistema para detectar objetos, sino también para generar información espacial consistente y utilizable por el manipulador. En consecuencia, se establece una base confiable para la ejecución de tareas de manipulación, donde la estimación de posición del objeto es un factor determinante.

4.3 Objetivo específico 3

Se desarrollaron los componentes necesarios para la planificación y ejecución del ciclo Pick and Place, integrando la información proveniente del módulo de percepción con la lógica de selección del objetivo. La implementación del criterio de priorización basado en la distancia al centro de la imagen ($d2_center$) permitiendo estabilizar la selección del objeto en escenarios con múltiples detecciones.

Este enfoque reduce la variabilidad en la toma de decisiones del sistema y evita cambios abruptos de objetivo entre iteraciones, lo cual es fundamental para garantizar la coherencia en la ejecución del ciclo de manipulación.

Desde una perspectiva de control, este resultado demuestra la importancia de incorporar criterios de selección determinísticos en sistemas basados en percepción, ya que la detección por sí sola no garantiza una ejecución estable. La integración de este criterio contribuye a mejorar la robustez operativa del sistema, facilitando la ejecución continua de las tareas de agarre y posicionamiento.

4.4 Objetivo específico 4

El análisis experimental del sistema permitió identificar la relación directa entre la calidad de la percepción y el desempeño del ciclo Pick and Place. A partir de los ensayos realizados, se obtuvo una tasa global de éxito del 63.2 %, evidenciando que, aunque el sistema es funcional, su desempeño está condicionado por múltiples factores.

En particular, se identificó que la distancia entre la cámara y el objeto tiene un impacto significativo en la tasa de éxito. Mediante el ajuste de un modelo de regresión logística, se determinó la existencia de una zona crítica de operación en rangos cercanos, donde pequeñas variaciones en la distancia generan cambios importantes en la probabilidad de éxito del sistema.

Adicionalmente, el análisis del error en la medición de profundidad mostró que, aunque el sensor presenta una alta precisión global (MAE de 0.23 cm y RMSE de 0.30 cm), incluso errores pequeños pueden afectar la ejecución del agarre en distancias cortas.

Esto demuestra que el problema no se limita a la detección o a la estimación de coordenadas, sino a la sensibilidad del sistema frente a errores acumulados en la cadena percepción-acción.

Por otra parte, se evidenció que las características físicas de los objetos influyen directamente en el desempeño del sistema. En particular, el objeto con identificador 0 (automóvil de juguete) presentó una tasa de éxito significativamente menor debido a limitaciones en la adherencia de la ventosa, lo cual resalta la importancia de considerar no solo la percepción, sino también la interacción física entre el manipulador y el objeto.

En conjunto, estos resultados demuestran que el desempeño del sistema es el resultado de la interacción entre percepción, estimación espacial y manipulación física, siendo necesario abordar estos tres aspectos de manera integrada para mejorar la robustez del sistema.

4.5 Limitaciones

El sistema fue validado principalmente en un entorno controlado; por ello, la variabilidad de condiciones como iluminación, fondos, oclusiones y distribución de objetos puede ser limitada frente a escenarios reales. Asimismo, el alto desempeño del modelo YOLOv8 está asociado a las clases definidas y a la diferenciación visual de los objetos, por lo que la generalización a nuevos objetos o configuraciones requiere la ampliación del dataset y nuevas validaciones. Un aspecto adicional a considerar es la textura de los objetos, la cual inicialmente no se tuvo en cuenta: una superficie irregular puede ocasionar fallos en la succión de la ventosa del brazo robótico.

Finalmente, aunque se cuenta con evidencia de detección, estimación de coordenadas y priorización del objetivo, se requiere instrumentación adicional para registrar de manera sistemática métricas cuantitativas del ciclo completo Pick and Place (tiempo por ciclo, tasa de éxito, errores de colocación y clasificación de fallos).

4.6 Trabajo futuro

Como trabajo futuro se propone fortalecer el sistema en tres frentes principales: generalización del módulo de visión, robustez de la estimación espacial y mejora del control del manipulador. En primer lugar, se recomienda ampliar el dataset incorporando mayor diversidad de escenarios respecto a los utilizados en este trabajo, incluyendo variaciones de iluminación, cambios de fondo, oclusiones parciales, distintos ángulos de cámara, variación en la distancia de trabajo y una mayor densidad de objetos en escena. De igual forma, se sugiere incluir nuevas clases y variaciones de textura y material, ya que superficies irregulares o reflectivas pueden afectar tanto la detección como el desempeño del agarre por succión. Esta ampliación permitiría validar la capacidad de generalización del detector y reducir el riesgo de sobreajuste a condiciones controladas.

Se recomienda profundizar en el control del brazo robótico para mejorar la repetibilidad y reducir fallos durante el ciclo Pick and Place. Esto incluye instrumentar el sistema para registrar métricas del ciclo completo (tiempo por etapa, tasa de éxito, error de colocación y clasificación de fallos), incorporar rutinas de recuperación ante errores (detección fallida, agarre fallido o pérdida de objetivo) y ajustar parámetros dinámicos como velocidades, aceleraciones y alturas seguras según el tipo de objeto. Asimismo, sería pertinente evaluar estrategias de trayectoria más robustas y, de ser posible, integrar retroalimentación adicional para confirmar el agarre antes de ejecutar el traslado.

Finalmente, se sugiere realizar campañas de prueba en escenarios progresivamente más diversos y cercanos a condiciones reales, documentando el desempeño del sistema bajo variabilidad controlada. Estas pruebas permitirán identificar límites operativos, cuantificar mejoras e incrementar la aplicabilidad del prototipo tanto en contextos académicos o industriales.

Referencias Bibliográficas

- [1] Robotics Planning, Control and Innovation By WLKATA Mirobot.
- [2] RealSense, “Depth camera d435i.” <https://www.realsenseai.com/products/depth-camera-d435i/>, 2025. [Online; accessed 25-Feb-2026].
- [3] NVIDIA Corporation, “Jetson orin nano series modules data sheet,” Tech. Rep. DS-11105-001, NVIDIA, 2023. v1.2.
- [4] Deloitte and The Manufacturing Institute, “Us manufacturing could need as many as 3.8 million new employees by 2033.” Press release / web article, Apr. 2024.
- [5] P. Martínez Otero, A. Tellaeche, and M. Hernández Melero, “Performance analysis of the yolo object detection algorithm in embedded systems: Generated code vs. native implementation,” *Computation*, vol. 14, no. 3, p. 67, 2026.
- [6] A. J. Mahbuubi and C.-W. Lin, “Innovative syringe volume detection and calculation using yolov8 and embedded ai on jetson orin nano,” in *International Conference on Biomedical and Health Informatics 2024 (K.-P. Lin, R. Magjarević, and P. de Carvalho, eds.)*, vol. 118 of IFMBE Proceedings, pp. 255–261, Cham: Springer, 2025.
- [7] K. Sroymuk, K. Wongsaroj, A. Anusasananan, and P. Raksincharoensak, “Design and implementation of six degree of freedom robotic arm using image processing for automation in agriculture,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 6, pp. 12088–12096, 2022.
- [8] X. Li, L. Chen, Y. Wu, and Z. Liu, “Vision–language guided 6-dof grasp detection via negative sampling,” *Processes*, vol. 13, no. 5, p. 1598, 2025.
- [9] J. A. Fortoul-Diaz, L. A. Carrillo-Martinez, A. Centeno-Tellez, F. Cortes-Santacruz, I. Olmos-Pineda, and R. R. Flores-Quintero, “A smart factory architecture based on industry 4.0 technologies: Open-source software implementation,” *IEEE Access*, vol. 11, pp. 101727–101750, 2023.
- [10] J. Kovalčík, M. Straka, and J. Trojanowska, “Utilize educational robots to design logistic systems,” *Applied Sciences*, vol. 14, no. 5664, 2024.
- [11] WLKATA Mirobot Robotic Arm Programming and Control.
- [12] M. Pekarcikova, P. Trebuna, M. Kliment, and M. Dic, “Wlkata educational robots used in simulation in the laboratory conditions,” *Acta Simulatio*, vol. 9, no. 3, pp. 33–37, 2023.

- [13] M. Pekarcikova, P. Trebuna, M. Kliment, J. Kronova, and M. Matiscsak, "Educational robotics for industry 4.0 and 5.0 with wlkata mirobot in laboratory process modelling," *Machines*, vol. 13, no. 753, 2025.
- [14] U. Robots, "Sistema pick and place: qué es, funcionamiento y principales aplicaciones," *tech. rep.*, Universal Robots, 22 ene. 2024. [En línea]. Disponible: <https://www.universal-robots.com/mx/blog/sistema-pick-and-place-que-es-funcionamiento-y-principales-aplicaciones/>.
- [15] F. J. Martínez-Peral, H. Migallón, J. Borrell-Méndez, et al., "Manipulation order optimization in industrial pick-and-place operations: application to textile and leather industry," *The International Journal of Advanced Manufacturing Technology*, vol. 133, pp. 987–1010, 2024.
- [16] J. Borrell-Méndez, C. Perez-Vidal, J. V. S. Heras, and J. J. Pérez-Hernández, "Robotic pick-and-place time optimization: application to footwear production," *IEEE Access*, vol. 8, pp. 209428–209440, 2020.
- [17] J.-C. Chen and G.-R. Chen, "Applying an artificial neuromolecular system with autonomous learning capability to learn to control the movement of a six-axis robotic arm," in *IARIA Congress 2025: The 2025 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, pp. 8–9, 2025.
- [18] V. Tadic, A. Odry, I. Kecskes, E. Burkus, Z. Kiraly, and P. Odry, "Application of intel realsense cameras for depth image generation in robotics," *WSEAS Transactions on Computers*, vol. 18, pp. 107–111, 2019.
- [19] R. E. Wagner, *Stereo Active Vision and Peripheral Optical Flow: Computer Vision Applications of the Wide-Field Human Visual Representation. Phd dissertation, Boston University, 2004. Publication No. 3101097.*
- [20] C. G. A. Jain, H. Jain, and Mohana, "Real time object detection and tracking using deep learning and opencv," in *International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 2018.*
- [21] Intel Corporation, *Intel RealSense D400 Series / SR300 Viewer User Guide. Intel Corporation, 2018. Document Number: 337495-002, Revision 002, May 2018. [Online]. Available: https://www.manualpdf.es/intel/realsense-d435i/manual.*
- [22] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, 2021.
- [23] Ultralytics, "Performance metrics deep dive." *Ultralytics Documentation (Web)*. Accedido: 2026-03-18.

Autor 1

Nombre: *Angel Esteban Moreno Daza*

Programa: *Ingeniería Electrónica*

Universidad: *Universidad Santo Tomás – Seccional
Tunja*

Correo institucional: *angel.moreno@usantoto.edu.co*

Teléfono: *3227940703*

Autor 2

Nombre: *Marco Antonio Lasprilla Fernandez*

Programa: *Ingeniería Electrónica*

Universidad: *Universidad Santo Tomás – Seccional
Tunja*

Correo institucional: *marco.lasprilla@usantoto.com*

Teléfono: *3142737639*

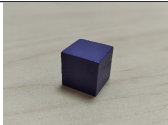
A nexos

Anexo A: Objetos de prueba

Este anexo presenta los objetos físicos utilizados durante las pruebas del sistema Pick and Place, incluyendo sus dimensiones y una referencia visual. En el documento principal se hace referencia a estos objetos como Objeto 1–Objeto 4.

Listado, dimensiones y apoyo visual

Tabla 4.1: Objetos de prueba utilizados en el sistema (con apoyo visual)

Imagen	ID	Objeto	Descripción	Dimensiones
	0	Auto de juguete	Auto de juguete (miniatura), de color verde y negro.	$5.5 \times 3.5 \times 2.7$ cm
	4	Carta (lado A)	Carta “As de corazones”.	8.3×5.8 cm
	5	Carta (lado B)	Carta con patrón de rombos azules y relieve blanco.	8.3×5.8 cm
	2	Cilindro	Cilindro naranja con orificio interno.	Diámetro: 2.2 cm; altura: 1.1 cm; orificio interno: 1.6 cm
	1	Cubo	Cubo morado.	$1.1 \times 1.1 \times 1.1$ cm

Nota sobre Objeto 2

El Objeto 2 presenta dos caras: el lado A corresponde al “As de corazones”. El lado B presenta un patrón de rombos azules con relieve blanco. Debido a que la variación visual del lado B es pequeña respecto al tamaño total de la carta, puede considerarse irrelevante

para el desempeño del sistema; sin embargo, en el entrenamiento del detector es posible tratar ambas caras como clases separadas si se requiere robustez frente a cambios de orientación del objeto.

Anexo B: Código fuente y datos del proyecto

Con el fin de garantizar trazabilidad, replicabilidad y facilitar la revisión técnica del trabajo, el código fuente desarrollado, junto con los archivos de salida (CSV) utilizados como evidencia en el Capítulo de Resultados, se encuentran disponibles en un repositorio externo.

B.1 Repositorio y versión de referencia

El repositorio del proyecto (código, registros y evidencia) se encuentra disponible en GitHub.

Repositorio **(GitHub):**
`<https://github.com/Angel26dz/tesis-pick-place.git>`

B.2 Estructura general del repositorio

La organización del repositorio se mantiene separando código, datos y evidencia, con el fin de facilitar su consulta:

```
tesis-pick-place/
|-- codigo/           (scripts Python)
|-- datos/           (CSV generados y utilizados en resultados)
|-- evidencias/      (imagenes y enlaces a videos)
'-- README.md        (guia de ejecucion)
```

B.3 Descripción de archivos principales

Tabla 4.2: Archivos principales del repositorio

Archivo	Descripción
<code>codigo/rutine.py</code>	Script principal. Integra detección YOLOv8, adquisición RGB-D, estimación de coordenadas y ejecución del flujo percepción-control.
<code>codigo/camera_settings.py</code>	Script de configuración y verificación de la cámara Intel RealSense D435i. Incluye utilidades de visualización para ajuste del plano de trabajo.
<code>codigo/wlkata_move_test.py</code>	Script de prueba de comunicación y movimientos básicos del brazo WLKATA Mirobot 6-axis.
<code>datos/camera_coords.csv</code>	Registro de coordenadas generadas por el módulo de percepción (centroide + profundidad).
<code>datos/rank_log.csv</code>	Registro del criterio de priorización del objetivo (ranking) cuando existen múltiples detecciones en escena.

Anexo C: Evidencia audiovisual del sistema

Los videos de evidencia del funcionamiento del sistema se almacenan externamente debido a limitaciones de tamaño y portabilidad del documento PDF. En este anexo se listan los enlaces y una descripción de la evidencia que aporta cada video.

C.1 Enlace a la carpeta de evidencia audiovisual

Carpeta / lista de reproducción: <https://drive.google.com/drive/folders/1wd64EbY054_ggKmR0264wHSL29mulPna?usp=drive_link>

C.2 Registro de videos

Tabla 4.3: Registro de evidencia audiovisual del sistema

Video	Duración	Evidencia / descripción
V1	01:24	En este video se presenta una prueba experimental en la que todos los objetos se encuentran ubicados a una misma altura dentro del área de trabajo. El objetivo de esta prueba es evaluar el funcionamiento del sistema de detección y la capacidad del manipulador robótico para ejecutar tareas de recogida en condiciones controladas y homogéneas.
V2	01:03	En este video se realiza una prueba bajo condiciones más exigentes, donde los objetos se disponen a diferentes alturas. Esta configuración permite analizar el desempeño del sistema frente a variaciones en la dimensión vertical, evaluando su capacidad de adaptación, precisión en la localización y efectividad en la manipulación de objetos.