



FACULTAD DE INGENIERÍA

Carrera de Ingeniería Mecatrónica

**DISEÑO Y PROTOTIPO DE UNA MÁQUINA
AUTOMÁTICA MEDIANTE VISIÓN
ARTIFICIAL PARA MEJORAR LA CALIDAD
DE SEGREGACIÓN DE RESIDUOS EN LA I.E.
3049 IMPERIO DEL TAHUANTINSUYO,
DISTRITO DE INDEPENDENCIA, LIMA**

Tesis para optar al título profesional de:

Ingeniero Mecatrónico

Autor:

Joseph Milton Calderon Espejo

Asesor:

Mg. Ing. Gabriel Tirado Mendoza

<https://orcid.org/0000-0001-8411-7684>

Lima - Perú

2025


JURADO EVALUADOR

Jurado 1 Presidente(a)	JOHNNY DAVID ARRUSTICO LOYOLA
	Nombre y Apellidos

Jurado 2	OSCAR ARTURO VASQUEZ MENDOZA
	Nombre y Apellidos

Jurado 3	ELISEO JUAN ZARATE PEREZ
	Nombre y Apellidos

Informe de Similitud

 Página 2 of 152 - Descripción general de integridad Identificador de la entrega: trnoid::1:3306274978




8% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe


- Bibliografía
- Texto citado

Fuentes principales

8%		Fuentes de Internet
0%		Publicaciones
0%		Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

-  **Caracteres reemplazados**
42 caracteres sospechosos en N.º de páginas
Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Dedicatoria

*Dedicado a mi querida familia, por
acompañarme en mi camino para forjar el
profesional que soy, por las experiencias que
me permitieron conocer la ciencia e
ingeniería.*

*A Dios y a todas las personas queridas que
me enseñaron algo valioso sobre lo que
podíamos hacer para ver un mundo un poco
aunque mejor cada día usando la ingeniería.*

Agradecimiento

Agradezco al Ing. Jhonatan Calderón Rosales por apoyarme compartiendo su sabiduría en la programación y a mi Asesor, Mag. Ing. Gabriel Tirado Mendoza, quién siempre con mucho entusiasmo supo guiarme en el desarrollo de esta investigación.

Tabla de contenidos

Índice de tablas	7
Índice de Figuras.....	8
Resumen	13
CAPÍTULO I: INTRODUCCIÓN.....	14
1.1. Realidad problemática	14
1.2. Formulación del problema.....	26
1.3. Objetivos.....	27
1.4. Hipótesis	27
CAPÍTULO II: METODOLOGÍA	29
CAPÍTULO III: RESULTADOS	40
CAPÍTULO IV: DISCUSIÓN Y CONCLUSIONES.....	194
REFERENCIAS	205
ANEXOS	209

Índice de tablas

Tabla 1	<i>Requerimientos funcionales y técnicos del prototipo</i>	53
Tabla 2	<i>Residuos segregados por los 15 alumnos del 5to grado “A” de secundaria de la I.E. 3049 Imperio del Tahuantinsuyo</i>	63
Tabla 3	<i>Recolección de datos de segregación mediante el proceso manual por los estudiantes de 5to año “A”</i>	146
Tabla 4	<i>Correspondencia entre clases y nombres de residuos</i>	153
Tabla 5	<i>Recolección de datos de segregación mediante el uso del prototipo por parte de los estudiantes de 5to año “A”</i>	155
Tabla 6	<i>Estadísticos descriptivos de la Calidad de la Segregación de Residuos antes y después de implementar el prototipo</i>	164
Tabla 7	<i>Resultados de la Prueba de Chi-Cuadrado de Independencia</i>	167
Tabla 8	<i>Matriz de trazabilidad de requerimientos para el diseño del Prototipo “EcoSplit”</i>	169
Tabla 9	<i>Evidencia de factibilidad del diseño de estructura del prototipo “EcoSplit”</i> .	172
Tabla 10	<i>Evidencia de factibilidad del diseño del sub-sistema de procesamiento y decisión del prototipo “EcoSplit”</i>	174
Tabla 11	<i>Evidencia de factibilidad del diseño del sub-sistema de actuación del prototipo “EcoSplit”</i>	176
Tabla 12	<i>Evidencia de factibilidad del diseño del sub-sistema de alimentación del prototipo “EcoSplit”</i>	179
Tabla 13	<i>Evidencia de factibilidad del diseño del sub-sistema de percepción del prototipo “EcoSplit”</i>	182
Tabla 14	<i>Evidencia de factibilidad del diseño del sub-sistema de señalización/activación del prototipo “EcoSplit”</i>	183

Índice de Figuras

Figura 1 <i>Sub-sistemas de prototipo</i>	55
Figura 2 <i>Diagrama de flujo del prototipo</i>	56
Figura 3 <i>Diagrama esquemático de todas las conexiones del prototipo</i>	57
Figura 4 <i>Cámara web usb modelo RT-A200 de la marca OEM</i>	58
Figura 5 <i>Soporte para cámara</i>	58
Figura 6 <i>Conexión entre el sub-sistema de percepción y el sub-sistema de procesamiento y decisión</i>	59
Figura 7 <i>Placa de prototipado Arduino Uno</i>	60
Figura 8 <i>Conexión entre Arduino Uno y el código python en la laptop</i>	61
Figura 9 <i>Conexiones del sub-sistema de señalización/activación</i>	62
Figura 10 <i>Código en python para la captura de imágenes</i>	65
Figura 11 <i>Captura de imágenes de residuos</i>	69
Figura 12 <i>Labels de los residuos a segregar</i>	70
Figura 13 <i>Etiquetado de residuos</i>	71
Figura 14 <i>Función de exportar labels en formato YOLO</i>	72
Figura 15 <i>Ejemplo de label de una imagen de un residuo con clase correspondiente al 12</i>	72
Figura 16 <i>Archivo classes.txt</i>	73
Figura 17 <i>Lista de archivos para data.zip</i>	74
Figura 18 <i>Archivo data.zip</i>	74
Figura 19 <i>Función sigmoide</i>	76

Figura 20 <i>Función Unidad lineal rectificada (ReLU)</i>	77
Figura 21 <i>Función Softmax</i>	79
Figura 22 <i>Resultado de la aplicación de un filtro para detectar bordes</i>	83
Figura 23 <i>Estructura de una red neuronal convolucional</i>	86
Figura 24 <i>YOLOv11: Precisión (mAP_{val}^{50-95}) vs. Latencia (ms/img) en COCO</i>	87
Figura 25 <i>Inicio del entorno de ejecución</i>	89
Figura 26 <i>Verificación de la disponibilidad de GPU</i>	89
Figura 27 <i>Cargar en el sistema de archivo de Colab el archivo comprimido data.zip</i>	90
Figura 28 <i>Descompresión del archivo data.zip</i>	90
Figura 29 <i>Creación de la carpeta train y validation</i>	91
Figura 30 <i>Instalación de la biblioteca Ultralytics</i>	91
Figura 31 <i>Creación del archivo data.yaml</i>	92
Figura 32 <i>Instrucción para iniciar el entrenamiento</i>	93
Figura 33 <i>Prueba del modelo</i>	94
Figura 34 <i>Resultados de prueba del modelo</i>	94
Figura 35 <i>Descarga del modelo entrenado</i>	95
Figura 36 <i>Resultados del entrenamiento</i>	95
Figura 37 <i>Programación del sub-sistema de procesamiento y decisión parte 1.</i> 100	
Figura 38 <i>Programación del sub-sistema de procesamiento y decisión parte 2.</i> 101	

Figura 39 <i>Mecanismos Iris diseñado. Fuente: Propia</i>	111
Figura 40 <i>Engranaje recto 24 dientes</i>	115
Figura 41 <i>Engranaje recto 107 dientes</i>	117
Figura 42 <i>Ensamblaje de mecanismo Iris parte 1</i>	118
Figura 43 <i>Posición del pin del segundo engranaje</i>	119
Figura 44 <i>Posición final de la paleta para hallar el radio</i>	120
Figura 45 <i>Radio de la circunferencia de la ranura</i>	121
Figura 46 <i>Configuración cerrada del mecanismo Iris</i>	122
Figura 47 <i>Configuración abierta del mecanismo Iris</i>	122
Figura 48 <i>Mecanismo re-dirigidor sin rampa</i>	123
Figura 49 <i>Mecanismo re-dirigidor con rampa</i>	124
Figura 50 <i>Código de la placa Arduino Uno parte 1</i>	125
Figura 51 <i>Código de la placa Arduino Uno parte 2</i>	126
Figura 52 <i>Conexión para la señal de los servomotores</i>	132
Figura 53 <i>Alimentación de servomotores</i>	136
Figura 54 <i>Contenedores</i>	137
Figura 55 <i>Parte superior</i>	138
Figura 56 <i>Mecanismos integrados</i>	138
Figura 57 <i>Ensamblaje final</i>	139
Figura 58 <i>Armado de estructura</i>	141
Figura 59 <i>Avance de parte inferior de estructura</i>	142

Figura 60 <i>Unión de la estructura inferior</i>	142
Figura 61 <i>Ensamblaje de la parte superior</i>	143
Figura 62 <i>Diseño de logo EcoSplit</i>	144
Figura 63 <i>Logo EcoSplit</i>	144
Figura 64 <i>Ensamblaje listo</i>	145
Figura 65 <i>Charla y pruebas sobre el uso correcto del prototipo</i>	154
Figura 66 <i>Registro automático del experimento parte 1</i>	161
Figura 67 <i>Registro automático del experimento parte 2</i>	162
Figura 68 <i>Gráfico de segregación de residuos mensual 2025-06-04 a 2025-07-04</i>	163
Figura 69 <i>Gráfico de segregación de residuos semanal 2025-06-27 a 2025-07-04</i>	163
Figura 70 <i>Resultado del Porcentaje de Segregación Efectiva antes y después del uso del prototipo</i>	165
Figura 71 <i>Categoría de residuos correspondientes a cada compartimento</i>	185
Figura 72 <i>Recepción de gaseosa incakola</i>	186
Figura 73 <i>Recepción de lata vacía</i>	186
Figura 74 <i>Identificación del residuo botella incakola y su categoría</i>	187
Figura 75 <i>Identificación del residuo lata y su categoría</i>	187
Figura 76 <i>Mecanismo re-dirigidor orientado al compartimento de residuos aprovechables para depositar la botella incakola</i>	188
Figura 77 <i>Mecanismo re-dirigidor orientado al compartimento de residuos</i>	

<i>aprovechables para depositar la lata vacía</i>	188
<i>Figura 78 Depósito de la botella incakola.....</i>	189
<i>Figura 79 Depósito de la lata vacía.....</i>	189
<i>Figura 80 Registro de los residuos automáticamente.....</i>	190

Resumen

La gestión inadecuada de residuos sólidos representa un desafío ambiental crítico, exacerbado por procesos de segregación manual ineficientes y propensos a errores. En el contexto de la I.E. 3049 Imperio del Tahuantinsuyo, se identificó que el método manual de clasificación de residuos presenta una efectividad de solo un 40.40%, resultando en una alta contaminación cruzada de materiales reciclables. Para abordar esta problemática, el presente estudio se centró en el diseño, prototipado y validación de un prototipo de máquina automática, "EcoSplit", basada en visión artificial y componentes de bajo costo, con el objetivo de mejorar la calidad de la segregación en la fuente. Se siguió un diseño de investigación pre-experimental, comparando el rendimiento del proceso manual (pre-test) con el del prototipo implementado (post-test). Se construyó un sistema mecatrónico integral que utiliza una arquitectura de hardware accesible, compuesta por una laptop, una cámara web y una placa Arduino Uno que controla mecanismos de actuación específicos (Mecanismo Iris y re-dirigidor) accionados por servomotores MG90S. El núcleo del sistema es un modelo de inteligencia artificial de detección de objetos, YOLOv11, entrenado para identificar y clasificar distintas categorías de 22 residuos basándose en sus características visuales complejas, como forma y textura. Los resultados de la validación experimental demostraron que el prototipo "EcoSplit" alcanzó un porcentaje de segregación efectiva del 95.62%, lo que representa una mejora neta de 55.22 puntos porcentuales sobre el método manual. Esta diferencia fue validada como estadísticamente significativa ($p < 0.001$). El estudio concluye que la integración de un modelo de IA avanzado con un diseño mecatrónico optimizado y de bajo costo es una solución altamente eficaz y superior a la automatización tradicional basada en sensores dedicados. El prototipo no solo valida una solución tecnológica viable y accesible, sino que también funciona como una herramienta de alto impacto para fomentar la educación ambiental y una cultura de reciclaje en el entorno educativo.

Palabras clave: segregación de residuos, visión artificial, inteligencia artificial, YOLO, mecatrónica, prototipo de bajo costo, reciclaje.

CAPÍTULO I: INTRODUCCIÓN

1.1. Realidad problemática

En la historia se ha visto la necesidad del hombre por recurrir a diversas maneras para lograr disminuir los tiempos y mejorar la efectividad de los procesos, sin embargo desde el siglo XX se empezó a aplicar la automatización para procesos destacando el uso de motores, relés, etc. Desde aquel entonces la tecnología de automatización ha ido tomando mayor relevancia hasta hoy en día, ya no solo teniendo aplicaciones en la industria sino aplicaciones domésticas, exploración, sociedad, etc (Agudelo et al., n.d.).

Cardona & Gratacós (2023) afirma que la automatización se usa para procesos regidos por reglas específicas o procesos mecánicos y repetitivos pero que gracias a nuevos avances tecnológicos se empiezan a desarrollar procesos automatizados que incorporan Inteligencia Artificial, esto permitiendo replicar decisiones que tomaría un humano, automatizando procesos más complejos.

Lo mencionado por estos autores se llega a complementar con lo expresado por Cáceres Tello & Cáceres Tello (2006) sobre la visión artificial y su relevancia hasta el día de hoy, siendo la visión artificial cada vez parte de más numerosas aplicaciones entre las que se destacan manipulación de materiales, control de calidad, monitorización de procesos, etc.

Dentro de las instituciones educativas de acuerdo con el Ministerio del Ambiente debe haber actividades relacionadas con la manipulación de materiales, como por ejemplo, en una institución de educación básica regular se debe llevar a cabo la segregación de residuos por parte de los miembros de la comunidad estudiantil mediante

la guía de clasificación de aquellos en depósitos de colores determinados para instituciones de este nivel de educación (Ministerio Del Ambiente, 2020).

A lo anterior se complementa con lo expresado por Sullca Estrada (2023) que explica en su investigación que la importancia de tener un método eficiente de segregación de residuos se debe a que esta forma se reduce los diferentes efectos perjudiciales en la salud de las personas y el medio de la contaminación ambiental.

Es usual pensar que el método de segregación de residuos proporcionado por la guía del Ministerio Del Ambiente actual mencionada es un proceso sencillo y que la calidad de la segregación de residuos sería alta. Sin embargo los resultados reales muestran que este método aplicado a nuestra realidad en el contexto educativo como el de la región Lima es ineficiente y genera una baja calidad de la segregación de residuos (Instituciones et al., 2023).

Los métodos de segregación de residuos similares aplicados a nivel nacional se desempeñan de igual manera mostrándose cifras preocupantes como una producción diaria de 21 mil toneladas de basura en el Perú, los cuales no son segregados correctamente debido a que el método actual también no es eficiente para la mayor parte de la sociedad peruana actual generando efectos negativos para todos los seres vivos (EL PERUANO, como se cita en Sullca Estrada, 2023).

En la Institución Educativa de educación básica regular 3049 Imperio del Tahuantinsuyo el problema que se presenta es que el método de segregación de residuos aplicado ha generado una baja calidad en la segregación de residuos siendo ineficiente. Este problema se debe a que el actual método aplicado de segregación de residuos por esta institución y sugerido por el Ministerio del Ambiente para este tipo de instituciones no es llevado a cabo correctamente por la comunidad estudiantil por razones como la

falta de una educación medio ambiental y la dificultad para poder diferenciar entre una gran cantidad de residuos causando que muchos de estos que necesariamente no deban estar mezclados, lo estén. Además este método de segregación aplicado causa que la tarea de los recicladores autorizados de recolectar lo segregado sea más dificultosa y peligroso, pues al no estarlo tratan ellos de clasificarlos manualmente, generando un mayor uso de su tiempo y energía, más un claro riesgo de su salud al no saber a que se exponen, claramente estos esfuerzos también son infructíferos, pues según los resultados de los indicadores RRSS año 2019 en Independencia incluyendo también el rendimiento bajo en la segregación de residuos, solo el 1.29% de residuos aprovechables anualmente se lograba valorizar o reutilizar, y en el Perú solo el 1.4% se lograba valorizar, esto claramente contribuye negativamente a la eficacia de un sistema integrado de gestión de residuos en nuestro país (Minam, 2020).

En dicho sentido, si es que no se logra resolver este problema, las consecuencias serían menor cantidad de residuos aprovechables y orgánicos no valorizados, que a su vez al no ser valorizados acaban en lugares inadecuados generando una mayor propagación de enfermedades en los alrededores de la comunidad educativa, también he de mencionar que esto genera mayor cantidad de residuos destinados a los rellenos sanitarios contribuyendo a la generación de efectos negativos en nuestra atmósfera, aumento de la temperatura del planeta y por ende condiciones adecuadas para vida microorgánica peligrosa, sus vectores y hospederos de estos últimos, lo cual también incrementa la transmisión de enfermedades y ocasiona graves consecuencias sobre la salud de seres vivos a nivel no solo regional o nacional sino global, mayor tiempo y energía de empleo por parte de los recicladores autorizados y por ende mayor cantidad de incidentes y accidentes o perjuicios a la salud de los recicladores autorizados.

En el contexto internacional, en Ecuador, Alejandro et al. (2024), en su tesis tuvo como objetivo diseñar e implementar un sistema de clasificación automática de botellas a través del reconocimiento de imágenes mediante visión artificial para contribuir al tratamiento adecuado de residuos sólidos reciclables. La metodología aplicada fue de tipo experimental, estructurada en tres fases: investigación documental y análisis de algoritmos, entrenamiento del modelo de detección basado en YOLOv5s mediante Google Colab, e implementación de un prototipo físico que integra una banda transportadora con un sistema de clasificación automatizado. La visión artificial se ejecutó con lenguaje Python desde una computadora, mientras que la activación de los mecanismos físicos (motor de la banda, pistón de clasificación) fue controlada por un microcontrolador ESP32 y el entorno de desarrollo de Arduino, comunicándose vía serial con el sistema principal. Entre los principales resultados, se logró una detección y clasificación eficaz de botellas bajo condiciones controladas, con un alto nivel de acierto. Sin embargo, también se identificó que el rendimiento disminuye conforme aumenta la cantidad de botellas en simultáneo a identificar, debido a la limitación del sistema en entornos de alta densidad. Esta investigación demuestra que es viable implementar sistemas de clasificación inteligente de residuos mediante visión artificial y microcontroladores de bajo costo. Además, valida el uso de una arquitectura distribuida donde la detección se realiza en un ordenador con Python y el control de actuadores se gestiona desde un microcontrolador, lo cual permite modularidad, bajo costo y adaptabilidad para prototipos de futuras investigaciones relacionadas.

Del mismo modo, Romero (2023) en su investigación tuvo como objetivo desarrollar y evaluar un sistema de detección y clasificación automática de residuos utilizando técnicas de visión por computador y aprendizaje profundo, con el fin de

contribuir a una gestión de residuos más eficiente y sostenible. Para ello, se creó desde cero una base de datos pública compuesta por más de 1300 imágenes etiquetadas que representan seis clases de residuos: botellas de plástico, latas, envases de yogur, papel, cartón y vidrio. Este estudio fue experimental consistiendo en el uso de redes neuronales convolucionales (CNN), implementadas bajo la arquitectura YOLOv8, entrenadas en Google Colab. Se aplicaron técnicas de preprocesamiento y aumento de datos usando herramientas como Roboflow, y se construyó un modelo de clasificación en tiempo real. Los resultados mostraron una alta precisión y recall en la detección de objetos, con un desempeño robusto incluso ante objetos superpuestos. Finalmente, se implementó el modelo en una aplicación práctica con una maqueta física controlada por Arduino que simula la clasificación automática de residuos. Esta investigación demuestra que Google Colab, es una valiosa plataforma colaborativa basada en la nube que permite ejecutar procesos complejos de entrenamiento utilizando GPU, sin necesidad de recursos locales avanzados, así permitiendo entrenar mediante el uso de cuadernos de Yolo modelos robustos y eficientes desde entornos accesibles y replicables.

Llerena Buenaño & Salazar Villamar (2022), en su tesis titulada “Automatización de un sistema identificador y posicionador de objetos a través de un brazo robótico mediante visión artificial con lenguaje Python”, se planteó como objetivo automatizar la clasificación de objetos por color en una línea de producción, mediante un brazo robótico controlado con visión artificial, con el fin de mejorar las prácticas de laboratorio de control en modalidad remota y promover el aprendizaje interactivo en contexto de restricciones sanitarias. La metodología utilizada fue de tipo experimental, enfocada en el diseño y desarrollo de un sistema funcional basado en Raspberry Pi, cámara Logitech, y un brazo robótico Lewansoul xArm1S, programado con Python y OpenCV. Los

resultados evidenciaron que el sistema fue capaz de detectar objetos por color en tiempo real, identificar su ubicación, y posicionarlos automáticamente en lugares específicos, logrando una clasificación precisa y reduciendo el margen de error humano. Se concluyó que el uso de visión artificial resultó altamente favorable, ya que permitió identificar y clasificar objetos por color de forma automática y útil para detección en tiempo real, mejorando la precisión y eficiencia del sistema. Asimismo, la integración de servomotores y el módulo LM2596, un convertidor de voltaje DC-DC Step-Down para la alimentación eléctrica regulada de este, facilitó movimientos controlados y exactos del brazo robótico, fundamentales para posicionar los objetos correctamente.

En un contexto similar, Arishi (2025), tuvo como objetivo desarrollar un sistema de clasificación y detección de residuos domésticos en tiempo real, utilizando el modelo YOLOv8, con el fin de mejorar la eficiencia del reciclaje sostenible. El estudio fue de tipo experimental, y la metodología consistió en el diseño de una red de aprendizaje profundo entrenada sobre un dataset personalizado de 3775 imágenes distribuidas en 17 clases de residuos reciclables, no reciclables y peligrosos. Se aplicaron técnicas de aumento de datos y se incorporaron mecanismos de atención (SE y CBAM) para mejorar la precisión del modelo. Como resultado, el modelo mejorado YOLOv8-CBAM alcanzó una precisión media (mAP) del 89.5%, superando ampliamente a modelos tradicionales como Mask R-CNN y Faster R-CNN. Este alto rendimiento demostró la eficacia del sistema para clasificar residuos de manera rápida y precisa, incluso en condiciones reales con ruido de fondo. El uso del algoritmo YOLOv8 se consideró favorable, ya que mejoró significativamente la velocidad y precisión del proceso de detección, siendo ideal para aplicaciones en sistemas de reciclaje automatizados, como contenedores inteligentes o recolectores robóticos. El estudio concluye recomendando el uso de visión artificial

basada en YOLOv8 como una solución efectiva para optimizar la gestión de residuos, reducir la intervención humana y facilitar el reciclaje en entornos domésticos e industriales, siendo así una opción a considerar entre los modelos de detección de objetos para trabajos futuros de segregación de residuos.

También, Aydin Adnan & Rehman (2022), en su tesis tuvo como objetivo analizar el funcionamiento de las máquinas de clasificación óptica en industrias como la alimentaria, minera y de reciclaje, y desarrollar un modelo conceptual optimizado para el sector alimentario. La metodología empleada combinó revisión bibliográfica, visitas a plantas de procesamiento de alimentos y el diseño de un prototipo mediante el software SolidWorks. El modelo propuesto se basó en un sistema de alimentación por tolva de caída libre y un sistema de expulsión por válvulas neumáticas. Los resultados demostraron que esta configuración ofrece ventajas en mantenimiento, precisión multivista, costo y facilidad de montaje frente a otros sistemas como el de cinta transportadora. Se concluye que la aplicación de clasificadoras ópticas mejora la eficiencia, reduce el contacto humano con materiales contaminados y promueve una mayor seguridad y sostenibilidad en los procesos industriales. Este estudio es de gran importancia por establecer cuales son las etapas generales recomendadas de una máquina de clasificación óptica que podría usar en la parte de identificación cámaras de color, y en conjunto con los aportes de los otros autores se podría integrar a eso un modelo de red neuronal convolucional para la detección de objetos.

Del mismo modo, Liseth (2025), tuvo como objetivo principal diseñar e implementar un sistema automatizado capaz de identificar y clasificar residuos como plástico, papel, vidrio y materia orgánica. El estudio fue de tipo experimental y para ello se empleó una metodología mixta: revisión bibliográfica sobre visión artificial y

aprendizaje profundo; desarrollo de un modelo de red neuronal convolucional en TensorFlow (Python) entrenado con imágenes capturadas por un módulo ESP32-CAM; y montaje de un prototipo que integra servomotores controlados por Arduino para el depósito automatizado de desechos. La comunicación serial entre Arduino y el script de Python demostró ser un método válido y escalable para futuros prototipos de bajo costo, ya que la adquisición de imágenes del ESP32-CAM es usada por el modelo de detección en una laptop, facilitando así actualizaciones o migraciones a otros entornos de cómputo válidos. Las pruebas en diversas condiciones de iluminación arrojaron una precisión superior al 90% en la clasificación, confirmando la eficacia del enfoque. Además, para optimizar el rendimiento en escenarios ya no de imágenes estáticas sino detección en tiempo real, arquitecturas de detección como YOLOv8 podrían implementarse con la cámara, ya que ofrecen inferencia más rápida y la capacidad de procesar múltiples clases de objetos en una sola pasada, lo que potenciaría aún más la velocidad y robustez del sistema.

En el contexto nacional, Colmenares Velasquez & Chavez Vasquez (2024), tuvo como objetivo del estudio fue diseñar y simular un sistema robótico capaz de clasificar residuos sólidos reciclables utilizando inteligencia artificial para mejorar el proceso de reciclaje en la ciudad de Cajamarca. Su estudio fue de tipo no experimental y la metodología se basó en la simulación de un sistema en el software CoppeliaSim, donde se integró un robot delta ABB IRB360, sensores de proximidad, fajas transportadoras, una cámara lineal y un modelo de red neuronal ligera FOCUS-RCNet entrenado con más de 1,600 imágenes. La conexión con el modelo de IA se realizó mediante Python a través de la API de CoppeliaSim. Los resultados evidenciaron que el sistema logró una precisión de clasificación del 92% y una capacidad operativa de hasta 7 objetos por minuto. Se

concluye que el uso combinado de inteligencia artificial y robótica delta permite optimizar el proceso de clasificación de residuos, reduciendo errores y tiempos de trabajo, además de ofrecer una alternativa escalable y eficiente para mejorar la sostenibilidad en la gestión de residuos urbanos.

También, Belji et al. (2021), tuvo como objetivo diseñar una máquina que permita clasificar automáticamente botellas PET según su color (transparente, verde, azul y rojo-joyudo), con el fin de optimizar el proceso de reciclaje y reducir la carga de trabajo manual. El estudio fue de tipo no experimental y la metodología se basó en un enfoque tecnológico, junto con simulaciones en Inventor y programación en el software CoDeSys con lenguaje Ladder. El sistema de control fue desarrollado con un PLC Siemens S7-1200, y para la detección de colores se emplearon sensores ópticos colorSENSOR OT-3-MA conectados mediante interfaz RS232, que activaban cilindros neumáticos para la separación de las botellas. Los resultados demostraron que el sistema automatizado permitía una clasificación precisa y eficiente, reduciendo tiempos y mejorando el orden del reciclaje. Este trabajo evidencia la viabilidad de usar controladores industriales robustos en proyectos de automatización industriales, sin embargo, para fines de prototipos no industriales, el uso de microcontroladores como Arduino también resulta recomendable, como se evalúa en el presente estudio.

En un contexto similar, Meza et al. (2025), tuvo como objetivo desarrollar un sistema automatizado basado en inteligencia artificial para la clasificación de residuos orgánicos e inorgánicos, con el propósito de mejorar su gestión y promover la sostenibilidad ambiental. La metodología fue de enfoque cuantitativo y diseño preexperimental, utilizando un conjunto de datos de 1,298 imágenes provenientes de repositorios públicos. Se implementó un sistema de visión artificial con el modelo

YOLOv8 para detección en tiempo real, una interfaz gráfica en Python (Tkinter) y procesamiento de imágenes con OpenCV. Los resultados mostraron un aumento del 11.52% en precisión, mejora del 23.61% en velocidad de detección y una reducción del 24.13% en la tasa de error, validando la efectividad del sistema. Aunque se empleó hardware de cómputo estándar y software especializado, el uso de microcontroladores como Arduino también es recomendable para prototipos o entornos educativos, ya que permite integrar sensores de bajo costo y comunicación serial con modelos de IA ejecutados en una PC, facilitando el desarrollo de soluciones accesibles y escalables.

Es de suma importancia describir las variables que fueron consideradas en mi estudio, siendo estas: Máquina automática para segregación de residuos mediante visión artificial y segregación de residuos. La variable independiente de este estudio es una máquina automática mediante visión artificial para la segregación de residuos el cual es un sistema de segregación basado en un sensor óptico el cual puede constar de manera general de cuatro etapas, alimentación, presentación, identificación y separación (Aydin Adnan & Rehman, 2022). La variable dependiente de este estudio es la calidad de la segregación de residuos la cual se define como el grado de efectividad con que los residuos son segregados en el punto de generación de estos en sus respectivos contenedores de acuerdo a su tipología siguiendo criterios técnicos y ambientales (Gutiérrez Rodríguez, 2018).

Las dimensiones de la variable independiente en mi estudio son los subsistemas que conforman el prototipo de mi máquina. El primer subsistema y dimensión es el de percepción, en este subsistema se encuentra el componente hardware responsable de capturar la información visual para entregarla al subsistema de procesamiento y decisión, el componente principal aquí sería la cámara web usb modelo RT-A200 de la marca

OEM, los indicadores de esta dimensión sería la resolución de la cámara que sería 1920 x 1080 píxeles y la tasa de fotogramas por segundo que sería 30 fps. La segunda dimensión y subsistema es el de procesamiento y decisión, aquí es donde se ejecuta el modelo re-entrenado de inteligencia artificial YOLO V11 para analizar los datos visuales y determinar un resultado de detección a partir de lo que recibe del subsistema de percepción. El componente hardware de este subsistema para el prototipo de la máquina fueron, una laptop ASUS TUF Dash F15 modelo FX516P que cuenta con un procesador Intel Core i7-11375H de cuatro núcleos y una frecuencia máxima de 5.0 GHz. Está equipada con una tarjeta gráfica NVIDIA GeForce RTX 3050 para laptops, con 4 GB de memoria GDDR6. En cuanto a memoria RAM, dispone de 16 GB DDR4 a 3200 MHz. El componente software de este subsistema fue un modelo re-entrenado de red neuronal enfocado en la detección de objetos, específicamente YOLOV11 (You only look once). Los indicadores del subsistema de procesamiento y decisión son la precisión de clasificación (metrics/precision(B) y val/cls_loss), la tasa de detección (metrics/recall(B)), la precisión de localización (val/box_loss y metrics/mAP50-95(B)), y el rendimiento general del modelo (metrics/mAP50(B)). El tercer subsistema y dimensión de la variable independiente es el subsistema de actuación, es el conjunto de componentes hardware responsables de ejecutar la acción física de segregación. Los principales componentes son la placa Arduino uno que se encarga de recibir mediante comunicación serial la decisión tomada por el subsistema de procesamiento y decisión en la laptop, en la ejecución de un código python, para que Arduino uno inmediatamente a partir de su programación envíe la señal de posición angular a los segundos componentes hardware importantes, los servomotores MG90S, los cuales se encuentran interrelacionados con el tercer componente, los cuales son los mecanismos de apertura, cierre y redirección, con el cambio de posición angulos de los servomotores, estos

mecanismos de apertura, cierre y redirección toman posiciones particulares y determinadas que permiten que los residuos se depositen en los espacios determinados y adecuados. El indicador de este tercer subsistema y dimensión es el tiempo de respuesta mecánica a adoptar las posiciones específicas de los diferentes mecanismos cuando la placa Arduino uno envía las señales a los servomotores MG90S. El cuarto subsistema y dimensión es el subsistema de señalización/activación de la máquina, el primer componente hardware en este subsistema, es el LED (Diodo Emisor de Luz) y el segundo componente hardware es el interruptor de presión, el primer indicador de este subsistema son los colores de los dos LED'S los cuales son rojo y verde, y el segundo indicador sería el voltaje en ambos LED'S el cual debe ser 1.6 V y 2.4 V respectivamente. Estos LED'S cumplen una función de señalar la actividad del sistema, el LED rojo indica que el máquina no esta en actividad y se encuentra esperando por la interacción con un usuario, mientras que el LED verde indica que la máquina se encuentra en actividad, realizando procesos en los subsistemas, el tercer indicador es el estado normal del interruptor el cual es normalmente abierto, este interruptor de presión es usado para iniciar el proceso de segregación automático de un residuo. El quinto subsistema y dimensión es el de alimentación, este subsistema provee la energía eléctrica necesaria para el funcionamiento de los subsistemas, los principales componentes hardware en este subsistema, son la fuente de alimentación externa y el Convertidor DC-DC regulable para los servomotores, mientras que la alimentación de la cámara y placa Arduino uno, que a su vez alimenta los LED'S, es realizada por la alimentación de la propia laptop, los indicadores de este subsistema es el voltaje y corriente soportada por la fuente de alimentación externa que es de 24V-5A, y del puerto USB de la laptop que es 5V a 1.5 A, mientras que otro indicador sería también la capacidad de regulación del convertidor DC-DC XL4015 que soportaría un voltaje de entreda de entre 4 -38 V y un voltaje de

salida de entre 1.25 -32 V a un máximo de 5A. La dimensión de mi variable dependiente es la efectividad de la segregación la cual se define como el grado de acierto con el que los residuos sólidos son identificados y depositados en el contenedor que corresponde según su tipo (orgánico, inorgánico, aprovechable o peligroso). Esta dimensión evalúa si la máquina o el procedimiento utilizado logra una segregación adecuada y sin errores. El indicador de esta dimensión se llama “Porcentaje de Segregación Efectiva” la cual es una medida cuantitativa que representa la proporción de residuos correctamente clasificados y segregados frente al total de residuos procesados, indicando la eficacia del sistema o procedimiento de segregación.

1.2. Formulación del problema

1.2.1 Problema general

¿Cómo diseñar y prototipar una máquina automática mediante visión artificial que permita mejorar la calidad de segregación de residuos en la I.E. 3049 Imperio del Tahuantinsuyo, distrito de Independencia, Lima?

1.2.1 Problemas específicos

- ¿Cuáles son los requerimientos funcionales y técnicos que se deben analizar y definir para el diseño de un prototipo de máquina automática mediante visión artificial para la segregación de residuos?
- ¿De que manera se deben diseñar los subsistemas del prototipo para que funcione de manera integrada y efectiva?
- ¿Cómo se debe construir físicamente el prototipo, integrando sus sub-sistemas, para asegurar su operatividad?
- ¿Cuál es el nivel de efectividad del prototipo en la mejora de la calidad de segregación de residuos, medido por el indicador “porcentaje de segregación

efectiva” en comparación con el proceso manual?

1.3. Objetivos

1.3.1 Objetivo general

La implementación de un prototipo de máquina automática mediante visión artificial mejorará significativamente la calidad de segregación de residuos en la I.E. 3049 Imperio del Tahuantinsuyo, distrito de Independencia, Lima.

1.3.2 Objetivos específicos

- Analizar y definir los requerimientos funcionales y técnicos necesarios para el diseño de un prototipo de una máquina automática mediante visión artificial para mejorar la calidad de segregación de residuos.
- Diseñar los subsistemas del prototipo de la máquina automática mediante visión artificial para mejorar la calidad de segregación de residuos.
- Prototipar físicamente la máquina automática mediante visión artificial para mejorar la calidad de segregación de residuos.
- Evaluar la efectividad del prototipo en la mejora de la calidad de segregación de residuos mediante el indicador porcentaje de segregación efectiva comparándolo con el del proceso manual.

1.4. Hipótesis

1.4.1 Hipótesis general

La implementación de un prototipo de máquina automática mediante visión artificial mejorará significativamente la calidad de segregación de residuos en la I.E. 3049 Imperio del Tahuantinsuyo, en el distrito de Independencia, Lima.

1.4.2 Hipótesis específicas

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

- Si se analizan y definen correctamente los requerimientos funcionales y técnicos, entonces se podrá establecer una base sólida y adecuada para el diseño del prototipo de la máquina automática.
- Si se diseñan adecuadamente los subsistemas del prototipo, entonces se logrará la construcción de un prototipo funcional e integrado.
- Si se prototipa en físico adecuadamente la máquina automática, integrando sus componentes, entonces será capaz de ejecutar la segregación de residuos.
- Si se utiliza el prototipo en la I.E. 3049 Imperio del Tahuantinsuyo, entonces el indicador porcentaje de segregación efectiva será estadísticamente superior al obtenido mediante el proceso manual.

CAPÍTULO II: METODOLOGÍA

2.1 Enfoque de investigación

Esta investigación tiene un enfoque cuantitativo, analizando la realidad objetivamente para obtener determinados resultados. El enfoque cuantitativo representa un conjunto de procesos secuenciales y probatorios, En este tipo de enfoque no se puede “brincar” etapas o pasos, se realizan mediciones de las variables en un contexto específico mediante herramientas estadísticas, y a partir de ellas se consiguen las conclusiones (Hernández et al., 2014).

2.1.1 Nivel de investigación

El nivel de investigación de este trabajo es aplicada usando las ciencias aplicadas aprendidas para solucionar la problemática y obtener resultados, a si mismo la dificultad más considerable será la elaboración del diseño y prototipado de la máquina para mejorar la calidad de segregación de residuos en la mencionada institución educativa.

2.1.2 Diseño de la investigación

Esta investigación es de diseño pre-experimental debido a que en este trabajo se busca modificar la variable independiente de forma intencional para poder observar las consecuencias que esto tiene sobre la variable dependiente (Hernández et al., 2014).

2.1.3 Alcance de la investigación

El alcance de esta investigación es descriptivo-explicativo debido a que se reconoce que el proyecto primero describe una solución tecnológica y su proceso

de creación, para luego explicar y demostrar su efectividad y el impacto que tiene sobre el problema planteado.

2.2 Población

La población está conformada por los residuos sólidos generados en la I.E. 3049 Imperio del Tahuantinsuyo, en el distrito de Independencia, Lima, e indirectamente por la comunidad educativa involucrada en el proceso de segregación de dichos residuos.

2.2.1 Muestra

La muestra está compuesta por los residuos segregados en un único salón conformado por 15 estudiantes del 5to grado de secundaria de la sección “A” recolectados durante un periodo de 40 días. No se seleccionó la muestra al azar, sino que utilizó un muestreo no probabilístico por conveniencia. Los criterios y las características para esta selección son los siguientes:

- **Criterios de Selección**

Accesibilidad y Viabilidad: Se consideró factible y manejable trabajar con un grupo pequeño y controlado que con toda la institución educativa.

Grupo Controlado: El enfocarse en un solo salón, contribuye a que se pueda dar instrucciones claras y uniformes, supervisar directamente el proceso y garantizar que las condiciones del experimento, tanto manual como con el prototipo, sean consistentes.

Representatividad Funcional: Aunque no es estadísticamente representativo de toda la escuela, el tipo de residuos generados por un salón de secundaria como: envolturas, botella de gaseosa, papel, restos de

comida orgánica es lo suficientemente típico como para probar la funcionalidad del prototipo en un entorno real y relevante.

Consistencia: Al utilizar el mismo grupo de estudiantes para evaluar tanto el proceso manual como el del prototipo, se asegura que la fuente generadora de residuos sea la misma, lo que permite una comparación más justa y directa entre ambos métodos.

- **Características de los Elementos de la Muestra**

Los elementos de la muestra no fueron todos los residuos que generaron, sino aquellos que cumplían con las siguientes características:

Frecuencia de Generación: Se seleccionaron los tipos de residuos más comúnmente generados por los estudiantes de esa sección. Esto asegura que el experimento se centre en los desechos más relevantes y abundantes del contexto.

Clasificación Predefinida: Los residuos seleccionados debían poder ser clasificados en las cuatro categorías de interés para el estudio: Aprovechable, No aprovechable, Orgánico y Peligroso.

Viabilidad para la Visión Artificial: Implícitamente, los objetos debían tener características visuales como formas regulares, colores no totalmente transparentes, texturas continuas y apreciables lo suficientemente distintivas para que el modelo de visión artificial pudiera ser entrenado para reconocerlos.

Manipulables: Los residuos debían tener tamaños que no excedan un ancho y largo de 10 cm y un alto de 24 cm debido a que esan eran las

dimensiones del espacio en el que se depositaban los residuos, esto para ser manipulados correctamente tanto por los estudiantes como por los mecanismos del prototipo.

2.3 Técnica de recolección de datos

Fichaje

En el presente trabajo de investigación se eligió usar el fichaje como técnica de recolección de datos, esto con el fin de recolectar datos del indicador de “porcentaje de segregación efectiva”.

De acuerdo con Loayza Maturrano (2021), el fichaje se utiliza como técnica para recolectar y guardar información, posibilitando el ordenar los datos y el recogerlos de forma independiente sobre los indicadores.

2.4 Instrumentos de recolección

Ficha de recolección de datos

Según Robledo Mérida (2006) las llamadas fichas de registro son instrumentos que nos permiten el registrar y hacer la identificación de fuentes de información, acopio de datos y evidencias.

Para la presente investigación se diseñó y utilizó una ficha de registro para evaluar el indicador "Porcentaje de segregación efectiva" de la variable dependiente "Calidad de segregación de residuos" para cuando se usa el proceso manual y para cuando se usa el prototipo, para luego comparar los resultados en dicho indicador por ambos métodos de forma descriptiva e inferencial y así validar la hipótesis: Si se utiliza el prototipo en la I.E. 3049 Imperio del Tahuantinsuyo, entonces el indicador porcentaje de segregación efectiva será superior al obtenido

mediante el proceso manual, mejorando así la calidad de segregación de residuos en esta institución educativa.

El rellenado de la ficha de datos se realizó por mi persona bajo mi supervisión en el aula sobre todos los eventos de segregación ocurridos. Dicha ficha de registro cuenta con las siguientes columnas a rellenar: Fecha, Tipo de Residuo Observado, Predicción, ¿Fue correctamente segregado? (Si/No), Contenedor al que fue dirigido, Contenedor esperado. La fecha corresponde al día en el que ocurrió el evento de segregación, el tipo de residuo observado es el residuo real que yo como supervisor vi que se depositó tanto el proceso manual como el proceso con el prototipo, predicción es el resultado de la inferencia por parte del prototipo el cual lo obtuve mediante el chequeo de datos que guardaba automáticamente el programa python en el archivo excel de su registro, es importante señalar que dicha columna solo tiene valores representativos para el método con el prototipo pues en los eventos con el proceso manual no se obtiene este valor ya que no es de su naturaleza por lo tanto se rellena con “Ninguno” en dicha columna para el proceso manual, ¿Fue correctamente segregado? (Si/No) es el valor que corresponde a si el residuo segregado tanto en el proceso manual como con el prototipo se observó que se depositó en el contenedor correcto de los contenedores de aprovechable, no aprovechable, orgánico y peligroso. “Contenedor al que fue dirigido” es el dato del contenedor en el que se observó que fue depositado. “Contenedor esperado” es el dato del contenedor correcto al que pertenecía dicho residuo segregado. Esta ficha se diseñó en base a los instrumentos usados en trabajos anteriores como el de Colmenares Velasquez & Chavez Vasquez (2024), Diseño y simulación de máquina clasificadora de

residuos reciclables (cartón, metal, plástico y vidrio) usando inteligencia artificial y robot delta en la provincia de Cajamarca, y Meza et al. (2025), Sistema basado en inteligencia artificial para la detección de residuos orgánicos e inorgánicos.

Para completar estas fichas se utilizaron los datos de los eventos de segregación ocurridos en el salón de 5to año “A” con el proceso manual durante 20 días desde el 12 de mayo hasta el 6 de junio obteniendo los datos de ese método con cuatro tachos correspondientes a las cuatro categorías de residuos en instituciones educativas (Aprovechables, No aprovechables, Orgánicos, Peligrosos), posteriormente se utilizaron los datos de los eventos de segregación con el uso del prototipo por el salón ocurridos durante 20 días desde el 9 de junio hasta el 4 de julio con los cuatro contenedores interiores, correspondiente igualmente a las cuatro categorías de residuos, con los que cuenta el prototipo. Posteriormente se calculó el indicador porcentaje de segregación efectiva para todos los eventos de segregación de cada ficha, la cual está determinada por la siguiente ecuación:

$$\text{Porcentaje de segregación efectiva} = \frac{\text{Número de residuos correctamente segregados}}{\text{Total de residuos observados}} \times 100$$

Donde el “número de residuos correctamente segregados” corresponde al número de valores “Si” en la columna “¿Fue correctamente segregado”, y “Total de residuos observados” corresponde al número total de eventos de segregación de la tabla 3 y 5, la cual varía naturalmente para cada proceso ya que no se puede controlar la cantidad de desechos que se generen en una misma cantidad de días para ambos métodos tanto manual como con el prototipo.

2.5 Procesamiento de recolección de datos

Para esta investigación el enfoque fue cuantitativo, por lo tanto se realizó la comparación entre el pre_test (resultados de los eventos de segregación mediante el proceso manual) y el post_test (resultados de los eventos de segregación con el uso del prototipo) basandome en el llenado de fichas cada una por un total de 20 fechas diferentes para cada ficha.

Según el libro metodología de la investigación de Hernández Sampieri, nos recomienda que los análisis cuantitativos de datos sean procesados por una computadora, pues en los últimos años hay una tendencia a no realizar este proceso de forma manual, ni operando fórmulas manualmente, esto sucede aún más cuando hay un volumen grande de datos. (Hernández Sampieri et al., n.d.).

Para el tratamiento y análisis de los datos recopilados durante las fases de pre-test y post-test, se utilizó el software estadístico IBM SPSS Statistics. Esta herramienta es ampliamente reconocida en la investigación científica por su capacidad para ejecutar análisis estadísticos complejos y gestionar datos de manera eficiente, facilitando la obtención de resultados fiables. Las variables principales del estudio son la Variable Independiente, “Máquina automática para segregación de residuos mediante visión artificial”, y la Variable Dependiente, “Calidad de la segregación de residuos”. La operacionalización de estas variables para el análisis se realizó directamente a partir de la ficha de recolección de datos:

- La Variable Independiente fue definida por la fase experimental, correspondiendo al método ‘Manual’ (datos del pre-test, Tabla 3) y ‘Prototipo’ (datos del post-test, Tabla 5).

- La Variable Dependiente se midió directamente a partir de la columna ‘¿Fue correctamente segregado? (Sí/No)’ de dicho instrumento, generando así las categorías ‘Correcto’ e ‘Incorrecto’ para el análisis.

Dado que estas variables operacionales son de naturaleza categórica y nominal, no es aplicable una prueba de normalidad como Shapiro-Wilk, la cual está diseñada para variables cuantitativas continuas. En su lugar, para contrastar la hipótesis principal y comparar la efectividad de ambos métodos, se seleccionó la prueba de Chi-cuadrado (χ^2) de independencia. Esta prueba es la más adecuada por dos razones fundamentales:

- Permite determinar si existe una asociación estadísticamente significativa entre dos variables categóricas.
- Es la prueba idónea para comparar dos muestras independientes, como lo son el conjunto de residuos evaluados en el pre-test (N=99) y el conjunto de residuos evaluados en el post-test (N=92).

El análisis de datos se abordó desde un doble enfoque: descriptivo e inferencial. A nivel descriptivo, se calcularon las frecuencias y los porcentajes para resumir y organizar los datos, lo que permitió cuantificar la efectividad de cada método. Esto se evidencia en el cálculo del "Porcentaje de Segregación Efectiva" para la fase manual (40.40%) y la fase con el prototipo (95.62%). Por su parte, el análisis inferencial, mediante la prueba de Chi-cuadrado, se utilizó para generalizar estos resultados y determinar si la mejora observada era estadísticamente significativa y no producto del azar, permitiendo así validar la cuarta hipótesis de la investigación sobre el impacto positivo del prototipo "EcoSplit".

2.6 Comprobación de Hipótesis

Se describen los procedimientos metodológicos específicos que se seguirán para la validación de cada una de las hipótesis del estudio.

Para la Hipótesis 1 (Sobre Requerimientos y Diseño):

Se aplicará un método de demostración causal basado en la trazabilidad, dicha matriz de trazabilidad se encuentra en la sección de resultados en la tabla 8. El procedimiento consistirá en: (1) documentar los requerimientos funcionales y técnicos derivados del análisis del problema y (2) elaborar una matriz de trazabilidad que conecte cada requerimiento con una decisión de diseño específica, demostrando que la base del diseño es una consecuencia directa de dichos requerimientos.

Para la Hipótesis 2 y 3 (Sobre Diseño, Prototipado y Ejecución):

Se utilizará un método de demostración empírica y validación funcional. El procedimiento se dividirá en: (1) el diseño de los cinco subsistemas funcionales mediante herramientas de ingeniería; (2) la construcción física y el ensamblaje del prototipo como prueba de la viabilidad del diseño; y (3) la ejecución de pruebas funcionales para verificar que el prototipo puede realizar cada ciclo de segregación de manera correcta.

Para la Hipótesis 4 (Sobre la Comparación con el Método Manual):

La Hipótesis 4 ("Si se utiliza el prototipo en la I.E. 3049 Imperio del Tahuantinsuyo, entonces el indicador porcentaje de segregación efectiva será estadísticamente superior al obtenido mediante el proceso manual.") se validará

mediante un diseño pre-experimental de pre-prueba y post-prueba. El método será el siguiente:

Fase de Pre-prueba (Manual): Se medirá el "Porcentaje de Segregación Efectiva" del proceso manual durante un período de 20 días.

Fase de Post-prueba (Prototipo): Tras la intervención, se medirá el "Porcentaje de Segregación Efectiva" utilizando el prototipo "EcoSplit" durante un período de 20 días.

Análisis Estadístico Comparativo: Se realizará un análisis estadístico de los resultados cuantitativos de ambas fases, el cual constará de dos niveles:

- **A nivel descriptivo:** Se calcularán los porcentajes de segregación efectiva totales para cada fase y la mejora neta porcentual. Los resultados se presentarán visualmente mediante tablas y gráficos de barras para ilustrar la magnitud de la diferencia.
- **A nivel inferencial:** Se aplicará una prueba estadística apropiada (Una prueba de chi-cuadrado) para determinar si la mejora observada es estadísticamente significativa. La hipótesis se considerará validada si se demuestra que la diferencia entre el rendimiento del prototipo y el método manual no se debe al azar (p. ej., con un nivel de significancia $p < 0.05$).

Los resultados, datos y evidencias obtenidos tras la aplicación de esta metodología se presentarán de manera detallada en el Capítulo III.

2.7 Aspectos éticos

Durante el desarrollo de esta investigación se han respetado los principios éticos fundamentales, garantizando en todo momento la integridad, la seguridad y el bienestar de los participantes y del entorno. No se ha involucrado el uso de datos personales sensibles ni la intervención directa de personas sin consentimiento. El prototipo fue diseñado para operar en condiciones seguras, sin representar riesgos eléctricos o mecánicos para los estudiantes. Asimismo, se ha procurado utilizar materiales accesibles y prácticas sostenibles que no generen impacto ambiental negativo. El objetivo final del proyecto es contribuir a la educación ambiental y la mejora de la gestión de residuos sólidos, promoviendo una cultura responsable en el tratamiento de desechos en la comunidad educativa. Se respetaron en todo momento las normativas institucionales y se actuó con total transparencia en la recolección y presentación de resultados. También se respetó una redacción siguiendo el manual de normas APA 7ma Edición y se realizó la redacción de todos los pasos del desarrollo del prototipo, recolección de datos y procesamiento de estos mismos.

CAPÍTULO III: RESULTADOS

3.1 Instrumentos, materiales y métodos

3.1.1 Instrumentos

Los instrumentos utilizados en el desarrollo de este trabajo fueron los siguientes:

- Ficha de recolección de datos: Se diseñó un instrumento que evalúe el indicador "Porcentaje de segregación efectiva" de la variable dependiente "Calidad de segregación de residuos". Esta ficha se diseñó en base a los instrumentos usados en trabajos anteriores como el de Colmenares Velasquez & Chavez Vasquez (2024), Diseño y simulación de máquina clasificadora de residuos reciclables (cartón, metal, plástico y vidrio) usando inteligencia artificial y robot delta en la provincia de Cajamarca, y Meza et al. (2025), Sistema basado en inteligencia artificial para la detección de residuos orgánicos e inorgánicos. Mediante esta ficha se pudo realizar un análisis descriptivo e inferencial de los datos y también validar la hipótesis: Si se utiliza el prototipo en la I.E. 3049 Imperio del Tahuantinsuyo, entonces el indicador porcentaje de segregación efectiva será superior al obtenido mediante el proceso manual, mejorando así la calidad de segregación de residuos en esta institución educativa.
- Software CAD Autodesk Inventor: Autodesk Inventor es uno de los software para diseño 3D en ingeniería más usados en el mundo, el uso de este software fue esencial para el diseño del prototipo, su uso en este trabajo se puede dividir en dos etapas, la primera parte en la que se utilizó este software fue en el modelado 3D completo del prototipo, tanto de los componentes

electrónicos, mecánicos, estructurales y de su respectiva ubicación dentro del prototipo, ayudo a pre-visualizar en tres dimensiones los bosquejos en papel con todos los aspectos físicos de la máquina para encontrar una adecuada organización, la segunda parte en la que este software fue útil fue en el diseño de los mecanismos, exactamente al dimensionar y diseñar la cinemática requerida en los mecanismos que se necesitaban, como lo es el del mecanismo redirigidor y el mecanismo iris de apertura y cierre.

- Visual Studio Code: Es un editor de código fuente para programadores para escribir y editar código en diferentes lenguajes de programación. En el presente trabajo el lenguaje de programación que se uso para el sub-sistema de procesamiento y decisión, fue Python.

- Python: Es un lenguaje de programación de alto nivel, sencillo y versátil. En el presente trabajo se lo utilizó en el sub-sistema de procesamiento y decisión, detectando residuos en tiempo real usando un modelo re-entrenado para este trabajo conocido como YOLOv11 procesando las imágenes con OpenCV. En una segunda parte se lo utilizó para comunicarse con el sub-sistema de actuación y señalización/activación, recibiendo y enviando señales a la placa Arduino uno mediante comunicación serial activando el mecanismo re-dirigidor, el mecanismo iris de apertura y cierre, y activando los led's de actividad del prototipo. En una tercera parte se uso este lenguaje para registrar los residuos detectados en un archivo Excel y generar gráficos semanales y mensuales actualizándolos automáticamente en el mismo archivo.

- Computer Vision Annotation Tool (CVAT): Es una herramienta de anotación de imágenes y videos de código abierto desarrollada por Intel, actualmente esta herramienta se puede usar de manera online, permite

etiquetar objetos en imágenes para entrenar modelos de visión artificial. Fue diseñada específicamente para crear conjuntos de datos que puedan ser usados en aplicaciones como detección de objetos. En mi trabajo fue usada para etiquetar manualmente las imágenes de los residuos de mi base de datos, es decir, para marcar los distintos tipos de residuos presentes en cada imagen que recolecté, de dicha manera generé los labels necesarios exportados al final en el formato de entrenamiento para el modelo YOLOv11 y así pudiera aprender a detectar correctamente los residuos en tiempo real.

- **Arduino IDE:** Es un entorno de desarrollo utilizado para programar placas Arduino. Permite escribir, compilar y cargar código en la placa a través de un puerto USB, usando un lenguaje basado en C/C++. En mi trabajo se usó para programar la placa Arduino uno de manera que el primer uso que se le dio es que reciba señales desde Python para modificar la posición angular para el mecanismo redirigidor y el mecanismo iris de apertura y cierre, y el segundo uso fue para el sub-sistema de señalización/activación, activando los led's verde y rojo según la señal de actividad del sistema que reciba Arduino uno.
- **Excel:** Es un programa de hojas de cálculo que permite organizar y analizar datos de forma tubular. Para mi trabajo Excel se usó como base de datos de las detecciones realizadas y como visualizador de los datos mediante graficos de barras de los residuos detectados semanalmente y mensualmente. Se registró automáticamente cada detección realizada por el sub-sistema de procesamiento y decisión guardando la fecha, hora, nombre del objeto y su categoría (aprovechable, no aprovechable, orgánico o peligroso).
- **Impresora 3D Bambulab A1:** Es una impresora 3D que permite fabricar objetos capa por capa usando filamento termoplástico como PLA. Se destaca

por su alta velocidad y alta precisión. En este trabajo se usó el servicio de impresión de 3D con este modelo en particular para obtener los dos mecanismos usados y todas sus piezas.

3.1.2 Materiales

Los materiales usados en este trabajo fueron:

- You only look once (YOLOV11): Es un avanzado algoritmo de detección de objetos en tiempo real que, a diferencia de otros métodos, utiliza una única red neuronal para analizar una imagen completa en una sola pasada, haciéndolo extremadamente rápido y eficiente. Se empleó su versión YOLOV11 como el cerebro del sistema de clasificación automática, para lo cual fue reentrenado con una base de datos de 11,000 imágenes de residuos locales. Durante su funcionamiento, el modelo YOLOV11 analizaba instantáneamente la imagen de un desecho capturada por una cámara, lo identificaba y daba un resultado de que residuo infería, para luego enviar una señal a los mecanismos de la prototipo, que dirigían físicamente el residuo al contenedor correcto.
- Laptop ASUS TUF Dash F15 modelo FX516P: Fue el componente hardware del sub-sistema de procesamiento y decisión para el prototipo de la máquina fueron, cuenta con un procesador Intel Core i7-11375H de cuatro núcleos. En cuanto a memoria RAM, dispone de 16 GB DDR4 a 3200 MHz. También se usó como fuente de alimentación para la placa Arduino uno y la cámara usb, es decir para alimentar en parte al sub-sistema de actuación ya que en este caso solo alimentaba la placa Arduino uno y la cámara usb, luego Arduino uno al estar alimentado enviaba las señales a los servomotores de

posición angular y también activaba los led's rojo y verde. Sin embargo para la alimentación de los servomotores MG90S se usó otra fuente de alimentación externa y así cambiaban su posición angular a la indicada por la señal de la placa Arduino uno.

- Cámara web usb modelo RT-A200 de la marca OEM: Es una cámara usb de resolución de 1920 x 1080 píxeles y tasa de fotogramas por segundo de 30 fps. Fue utilizada para el sub-sistema de percepción encargado de capturar las imágenes en tiempo real y enviarlas al sub-sistema de procesamiento y decisión.

- Placa Arduino uno: Es una placa de desarrollo electrónica de código abierto basada en el microcontrolador ATmega328P. Es muy utilizado en el prototipado de proyectos de electrónica, automatización y robótica. La placa Arduino uno en mi trabajo recibe mediante comunicación serial la decisión tomada por el sub-sistema de procesamiento y decisión en la laptop, en la ejecución de un código python, para que Arduino uno inmediatamente a partir de su programación envíe la señal de posición angular a los segundos componentes hardware importantes, los servomotores MG90S, los cuales se encuentran interrelacionados con el tercer componente, los cuales son los mecanismos de apertura, cierre y redirección, con el cambio de posición angular de los servomotores, estos mecanismos de apertura, cierre y redirección toman posiciones particulares y determinadas que permiten que los residuos se depositen en los espacios determinados y adecuados. Otras dos funciones de la placa Arduino uno en mi trabajo son los de activar los led's rojo y verde del sub-sistema de señalización/activación indicando si el prototipo está en actividad o está esperando, y un último uso es el de recibir la

señal de activación del prototipo registrando si el botón de inicio ha sido presionado y ejecutando parte de la programación que se activa bajo esa condición, para luego enviar una señal de ejecución de código mediante comunicación serial a python para que detecte el residuo ingresado.

- Fuente de alimentación externa: Es la fuente de alimentación externa que provee 24 V a un máximo de 5 A a parte del sub-sistema de actuación, pues se usó para alimentar los cuatro servomotores MG90S cuando dicho voltaje era reducido por el convertidor DC-DC regulable XL4015 el cual regulaba el voltaje a 6 V y así se alimentara correctamente los servomotores y estos puedan cambiar su posición angular haciendo que los mecanismos tomen una configuración particular y útil.
- Convertidor DC-DC regulable XL4015: Es un módulo electrónico que permite reducir y regular el voltaje de una fuente de corriente continua, en el modelo particular que se usó soporta un voltaje de entrada de entre 4 -38 V y un voltaje de salida de entre 1.25 -32 V a un máximo de 5 A, de esta forma se lograba alimentar con el voltaje correcto a los servomotores MG90S.
- Servomotor MG90S: Es un pequeño motor eléctrico de precisión realimentado que puede girar en un ángulo controlado entre 0° y 180° y se utiliza ampliamente en proyectos de prototipado de electrónica. Al ser realimentado comprueba y ajusta su posición angular hasta la indicada por el controlador. Es de tamaño compacto y resistente debido a sus engranajes metálicos a diferencia del servo-motor SG90 cuyos engranajes son de nylon. Su voltaje de operación es entre 4.8V a 6V y dependiendo de la carga la corriente eléctrica que puede darse en reposo o operación esta entre 10 mA a 1 A. Su uso en este trabajo fue fundamental ya que son los actuadores que

generaron una entrada de movimiento rotacional en los mecanismos iris y re-dirigidor para que estos tomen configuraciones particulares necesarias.

- Resistencia: Es un componente electrónico, el cual se opone al flujo de corriente eléctrica en un circuito. Su función principal es el de limitar la corriente y disipar la energía en forma de calor, ayudando a proteger otros componentes. La resistencia también como magnitud física se mide en ohmios (Ω). En este trabajo se los uso para reducir el voltaje en la parte de alimentación de los LED's, así proveer una alimentación adecuada evitando dañar los led's rojo y verde.
- Cables jumpers: Los cables jumpers son cables eléctricos pequeños y flexibles que se utilizan para realizar conexiones para prototipados entre los componentes. Las conexiones del prototipo entre los componentes electrónicos y mecánicos fueron realizadas mediante cables jumpers.
- Diodo emisor de luz (LED): Es un componente electrónico que emite luz cuando circula corriente eléctrica a través de él en la dirección correcta, sus siglas significan diodo emisor de luz, comercialmente hay distintos led's que emiten diferentes colores, en el presente trabajo se usaron los colores rojo y verde para señalar la actividad del prototipo.
- Interruptor de presión: Es un interruptor momentáneo el cual solo permite el paso de corriente eléctrica mientras está presionado. Al soltarlo, vuelve a su estado abierto.
- Mecanismo re-dirigidor: Es un mecanismo espacial diseñado de dos grados de libertad, cuyas dos entradas de movimiento rotacional son producidas por dos servomotores MG90S, la configuración determinada en

este mecanismo es usada para poder ubicar en el contenedor correcto a los residuos según su categoría. En este trabajo para el diseño cinemático del mecanismo re-dirigidor se uso el método gráfico validado con el uso de simulación de movimiento en el software Autodesk Inventor, ya que los métodos avanzados no fueron necesarios para el diseño de este mecanismo espacial particular, el método gráfico es una útil herramienta para casos donde un análisis extenso no aportaría una mayor precisión considerable en la función del mecanismo frente al uso del método gráfico mediante un software CAD/CAE como Autodesk Inventor (*Maquinas y Mecanismos*, 2011).

- Mecanismo iris: Es un mecanismo plano, en el cual todos los movimiento relativos de todos los eslabones se encuentran en el mismo plano y pueden ser graficados desde solo una perspectiva. La función del mecanismo iris es permitir el ingreso o no, mediante la apertura y cierre en un orificio central, estas dos configuraciones particulares se logran con la entrada de movimiento rotacional del servo-motor MG90S, por lo tanto este mecanismo solo cuenta con un grado de libertad para llegar a las dos configuraciones necesarias. En el presente trabajo se diseño la cinemática de este mecanismo iris haciendo uso de un método analítico, sin embargo también se realizó el uso de un método gráfico alternativo en cierta etapa para validar que ambos métodos en el diseño de ciertos mecanismos pueden ser adecuados cuando se obtiene la misma funcionalidad sin volver tan complejo el proceso de diseño (Shigley et al., 1996).

3.1.3 Método

El proceso de diseño y prototipado de la máquina automática mediante visión artificial para mejorar la calidad de segregación de residuos en la I.E. 3049 Imperio del Tahuantinsuyo, distrito Independencia, Lima, se realizó mediante el método sistemático.

3.1.3.1 Requerimientos

Teniendo en cuenta los antecedentes internacionales y nacionales mencionados y el análisis de la realidad problemática se puede formular esta sección para la solución de esta. Como se busca diseñar un prototipo de una máquina capaz de segregar los residuos automáticamente se debe tener un primer acercamiento mediante la literatura sobre las máquinas que comparten este propósito.

Según Aydin Adnan & Rehman (2022) para el tipo de máquinas automáticas que separen residuos un tipo serían las máquinas que utilicen sensores ópticos, las cuales comúnmente tienen cuatro etapas, alimentación, presentación, identificación y separación. Menciona que la etapa de alimentación y presentación son fundamentales ya que al usar los sensores ópticos la eficiencia de la distinción se ve afectada por como presentamos los objetos ante el sensor óptico. Estas primeras etapas según el investigador se diseñan según las necesidades del usuario y del material a separar. En la etapa de identificación se menciona que puede consistir de dos partes, lo que es conjunto de sensores y la iluminación. Entre los conjuntos de sensores se menciona que pueden usarse sensores de conductividad, cámaras monocromáticas o de color, o el sensor óptico adecuado según la necesidad. La iluminación es un factor importante en algunos casos se usan lámparas

fluorescentes o halógenas, en otros se usan diodos emisores de luz, pero también su selección depende del ambiente y la necesidad que se este resolviendo. El autor menciona que es importante que exista un buen equilibrio entre la velocidad de procesamiento y la precisión de los datos obtenidos, ya que es esencial para la eficiencia del proceso de identificación. La etapa de separación se diseña de acuerdo a los pesos de los objetos que se separarán, en algunos casos se usan válvulas de chorro de agua, válvulas de succión, válvulas mecánicas y válvulas de aire según el peso y el tipo de objeto. Menciona que el aumento en el uso de sistemas de separación en la industria se debe a su rapidez, precisión y eficiencia.

Lo mencionado por el autor anterior se llega a complementar con el trabajo de Alejandro et al. (2024), en el cual diseñó un prototipo para la clasificación de botellas de materiales como vidrio, plástico u otro material usando el reconocimiento de imágenes. El sensor óptico usado en este prototipo fue una cámara de color en conjunto con un modelo re-entrenado de detección de objetos YOLO, esto debido a que a diferencia de otros sensores como el de conductividad o monocromático, una cámara de color junto a un modelo re-entrenado de detección de objetos como el mencionado puede llegar a distinguir muchos más objetos de diferentes materiales al analizar más propiedades como su tamaño, forma, color y brillo. Dicho aporte es la principal razón por la que en el presente trabajo tenga en cuenta el uso de una cámara de color para la etapa de identificación de objetos ya que se busca distinguir un conjunto diferente de residuos de múltiples formas y colores, de

esta forma con la estructura general de Rehman para este tipo de máquina se concibe una noción del diseño del prototipo.

Un modelo de red neuronal convolucional para detección de objetos nos permite a partir de los objetos que buscamos que se detecte automáticamente y una base de datos de imágenes para el entrenamiento y validación, re-entrenar los modelos para los objetos con los que estemos trabajando y así poderlos aplicar en la solución de un problema.

Según lo expuesto en el trabajo de Arishi (2025), se diseñó un sistema de clasificación y detección de residuos domésticos en tiempo real, utilizando el modelo YOLOV8. Como resultado, el modelo re-entrenado YOLOV8 alcanzó una precisión media (mAP) del 89.5%, superando ampliamente a modelos tradicionales como Mask R-CNN y Faster R-CNN. Este alto rendimiento demostró la eficacia del sistema para las aplicaciones de detección de objetos en tiempo real. El estudio concluyó recomendando el uso de visión artificial basada en YOLOV8 como una solución efectiva para las actividades de segregación, ya que a parte de tener una alta precisión de detección también es mucho más simple de entrenar que modelos como Tensorflow, y más eficaz en la detección en tiempo real que de manera estática como Tensorflow. Debido a que se busca realizar detección en tiempo real con alta precisión se considerará el uso de un modelo YOLO para la detección de residuos en este trabajo.

Asimismo, Alejandro et al. (2024), en su tesis que tuvo como objetivo diseñar e implementar un sistema de clasificación automática de botellas a través del reconocimiento de imágenes mediante visión artificial estructuró las

cuatro etapas generales de su máquina de forma que la etapa de identificación y el modelo YOLO usado se procesará en un ordenador externo mientras que la activación de los mecanismos físicos (motor de la banda, pistón de clasificación) fue controlada por un microcontrolador ESP32 y el entorno de desarrollo de Arduino, comunicándose vía serial con el sistema principal. Esta investigación demuestra que es viable implementar sistemas de clasificación inteligente de residuos mediante visión artificial y microcontroladores de bajo costo. Este hecho, valida el uso de una arquitectura distribuida donde la detección se realiza en un ordenador con Python y el control de actuadores se gestiona desde un microcontrolador, lo cual permite modularidad, bajo costo y adaptabilidad para mejoras posteriores en el prototipo.

Dado el análisis anterior, se propone los siguientes puntos para los requerimientos, debido a que el ambiente en el que se probará el prototipo es una institución educativa secundaria no se puede ocupar un gran espacio, entonces las dimensiones deben ser no tan lejanas a la de los tachos convencionales de las instituciones pero lo suficiente para albergar los residuos que se segregarán, en base a los tachos utilizados en esta institución de proponer tomar como base las dimensiones de dichos tachos, las cuales son de un tacho de 100 L Reyplast (53x51x84 cm), donde las primeras dos medidas corresponden al ancho y largo mientras que la última corresponde al alto, dichas medidas servirán mas adelante para ajustar estas dimensiones a las requeridas según otros criterios en el diseño de la estructura.

Como se busca segregar los residuos con una alta precisión se propone una alimentación individual de los residuos de los estudiantes de 5to año de

secundaria cuando ellos tengan algún residuo a separar, la alimentación individual aporta que el prototipo pueda segregar los residuos con mayor precisión lo cual para la forma en que se lleva a cabo es adecuado ya que no se presentaban tasas altas de segregación simultánea, esto también beneficia en el hecho de que el prototipo no tendría una alta potencia para su funcionamiento en todo momento.

Para la parte de alimentación al prototipo se propone que al tener una tasa baja de segregación simultánea los estudiantes puedan ingresar sus residuos individualmente por una apertura producida por un mecanismo de ingreso al prototipo, así de esta forma los residuos llegarían individualmente a la etapa de presentación para posteriormente iniciar con la etapa de identificación. Al estar los residuos ya ubicados en el espacio de presentación serían identificados por el sensor óptico seleccionado enviando las imágenes capturas a una laptop externa para su procesamiento, en este caso se propone usar una cámara web usb modelo RT-A200 de la marca OEM, el cual estaría ubicado en un soporte para cámaras apuntando en la dirección del espacio de presentación donde se encuentre el residuo que haya sido ingresado. Una vez que se haya identificado el residuo, para la parte de segregación se propone usar el mismo mecanismo de apertura y cierre para que se envíe a un mecanismo que re-dirija el residuo al contenedor correspondiente dentro del prototipo, de esta manera segregando el residuo correctamente.

En base al análisis reciente de los antecedentes y el análisis del contexto donde se busca aplicar el prototipo se elabora la tabla 1 que resume los

requerimientos funcionales y técnicos que se debe tener en cuenta en el diseño del prototipo.

Tabla 1

Requerimientos funcionales y técnicos del prototipo

Requerimientos Funcionales	Requerimientos Técnicos
<ul style="list-style-type: none"> • Tener compartimientos internos para las cuatro categorías de residuos (Aprovechable, No Aprovechable, Orgánico y Peligroso). • Recibir y procesar un residuo a la vez por cada usuario. • Realizar la identificación del residuo de forma automática. • Depositar cada residuo identificado en el compartimiento correcto. • Registrar la fecha y hora, resultado de la inferencia, y categoría correspondiente de la clase de cada evento de segregación de forma automática. 	<ul style="list-style-type: none"> • El prototipo debe tener dimensiones aproximadas a la del tacho Reyplast usado en la institución por los alumnos 53x51x84 cm . • Contar con un espacio cuyo volumen sea suficiente para la etapa de presentación de los residuos seleccionados a la cámara. • Diseñar mecanismos de apertura y cierre que aprovechen el giro de servomotores de bajo costo para el ingreso y salida de los residuos al espacio de presentación ante la cámara y fuera de él a la etapa de separación. • Diseñar un mecanismo de redirección que aproveche el giro de servomotores de bajo costo para el depósito de cada residuo a los compartimientos correspondientes según la inferencia del prototipo. • Proveer retroalimentación visual al usuario sobre la actividad del prototipo mediante LED's. • Ser de fácil interacción para los usuarios durante el evento de segregación con solo el uso de un único botón.

	<ul style="list-style-type: none">• Definir la versión del modelo de red neuronal convolucional para procesamiento de video en tiempo real (YOLO) para la detección de residuos por la cámara.• Uso de una cámara de color.• Emplear una arquitectura de hardware de bajo costo suficiente basada en el uso de una laptop para el procesamiento y la placa Arduino Uno para el control de los actuadores.• Seleccionar la fuente de alimentación en base a los actuadores, elementos de señalización y cámara.• Debido a que se necesita poder supervisar los procesos del prototipo para verificar su correcto funcionamiento y diagnóstico de problemas, seleccionar un material translúcido.• Diseñar el prototipo con tal de que se puedan retirar los residuos fácilmente.• Disponer al prototipo de elementos que le permitan ser fácil movilización.
--	---

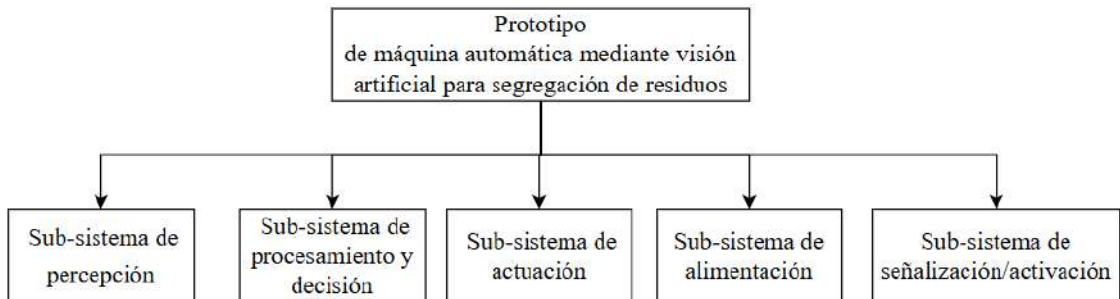
3.1.3.2 Diseño del prototipo de máquina automática mediante visión artificial para segregación de residuos

3.1.3.2.1 Diseño general

El diseño general del prototipo esta compuesto por cinco sub-sistemas: percepción, procesamiento y decisión, actuación, señalización/activación, y alimentación.

Figura 1

Sub-sistemas de prototipo

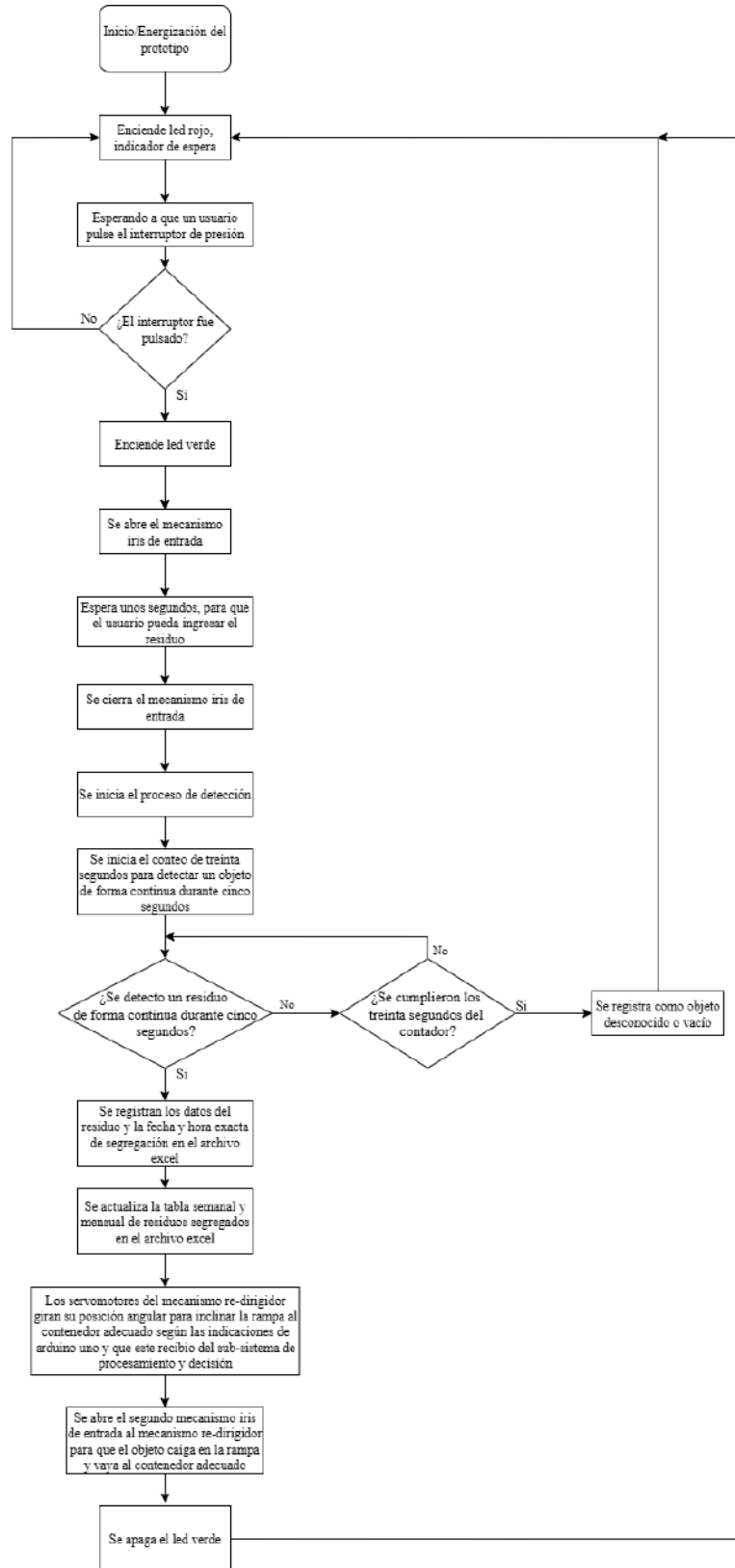


El diagrama de flujo del prototipo propuesto muestra todas las actividades que se dan en el funcionamiento de este como se puede observar en la figura

2.

Figura 2

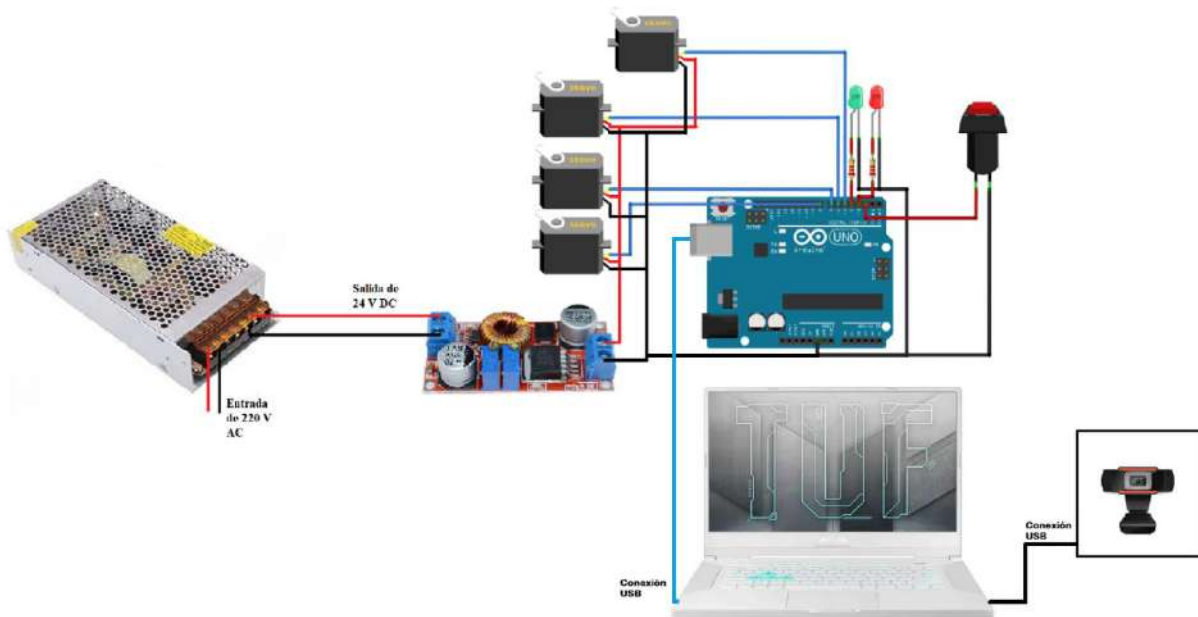
Diagrama de flujo del prototipo



Se presenta en la figura 3 todas la conexiones realizadas entre todos los componentes del prototipo, dichas conexiones entre todos los sub-sistemas serán explicadas en el apartado del diseño de cada sub-sistema:

Figura 3

Diagrama esquemático de todas las conexiones del prototipo



3.1.3.2.2 Diseño del subsistema de percepción

El subsistema de percepción está compuesto por los componentes hardware que se usarán para la captura de imágenes, en este trabajo se está usando la cámara web USB modelo RT-A200 de la marca OEM debido a que como se indicó anteriormente una cámara de color en conjunto con un modelo de red neuronal como YOLO son un método con una gran precisión en la detección de objetos de distintos colores, tamaños y formas. La cámara utilizada tiene una resolución de 1920 x 1080 píxeles y la tasa de fotogramas por segundo que sería 30 fps, cuya calidad de imagen es suficiente para obtener los detalles suficientes para la distinción (texturas, bordes finos, colores precisos), acompañada de la tasa de fps que es un estándar para video

en tiempo real, garantizando una captura fluida sin perder información. También para posicionar correctamente la cámara se está usando un soporte para cámaras como el que se puede observar en la imagen 4. La cámara se encargará de capturar las imágenes en tiempo real para enviarlas mediante usb al modelo re-entrenado YOLOV11 que se estará ejecutando en la laptop del sub-sistema de procesamiento y decisión. La cámara es alimentada a su vez por la laptop del sub-sistema de procesamiento y decisión.

Figura 4

Cámara web usb modelo RT-A200 de la marca OEM



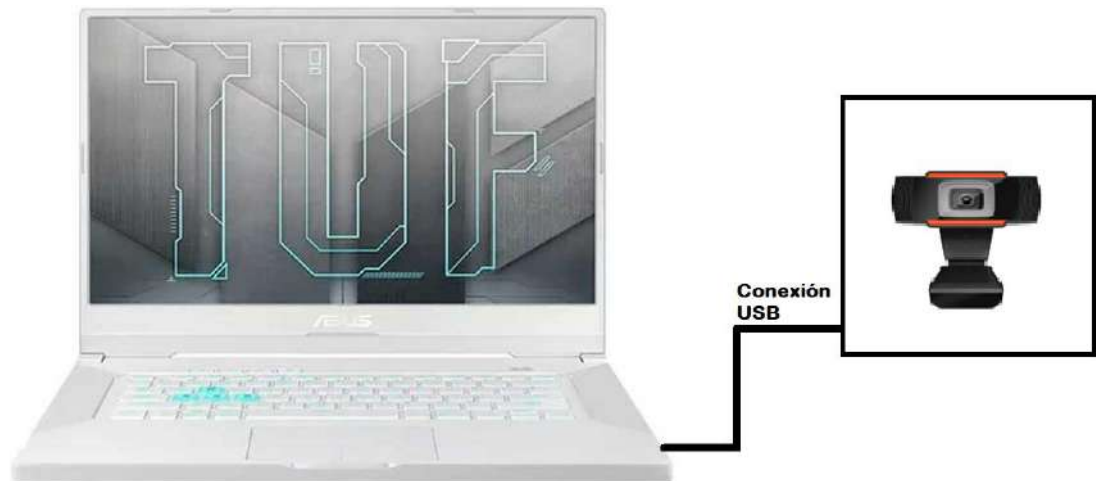
Figura 5

Soporte para cámara



Figura 6

Conexión entre el sub-sistema de percepción y el sub-sistema de procesamiento y decisión



3.1.3.2.3 Diseño del subsistema de señalización/activación

A partir del análisis realizado en los requerimientos se propuso para el diseño del prototipo el uso de la combinación de un ordenador externo como una laptop para la ejecución del modelo re-entrenado YOLOV11 mediante python y el uso de una tarjeta de prototipado de bajo costo con un microcontrolador como Arduino uno suficiente para el envío de las señales a los servomotores MG90S, los led's de señalización, y el recibimiento de la señal del interruptor de presión para el inicio del proceso de segregación y su posterior envío de una señal mediante comunicación serial a la laptop para la ejecución del modelo de detección de residuos. Es decir la placa Arduino uno se usará para el sub-sistema de actuación y señalización/activación. La selección de la placa Arduino uno se debió a que para las necesidades de control de los componentes mencionados cuenta con la cantidad de pines

suficiente y la memoria suficiente, al mismo tiempo una placa de Arduino uno para prototipado es de muy bajo costo comparado a otras tarjetas de prototipado como lo serían ESP32 cuyas capacidades sobrepasarían las necesidades para esta etapa.

Figura 7

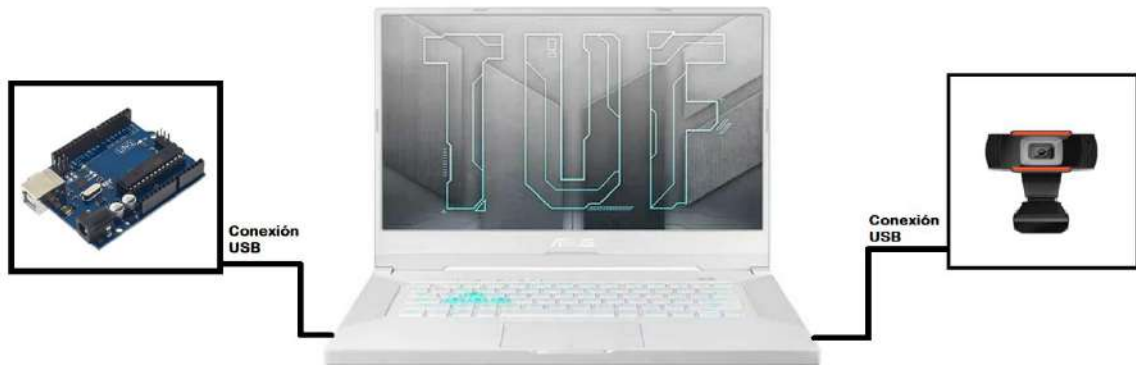
Placa de prototipado Arduino Uno



Como se propone usar dos led's para indicar los dos estados del prototipo, se eligieron los colores verde y rojo, el color rojo para representar el estado en el que el prototipo esta esperando por algún usuario que pulse el interruptor de presión para iniciar los procesos, mientras que el color verde se usó para representar el estado en el que un usuario ya pulsó el interruptor y se han iniciado los procesos del prototipo, dicho led verde estará encendido hasta que vuelva al primer estado. Se propone que la programación de Arduino Uno trabaje en conjunto con la programación en el lenguaje Python en la laptop comunicados serialmente para así coordinar el orden de la ejecución de los procesos.

Figura 8

Conexión entre Arduino Uno y el código python en la laptop



Los ánodos de los led's verde y rojo estarán conectados a los pines 4 y 3 respectivamente desde donde serán alimentados y encendidos según la programación de Arduino Uno acompañados de dos resistencias de 220Ω , mientras que sus cátodos estarán conectados a tierra, uno de los pines del interruptor de presión estará conectado al pin 2 de la placa mientras que el otro pin del interruptor estará conectado a tierra, dicho valor del pin será leído para iniciar el proceso mediante una condicional If, para lo cual debemos establecer un valor predeterminado cuando el interruptor no este pulsado, el cual se logra mediante la asignación del modo INPUT_PULLUP a dicho pin, esto para que dicho pin internamente en el microcontrolador cuando no este conectado a nada tenga un voltaje de 5 V a través de una resistencia de $20 \text{ k}\Omega$, pero cuando sea pulsado el valor del pin sea de 0 V y al leer dicho valor con la condicional If se inicie los procesos del prototipo.

- **Cálculo de resistencias para los led's verde y rojo**

Para los led's rojos y verdes de 5 mm podemos asumir un caída de voltaje adecuada de 1.8 V de acuerdo a las especificaciones. Debido a que los pines a los que estaran conectados estos led's tienen un voltaje de 5 V se tiene que la caída de voltaje en la resistencia sera de 3.2 V.

$$V_r = 5 \text{ V} - 1.8 \text{ V}$$

$$V_r = 3.2 \text{ V}$$

De acuerdo a las especificaciones de los colores de estos led's una corriente eléctrica de 20 mA es adecuada para un brillo apreciable.

$$I_r = 20 \text{ mA} = 0.02 \text{ A}$$

Por lo tanto la resistencia eléctrica de ambos resistores por medio de la ley de ohm sería:

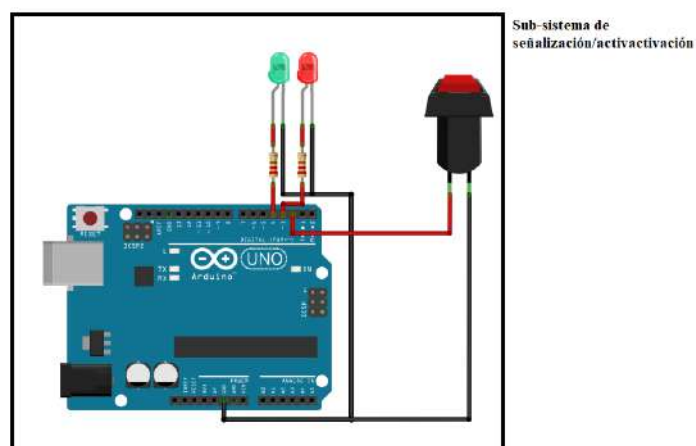
$$R_r = \frac{V_r}{I_r} = \frac{3.2 \text{ V}}{0.02 \text{ A}} = 160 \Omega$$

Sin embargo el valor de resistencia comercial mas cercano sería 180 Ω . Pero por un margen de seguridad se toman valores un poco mayores como de 220 Ω ya que a mayor resistencia menor corriente, sin embargo este valor aún así aporta un brillo moderado y apreciable.

Una vez definido los detalles de los componentes y sus conexiones se puede hacer el esquema de conexiones.

Figura 9

Conexiones del sub-sistema de señalización/activación



3.1.3.2.4 Diseño del subsistema de procesamiento y decisión

En esta etapa se deben ejecutar procesos importantes en la laptop pero antes de ello debe recibir la señal de que el interruptor de presión ha sido pulsado por un usuario por parte la placa Arduino Uno del sub-sistema de señalización/activación, esto se logra mediante la comunicación serial entre el código python y la placa Arduino Uno mandando un mensaje que activa la ejecución del bloque de detección de residuos. Para comenzar con el diseño de esta etapa se debe definir que objetos serán segregados, para esto se selecciono algunos de los residuos que generan los alumnos de una sección de quinto año de secundaria, los cuales deben estar agrupados según la norma técnica sugerida por el Ministerio del Ambiente.

Tabla 2

Residuos segregados por los 15 alumnos del 5to grado “A” de secundaria de la I.E. 3049 Imperio del Tahuantinsuyo

Tipo de Residuo	Residuos
Aprovechable	<ul style="list-style-type: none"> - Botella plástica Inca Kola 500 mL - Gaseosa en Lata Oro Kola 355 mL - Retazos de cartón - Bolas de papel - Late de leche bonle defensis 390 g - Late de leche sin etiqueta
No aprovechable	<ul style="list-style-type: none"> - Vaso de poliestireno blanco pequeño - Vaso de poliestireno con diseño - Vaso de plástico transparente - Envoltura de galleta ritz 52.5 g - Envoltura de galleta chips ahoy 37 g

Orgánico	<ul style="list-style-type: none"> - Cascara de huevo partida a la mitad - Cascara de platano - Pan integral - Sobres de anís - Restos de manzana
Peligroso	<ul style="list-style-type: none"> - Tabletas de pastillas - Foco LED - Pila panasonic - Envase de agua oxigenada 120 mL - Envoltura panadol antigripal verde - Jeringa

Teniendo definido los residuos que se segregarán, el siguiente paso es la recolección de la base de datos de imágenes, para esto se capturaron 500 imágenes de cada residuo en diferentes posiciones. Debido a que en este prototipo la estructura del prototipo es transparente para que se puedan apreciar los sub-sistemas un factor a tener muy cuenta fue la iluminación, según los antecedentes se recomienda que las imágenes sean capturadas en un entorno similar a donde serán probadas o en el mismo entorno para que ya en la experimentación se tenga una alta precisión de detección, debido a que el prototipo será usado durante el turno de la sección de quinto de secundaria de la mañana, la captura de imágenes de los residuos se realizó en horas de la mañana por obtener una iluminación natural similar al del entorno de experimentación. También las imágenes serán capturadas en la ubicación donde serán presentados a la cámara, donde para evitar que la cámara pueda confundir el fondo con parte del objeto se recubrió el espacio de presentación de pintura en spray negra. La captura de imágenes se realizó casi de forma automática mediante un código python donde se puede seleccionar la cantidad de imágenes que se quiere capturar mediante la cámara usb. Dicha captura de

imágenes también se realizó con la misma cámara que se realizaría la detección para poder asegurar que la precisión de detección no se vea afectada por usar una cámara diferente con especificaciones diferentes.

- **Código python para la captura de imágenes**

El presente código de python se realizó y ejecuto en el software Visual Studio Code, tiene la función de capturar imágenes para el entrenamiento del modelo.

Figura 10

Código en python para la captura de imágenes

```
import cv2
import os
import time

#Se configura la carpeta donde se guardarán las imágenes
output_folder = "C:/Users/Calderon/Desktop/imagenes"

# Se inicializa la cámara, en este caso se usa 0 como argumento para indicar que use la cámara predeterminada
cap = cv2.VideoCapture(0)

#Verifica si la cámara se abrió correctamente
if not cap.isOpened():
    print("La cámara no se pudo abrir")
    exit()

#Configuración de número de imágenes y pausa de tiempo
num_images = 500 #número total de imágenes
intervalo_segundos = 0.09 #tiempo entre capturas en segundos

count = 0 #Es el contador de imágenes que se hicieron, al inicio esta en cero

#mensajes de detalles y de la tecla para detener la ejecución del código
print(f"Capturando {num_images} imágenes cada {intervalo_segundos} segundos.")
print("Presiona 'q' para detener el proceso manualmente.")

while count < num_images: #Este bucle se usa para capturar el número de imágenes que se quieren
    ret, frame = cap.read()

    ...

    Guarda en ret una valor de True o False según se haya capturado o no la imagen,
    es la imagen almacenada como una matriz que contiene los valores de color de cada pixel
    ...

    if not ret:
        print("Error, no se pudo leer el frame")
        break

    #Son las direcciones de donde se guardará el archivo, incluye hasta definir el nombre del archivo
    image_path = os.path.join(output_folder, f"imagen_{count:04d}.jpg") #une la carpeta donde se guardara el archivo con el nombre del archivo según la cuenta
    cv2.imwrite(image_path, frame)

    #es una función de cv2, guarda la imagen en un archivo, el primer argumento es la ruta y el segundo es la imagen como matriz de colores de cada pixel

    #Muestra una ventana donde se pueden ver las imágenes que son capturadas
    cv2.imshow("Capturando...", frame)
    #El primer argumento es el nombre de la ventana, el segundo es la imagen que se muestra

    count += 1 #Actualiza el valor de la cuenta sumando uno
    ...

    Se usa una condicional en caso se desee detener el proceso de captura,
    sino se espera el tiempo definido antes de volver al inicio del bucle y capturar la siguiente imagen
    ...

    if cv2.waitKey(1) & 0xFF == ord('q'):
        print("Se ha denido la captura de imágenes por presionar q")
        break

    time.sleep(intervalo_segundos) #Es el tiempo antes de tomar la siguiente imagen

print(f"Se capturaron {count} imágenes en la carpeta '{output_folder}'.") #imprime este mensaje para indicar las imágenes capturadas y la carpeta donde se guardo

#libera la cámara y cierra las ventanas abiertas por opencv
cap.release()
cv2.destroyAllWindows()
```

Este código fue diseñado para capturar automáticamente un número específico de imágenes desde la cámara usb conectada a la laptop y guardarlas en una carpeta específica. Primero, se importan los módulos necesarios: cv2 para usar OpenCV, es decir para el procesamiento de imágenes, os para manejar rutas de archivos y carpetas, y time para controlar el intervalo entre capturas. Luego, se define una carpeta de destino output_folder donde se almacenarán las imágenes capturadas. Se crea el objeto cap de la clase VideoCapture con cv2.VideoCapture(0), al hacer esto inicializa la cámara, donde 0 representa la cámara predeterminada, en este caso la laptop la cámara usb es la cámara predeterminada, se verifica si se abrió correctamente con cap.isOpened(), esto es un método de la clase VideoCapture, que se usa para verificar si la conexión con la cámara, devuelve un valor de True en caso de que se haya abierto correctamente la cámara y False en caso contrario, el bloque condicional If a su lado se usará para detener el programa en caso de que dicho método devuelva False, para esto se agrega not, pues al devolver False por falla de conexión se convertirá en True con el not y se ejecutará el bloque If, dentro del bloque If se imprime un mensaje de error y se termina el programa con exit(). Se configuran dos parámetros clave: el número total de imágenes que se desean capturar num_images y el intervalo en segundos entre cada captura intervalo_segundos. Se inicia un bucle while que corre hasta alcanzar la cantidad deseada de imágenes; en cada iteración, se captura un fotograma en frame usando el método cap.read(), si en caso ret recibe un valor False indicador de que la lectura falló, se detiene el programa gracias a la condicional If a continuación, que al recibir el valor False de ret con not se convierte en True y así ejecuta el bloque If para detener el programa. Si es

exitosa, salta ese bloque If, y se guarda la imagen en la carpeta especificada con un nombre progresivo como imagen_0000.jpg, imagen_0001.jpg.... utilizando el dato de count, y guardandoló usando cv2.imwrite(). Además, se muestra la imagen en una ventana con cv2.imshow() para retroalimentación visual en tiempo real. Se usa cv2.waitKey(1) & 0xFF == ord('q') para detectar si el usuario presiona la tecla 'q' y detener la captura manualmente, para poder realizar esto se necesita detectar que se pulso una tecla específica para esto usamos cv2.waitKey(1), una función que nos devuelve un entero decimal que es expresado internamente en 32 bits pero solo los últimos 8 bits corresponden al código ASCII de la tecla presionada que se necesitan comparar con los bits del entero que devuelve ord('q') que pertenecen al código ASCII de la tecla q, por ello se deben eliminar los otros 24 bits iniciales que representan el entero decimal que devuelve cv2.waitKey(1) para ello se hace una operación & bit a bit con 0xFF, un valor entero hexadecimal, el cual es el entero decimal 255 que en binario es 11111111, entonces la operación & pone cero en todos los bits donde ambos no sean 1 eliminando los 24 bits iniciales y manteniendo igual los últimos 8 bits de la tecla que se presiono, así para que ya se compare mediante el operador == si es que ambos enteros son iguales, si es que esto se vuelve True entonces se ejecuta el bloque dentro de If, deteniendo la captura de imágenes. Después de cada imagen, se espera el tiempo especificado con time.sleep() antes de continuar. Finalmente, cuando el contador ya indica que se llevo al número de imágenes que se quería salta el bucle while y se imprime un mensaje con el total de imágenes capturadas, se libera la cámara con el método cap.release() y se cierran las ventanas abiertas por OpenCV con cv2.destroyAllWindows() para limpiar los recursos utilizados.

- **Uso del programa python para la captura de imágenes**

Una vez realizado el programa nos queda realizar la base de imágenes. Para cada uno de los objetos que se quiere detectar se entrenó un total de 500 imágenes ubicados en el espacio de identificación y expuesto a las condiciones de iluminación en las que serán probadas, es muy importante que en las imágenes que se tomen los residuos tienen que ser girados para capturar otros todos los lados y detalles del objeto ya que al ingresar el residuo puede entrar de cualquier lado.

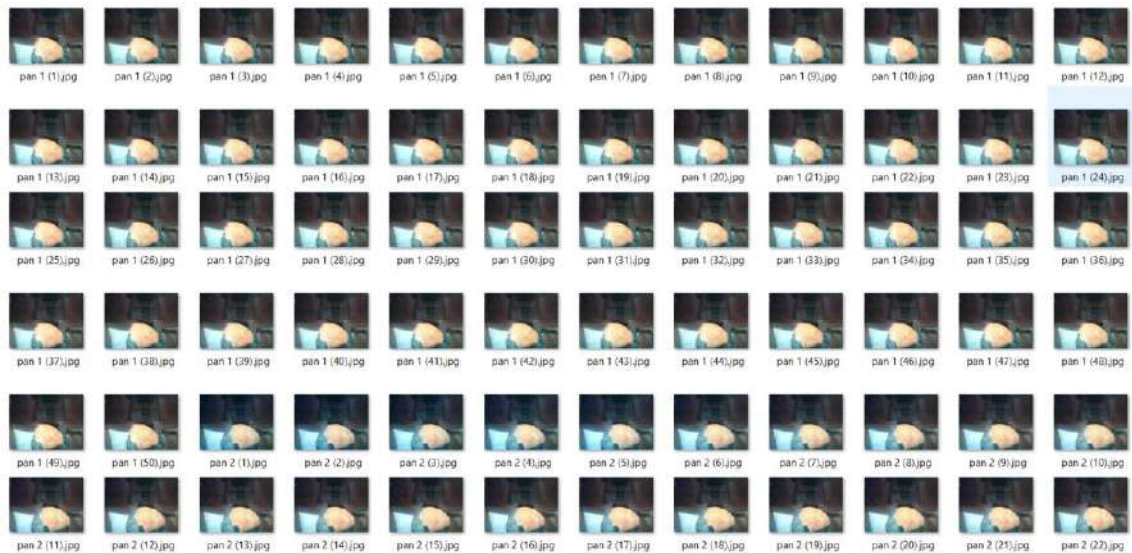
En el presente trabajo como se quiere de mejorar la calidad de segregación mediante un prototipo de una máquina automática mediante visión artificial, la actividad de segregación se llevará acabo dentro del prototipo por ello el objetivo no es lograr una generalización amplia para entornos altamente variables, ya que no se busca la segregación en entornos muy diferentes, se prefiere más enfocarse en la identificación en un escenario controlado, es decir entrenar un modelo robusto y preciso en un escenario predecible y repetitivo, es decir dentro un espacio controlado en el prototipo, donde las condiciones de captura coinciden con las condiciones operativas, por ello capturar múltiples imágenes por ángulos similares permite que el modelo refuerce su capacidad de detección para esas vistas particulares, que son las que encontrará en la práctica.

En el presente trabajo, se optó por capturar 50 imágenes por cada ángulo representativo del objeto con condiciones de iluminación y fondo constantes a la de la práctica hasta llegar a las 500 imágenes. Esta estrategia permite optimizar la eficiencia del entrenamiento, enfocando el aprendizaje del modelo en las condiciones reales de uso del prototipo. Una vez que se tenía un

conjunto de 50 fotos listas de un residuo se cambiaba la ubicación de guardado de las imágenes para que el siguiente conjunto de fotos con los mismos nombres de la primera no las sobre-escribiera, cuando ya se tenían los 10 conjuntos de 50 fotos de un residuo se les cambiaba el nombre de conjunto en conjunto, por ejemplo al primer conjunto todas las imágenes se les cambiaba el nombre a “pan 1” esto causaba que sus nombres empezarán con “pan 1” pero con un numerador de 1 a 50, las imágenes del siguiente conjunto de 50 imágenes era cambiado con el nombre “pan 2” causando el mismo efecto de nombramiento, esto para que al momento de unir todos los conjuntos de 50 fotos en una sola carpeta las imágenes siguieran un orden en cuanto al ángulo en el que fueron tomado y no se desordenaran facilitando la etapa siguiente del etiquetado.

Figura 11

Captura de imágenes de residuos



Las carpetas donde se unirán todas las imágenes de un tipo de residuo llevarán un nombre que las identifique por ejemplo “pan”, de la misma forma se haría para los otros tipos de residuos. Luego de esta etapa se creará una

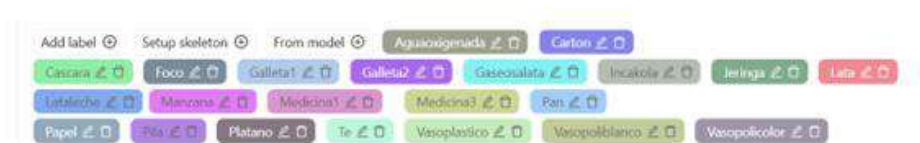
carpeta donde se juntarán las 500 imágenes de cada residuo, teniendo en cuenta que no se desordenarán debido al nombramiento que se les dio. Esta carpeta fue llamada “images” y todo el contenido de esta se subirá a el servicio gratuito de etiquetado manual Computer Vision Annotation Tool (CVAT).

- **Creación de las etiquetas con Computer Vision Annotation Tool (CVAT)**

La etapa del etiquetado es la siguiente etapa a la creación de la base de imágenes, esto debido a que en esta etapa se señalará en que parte de la imagen esta el residuo que deberá identificar para así en el entrenamiento extraiga las características de ese objeto y también para que ponga a prueba el modelo en la etapa de validación usando la zona donde se etiqueto como una comparación para ver si identifico correctamente la zona y el objeto, obteniendo datos como la precisión del modelo. En esta primera parte se deben definir lo que son los labels que se usarán para entrenar el modelo, los labels son las etiquetas o nombres que indican que objeto es el elemento que ha sido indicado según la caja que encierre al residuo en la imagen. Para este trabajo se crearon los siguientes labels en el servicio de etiquetado CVAT:

Figura 12

Labels de los residuos a segregar

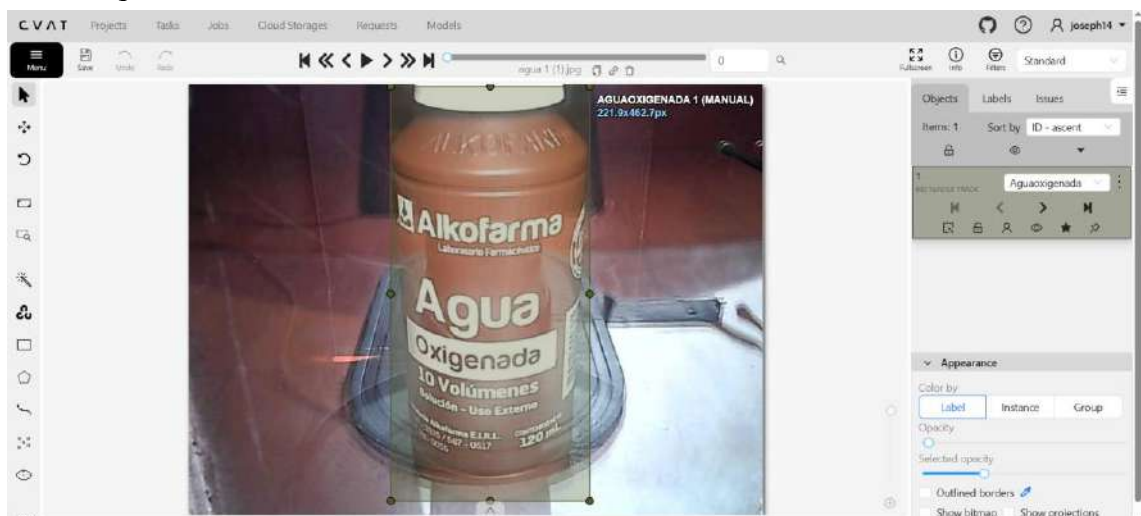


A continuación se realizan los etiquetados con la herramienta box y tracking de CVAT, de esta forma el recuadro que se haga en una imagen se mantendrá en el siguiente para ajustarse en caso el residuo se haya movido a algún lado, en esta etapa es de suma importancia seleccionar lo más que pueda solo el espacio donde se encuentre el residuo para evitar confundir al residuo

con elementos del fondo y así evitar tener menos precisión en la detección, también el orden que tuvimos en el nombramiento de las imágenes nos ayuda muchísimo a ahorrar tiempo en el etiquetado teniendo que ajustar poco los recuadros a medida que avanzamos en las imágenes.

Figura 13

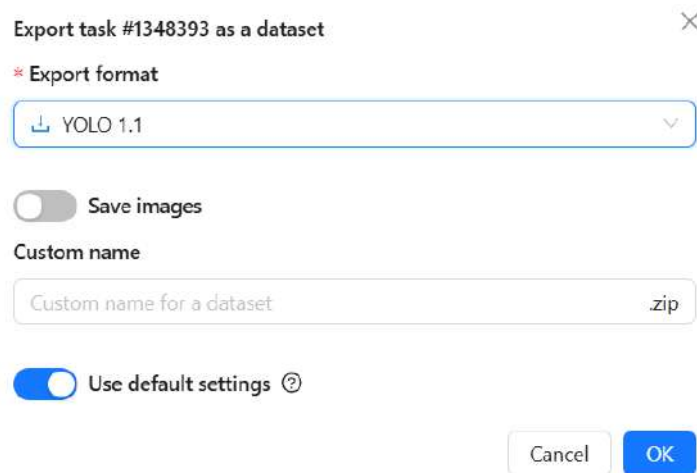
Etiquetado de residuos



Una vez se haya realizado el etiquetado de todos los residuos en las imágenes correctamente, se tiene que exportar los labels con la siguiente función, eligiendo el formato de YOLO 1.1 ya que se usara justamente un modelo YOLO para el re-entrenamiento.

Figura 14

Función de exportar labels en formato YOLO



Cada box que se creo en la carpeta de etiquetado brindará cinco datos al exportarse en formato YOLO, por ejemplo de pan 1 (1).jpg creará un archivo llamado pan 1 (1).txt, este archivo .txt es el label de dicha imagen, el primer dato indica la clase a la que pertenece, el segundo dato indica la coordenada x del centro del bounding box, el tercer dato indica la coordenada y del centro del bounding box, mientras que el cuarto y quinto dato indican el ancho y alto del recuadro que se miden a partir del punto definido por la coordenada x e y. Es muy importante recordar que si el primer dato es cero, eso indica que el label de una imagen fue creada con la primera clase que se creo en CVAT, es decir si la primera clase que se creo fue pan, todos los archivos .txt que contengan cero al inicio indicarán que los recuadros de las imágenes a las que pertenecen esos archivos .txt pertenecerán a la clase o residuo pan.

Figura 15

Ejemplo de label de una imagen de un residuo con clase correspondiente a 12

12 0.614586 0.766135 0.424797 0.467729

Todos los labels que nos haya entregado CVAT los debemos colocar en una carpeta llamada labels. Ahora en esta etapa tendremos una carpeta llamada images con todas las imágenes en dicha carpeta, y una carpeta llamada labels en la cual estarán todos los archivos .txt. El siguiente paso para preparar nuestra base de imágenes es crear un archivo texto llamado “classes.txt” en la cual debemos escribir las clases que creamos en CVAT en el mismo orden en el que se crearon ahí, este paso es muy importante para que en el entrenamiento las imágenes y labels sean relacionadas correctamente con el nombre de clase correcta.

Figura 16

Archivo classes.txt



```
classes.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Aguaoxigenada
Carton
Cascara
Foco
Galleta1
Galleta2
Gaseosalata
Incakola
Jeringa
Lata
Lataleche
Manzana
Medicina1
Medicina3
Pan
Papel
Pila
Platano
Te
Vasoplastico
Vasopoliblanco
Vasopolicolor
```

Como último paso para completar nuestra base de imágenes debemos comprimir las dos carpetas y el archivo classes.txt en un archivo .zip que llamaremos data.

Figura 17

Lista de archivos para data.zip

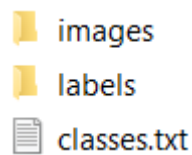
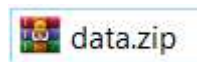


Figura 18

Archivo data.zip



De esta forma ya habremos completado nuestra base de imágenes con sus respectivos labels y su archivo de texto classes.txt para comenzar con el re-entrenamiento del modelo YOLOV11.

- **Re-entrenamiento del Modelo convolucional YOLOV11**

Para poder comprender lo que se hace en esta parte se debe tener claro ciertos conceptos relacionado al campo de la informática.

- **Inteligencia Artificial**

Es una rama de la informática que busca desarrollar sistemas capaces de realizar tareas que requieren inteligencia humana, es decir que las máquinas simulen lo que es el aprendizaje, comprensión, la resolución de problemas y la toma de decisiones de la capacidad humana. (Russell, S., & Norvig, 2020).

- **Machine Learning**

Dentro de la inteligencia artificial, el machine learning se enfoca en permitir que los computadoras "aprendan" o imiten la forma en la

que aprende un humano a partir de datos, mejorando su rendimiento con la experiencia (Mitchell, 1997).

- **Deep Learning**

Es una subcategoría del machine learning en la cual se utiliza redes neuronales multicapa, que son llamadas también redes neuronales profundas y todas en conjunto se usan en la simulación del complejo poder de toma de decisiones del humano (Goodfellow et al., 2016).

- **Red neuronal regular y su entrenamiento**

Fueron el primer modelo matemático de una red neuronal multicapa también llamada perceptrón multicapa, se inspiró en la estructura del cerebro humano, específicamente en el funcionamiento de las neuronas biológicas. Su propósito principal es aprender patrones a partir de datos para realizar tareas como segregación. Está compuesta por una serie de capas de neuronas, organizadas en una arquitectura que incluye una capa de entrada, una o más capas ocultas y una capa de salida. El entrenamiento de una red neuronal convolucional se basa en parte al de uno regular, por eso primero se tiene que entender como se entrena un modelo regular y cuales son sus desventajas para usar uno convolucional (Goodfellow et al., 2016).

Todo comienza con una red neuronal regular, cuyo propósito es aprender patrones en los datos y hacer predicciones. Si queremos entrenar esta red para clasificar tipos de residuos, necesitamos transformar cada imagen de entrada en una secuencia de números, estos son los valores de entrada, por ejemplo, si la imagen tiene 28×28 píxeles, se aplanan en un vector de 784 números, donde cada número

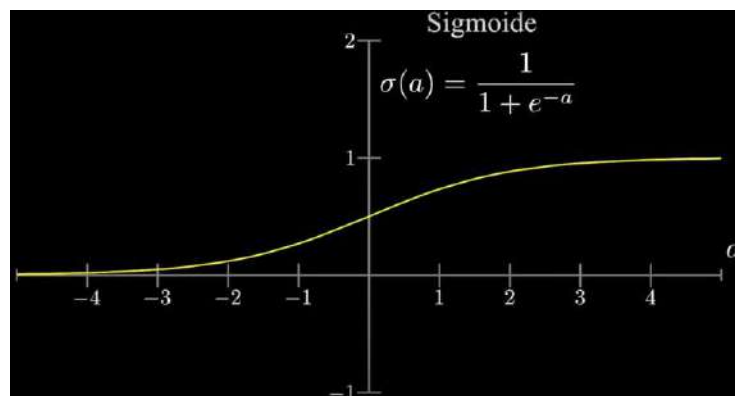
representa la intensidad del color de un píxel, estas formarán las neuronas de entrada. Cada neurona de la primera capa oculta esta conectada a todas las entradas, y cada una de esas conexiones tiene un número asignado llamado peso, que indica la importancia relativa de esa entrada para esa neurona. Para cada neurona de esta capa oculta, se realiza una operación matemática llamada suma ponderada, en la cual cada entrada se multiplica por su peso correspondiente, y todos estos productos se suman. A esta suma se le añade un sesgo, que es un valor constante que permite que la neurona se active incluso si la entrada no es muy fuerte, proporcionando más flexibilidad. Esta suma total se llama z y se expresa como:

$$z = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_n \cdot w_n + b$$

Pero z por sí sola no determina la salida, esto se pasa por una función de activación, en los inicios se usaba la función sigmoide para aplastar la suma ponderada y poder llevarla a un valor entre cero y uno motivadas por la analogía biológica ser ser activas o inactivas.

Figura 19

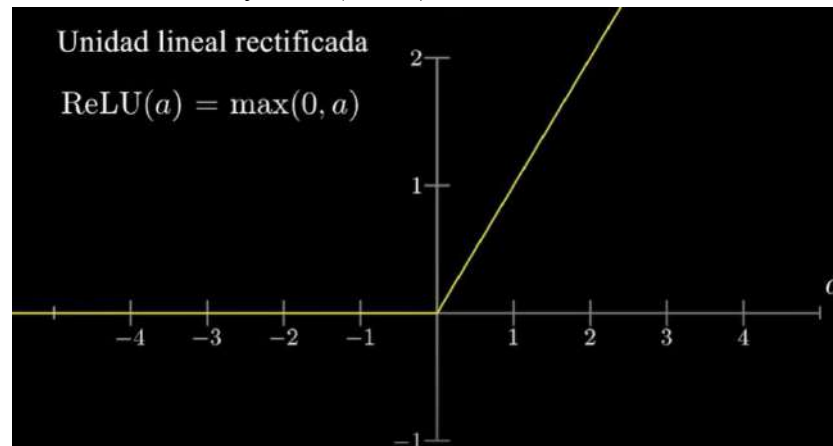
Función sigmoide



Sin embargo pocas de las redes modernas ya no usan esta función de activación debido a que se usa otra función de activación llamada ReLU (Unidad lineal rectificada) con la que es más fácil de entrenar.

Figura 20

Función Unidad lineal rectificada (ReLU)



En esta función ReLU toma el máximo valor entre cero y un valor a , donde a es igual a la suma ponderada z , es decir si a es un número negativo la salida de la función será siempre cero, cuando a es cero la salida de la función es cero y cuando a es positiva la salida de la función es dicho valor positivo, esto también fue motivado parcialmente por una analogía biológica de como las neuronas estarían o no activadas y si están pasan un cierto límite la función ReLU sería igual a la función identidad, es decir que la entrada a en la función sería también la salida, pero si no pasa cierto límite, simplemente no serían activadas y serían cero, sería como una simplificación. Sobre el uso de sigmoides se sabe que no ayudó en los entrenamientos pues fue muy difícil usarlos en cierto punto de modelos profundos por como estaba definida y causaba un aprendizaje lento, es cuando el uso de la función

ReLU por como esta definida funciona mejor para redes neuronales increíblemente profundas.

Esta función toma a z como el valor a , si z es menor que 0, se convierte en 0, si es cero la salida también lo será, si es mayor que 0, se mantendrá dicho valor positivo como salida. El resultado es la activación de la neurona, que se transmite a la siguiente capa. A medida que la información avanza por las capas ocultas, se repiten estas operaciones de suma ponderada, adición de sesgo, función de activación una y otra vez. Las capas más profundas aprenden combinaciones más complejas de los datos: desde patrones simples hasta características más abstractas. Finalmente, en la capa de salida, la red da una activación por cada clase posible, es decir por ejemplo, cuatro valores si clasificamos en cuatro tipos de basura. Estos valores se interpretan como probabilidades entre 0 y 1 después de aplicar la función softmax. Softmax recibe las salidas de la última capa de la red neuronal antes de clasificarse, y transforma esos números en probabilidades que estén entre 0 y 1. La suma de todas esas probabilidades será exactamente 1. De esa forma se indica cuanta confianza tiene el modelo en que una imagen pertenece a cada clase.

En la siguiente imagen se ve la definición de la función softmax la cual toma las salidas de la última capa de la red.

Figura 21

Función Softmax

$$\text{softmax}(a_i) = \frac{e^{a_i}}{\sum_{j=1}^n e^{a_j}}$$

$$\mathbf{a} = [a_1, a_2, \dots, a_n]$$

Para que un modelo tenga una alta precisión debe ajustar los pesos y sesgos a valores adecuados, esto ocurre durante el proceso de entrenamiento.

El modelo necesita ver muchas imágenes, muchas veces, para ir ajustando sus pesos y sesgos y llegar a un modelo que pueda generalizar los resultados correctamente, a esto se le llama entrenar por épocas. Una época es una pasada completa por todo el conjunto de datos de entrenamiento, justamente una de las carpetas en las que se encuentran todas las imágenes de los residuos. Durante cada época se divide el conjunto de datos en mini lotes, por ejemplo de 32 o 64 imágenes a la vez, para luego realizar una predicción para cada imagen del lote, esto es una propagación hacia adelante, lo que hace es tomar una de las imágenes y predecir una clase con los pesos actuales para luego calcular el error comparando la predicción con la clase correcta indicada por la etiqueta mediante una función de costo o pérdida. Si el modelo se equivoca, el error será alto, entonces ocurre la propagación hacia atrás, donde se calcula cómo cada peso y sesgo contribuyó al error, para eso se usa el gradiente del error con respecto a cada peso y sesgo, lo que requiere aplicar la regla de la cadena de derivación, el

gradiente es un vector de derivadas parciales que indica en qué dirección y con qué fuerza debe ajustarse cada parámetro para reducir el error. Luego, se promedian todos esos gradientes, uno por cada imagen del mini lote, y se ajustan los pesos usando ese promedio para que el modelo pueda llegar en sus predicciones a reconocer la mayor cantidad de objetos. Esto permite que la red no se vea afectada por errores puntuales de una sola imagen, sino que ajuste sus parámetros en función de un grupo más representativo. Esto lo realiza hasta completar el número de imágenes de entrenamiento y finalizar la primera época. Después de entrenar con todos los minibatches de la primera época continúa con la evaluación del rendimiento del modelo, es decir evalúa el modelo con las imágenes de validación, para esto se calcula el error y/o la precisión. Esto permite ver cómo de bien está generalizando el modelo a imágenes que no ha visto durante el entrenamiento. Posteriormente de esto se inicia nuevamente el entrenamiento, pero ya se estaría en la segunda época, esto se realiza hasta llegar a la última época o sino gracias a la evaluación al final de cada época se detecta si el modelo está aprendiendo de verdad o si está empezando a sobreajustarse, de otra manera se podría entender como que esta memorizando las imágenes lo cual es diferente estar captando los patrones generales para aplicarlos correctamente a datos nuevos que nunca ha visto antes, sin embargo el impacto de este problema se reduce en casos donde los objetos que se detectarán siempre tendrán las mismas características y estarán en las mismas condiciones ambientales, como en el caso de algunos residuos con envolturas con

formas similares pero con diseños y colores particulares donde el sobreajuste no sería tan malo para el propósito del trabajo. La evaluación al final de cada época también permite guardar una copia del modelo si es el mejor resultado hasta el momento o también puede usarse para detener el entrenamiento si ya no hay mejoras. Sin embargo como se dijo al inicio también se presentaban ciertas desventajas del uso de redes regulares cuando se quiere detectar objetos, una de ellas es que las imágenes deben ser pequeñas, simples y bien alineadas porque en una red neuronal regular, cada neurona de entrada está conectada a todas las neuronas de la siguiente capa. Esto significa que el número de parámetros crece rápidamente con el tamaño de la imagen. Como se dijo una imagen de 28×28 píxeles tiene 784 entradas. Si la primera capa tiene 128 neuronas, necesitas $784 \times 128 = 100,352$ pesos solo en la primera capa. Una imagen de $224 \times 224 \times 3$ con tres canales de colores, RGB, son 150,528 entradas, para 128 neuronas tan solo de la primera capa la cantidad de pesos a ajustar es gigantesca:

$$150,528 \times 128 = 19.3 \text{ millones de pesos}$$

Por lo que el modelo sería computacionalmente inmanejable y propenso al sobreajuste. También las imágenes tienen que ser simples y tanto las imágenes con las que se entrenará y probará tendrán que tener los objetos posiciones similares porque este tipo de red regular no aprende características locales como bordes, texturas o patrones visuales, solo ve una combinación global de píxeles. Por eso, si los datos tienen muchos detalles, formas complejas o fondos variados, la red no puede generalizar correctamente y no es invariante a la

traslación, ni a la escala, ni a la rotación. Es decir, si se entrena la red con una imagen donde el objeto está centrado, y luego pruebas con el mismo objeto movido hacia un lado, la red fallará (Goodfellow et al., 2016).

Por ello las redes neuronales convolucionales son el siguiente punto ya que están diseñadas específicamente para datos con estructura espacial, como las imágenes. Son en parte basados en las redes regulares pero el uso de capas convolucionales es lo que la define, evitando las desventajas de las redes neuronales regulares.

- **Red neuronal convolucional y su entrenamiento**

Una red neuronal convolucional (CNN) es adecuada para la detección de objetos porque está diseñada para detectar patrones visuales jerárquicos directamente desde los píxeles de la imagen, sin necesidad de extracción manual de características. Además, gracias a operaciones como el Maxpooling, esta red logra cierta invariancia a traslaciones y escalas, haciendo que reconozca un objeto aunque aparezca en distintas posiciones o tamaños. A diferencia de una red densa, que trata cada pixel como una entrada independiente, la red neuronal convolucional aprovecha la estructura espacial de las imágenes y reduce significativamente la cantidad de parámetros, lo que permite entrenarla de forma más eficiente y generalizar mejor.

Una capa convolucional es el bloque básico de este tipo de red que mediante pequeños filtros o también llamados kernels, los cuales son ajustables de tamaño como por ejemplo 3×3 , escanea localmente un

volumen de entrada los cuales pueden ser la imagen de entrada o los mapas de activación, multiplicando cada bloque de píxeles por los pesos del filtro y sumando el resultado más un sesgo, de modo que genera un mapa de activación 2D donde cada valor indica la presencia de un patrón específico como bordes, texturas, etc.

Figura 22

Resultado de la aplicación de un filtro para detectar bordes



Posteriormente a esta etapa, se aplica ReLU (Unidad lineal rectificada), que introduce no linealidad al modelo esto debido a que las imágenes contienen relaciones complejas como curvas, bordes, texturas y formas compuestas. Cada vez que se aplica una ReLU en una capa, se crea una "esquina" en el espacio matemático, una decisión del tipo "¿esto me importa o no?", que se activa con base en los datos. Con miles de estas decisiones apiladas, el modelo puede representar funciones muy complejas, como "esto es una tapa de botella vista desde un ángulo", esto lo logra al convertir a cero todas las activaciones negativas y dejar pasar solo las positivas, posteriormente a esto MaxPooling toma bloques contiguos como por ejemplo de 2×2 para quedarse con el valor más alto de cada uno, reduciendo la resolución espacial y aportando invariancia a

pequeñas traslaciones. Los mapas de activación resultantes se apilan en un tensor 3D, en el cual las tres dimensiones representan altura \times ancho \times número de filtros, y sirven de entrada a la siguiente capa convolucional, cuyos filtros ahora de dimensiones de $3 \times 3 \times$ número de mapas de activación entrantes combinan las características previas para detectar un patrón mayor, como pueden ser tapas, logos etc. Con mas detalle se realiza al combinar las características de zonas de los mapas de activación entrantes y arrojando un número alto si es que se encontro una determinada combinación de características.

Se dan sucesivas capas que repiten el ciclo Convolución, ReLU, Pooling para formar primero bordes, luego formas geométricas, después partes de objetos y finalmente siluetas completas.

Una vez extraídas todas las características jerárquicas como por ejemplo, un tensor final de $4 \times 4 \times 256$, se aplica Flatten, que reorganiza ese volumen en un vector 1D como por ejemplo 4 096 elementos, cada uno representando la activación de un patrón visual aprendido, este vector alimenta entonces las capas densas donde cada neurona realiza una suma ponderada de todas las entradas más un sesgo y aplica de nuevo una función de activación ReLU para combinar esas características en representaciones aún más abstractas, hasta llegar a la capa de salida que, mediante softmax, convierte sus activaciones en probabilidades de clase normalizadas que suman 1. Durante el calculo de la salida para una entrada en el entrenamiento el modelo genera la predicción y se calcula la función de costo, y en la retropropagación se computan los gradientes $\partial \text{Error} / \partial \text{peso}$

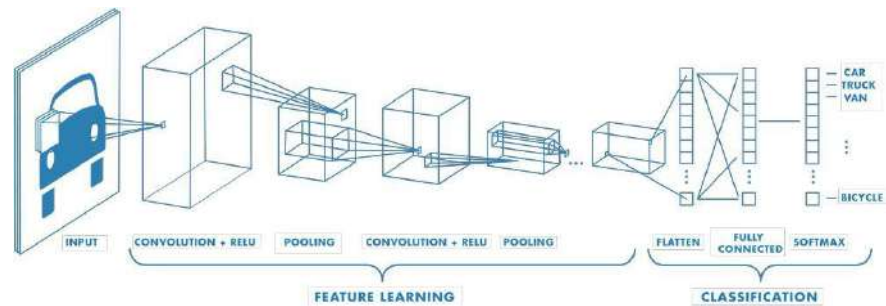
para cada filtro y conexión densa que indican cuánto cambiaría el error al modificar ligeramente ese peso, se promedian dentro de cada mini lote y se actualizan los pesos y sesgos en la dirección en que se reduce el error, este ciclo se repite en muchas épocas.

El funcionamiento de una modelo ya entrenado se podría entender que a la salida de la parte convolucional esos valores representan características complejas, es decir el modelo al tener la imagen de un perro en la salida de la parte convolucional y entrada a la red densa, los valores se podrían interpretar como “tengo una fuerte presencia de orejas puntiagudas, cuerpo alargado, textura peluda y hocico... parece un perro”. Esta parte es la que ya realiza la parte densa a partir de las características particulares que se activaron.

Si, en lugar de entrenar desde cero, se parte de un modelo de red neuronal pre-entrenado como YOLOV11 para detección de objetos de una base de imágenes como Imagenet, se congelan las capas iniciales pues ya capturan bordes y texturas de un banco de imágenes mayor y se reemplazan las últimas capas convolucionales y densas para adaptarlas a los objetos que se quieren detectar para luego mediante fine-tuning con las imágenes de la nueva base de imágenes, la retropropagación ajusta únicamente los filtros y pesos finales logrando re-entrenar el modelo para objetos nuevos.

Figura 23

Estructura de una red neuronal convolucional



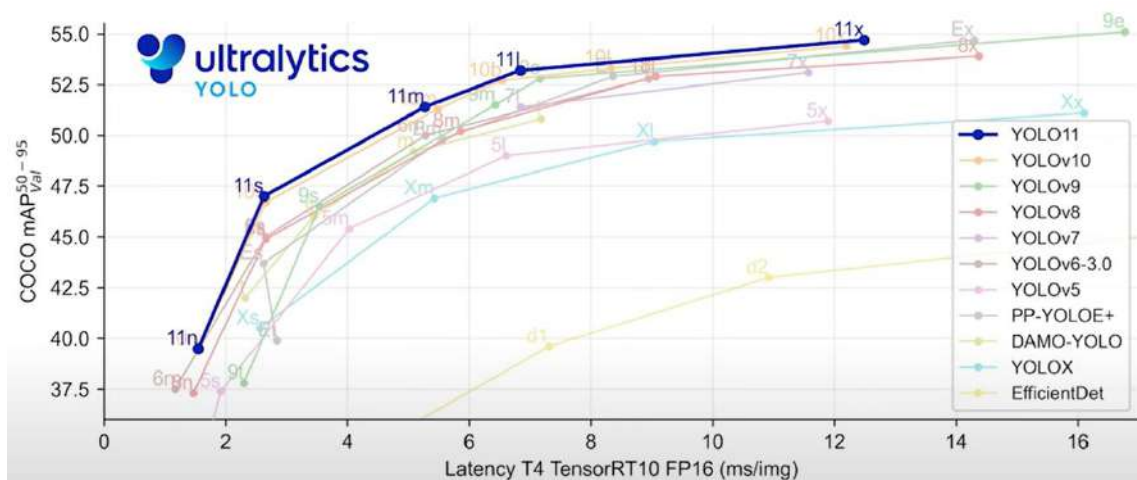
Para este trabajo se usó un cuaderno público de Ultralytics en Google Colab para el re-entrenamiento del modelo de red neuronal convolucional YOLOV11s. Esto se debe a que este servicio es muy útil cuando no se dispone de una computadora potente con GPU. Su principal ventaja es que ofrece acceso gratuito a aceleradores de hardware como GPUs o incluso TPUs, lo que acelera considerablemente el proceso de entrenamiento, que en una CPU convencional sería lento y poco práctico. Además, Colab ya viene con un entorno preconfigurado que incluye Python que se puede usar con Ultralytics, la cual es la organización detrás de las versiones modernas de YOLO que han sido rediseñadas para ser más accesibles, modulares y compatibles con entornos como Google Colab.

Para este trabajo es necesario cargar el archivo data.zip en un drive el cual se conectará al cuaderno de Google Colab para el re-entrenamiento del modelo YOLO. Se inició la prueba con el modelo YOLOV11s (small) el cual es un modelo ya entrenado con la base de imágenes COCO (Common Objects in Context) el cual fue entrenado con más de 118 000 objetos de 80 clases, es ideal para aplicaciones en tiempo real de segregación de residuos mediante visión artificial porque ya que aprendió

patrones generales del objetos reales, también ofrece un equilibrio entre velocidad y precisión según la figura 24. Con una latencia de solo 2 ms por imagen y un mAP cercano al 48%, supera a los modelos pequeños anteriores.

Figura 24

YOLOv11: Precisión (mAP_{val}^{50-95}) vs. Latencia (ms/img) en COCO

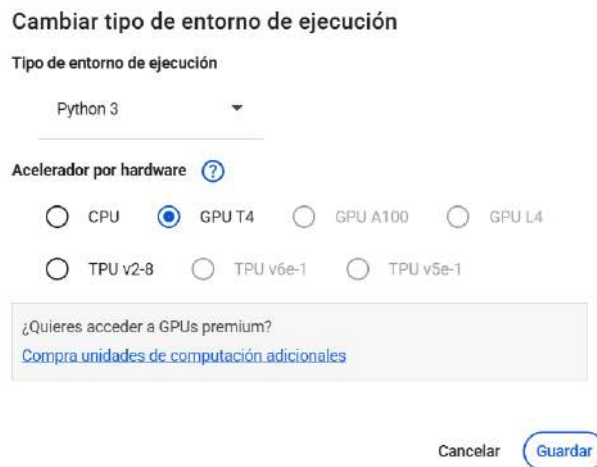


COCO mAPVal 50–95 es una métrica de rendimiento integral, no una simple exactitud. Corresponde a la mAP (mean Average Precision), una puntuación que evalúa la capacidad del modelo para minimizar tanto los falsos positivos como los falsos negativos dentro del exigente y complejo dataset COCO en su conjunto de validación (Val). El aspecto más crítico de esta métrica es el 50-95, que indica que la puntuación final es el promedio de 10 evaluaciones con umbrales de "Intersection over Union" (IoU) crecientemente estrictos, desde una superposición del 50% hasta una casi perfecta del 95%. Debido a que este método penaliza severamente cualquier imprecisión en la localización del objeto, un valor de 47.5 demuestra un rendimiento de alta fiabilidad en un amplio espectro de exigencia. Esta robusta precisión se complementa con una latencia de inferencia de aproximadamente 2.5 milisegundos por imagen, una velocidad extrema medida en un entorno de producción optimizado, lo que confirma su idoneidad para la tarea de detección de residuos en tiempo real que demandan tanto una respuesta instantánea como una correcta detección.

Lo primero es iniciar el entorno de ejecución python en el cuaderno de Google Colab como se puede observar en la siguiente imagen, y seleccionar la tarjeta gráfica:

Figura 25

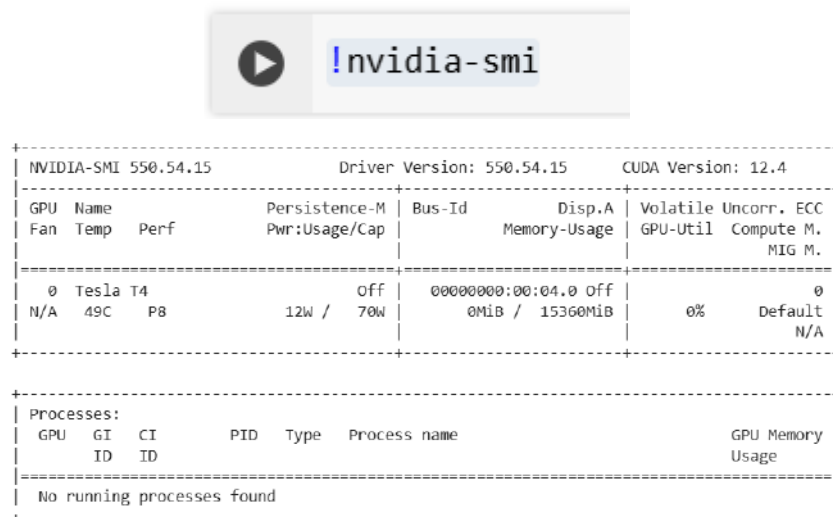
Inicio del entorno de ejecución



Una vez seleccionado ejecutamos la primera instrucción la cual verifica la disponibilidad de la tarjeta gráfica.

Figura 26

Verificación de la disponibilidad de GPU



Posteriormente se tiene que cargar el data.zip a una carpeta de drive, la carpeta se llamo “Hola” guardada en mi almacenamiento drive. Una vez que ya este guardado se procede a ejecutar la siguiente instrucción la cual carga desde Drive el archivo comprimido en el

sistema de archivos de Colab, para esto se debe definir la ruta en la que esta la carpeta que contiene el archivo data.zip.

Figura 27

Cargar en el sistema de archivo de Colab el archivo comprimido data.zip

```
[ ] from google.colab import drive
    drive.mount('/content/gdrive')

!cp /content/gdrive/MyDrive/Hola/data.zip /content
```

A continuación, con la siguiente instrucción se descomprime el archivo data.zip.

Figura 28

Descompresión del archivo data.zip

```
[ ] !unzip -q /content/data.zip -d /content/custom_data
```

Ultralytics requiere una estructura de carpetas específica para almacenar los datos de entrenamiento de los modelos. La carpeta raíz se llamará "data". Dentro, habrá dos carpetas principales:

Train: Contiene las imágenes utilizadas para entrenar el modelo. En una época de entrenamiento, cada imagen del conjunto de entrenamiento se pasa a la red neuronal. El algoritmo de entrenamiento ajusta los pesos de la red para que se ajusten a los datos de las imágenes.

Validation: Estas imágenes se utilizan para comprobar el rendimiento del modelo al final de cada época de entrenamiento.

En cada una de estas carpetas hay una carpeta "images" y una carpeta "labels", que contienen los archivos de imagen y los archivos de anotación, respectivamente.

La siguiente instrucción crea automáticamente la estructura de carpetas necesarias para que Ultralytics realice el entrenamiento correctamente, esta instrucción mueve aleatoriamente el 90 % del conjunto de datos a la carpeta "train" y el 10 % a la carpeta "validation".

Figura 29

Creación de la carpeta train y validation

```
[ ] !wget -O /content/train_val_split.py https://raw.githubusercontent.com/EdjeElectronics/Train-and-Deploy-YOLO-Models/refs/heads/main/Utils/train_val_split.py  
!python train_val_split.py --datapath="/content/custom_data" --train_pct=0.9
```

Posteriormente se ejecuta esta instrucción para instalar la biblioteca Ultralytics en esta instancia de Google Colab. Esta biblioteca de Python se usará para entrenar el modelo YOLO.

Figura 30

Instalación de la biblioteca Ultralytics

```
[ ] !pip install ultralytics
```

Antes de ejecutar el entrenamiento se debe crear el archivo YAML de configuración de entrenamiento de Ultralytics. Se debe ejecutar el siguiente bloque de código que especifica la ubicación de los datos de entrenamiento y validación, y también define las clases del modelo. El código genera automáticamente un archivo de configuración data.yaml mediante el uso del archivo classes.txt de data.zip.

Figura 31

Creación del archivo *data.yaml*

```
#Función de Python para crear automáticamente el archivo de configuración data.yaml
# 1. Lee el archivo "classes.txt" para obtener la lista de nombres de clases
# 2. Crea un diccionario de datos con las rutas correctas a las carpetas, el número de clases y sus nombres
# 3. Escribe datos en formato YAML en data.yaml

import yaml
import os

def create_data_yaml(path_to_classes_txt, path_to_data_yaml):

    # Leer class.txt para obtener los nombres de las clases
    if not os.path.exists(path_to_classes_txt):
        print(f'classes.txt file not found! Please create a classes.txt labelmap and move it to {path_to_classes_txt}')
        return
    with open(path_to_classes_txt, 'r') as f:
        classes = []
        for line in f.readlines():
            if len(line.strip()) == 0: continue
            classes.append(line.strip())
        number_of_classes = len(classes)

    # Crear diccionario de datos
    data = {
        'path': '/content/data',
        'train': 'train/images',
        'val': 'validation/images',
        'nc': number_of_classes,
        'names': classes
    }

    # Escribir datos en un archivo YAML
    with open(path_to_data_yaml, 'w') as f:
        yaml.dump(data, f, sort_keys=False)
        print(f'Created config file at {path_to_data_yaml}')

    return

# Define la ruta a classes.txt y ejecuta la función
path_to_classes_txt = '/content/custom_data/classes.txt'
path_to_data_yaml = '/content/data.yaml'

create_data_yaml(path_to_classes_txt, path_to_data_yaml)

print('\nFile contents:\n')
!cat /content/data.yaml
```

Ahora que los datos están organizados y el archivo de configuración creado, ya se puede entrenar, se tiene que definir el modelo que se usará, en este caso se usará yolov11s.pt. Luego se tiene que definir el número de épocas. Como se sabe una "época" es una sola pasada por todo el conjunto de datos de entrenamiento. La configuración del número de épocas determina la duración del

entrenamiento del modelo. En este trabajo el dataset cuenta con 11000 imágenes y se eligió una cantidad de 50 épocas para empezar. Se eligió entrenar por 50 épocas porque permite al modelo aprender en profundidad los patrones del dataset. Además los modelos pequeños necesitan más repeticiones para alcanzar el máximo de su capacidad, ya que tienen menos parámetros que modelos grandes como YOLOv11x. Otro parámetro a tener en cuenta es la resolución porque tiene un gran impacto en la velocidad y la precisión del modelo, pues un modelo de menor resolución tendrá mayor velocidad, pero menor precisión. Los modelos YOLO se suelen entrenar e inferir con una resolución de 640x640, la cual es la resolución tomada para este trabajo.

Figura 32

Instrucción para iniciar el entrenamiento

```
[ ] !yolo detect train data=/content/data.yaml model=yolo11s.pt epochs=50 imgsz=640
```

El algoritmo de entrenamiento analizará las imágenes en los directorios de entrenamiento y validación y luego comenzará a entrenar el modelo. Al final de cada época de entrenamiento, el programa ejecuta el modelo en el conjunto de datos de validación e informa datos como el mAP la cual a medida que continúa el entrenamiento debería aumentar con cada época. El entrenamiento finalizará una vez que alcance el número de épocas especificado. Los mejores pesos del modelo entrenado serán guardados junto a un archivo results.png que muestra cómo progresaron la pérdida, la precisión, la recuperación y el mAP en cada época. Para este trabajo el

entrenamiento y validación que se realizó tardo tres horas y media en completarse.

Una vez que se completo el entrenamiento y validación podemos probar el modelo, la siguiente instrucción ejecuta el modelo en las imágenes de la carpeta de validación y muestran los resultados de las primeras 10 imágenes dibujando el recuadro donde predeciria que esta el objeto junto con la clase de este.

Figura 33

Prueba del modelo

```
[ ] !yolo detect predict model=runs/detect/train/weights/best.pt source=data/validation/images save=True
```

```
[ ] import glob
from IPython.display import Image, display
for image_path in glob.glob(f'/content/runs/detect/predict/*.jpg')[:10]:
    display(Image(filename=image_path, height=400))
    print('\n')
```

Figura 34

Resultados de prueba del modelo



Ahora que el modelo personalizado ha sido entrenado, está listo para descargarse comprimido e implementarse mediante la siguiente instrucción. El código crea una carpeta llamada `my_model`, transfiere los pesos del modelo a ella y cambia el nombre de `best.pt` a `my_model.pt`, también agrega los resultados del entrenamiento.

Figura 35

Descarga del modelo entrenado

```
[ ] # Cree la carpeta "my_model" para almacenar los pesos del modelo y los resultados del entrenamiento
!mkdir /content/my_model
!cp /content/runs/detect/train/weights/best.pt /content/my_model/my_model.pt
!cp -r /content/runs/detect/train /content/my_model

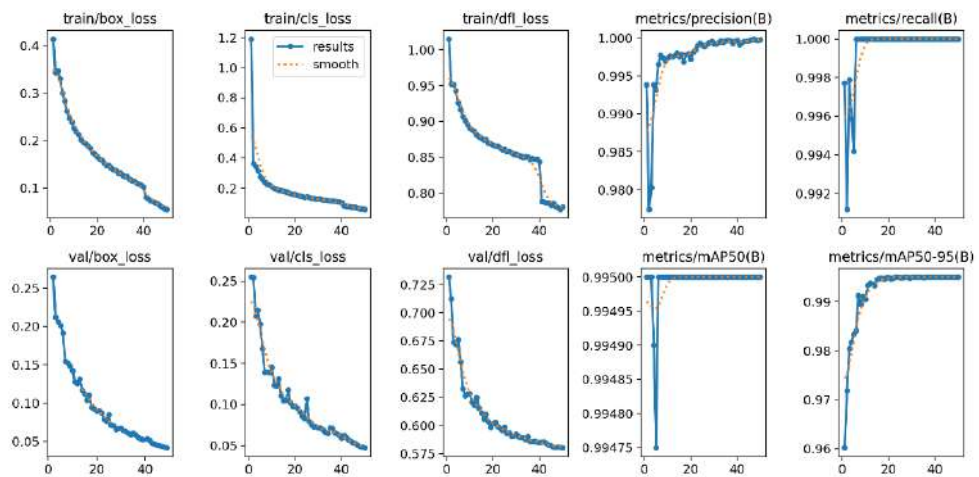
# Comprimir en "my_model.zip"
%cd my_model
!zip /content/my_model.zip my_model.pt
!zip -r /content/my_model.zip train
%cd /content
```

```
[ ] from google.colab import files

files.download('/content/my_model.zip')
```

Figura 36

Resultados del entrenamiento



Las métricas de entrenamiento o train muestran cómo se comporta el modelo con las imágenes de residuos que está utilizando para aprender en cada época asu vez aquí estan los indicadores del subsistema de procesamiento y decisión y otras métricas:

- `train/box_loss` (Pérdida de la Caja Delimitadora): Mide qué tan bien el modelo predice las coordenadas y el tamaño de la caja que encierra un residuo. Un valor más bajo significa que las cajas predichas están más cerca de las cajas reales.
- `train/cls_loss` (Pérdida de Clasificación): Mide qué tan bien el modelo identifica la categoría correcta del residuo dentro de la caja. Un valor bajo indica que el modelo clasifica los residuos correctamente.
- `train/dfloss` (Pérdida Focal de Distribución): Es un tipo específico y más avanzado de pérdida para la ubicación de la caja delimitadora, común en modelos recientes como YOLO. Ayuda al modelo a aprender la distribución de posibles ubicaciones para la caja, mejorando la precisión.
- `metrics/precision(B)` (Precisión): Esta puede responder a la siguiente pregunta, de todas las detecciones que hizo el modelo, ¿qué porcentaje fue correcto? Una precisión alta cercana a 1 significa que el modelo no suele dar falsos positivos.
- `metrics/recall(B)` (Exhaustividad o Sensibilidad): Esta también puede responder a la pregunta, de todos los residuos que realmente existían en las imágenes, ¿qué porcentaje fue capaz de encontrar el modelo? Un valor alto cercano a 1 indica que el modelo encuentra la mayoría de los residuos presentes y no se deja muchos sin detectar es decir pocos falsos negativos.

Las métricas de validación muestran el rendimiento del modelo en datos nuevos, lo que indica su capacidad para generalizar.

- `val/box_loss` (Pérdida de Caja en Validación): Igual que la de entrenamiento, pero calculada sobre el conjunto de validación.
- `val/cls_loss` (Pérdida de Clasificación en Validación): Igual que la de entrenamiento, pero calculada sobre el conjunto de validación.
- `val/dfl_loss` (Pérdida Focal de Distribución en Validación): Igual que la de entrenamiento, pero calculada sobre el conjunto de validación.
- `metrics/mAP50(B)` (Precisión Media Promedio con IoU > 0.50): Es una de las métricas más importantes en detección de objetos pues mide la precisión promedio del modelo considerando una detección como correcta si su caja delimitadora se superpone en al menos un 50% con la caja real. Un valor cercano a 1.0 es excelente.
- `metrics/mAP50-95(B)` (Precisión Media Promedio con IoU de 0.50 a 0.95): Esta es una métrica más estricta, pues mide la precisión promedio del modelo en diferentes umbrales de superposición (IoU), desde el 50% hasta el 95%. Para obtener una puntuación alta aquí, el modelo debe predecir cajas delimitadoras extremadamente precisas.

Basándome en las gráficas, puedo concluir lo siguiente sobre el entrenamiento del modelo:

- Todas las gráficas de pérdida (box_loss, cls_loss, dfl_loss), tanto en entrenamiento como en validación, muestran una clara tendencia a la baja. Esto representa que el modelo está aprendiendo correctamente, mejorando progresivamente en la localización y clasificación de residuos.
- No hay sobreajuste (Overfitting): Las curvas de pérdida de validación siguen de cerca a las curvas de pérdida de entrenamiento, en caso el modelo estuviera sobreajustado, veríamos que las pérdidas de validación empezarían a subir mientras las de entrenamiento siguen bajando. Como ambas disminuyen, indica que el modelo generaliza bien a datos nuevos.
- La precisión (precision) y la exhaustividad (recall) alcanzan rápidamente valores cercanos a 1.0, lo que significa que el modelo es muy fiable y detecta casi todos los residuos reales y casi nunca se equivoca.
- El mAP50 se estabiliza en un valor muy alto (aproximadamente 0.995), lo que indica una calidad de detección sobresaliente con un criterio estándar.
- El mAP50-95 también alcanza un valor excelente (superior a 0.99), lo que demuestra que el modelo no solo detecta los residuos, sino que lo hace con una precisión de localización muy elevada.

Con esta gráfica se puede concluir que el modelo convolucional resultante es robusto, preciso y generaliza bien, logrando un rendimiento de muy alto en la tarea de detección de residuos.

- **Programación en python del sub-sistema de procesamiento y decisión**

Teniendo el modelo convolucional listo para integrarse esta parte detalla la programación en python del sub-sistema de procesamiento y decisión que se encarga de:

- Integrar el modelo re-entrenado guardado en formato pytorch (.pt) que contiene los pesos ajustados del modelo y la estructura de este para la detección de residuos.
- Programar el envío y recibo de señales con la placa Arduino Uno mediante comunicación serial para coordinar las actividades de los componentes que estan conectados a dicha placa y la detección de residuos en python.
- Registrar cada detección en un archivo Excel, generando automáticamente gráficos de barras que permiten un análisis posterior del tipo y cantidad de residuos segregados a lo largo del tiempo.

Figura 37

Programación del sub-sistema de procesamiento y decisión parte 1

```
import os
import sys
import time
import cv2
import numpy as np
import serial
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
from io import BytesIO
from opencvxl import load_workbook
from opencvxl.drawing.image import Image as XLImage
from PIL import Image as PILImage
from ultralytics import YOLO

# Parámetros del modelo YOLOv11
model_path = "my_model.pt"
img_source = "usb0"
min_thresh = 0.6
user_res = None

# Ruta fija del archivo Excel
excel_path = "C:/Users/calderon/Desktop/registro_desechos.xlsx"

# Validación del modelo
if not os.path.exists(model_path):
    print('ERROR: Modelo no encontrado.')
    sys.exit(0)

# Cargar modelo YOLO
model = YOLO(model_path, task='detect')
labels = model.names

# Puerto serial con Arduino
arduino = serial.Serial('COM13', 9600, timeout=1)

# Zona de Interés
roi_x1, roi_y1 = 100, 0
roi_x2, roi_y2 = 500, 450

# Colores para las clases
bbox_colors = [(160,120,87), (68,148,228), (93,97,200), (178,182,133), (88,159,166),
               (96,202,231), (159,124,168), (169,162,241), (98,118,150), (172,176,184)]

# En caso el archivo excel no exista
if not os.path.exists(excel_path):
    df_init = pd.DataFrame(columns=["Fecha", "hora", "objeto", "categoria"])
    df_init.to_excel(excel_path, index=False)

# Función para registrar detección
def registrar_deteccion(objeto, categoria):
    fecha = datetime.now().strftime("%Y-%m-%d")
    hora = datetime.now().strftime("%H:%M:%S")
    df = pd.read_excel(excel_path)
    df.loc[len(df)] = [fecha, hora, objeto, categoria]
    df.to_excel(excel_path, index=False)
    generar_graficos(df)

# Función para generar gráficos y guardarlos en hojas por semana/mes
def generar_graficos(df):
    df['Fecha'] = pd.to_datetime(df['Fecha'])
    hoy = pd.Timestamp.now().normalize()

    # GRÁFICO SEMANAL
    inicio_semana = hoy - pd.Timedelta(days=7)
    fin_semana = hoy
    semanal_df = df[(df['Fecha'] >= inicio_semana) & (df['Fecha'] <= fin_semana)]
    semanal_cuenta = semanal_df['Categoria'].value_counts()

    fig1, ax1 = plt.subplots(figsize=(7, 5))
    semanal_cuenta.plot(kind='bar', ax=ax1, color='skyblue')
    ax1.set_title("Gráfico Semanal ((inicio_semana.date()) a {fin_semana.date()}))")
    ax1.set_ylabel("Cantidad")
    plt.tight_layout()
    buf1 = BytesIO()
    plt.savefig(buf1, format='png')
    plt.close()
    buf1.seek(0)

    # GRÁFICO MENSUAL
    inicio_mes = hoy - pd.Timedelta(days=30)
    fin_mes = hoy
    mensual_df = df[(df['Fecha'] >= inicio_mes) & (df['Fecha'] <= fin_mes)]
    mensual_cuenta = mensual_df['Categoria'].value_counts()

    fig2, ax2 = plt.subplots(figsize=(7, 5))
    mensual_cuenta.plot(kind='bar', ax=ax2, color='salmon')
    ax2.set_title("Gráfico Mensual ((inicio_mes.date()) a {fin_mes.date()}))")
    ax2.set_ylabel("Cantidad")
    plt.tight_layout()
    buf2 = BytesIO()
    plt.savefig(buf2, format='png')
    plt.close()
    buf2.seek(0)
```

Figura 38

Programación del sub-sistema de procesamiento y decisión parte 2

```
# Iniciar gráficos en el Excel
book = load_workbook(excel_path)

nombre_hoja_semanal = f"Semana_{inicio_semana.date()}_a_{fin_semana.date()}"
if nombre_hoja_semanal not in book.sheetnames:
    book.create_sheet(title=nombre_hoja_semanal)
sheet_sem = book[nombre_hoja_semanal]
chart_img1 = XlImage(buf1)
chart_img1.anchor = 'A1'
sheet_sem.add_image(chart_img1)

nombre_hoja_mensual = f"Mensual_{inicio_mes.date()}_a_{fin_mes.date()}"
if nombre_hoja_mensual not in book.sheetnames:
    book.create_sheet(title=nombre_hoja_mensual)
sheet_mes = book[nombre_hoja_mensual]
chart_img2 = XlImage(buf2)
chart_img2.anchor = 'A1'
sheet_mes.add_image(chart_img2)

book.save(excel_path)

# Bucle principal, espera la señal de Arduino para comenzar
while True:
    if arduino.in_waiting > 0:
        mensaje = arduino.readline().decode().strip()
        if mensaje == "Comenzar":
            print("Iniciando detección...")

            cap = cv2.VideoCapture(int(img_source[3])) if 'usb' in img_source else 0
            if user_res:
                res_h, res_w = map(int, user_res.split('x'))
                cap.set(3, res_h)
                cap.set(4, res_w)

            last_detected_class = None
            stable_start_time = None
            detection_confirmed = False
            detected_object_name = None
            start_time = time.time()

            while True:
                ret, frame = cap.read()
                if not ret:
                    print("Error de cámara")
                    break

                if not detection_confirmed and (time.time() - start_time >= 30):
                    mensaje_envio = "Vacío o desconocido"
                    arduino.write(mensaje_envio + "\n".encode())
                    print(f"Tiempo agotado. Mensaje enviado al Arduino: {mensaje_envio}")
                    registrar_deteccion("Ninguno o desconocido", "Ninguno o desconocido")
                    break

                results = model(frame, verbose=False)
                detections = results[0].boxes
                class_detected_this_frame = None

                for i in range(len(detections)):
                    xyxy = detections[i].xyxy.cpu().numpy().squeeze().astype(int)
                    xmin, ymin, xmax, ymax = xyxy
                    classidx = int(detections[i].cls.item())
                    classname = labels[classidx]
                    conf = detections[i].conf.item()
                    cx = (xmin + xmax) // 2
                    cy = (ymin + ymax) // 2

                    if conf > min_thresh and roi_x1 <= cx <= roi_x2 and roi_y1 <= cy <= roi_y2:
                        class_detected_this_frame = classname
                        color = bbox_colors[classidx % len(bbox_colors)]
                        cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), color, 2)
                        cv2.putText(frame, f'{classname}: {int(conf*100)}%', (xmin, ymin - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

                if class_detected_this_frame:
                    if class_detected_this_frame == last_detected_class:
                        if stable_start_time is None:
                            stable_start_time = time.time()
                        elif time.time() - stable_start_time >= 10 and not detection_confirmed:
                            detected_object_name = class_detected_this_frame
                            detection_confirmed = True

                            if detected_object_name in ["carton", "vasconalata", "mcahola", "lata", "latalecho", "popol"]:
                                categoria = "Aprovechable"
                            elif detected_object_name in ["balleta", "galleta", "vasoplastico", "vasopoliblanco", "vasopollcolor"]:
                                categoria = "No aprovechable"
                            elif detected_object_name in ["cascara", "manzana", "pan", "platano", "te"]:
                                categoria = "Organico"
                            elif detected_object_name in ["aguaoxigenada", "pila", "foco", "deringa", "medicinas", "medicinas2"]:
                                categoria = "Peligroso"
                            else:
                                categoria = "Desconocido"

                            arduino.write((categoria + "\n").encode())
                            print(f"Mensaje enviado: {categoria}")
                            registrar_deteccion(detected_object_name, categoria)
                            break
                    else:
                        last_detected_class = class_detected_this_frame
                        stable_start_time = time.time()
                        detection_confirmed = False
                else:
                    last_detected_class = None
                    stable_start_time = None
                    detection_confirmed = False

            cv2.rectangle(frame, (roi_x1, roi_y1), (roi_x2, roi_y2), (0, 255, 0), 2)
            cv2.putText(frame, "Zona de interes", (roi_x1, roi_y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
            cv2.imshow('Deteccion', frame)

            if cv2.waitKey(5) & notf == ord('q'):
                mensaje_envio = "Vacío o desconocido"
                arduino.write(mensaje_envio + "\n".encode())
                print(f"Tiempo agotado. Mensaje enviado al Arduino: {mensaje_envio}")
                break

            cap.release()
            cv2.destroyAllWindows()
            print("Esperando nueva señal...\n")
```

El flujo de trabajo general es: el sistema espera una señal de activación de haber pulsado el interruptor de presión vía Arduino Uno, inicia la captura de video, identifica el residuo en la zona de interés, valida la detección para asegurar su estabilidad, selecciona la categoría del objeto, envía una señal Arduino, y finalmente, registra el residuo y actualiza unas gráficas de barras semana y mensual.

Se importan múltiples librerías de Python, cada una con un rol específico:

- Interacción con el Sistema y Visión: Se usan las librerías os, sys, time y cv2 (OpenCV) para operaciones del sistema, control de tiempo y el procesamiento de video.
- Computación Numérica: Se usa numpy para operaciones matriciales eficientes, esenciales para el manejo de datos de imagen y coordenadas.
- Comunicación con Arduino Uno: Se usa serial para establecer la comunicación bidireccional con la placa Arduino.
- Análisis y Gestión de Datos: Se usa pandas para la manipulación de datos tabulares, matplotlib.pyplot para la generación de gráficos, y datetime para el sellado de tiempo preciso de cada evento.
- Integración con Excel: Se usa openpyxl y io para la manipulación avanzada de archivos .xlsx, permitiendo no solo escribir datos, sino también crear nuevas hojas e insertar imágenes de gráficos generados en memoria.

- **Visión Artificial:** Se usa ultralytics.YOLO para cargar y ejecutar el modelo de detección de objetos.

Lo primero es establecer las variables de configuración que gobiernan el comportamiento del sistema:

`model_path`: Especifica la ruta al archivo del modelo YOLO entrenado (`my_model.pt`), que es el cerebro del sistema de detección.

`min_thresh`: Define el umbral de confianza en 0.6. Esto significa que una detección solo será considerada válida si el modelo tiene al menos un 60% de certeza, filtrando así posibles falsos positivos.

`excel_path`: Fija la ruta del archivo Excel que funcionará como la base de datos o bitácora permanente del sistema.

Posteriormente, se cargo el modelo YOLO en memoria (`model = YOLO(model_path, task='detect')`) indicando la ruta al modelo y definiendo que clase de modelo es, es este caso es detección de objetos, con `detect` pues ya es una palabra reservada. De este modelo se extraen los nombres de las clases (`labels = model.names`), creando un mapa entre el índice numérico que el modelo predice (ej: 0) y su etiqueta correspondiente legible por humanos (ej: "Aguaoxygenada"). Se inicializa la conexión con el Arduino Uno en el puerto COM13 a una velocidad de 9600 baudios. Esta conexión es el canal a través del cual el código enviará las órdenes de segregación a la placa Arduino Uno. Se define una Zona de Interés (ROI) mediante coordenadas (`roi_x1`,

roi_y1) y (roi_x2, roi_y2). El sistema solo procesará objetos cuyo centroide se encuentre dentro de este rectángulo virtual en la imagen, esto evita que el sistema se distraiga con objetos en el fondo o en los bordes del campo visual de la cámara, enfocando el análisis únicamente en el área que importa.

En la siguiente sección del código se encarga del registro de los datos y su posterior análisis con gráficas de barras, convirtiendo los eventos de detección en información estructurada y visual.

Antes del bucle principal, un bloque condicional (if not os.path.exists(...)) verifica si el archivo Excel de registro existe. Si no es así, crea un nuevo archivo con las cabeceras "Fecha", "Hora", "Objeto" y "Categoría".

La función registrar_deteccion(objeto, categoria) es muy importante para el registro de datos. Su proceso consiste en Leer-Modificar-Escribir, y se da por lo siguiente:

- **Captura de Metadatos:** Obtiene la fecha y hora exactas del momento de la detección usando datetime.now() y las formatea en un estándar consistente (YYYY-MM-DD y HH:MM:SS).
- **Lectura:** Abre el archivo Excel existente y carga todo su contenido en un DataFrame de Pandas (df = pd.read_excel(excel_path)). Un DataFrame es como una hoja de cálculo dentro de Python.
- **Modificación:** Añade una nueva fila al final del DataFrame en memoria. La expresión df.loc[len(df)] es una forma de apuntar

a la primera posición vacía, donde se insertan la fecha, hora, y el objeto y categoría recibidos.

- Escritura: Guarda el DataFrame modificado de vuelta al archivo Excel, sobrescribiendo la versión anterior con la nueva información. `index=False` se usa para mantener un formato limpio sin índices numéricos de Pandas en la hoja de cálculo.
- Activación de Análisis gráfico: Inmediatamente después de registrar, llama a la función `generar_graficos`.

A continuación se define la función `generar_graficos` que recibe el `df` (DataFrame), que es la tabla completa con todo el historial de detecciones actualizado.

- Preparación de Datos: `pd.to_datetime...` convierte la columna "Fecha" de simple texto (ej: "2025-07-30") a un formato especial de "Fecha y Hora" que Pandas pueda entender y con el que puede operar matemáticamente, porque hasta antes de esto, el texto "2025-07-30" no es inherentemente "un día antes" que "2025-07-31", son solo cadenas de caracteres. Al convertirlas con `to_datetime`, ahora el programa sabe que son fechas y puede filtrar rangos, como "dame todo lo de los últimos 7 días".

La línea `hoy = pd.Timestamp.now().normalize()` establece el día actual a las 00:00:00, garantizando que los rangos de "últimos 7 días" y "últimos 30 días" sean consistentes independientemente de la hora de ejecución. Es una forma de decirle al programa: "Para mis cálculos, no me importa la hora

exacta del día. Solo quiero saber cuál es la fecha de 'hoy', considerando el día completo desde su inicio a medianoche" para que los últimos 7 días sean los mismos sin importar a que hora se ejecute el código.

- Filtrado y Agregación: Para los informes semanales y mensuales, filtra el DataFrame principal para crear subconjuntos de datos que contienen solo los registros del período de tiempo relevante. `semanal_df = df[(df['Fecha'] >= inicio_semana) & (df['Fecha'] <= fin_semana)]` selecciona únicamente las filas que cumplen esta doble condición. Luego con `value_counts()` cuenta las ocurrencias de cada categoría.
- Visualización en Memoria: Utilizando `matplotlib`, se generan los gráficos de barras. Una punto clave aquí es el uso de `io.BytesIO`. En lugar de guardar los gráficos como archivos de imagen temporales en el disco, se guardan directamente en un buffer en la memoria RAM (`plt.savefig(buf1, format='png')`) para luego poder pegarlo directamente en el informe final sin tenerlo que cargar desde el escritorio.
- Inserción en Excel: Usando `openpyxl`, el script carga el libro de trabajo de Excel y crea dinámicamente nuevas hojas con nombres descriptivos (como ejemplo: `Semanal_2025-07-23_a_2025-07-30`). Finalmente, inserta las imágenes de los gráficos (leídas desde el buffer de memoria) en la celda A1 de sus respectivas hojas, y guarda el libro de trabajo. Es como un

dashboard que se actualiza automáticamente con cada detección.

El bucle principal (while True) es la sección principal de este sub-sistema:

- Activación por Arduino: El sistema permanece en un estado de espera (if arduino.in_waiting > 0) hasta que recibe el mensaje "Comenzar" a través del puerto serie. Esto permite que la detección se active cuando el interruptor de presión haya sido pulsado.
- Ciclo de detección:
 - Inicialización: Se activa la cámara y se inicializan variables de estado, como last_detected_class y stable_start_time, que son importantes para la lógica de validación.
 - Procesamiento por Fotograma: El sistema entra en un bucle interno que lee la cámara fotograma a fotograma.
 - Inferencia del Modelo: Cada fotograma es enviado al modelo YOLO (results = model(frame, ...)), el cual devuelve las detecciones.
 - Filtrado de Detecciones: El código itera sobre las detecciones, aplicando el doble filtro de validación: la confianza debe ser mayor a 0.6 y el centro del objeto debe estar dentro de la ROI.

- Lógica de Estabilidad o podría entenderse como regla de los 10 segundos: Es un paso importante para la robustez del sub-sistema en un entorno real, no basta con detectar un objeto en un solo fotograma, el sistema requiere que el mismo objeto sea detectado de forma continua durante 10 segundos.

Si se detecta una clase, se compara con la del fotograma anterior. Si es la misma, se inicia o continúa un temporizador (`stable_start_time`). Solo si `time.time() - stable_start_time` supera los 10 segundos, la detección se considera confirmada (`detection_confirmed = True`). Si la clase detectada cambia o desaparece, el temporizador se reinicia. Esto previene clasificaciones erróneas debidas fallos momentáneos del modelo como cuando el objeto esta entrando en reposo.

- Acción y Registro: Una vez confirmada la detección, el código:
 - Asigna la categoría correspondiente ("Aprovechable", "Orgánico", etc.) basándose en el nombre del objeto.
 - Envía esta categoría como una cadena de texto al Arduino (`arduino.write((categoria + "\n").encode())`).
 - Llama a `registrar_deteccion` para guardar el residuo detectado.

- Rompe el bucle de detección (break) y vuelve al modo de espera.
- Mecanismo de Tiempo Agotado: Un temporizador de 30 segundos se asegura de que si ningún objeto es detectado de forma estable, el sistema no se quede bloqueado, sino que transcurrido este tiempo, envía "Vacio o desconocido" al Arduino Uno y registra el evento como tal, y se reinicia.
- A su vez se incorpora una funcionalidad de detenido manual del proceso a través del evento de teclado (`cv2.waitKey(5) & 0xFF == ord('q')`). La pulsación de la tecla 'q' durante un ciclo de detección es un mecanismo de control que ejecuta una secuencia de finalización controlada, dentro de este bloque If lo primero que se realiza es el envío de la señal "Vacio o desconocido" a la placa Arduino Uno para reestablecerlo en su configuración de espera y por último se interrumpe el bucle de detección actual de forma inmediata, sin necesidad de esperar la confirmación de un objeto o el vencimiento del temporizador de 30 segundos, sin haber registrado nada en este caso en el excel.

- Liberación de Recursos: Al finalizar cada ciclo de detección, `cap.release()` y `cv2.destroyAllWindows()` liberan la cámara y cierran las ventanas abiertas por OpenCV, asegurando que los recursos del sistema se gestionen correctamente y estén listos para la siguiente activación.

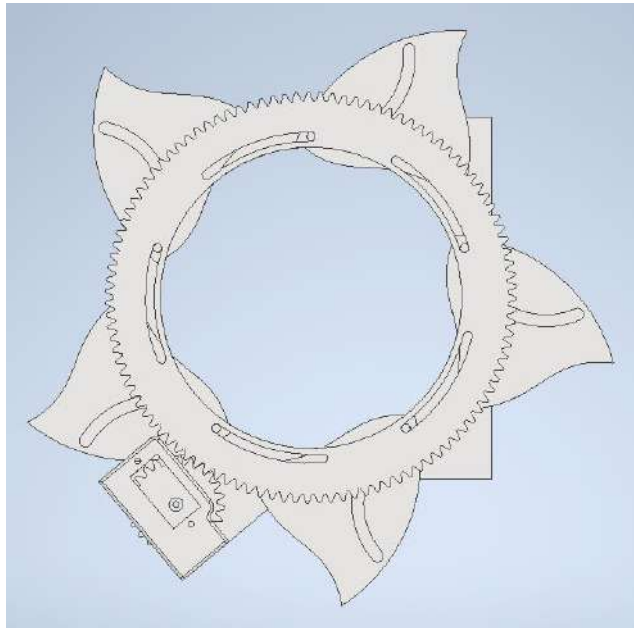
3.1.3.2.5 Diseño del subsistema de actuación

En esta etapa se desarrollo el código arduino que controlaba los servomotores usados en los mecanimos, se estableció la comunicación serial con python por para recibir la categoría de residuo que debían segregar, también controlaban los led's de señalización y el detectar si el interruptor de presión había sido pulsado. Asu vez también se diseño los mecanismos usados para completar la tarea de segregación.

Como se había determinado antes para la etapa de presentación se usaria un mecanismo que cumple la función de una puerta, el tipo de mecanismo seleccionado para esta función es uno del tipo llamado Iris.

Figura 39

Mecanismos Iris diseñado. Fuente: Propia



Para el accionamiento de los mecanismos Iris y re-dirigidor, se seleccionaron los servomotores modelo MG90S. La elección de este componente específico se basó en una evaluación de los requisitos de la aplicación, funcionamiento de los mecanismos, priorizando la fiabilidad, la suficiencia de torque y la facilidad de integración dentro del contexto de un prototipo funcional. La justificación se detalla en los siguientes puntos:

- Giro angular permitido: El servomotor MG90S fue seleccionado debido a que con el giro angular que se les permite se pueden diseñar mecanismos eficientes que aprovechen esa característica para tener las configuraciones útiles, a su vez también teniendo en cuenta la relación con el costo, como se ve en el siguiente punto.
- Relación Costo-Beneficio: En el contexto de un prototipo de bajo costo, el

MG90S ofrece un excelente equilibrio entre un precio asequible y las ventajas significativas en giro angular permitido, durabilidad que aportan sus engranajes metálicos, representando una mejora sustancial en fiabilidad por una diferencia de coste marginal en comparación con las alternativas de plástico, facilidad de integración para su control.

- Durabilidad y Fiabilidad (Engranajes Metálicos): A diferencia de otros servomotores de tamaño similar, como el popular SG90 que utiliza engranajes de nylon, el MG90S cuenta con un tren de engranajes metálicos. Esta característica fue un factor decisivo, ya que proporciona una durabilidad y resistencia al desgaste significativamente mayores. Considerando que el prototipo operará de forma repetitiva, los engranajes metálicos previenen fallos por rotura o barrido de dientes, un riesgo común en actuadores con engranajes de plástico bajo cargas cíclicas o bloqueos inesperados, garantizando así una mayor vida útil y fiabilidad para el prototipo.

- Suficiencia de Torque para la Aplicación (Estimación de Carga): Se realizó una estimación cualitativa de las cargas de trabajo. Las tareas de los servomotores consisten en mover los componentes de los mecanismos, fabricados en material ligero (PLA de una Impresora 3D Bambulab A1), y redireccionar residuos individuales cuyo peso es mínimo (ej. una botella de plástico vacía, una cáscara de fruta). El torque nominal del MG90S (aproximadamente 2.2 kg-cm a 6V) excede con un amplio margen de seguridad las bajas demandas de torque

requeridas por la aplicación. Por lo tanto, se determinó que no era necesario un análisis dinámico complejo para validar su suficiencia, siendo una elección adecuada y eficiente para la escala del prototipo.

- **Facilidad de Integración y Control:** El MG90S es un servomotor estándar que opera en un rango de voltaje (4.8V a 6V) compatible con el subsistema de alimentación diseñado (regulador DC-DC ajustado a 6V). Además, es totalmente compatible con la biblioteca Servo.h de Arduino, lo que simplificó enormemente el desarrollo del software de control y permitió una rápida implementación y ajuste de los movimientos de los mecanismos.

Sobre la selección de este tipo de mecanismo Iris se lo eligió por el espacio reducido en el que llega a cumplir su funcionamiento y ya que al ser de un grado de libertad solo se necesita una entrada para definir las dos configuraciones útiles, produciendo una abertura cuando el actuador, un servomotor MG90S este en 180° y el cierre la abertura cuando este en 0° . Es un diseño inteligente que utiliza pares cinemáticos como el de engranaje y otros únicos que como recalca el famoso libro Teoría de maquinas y mecanismos de Joseph Edward Shigley lo más importante para hallar los grados de libertad de un mecanismo, no es la aplicación de la formula de gruebler, ya que esta aunque funciona para ciertos para otros no da la cantidad correcta porque en el diseño de esta formula no contempla aspectos importantes como la forma de las uniones ni la geometría particular de los eslabones, entonces siempre se sugiere el analizar el movimiento que se producen sobre todo para llegar al número correcto de grados de libertad (Shigley et al., 1996).

En un mecanismo tipos Iris el primer elemento a diseñar es el tren de engranajes rectos, el tamaño de estos se define en base al espacio que deban ocupar, en este caso se deben diseñar dos engranajes, el primero que estara conectado directamente a la salida del servomotor MG90S y el otro acoplado a este para que transmita el movimiento de salida a las paletas que produzcan la abertura o cerradura en el espacio.

Los cálculos para el primer engranaje se diseñaron bajo la norma ANSI/AGMA 2101-d04 para engranajes rectos y fueron:

- Diámetro exterior y módulo:

$$D_e = 32.5 \text{ mm} \quad M = 1.25 \text{ mm}$$

Teniendo el módulo y el diámetro exterior se halla el número de dientes:

$$Z = \frac{D_e}{M} - 2 \rightarrow Z = 24 \text{ dientes}$$

Con estos datos ya podemos hallar los otros datos necesarios para dimensionar el engranaje.

- Diámetro primitivo

$$D_p = Z \times M \rightarrow D_p = 30 \text{ mm}$$

- Paso circular:

$$PC = \pi \times M \rightarrow PC = 3.93 \text{ mm}$$

- Cabeza del diente:

$$A = M \rightarrow A = 1.25 \text{ mm}$$

- Cabeza del diente:

$$S = PC \times \frac{19}{40} = 1.87 \text{ mm}$$

- Pie del diente:

$$PD = 1.25 \times M = 1.56 \text{ mm}$$

- Radio de entalle:

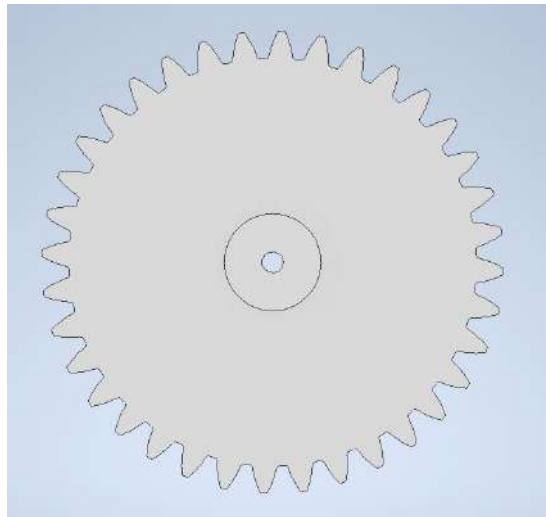
$$R = \frac{PC}{12} = 0.33 \text{ mm}$$

- Angulo del espesor:

$$AS = \frac{360 \times S}{2 \pi \times \frac{DP}{2}} = 7.14^\circ$$

Figura 40

Engranaje recto 24 dientes



Las cálculos para el segundo engranaje recto más grande y de salida fueron:

- Diámetro exterior y módulo:

$$D_e = 136.25 \text{ mm} \quad M = 1.25 \text{ mm}$$

Teniendo el módulo y el diametro exterior se halla el número de dientes:

$$Z = \frac{D_e}{M} - 2 \rightarrow Z = 107 \text{ dientes}$$

Con estos datos ya podemos hallar los otros datos necesarios para dimensionar el engranaje.

- Diámetro primitivo

$$D_p = Z \times M \rightarrow D_p = 137.75 \text{ mm}$$

- Paso circular:

$$PC = \pi \times M \rightarrow PC = 3.93 \text{ mm}$$

- Cabeza del diente:

$$A = M \rightarrow A = 1.25 \text{ mm}$$

- Cabeza del diente:

$$S = PC \times \frac{19}{40} = 1.87 \text{ mm}$$

- Pie del diente:

$$PD = 1.25 \times M = 1.56 \text{ mm}$$

- Radio de entalle:

$$R = \frac{PC}{12} = 0.33 \text{ mm}$$

- Angulo del espesor:

$$AS = \frac{360 \times S}{2 \pi \times \frac{DP}{2}} = 1.60^\circ$$

Figura 41

Engranaje recto 107 dientes



Es importante ahora también calcular cuanto girará el segundo engranaje de 107 dientes cuando el primero de 24 dientes giré 180° .

$$\theta_1 \times N_1 = \theta_2 \times N_2$$

Sabiendo que N_1 es 24 dientes y N_2 es 107 dientes, y que θ_1 es 180° :

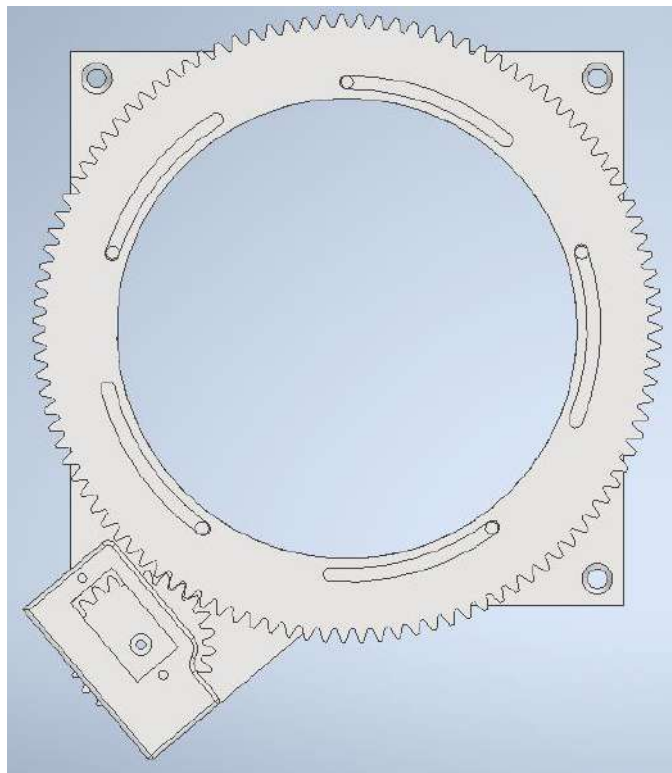
$$\theta_2 = 40.37^\circ$$

Este dato nos será importante para el diseño de las paletas. El siguiente paso es el diseño de la bancada, esta es el eslabón fijo respecto al cual todos los demás eslabones se mueven, como este es un mecanismo plano donde todos los movimiento de los eslabones deben estar en planos paralelos se tiene que una junta entre el segundo eslabón y la bancada, esto se logra por medio de unas ranuras a través de las cuales el unico movimiento respecto a la bancada de este es el de

rotar alrededor del centro del engranaje, para que este sujeto al eslabón fijo habrán unas entradas donde unos tornillos ingresan y sujetan el segundo engranaje, de la misma manera se acopla a este el primer engranaje a la distancia que corresponde entre sus centros, uniendo también el primer engranaje a la bancada y diseñando el espacio donde ingresará el servomotor MG90S.

Figura 42

Ensamblaje de mecanismo Iris parte 1

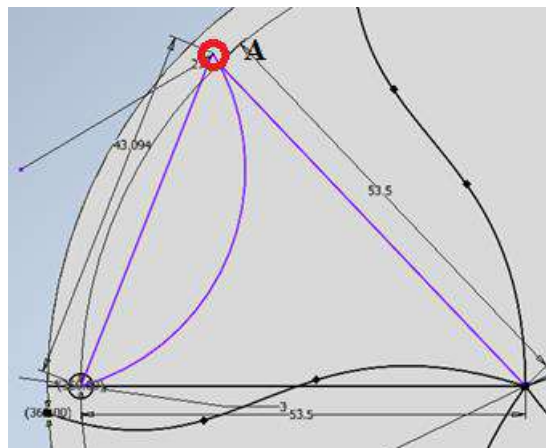


El siguiente paso es el dimensionamiento de las paletas para esto se eligió un número de cinco paletas, para el diseño de estas se debe dividir el espacio central en cinco partes iguales. Hasta el instante anterior solo hay un grado de libertad, cuando se agrega una paleta se tiene que por la necesidad de que este giré solo alrededor de un eje, se lo conecta al pin que va hacia la bancada, siendo entonces que solo puede desarrollar un movimiento de rotación hasta ese

momento, pero como necesitamos conectar el movimiento que recibe el segundo engranaje para que estas paletas se muevan como eslabones de salida se tiene que formar un par entre las paletas y el segundo eslabón, esto se realiza por medio de una ranura en cada paleta cuya forma es curva debido a que así se aprovecha más ángulo que gira el segundo engranaje para producir la mayor abertura posible. La última parte es dimensionar esta paleta, se sabe que en el segundo engranaje habrá un pin que conectará en la ranura de la paleta e impulsará hacia los costados, dicho pin se definirá de forma tentativa pero en todo momento estará dentro de una parte de una circunferencia, donde también está el punto central de donde se conecta la paleta con la bancada. Para saber el radio de la circunferencia de la ranura necesitamos definir la posición del punto final A ya que este en todo momento formará el mismo ángulo con la longitud inferior 53.5 mm.

Figura 43

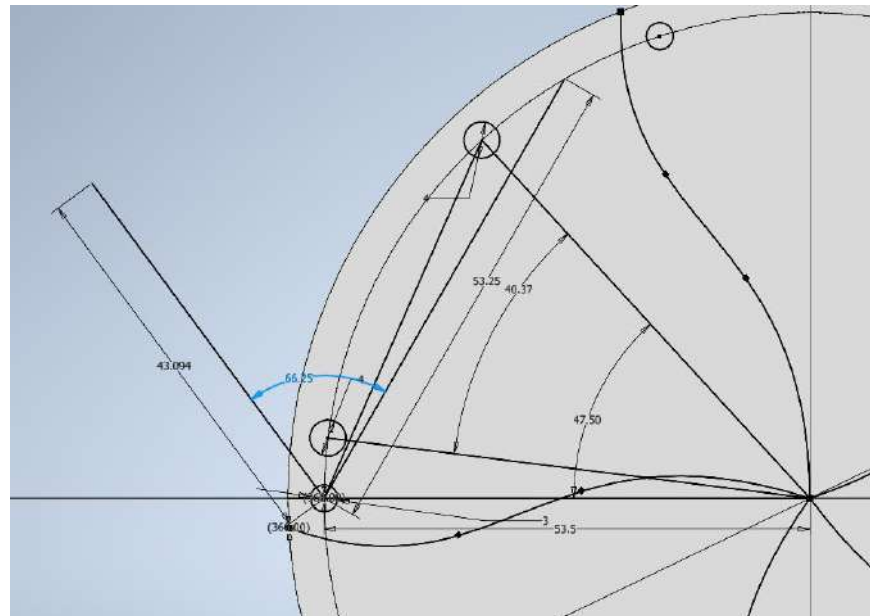
Posición del pin del segundo engranaje



Así se vuelve a construir el triángulo en la posición final necesaria del punto central, que indicaría la abertura que se produjera, como se puede ver en la siguiente imagen donde se mantiene el ángulo 66.25° :

Figura 44

Posición final de la paleta para hallar el radio



Se observa que en ese punto la parte inferior de la paleta habría girado suficiente como para despejar la zona de en medio, también es importante notar que la posición inicial de pin del segundo engranaje se desplazara la misma distancia angular 40.37° donde ya estará en su posición final y se supone que ahí ya debe producirse la abertura, sabiendo la posición final necesaria del punto A, el punto final de pin, y recordando que siempre también el punto donde rota la paleta es también un punto de la circunferencia, se pueden usar dos métodos para hallar el radio de la circunferencia para validarse, uno es un método gráfico y el segundo un método analítico que halla el radio de la circunferencia a partir de las coordenadas de tres puntos que lo forman.

- Método gráfico

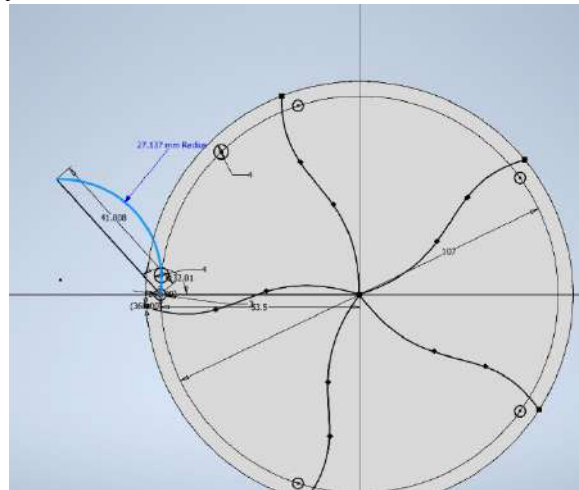
El autor del libro maquinas y mecanismos David Myszka destaca el uso de métodos gráficos frente a analíticos cuando ofrecen mayor

simplicidad en el cálculo de importantes dimensiones sin afectar la precisión de dichas dimensiones usando herramientas como Autodesk Inventor (*Maquinas y Mecanismos*, 2011).

Usando la herramienta arco con tres puntos de Inventor se puede hallar el radio nesario de esa circunferencia que en este caso es, 27.137 mm.

Figura 45

Radio de la circunferencia de la ranura



Así de esta forma ya se puede realizar el ensamblaje completo, donde el mecanismo resultantes es uno de un grado de libertad.

Para validar el mecanismo se uso la herramienta de simulación de movimiento donde los resultados son las siguientes figuras:

Figura 46

Configuración cerrada del mecanismo Iris

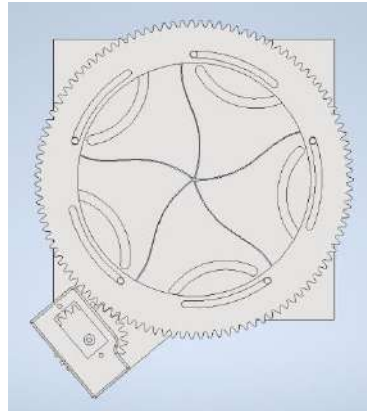
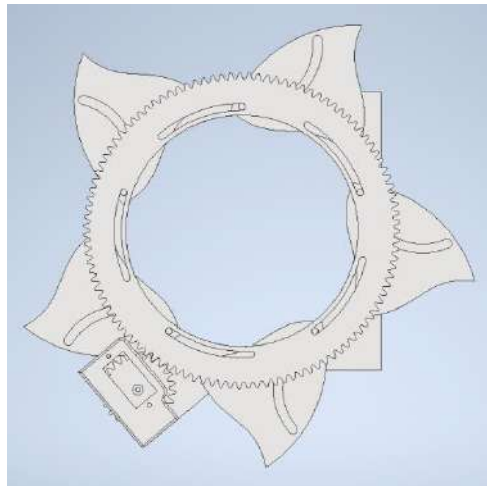


Figura 47

Configuración abierta del mecanismo Iris



El diseño del segundo mecanismo re-dirigidor es más simple pero aún así lo suficientemente efectivo para cumplir con el propósito, es un mecanismo espacial ya que los movimientos de los eslabones no están en planos paralelos, como se necesita un mecanismo que mande los residuos cuatro espacios se aprovechó las características de los servomotores MG90S ya que estos pueden girar 180 grados sexagesimales, suficiente para que con dos servomotores de este tipo se logre diseñar una rampa que se incline gracias a posiciones angulares específicas de ambos servomotores en coordinación que guían la

rampa a uno de los cuatros compartimentos, se logra cuando el servomotor inferior de la base se posiciona a 45° orientando la rampa hacia dos posibles compartimentos, el compartimento final al que se incline es determinado por el segundo servomotor que también se posiciona a 45° inclinándose a un determinado compartimento mientras que si se inclina a 135° la rampa se orientará al otro compartimento, se logrará apuntar a los últimos dos compartimentos cuando el servomotor de la base inferior se inclina a 135° , dicha estructura del mecanismo se puede observar en siguiente figura:

Figura 48

Mecanismo re-dirigidor sin rampa

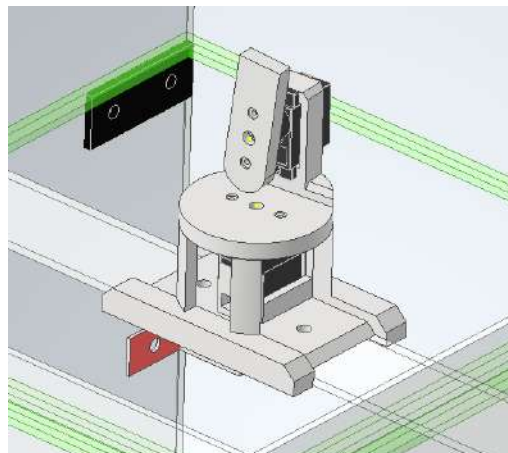
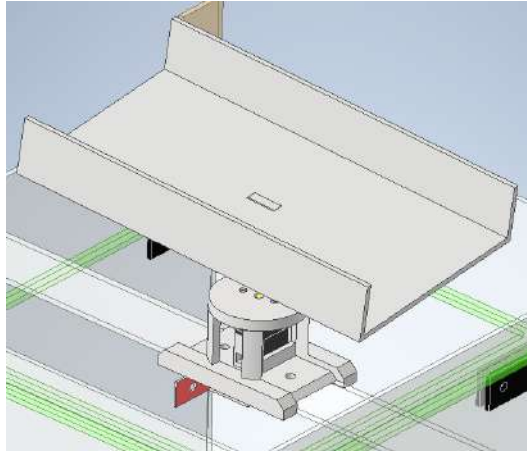


Figura 49

Mecanismo re-dirigidor con rampa



La siguiente parte de diseño es el código de programación de la placa Arduino:

Figura 50

Código de la placa Arduino Uno parte 1

```
#include <Servo.h>

const int boton = 2; // Aquí se conecta el botón 2
bool esperando = true;
const int ledverde = 4;
const int ledrojo = 3;
const int servo1pin = 5; //Servo de puerta superior
const int servo2pin = 6; //Servo de puerta inferior
const int servo3pin = 7; //Servo inferior de la base
const int servo4pin = 8; //Servo superior de la base

Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

void setup() {
  pinMode(boton, INPUT_PULLUP); // Se configura el pin 2 como input y en el modo pull up
  pinMode(ledverde, OUTPUT);
  pinMode(ledrojo, OUTPUT);
  servo1.attach(servo1pin);
  servo2.attach(servo2pin);
  servo3.attach(servo3pin);
  servo4.attach(servo4pin);
  Serial.begin(9600); // Se inicia la comunicación serial
}

void loop() {
  if (digitalRead(boton) == HIGH && esperando) { // El botón ha sido presionado para iniciar el proceso
    esperando = false;
    digitalWrite(ledverde, HIGH); //Se prende el led verde
    digitalWrite(ledrojo, LOW); //Se enciende el led rojo
    delay(3000);
    servo1.write(180); //Se abre el mecanismo 1
    delay(10000);
    servo1.write(0); //Se cierra el mecanismo 1
    Serial.println("Comenzar"); // Enviar mensaje de inicio a Python
  }

  else if (esperando == true) { // En este else if se encendera el led rojo y se apagara el led verde con la unica condicion de que esperando == True
    servo1.write(0);
    servo2.write(0);
    delay(450);
    servo2.detach();
    servo3.write(0);
    servo4.write(96);
    delay(500);
    servo3.detach();
    delay(2000);
    digitalWrite(ledverde, LOW);
    digitalWrite(ledrojo, HIGH);
  }

  if (Serial.available() > 0) { // Verificar respuesta de Python
    String mensaje = Serial.readStringUntil('\n');
    if (mensaje == "Aproachable") {
      servo3.attach(servo3pin);
      servo3.write(45);
      servo4.write(60);
      delay(4000);
      servo2.attach(servo2pin);
      servo2.write(180);
      delay(450);
      servo3.detach();
      delay(4000);
      servo2.attach(servo2pin);
      servo2.write(0);
      delay(450);
      servo2.detach();
      delay(2000);
      esperando = true; // Si el valor de mensaje coincide con el código de la posición de servomotores, entonces configura los servomotores y abre la segunda puerta, luego cierra la segunda puerta y restablece
    } //esperando = true; // Volver a esperar el botón
  }
}
```

Figura 51

Código de la placa Arduino Uno parte 2

```
else if (mensaje == "No aprovechable") {
  servo3.attach(servo3pin);
  servo3.write(145);
  servo4.write(60);
  delay(4000);
  servo2.attach(servo2pin);
  servo2.write(180);
  delay(450);
  servo2.detach();
  delay(4000);
  servo2.attach(servo2pin);
  servo2.write(0);
  delay(450);
  servo2.detach();
  delay(2000);
  esperando = true;
}

else if (mensaje == "Organico") {
  servo3.attach(servo3pin);
  servo3.write(45);
  servo4.write(135);
  delay(4000);
  servo2.attach(servo2pin);
  servo2.write(180);
  delay(450);
  servo2.detach();
  delay(4000);
  servo2.attach(servo2pin);
  servo2.write(0);
  delay(450);
  servo2.detach();
  delay(2000);
  esperando = true;
}

else if (mensaje == "Peligroso") {
  servo3.attach(servo3pin);
  servo3.write(145);
  servo4.write(135);
  delay(4000);
  servo2.attach(servo2pin);
  servo2.write(180);
  delay(450);
  servo2.detach();
  delay(4000);
  servo2.attach(servo2pin);
  servo2.write(0);
  delay(450);
  servo2.detach();
  delay(2000);
  esperando = true;
}

else if (mensaje == "Vacio o desconocido") {
  delay(2000);
  esperando = true;
}
}
```

1. Definición de Componentes y Variables Globales

El código comienza con la inclusión de la librería Servo.h, esencial para el control de los servomotores. Posteriormente, se declaran las constantes y variables globales que definen la arquitectura del sistema:

- Pines de Entrada/Salida: Se asignan pines digitales específicos para cada componente mediante constantes (const int), lo cual mejora la legibilidad y facilita el mantenimiento del código. Se define el boton en el pin 2, y los diodos LED de estado ledverde y ledrojo en los pines 4 y 3, respectivamente. Los pines 5, 6, 7 y 8 se asignan a los cuatro servomotores que controlan el mecanismo de clasificación.
- Objetos Servo: Se crean cuatro instancias de la clase Servo (servo1, servo2, servo3, servo4), correspondiendo a cada uno de los actuadores físicos.
- Variable de Estado: La variable booleana global esperando es el pilar de la lógica del programa. Funciona como el gestor de una máquina de estados finitos con dos estados principales:

esperando = true: El sistema está en reposo, listo para recibir un nuevo residuo.

esperando = false: El sistema está en un ciclo activo de procesamiento, esperando el análisis de Python o ejecutando una acción de segregación.

2. Inicialización del Sistema (void setup())

La función setup() se ejecuta una sola vez al energizar o reiniciar el microcontrolador. Su propósito es configurar el estado inicial de todos los componentes de hardware y la comunicación:

- Configuración de Pines:

`pinMode(boton, INPUT_PULLUP)`: Se configura el pin 2 como una entrada digital con una resistencia de pull-up interna activada. Esta configuración es fundamental, ya que asegura que el pin tenga un estado lógico ALTO (HIGH) por defecto y pase a BAJO (LOW) solo cuando se presiona el botón (conectado a tierra), evitando la necesidad de una resistencia externa y previniendo estados flotantes.

`pinMode(ledverde, OUTPUT)` y `pinMode(ledrojo, OUTPUT)`: Se configuran los pines de los LEDs como salidas para la retroalimentación visual del estado del sistema.

Vinculación de Servomotores: Las líneas `servoX.attach(servoXpin)` asocian cada objeto Servo con su pin de control físico correspondiente.

Inicio de Comunicación Serial: `Serial.begin(9600)` inicializa la comunicación serial a una velocidad de 9600 bits por segundo (baudios). Esta velocidad es idéntica a la configurada en el código de Python para garantizar una comunicación sin errores.

3. Lógica Operacional y Ciclo Principal (`void loop()`)

La función `loop()` contiene la lógica operativa que se ejecuta continuamente. Se estructura en tres bloques condicionales que gestionan los diferentes estados y eventos del sistema.

```
if (digitalRead(boton) == HIGH && esperando) { ... }
```

Este bloque se activa únicamente cuando se cumplen dos condiciones simultáneas: el botón es presionado (`digitalRead(boton) == HIGH`) y el sistema se encuentra en estado de espera (`esperando == true`). Esta doble condición previene que el ciclo se reinicie si el botón se presiona accidentalmente mientras un residuo ya está siendo procesado.

La secuencia de acciones es la siguiente:

Cambio de Estado: `esperando = false`; El sistema pasa a estado "activo".

Retroalimentación Visual: Se enciende el LED verde y se apaga el rojo, indicando al usuario que el proceso ha comenzado.

Apertura del Depósito: `servo1.write(180)`; El `servo1`, que controla la compuerta superior, se abre para permitir la entrada del desecho al área de análisis.

Tiempo de Depósito y Cierre: Tras una pausa de 10 segundos (`delay(10000)`), tiempo suficiente para que el objeto caiga, la compuerta se cierra (`servo1.write(0)`).

Señal a Python: `Serial.println("Comenzar")`; El Arduino Uno envía el mensaje "Comenzar" a través del puerto serie. Este mensaje es la señal de activación que el script de Python está esperando para iniciar la captura de video y el análisis por parte del modelo YOLO.

```
else if (esperando == true) { ... }
```

Este bloque define el comportamiento del sistema cuando está en estado de espera. Su principal función es asegurar que todos los actuadores estén en su posición inicial:

Posición por Defecto: Los servomotores se mueven a sus posiciones iniciales (`servo1.write(0)`, `servo3.write(0)`, etc.). La posición del servo4 a 96 grados es su posición de reposo específica.

Desvinculación de Servos: Se utiliza `servoX.detach()`. para la eficiencia energética y la protección del componente. Un servo adjunto podría intentar mantener activamente su posición, consumiendo corriente y pudiendo generar un "zumbido" o "vibración" si encuentra resistencia. Al desvincularlo, el servo deja de recibir la señal de control y se relaja, reduciendo el consumo a cero y eliminando el estrés mecánico. Se volverá a vincular (`attach`) solo cuando se necesite moverlo de nuevo.

Retroalimentación Visual: Se apaga el LED verde y se enciende el rojo, indicando claramente que el sistema está en modo de espera y listo para un nuevo ciclo.

```
if (Serial.available() > 0) { ... }
```

Este bloque gestiona la comunicación entrante desde Python. El microcontrolador sondea continuamente el buffer de recepción serial.

Lectura del Mensaje: Si hay datos disponibles, `Serial.readStringUntil('\n')` lee la cadena de caracteres enviada por Python hasta encontrar un carácter de nueva línea. Este mensaje corresponde a la categoría del desecho

("Aprovechable", "Organico", etc.).

Lógica de Decisión y Actuación: Una serie de condicionales if-else if compara el mensaje recibido con las categorías predefinidas. Cada bloque ejecuta una secuencia de movimientos de servomotor única y precisa para dirigir el residuo al contenedor correcto.

Por ejemplo para "Aprovechable":

`servo3.attach(servo3pin);` Se reactiva el control del servo.

`servo3.write(45); servo4.write(60);` Los servos de la base (`servo3` y `servo4`) se mueven a las coordenadas angulares (45°, 60°) que posicionan la plataforma de clasificación sobre el compartimento de "Aprovechables".

`servo2.write(180);` Tras un delay para permitir el posicionamiento, el `servo2` (compuerta inferior) se abre para soltar el residuo.

`servo2.write(0);` La compuerta se cierra.

`esperando = true;` Una vez completada la secuencia de clasificación, la variable de estado se restablece a true, lo que hace que el sistema vuelva al estado de reposo definido en el bloque else if (`esperando == true`), preparándolo para el próximo ciclo.

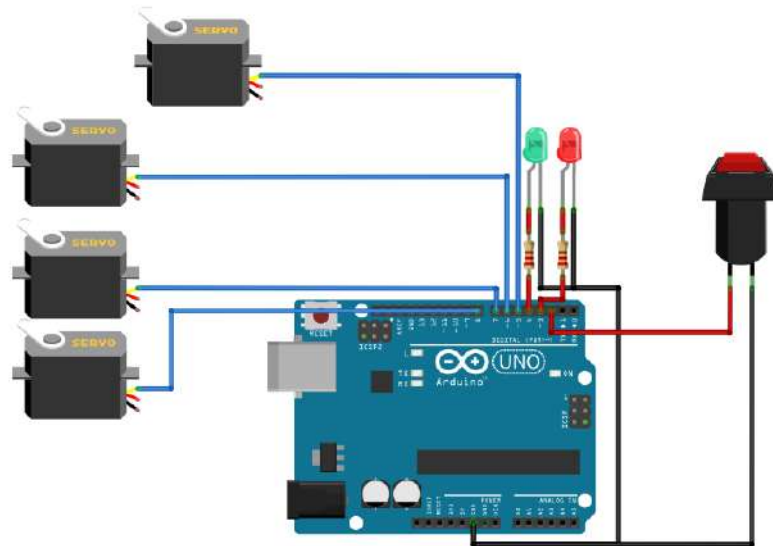
Este patrón se repite para cada categoría, con diferentes coordenadas

angulares para los servos de la base, demostrando un sub-sistema de actuación de segregación. La respuesta a "Vacio o desconocido" simplemente reestablece el estado de espera sin ninguna acción física, manejando correctamente en dicho caso.

La conexión de la señal que se envía a los servomotores es la siguiente:

Figura 52

Conexión para la señal de los servomotores



3.1.3.2.6 Diseño del subsistema de alimentación

Como se comentó anteriormente la laptop en este trabajo alimenta a la placa Arduino Uno y a la cámara USB, al estar Arduino Uno alimentado esto alimenta a los dos LEDs de 5 mm, también permite verificar el estado del interruptor de presión y por último puede enviar señales PWM correctamente a los servomotores para su control. Sin embargo los servomotores no son alimentados por Arduino pues no podría suministrar ni soportar la corriente eléctrica necesaria, lo cual se explica en la siguiente sección:

- **Análisis de la Corriente de Consumo de los Servomotores MG90S:** La selección de una fuente de alimentación externa se fundamenta en la comparación entre la demanda de corriente de los servomotores y la capacidad de suministro de la placa Arduino Uno.

- **Corriente de Consumo del Servomotor MG90S:** Un solo servomotor MG90S presenta un consumo de corriente variable dependiendo de su estado operativo:

- > Corriente en Reposo (Stall Current): Aproximadamente 10-20 mA.

- > Corriente de Operación (Sin Carga): Puede variar entre 100 mA y 250 mA mientras el motor se desplaza.

- > Corriente de Pico o de Arranque (Stall/Peak Current): Al iniciar un movimiento, al cambiar de dirección bruscamente o si se encuentra con una resistencia mecánica (bloqueo), el motor puede demandar una corriente de pico que puede alcanzar hasta 1 A (1000 mA) por un breve instante.

- **Capacidad de Suministro del Arduino Uno:** La placa Arduino Uno, cuando es alimentada por USB o por su conector de barril, tiene límites estrictos de suministro de corriente:

- > Suministro Total del Regulador de 5V: El regulador de voltaje interno de la placa puede suministrar un máximo de aproximadamente 500 mA a 800 mA en total antes de sobrecalentarse y fallar.

- > Límite por Pin de E/S: Cada pin digital o de 5V individual no debe

exceder los 40 mA de forma continua.

- **Justificación Técnica de la Fuente Externa:** Al considerar el peor escenario operativo, donde los cuatro servomotores podrían activarse simultáneamente o experimentar picos de corriente por atascamiento, la demanda total del sistema de actuación podría teóricamente acercarse a 4 A (4 x 1 A).

Demanda de Corriente Máxima (Teórica) \approx 4000 mA

Esta demanda de 4000 mA es drásticamente superior a la capacidad máxima de suministro del regulador del Arduino Uno (\approx 500 mA). Intentar alimentar los servos desde la placa Arduino habría provocado los siguientes fallos catastróficos:

- **Caída de Tensión (Brownout):** La alta demanda de corriente habría causado una caída de tensión en el riel de 5V, provocando que el microcontrolador ATmega328P se reiniciara constantemente (condición de "brownout"), haciendo el sistema inoperable.
- **Daño Físico a la Placa:** Forzar al regulador de voltaje a suministrar una corriente tan elevada habría causado su sobrecalentamiento y daño permanente, inutilizando la placa Arduino.

Por estas razones, se diseñó un subsistema de alimentación robusto y desacoplado, compuesto por una fuente de alimentación externa de 24V y 5A y un regulador de voltaje DC-DC Step Down ajustable modelo XL4015. Este diseño fue adecuado porque:

- **Suministra Corriente Suficiente:** La fuente de 5A provee la corriente necesaria para cubrir incluso los picos de consumo simultáneos de los

cuatro servos, garantizando un funcionamiento estable y sin caídas de tensión.

- Proporciona el Voltaje Correcto: El regulador XL4015 fue elegido por su alta eficiencia de conversión, que supera el 90 %, y por su capacidad para entregar hasta 5 A de corriente, también se lo configuró para reducir el voltaje de 24V a 6V, el voltaje óptimo de operación para los servomotores MG90S, asegurando que entreguen su máximo torque y rendimiento. Además, este regulador incorpora protecciones contra sobretensión, sobrecorriente y sobrecalentamiento, asegurando así la protección de los componentes y una regulación estable del voltaje.

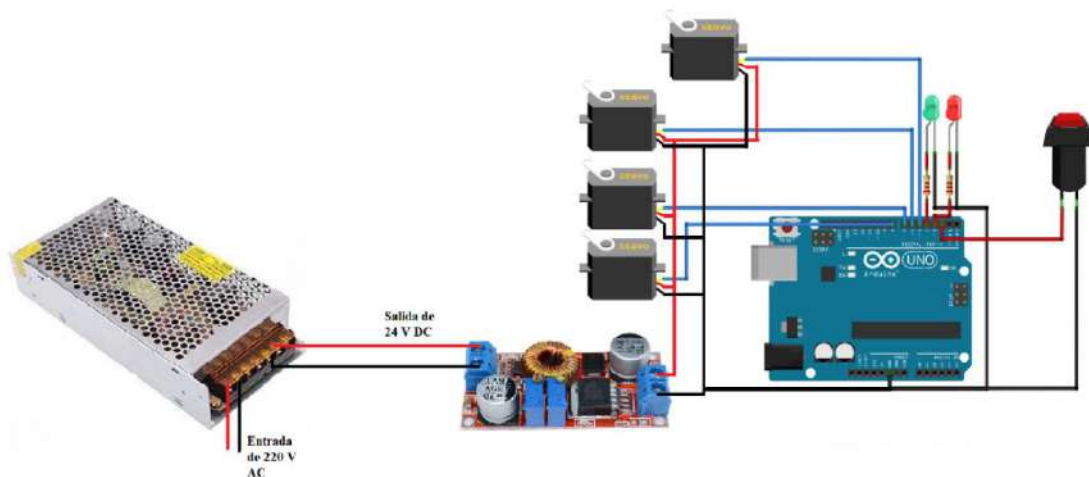
Esta separación de las fuentes de alimentación garantiza la estabilidad del sub-sistema, protege los componentes que se alimentan, asegurando que los actuadores dispongan de la potencia necesaria para operar de manera fiable.

También para garantizar un funcionamiento fiable de este sub-sistema, fue imperativo establecer una referencia de potencial eléctrico común mediante la conexión directa entre el pin de tierra (GND) de la placa Arduino uno y el terminal negativo del convertidor DC-DC externo. La señal de control enviada al servomotor, una Modulación por Ancho de Pulso (PWM), consiste en una serie de pulsos de 5V cuyo voltaje se genera y se mide intrínsecamente con respecto a la tierra del propio Arduino. Sin una conexión de tierra común, el circuito del servomotor, alimentado por el convertidor, carecería de esta referencia compartida, provocando que la señal PWM sea interpretada contra un potencial flotante y desconocido. Esta discrepancia anula la integridad de la señal, ya que

no se puede establecer un circuito de retorno para la corriente de control, resultando en un comportamiento errático o nulo del actuador. Al unir ambas tierras, se asegura que el "cero voltios" sea idéntico para el generador de la señal (Arduino Uno) y el receptor de la misma (servomotor), permitiendo una decodificación precisa de la duración del pulso y, por ende, un control de posición exacto y predecible.

Figura 53

Alimentación de servomotores



3.1.3.2.7 Diseño de la estructura

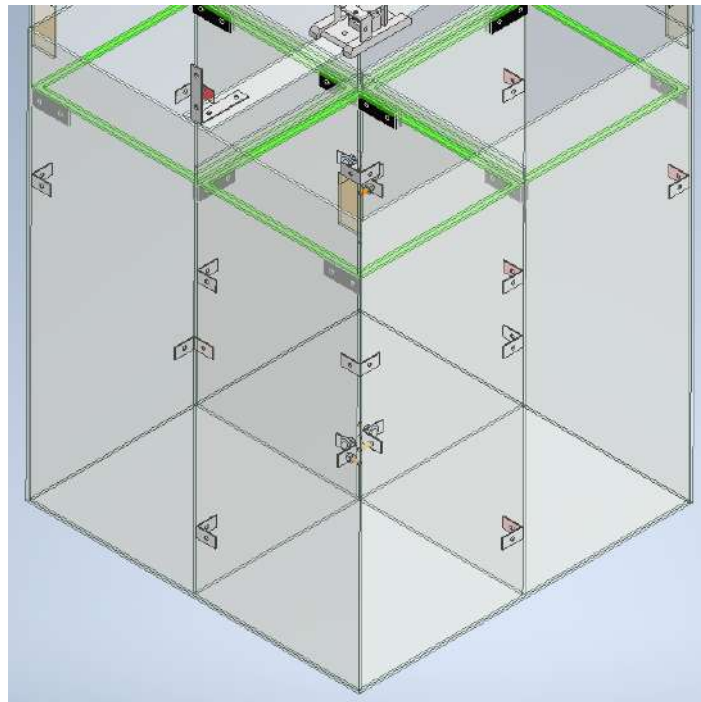
Una vez teniendo las dimensiones tentativas se hizo un diseño en Autodesk Inventor con todos los componentes que le corresponde. Las dimensiones del ancho, largo y alto se basaron a los tachos de 100 L Reyplast (53x51x84 cm) que utilizaba la institución educativa pues dichos tachos ya eran de fácil manipulación por los estudiantes y por el personal de limpieza, y se adecuaba al espacio disponible en el aula, por ello las dimensiones tomadas para el prototipo se aproximaron a estas siendo modificadas lo suficiente para tener suficiente espacio

en otras zonas, como el espacio de identificación o el espacio necesario para los mecanismos resultando en un tamaño de 46x46x100 cm.

El diseño se puede dividir en dos partes, se dividió el espacio de esta parte para el depósito final de los residuos en cuatro partes iguales; correspondientes a las cuatro categorías; para que todos los compartimentos tengan la misma capacidad de guardar los residuos.

Figura 54

Contenedores

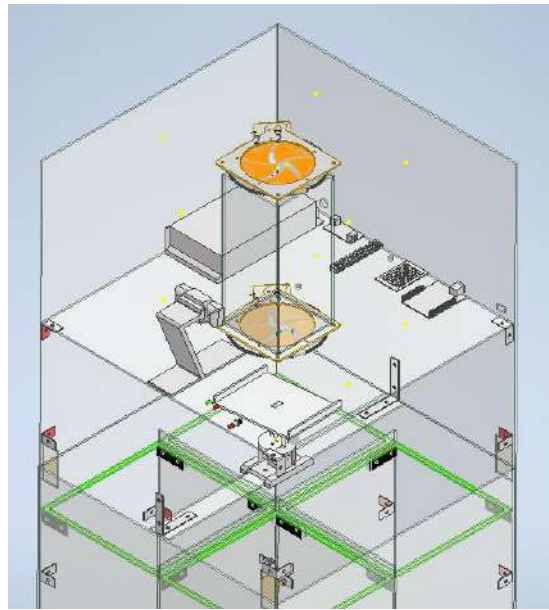


La segunda parte se dividió el espacio total del prototipo en dos partes con tal de que los residuos puedan ser retirados del prototipo fácilmente, mientras que la parte donde se encuentren todos los componentes físicos no puedan verse afectados por esa acción para luego se pueda volver a acoplar la parte superior mediante un encaje, también de acuerdo al tamaño de los residuos elegidos para

la identificación se determino un espacio para la etapa de presentación de residuos de 10x10x24 cm como se observa en la siguiente figura:

Figura 55

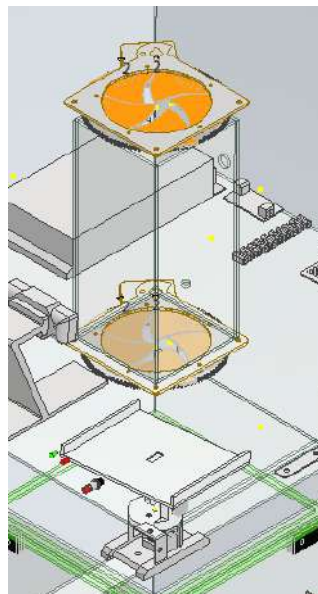
Parte superior



Como se observa también se ubicaron los mecanismos diseñados:

Figura 56

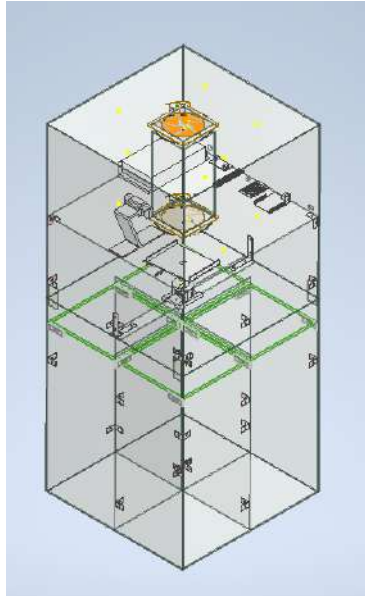
Mecanismos integrados



Como ultima representación esta el ensamblaje completo con todas sus partes, unidas por pernos, tuercas y ángulos:

Figura 57

Ensamblaje final



Se dispuso en el diseño de también utilizar ruedas locas para el fácil traslado del prototipo por la institución hasta el aula correspondiente, ahorrando el trabajo de tener que cargarlo al movilizarlo, dos de las cuales tenían la opción de frenado para quedarse posicionado en un solo lugar.

3.1.3.3 Implementación

Una vez teniendo todos los diseños de todos los sub-sistemas se procedió con la integración de los elementos físicos. Para la estructura principal, los contenedores y los paneles de soporte, se seleccionó acrílico transparente de 3 mm de espesor. Esta elección se sustenta en dos ventajas metodológicas clave para este proyecto:

- Visibilidad para la Observación y Diagnóstico: El objetivo principal de este prototipo es también validar un proceso. La naturaleza transparente

del acrílico fue un requisito de diseño fundamental, ya que permite la observación directa y sin obstrucciones de todos los procesos internos en tiempo real. Esto facilitó enormemente las fases de prueba, y validación, permitiendo verificar visualmente la cinemática de los mecanismos, el flujo de los residuos y el funcionamiento de los componentes electrónicos. Esta visibilidad es crucial para el análisis de fallos y la identificación de futuras mejoras.

- **Facilidad y Precisión de Fabricación:** El acrílico es un material ideal para el corte por láser, una tecnología de fabricación rápida y de alta precisión. Al basar el diseño estructural en este material, se garantizó que todas las piezas se fabricaran con las dimensiones exactas especificadas en los modelos CAD de Autodesk Inventor. Esto aseguró un ensamblaje correcto, con una alineación correcta de todos los orificios para pernos y soportes de componentes, reduciendo significativamente el tiempo de construcción y eliminando errores de fabricación manual.
- Además, el acrílico ofrece una rigidez estructural suficiente para la escala y las cargas ligeras del prototipo, a un costo asequible

Se empezó con el armado de la estructura de acrílico la cual se corto mediante laser en los diseños de la estructura hechos, posteriormente para su unión mecánica con pernos y tuercas, los mecanismos fueron impresos en 3D usando la impresora bambulab A1 por su rapidez y calidad de acabado. Se implementaron todos los componentes electrónicos y mecánicos para posteriormente ser probado.

Figura 58

Armado de estructura



Figura 59

Avance de parte inferior de estructura



Figura 60

Unión de la estructura inferior



Figura 61

Ensamblaje de la parte superior



También se diseño un logo para el prototipo en Adobe Illustrator que representara el propósito de segregar correctamente con la palabra EcoSplit, que significaría una segregación para la ecología, este sería el nombre del prototipo:

Figura 62

Diseño de logo EcoSplit



Figura 63

Logo EcoSplit

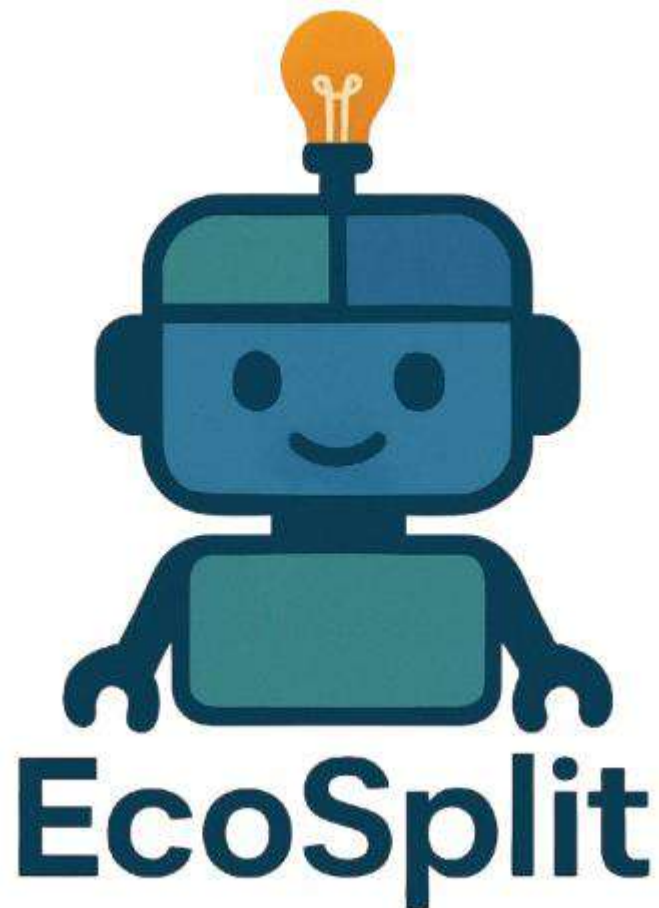


Figura 64

Ensamblaje listo



3.1.3.4 Recolección de datos

La recolección de datos de eventos de segregación mediante el proceso manual para el relleno de la primera ficha se llevó a cabo durante 20 días empezando el 12 de mayo en el aula, donde se supervisó a los 15 alumnos del aula durante todo el turno mañana de lunes a viernes. Los alumnos de dicha sección contaban con cuatro recipientes correspondientes a las cuatro categorías (Aprovechables, No Aprovechables, Orgánico y Peligrosos) para segregar sus residuos. Cada evento de segregación durante dichos días se registró completando los datos en las columnas correspondientes a la primera ficha hasta el último día, 6 de junio. Se debe notar que en la columna de “Predicción” todos los valores son “Ninguno” ya que esta propiedad no es de la naturaleza del proceso manual, y solo el proceso con el prototipo tiene valores en esta columna que son el dato del residuo que infiere el prototipo.

Tabla 3

Recolección de datos de segregación mediante el proceso manual por los estudiantes de 5to año “A”

Fecha	Tipo de residuo observado	Predicción	¿Fue correctamente segregado? (Sí/No)	Contenedor al que fue dirigido	Contenedor esperado
2025-05-12	Retazos de cartón	Ninguna	No	Orgánico	Aprovechable
2025-05-12	Gaseosa en Lata Oro Kola 355 mL	Ninguna	No	No aprovechable	Aprovechable
2025-05-12	Vaso de poliestireno blanco pequeño	Ninguna	Si	No aprovechable	No aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-05-12	Envoltura de galleta chips ahoy 37 g	Ninguna	No	Orgánico	No aprovechable
2025-05-13	Pan integral	Ninguna	No	No aprovechable	Orgánico
2025-05-13	Vaso de poliestireno blanco pequeño	Ninguna	No	Aprovechable	No aprovechable
2025-05-13	Botella plástica Inca Kola 500 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-05-13	Late de leche bonle defensis 390 g	Ninguna	Si	Aprovechable	Aprovechable
2025-05-14	Tabletas de pastillas	Ninguna	Si	Peligroso	Peligroso
2025-05-14	Vaso de poliestireno con diseño	Ninguna	No	Peligroso	No aprovechable
2025-05-14	Envoltura de galleta ritz 52.5 g	Ninguna	Si	No aprovechable	No aprovechable
2025-05-14	Envoltura panadol antigripal verde	Ninguna	No	Orgánico	Peligroso
2025-05-14	Cascara de huevo partida a la mitad	Ninguna	No	Peligroso	Orgánico
2025-05-15	Restos de manzana	Ninguna	Si	Orgánico	Orgánico
2025-05-15	Sobres de anís	Ninguna	No	Peligroso	Orgánico
2025-05-15	Cascara de platano	Ninguna	Si	Orgánico	Orgánico
2025-05-15	Envase de agua oxigenada 120 mL	Ninguna	Si	Peligroso	Peligroso
2025-05-16	Botella plástica Inca Kola 500 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-05-16	Late de leche bonle defensis 390 g	Ninguna	No	Peligroso	Aprovechable
2025-05-16	Bolas de papel	Ninguna	No	Peligroso	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-05-16	Retazos de cartón	Ninguna	No	No aprovechable	Aprovechable
2025-05-16	Gaseosa en Lata Oro Kola 355 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-05-16	Vaso de poliestireno blanco pequeño	Ninguna	No	Peligroso	No aprovechable
2025-05-19	Foco LED	Ninguna	Si	Peligroso	Peligroso
2025-05-19	Vaso de poliestireno con diseño	Ninguna	No	Aprovechable	No aprovechable
2025-05-19	Vaso de poliestireno blanco pequeño	Ninguna	Si	No aprovechable	No aprovechable
2025-05-19	Vaso de plástico transparente	Ninguna	Si	No aprovechable	No aprovechable
2025-05-20	Cascara de huevo partida a la mitad	Ninguna	No	Aprovechable	Orgánico
2025-05-20	Envoltura panadol antigripal verde	Ninguna	Si	Peligroso	Peligroso
2025-05-20	Sobres de anís	Ninguna	Si	Orgánico	Orgánico
2025-05-20	Tabletas de pastillas	Ninguna	No	Aprovechable	Peligroso
2025-05-20	Pan integral	Ninguna	Si	Orgánico	Orgánico
2025-05-21	Vaso de poliestireno blanco pequeño	Ninguna	No	Aprovechable	No aprovechable
2025-05-21	Envoltura de galleta ritz 52.5 g	Ninguna	Si	No aprovechable	No aprovechable
2025-05-21	Retazos de cartón	Ninguna	No	No aprovechable	Aprovechable
2025-05-21	Cascara de huevo partida a la mitad	Ninguna	Si	Orgánico	Orgánico

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-05-22	Restos de manzana	Ninguna	No	Aprovechable	Orgánico
2025-05-22	Sobres de anís	Ninguna	No	No aprovechable	Orgánico
2025-05-22	Cascara de platano	Ninguna	Si	Orgánico	Orgánico
2025-05-22	Pan integral	Ninguna	No	No aprovechable	Orgánico
2025-05-22	Retazos de cartón	Ninguna	Si	Aprovechable	Aprovechable
2025-05-23	Botella plástica Inca Kola 500 mL	Ninguna	No	Orgánico	Aprovechable
2025-05-23	Late de leche bonle defensis 390 g	Ninguna	Si	Aprovechable	Aprovechable
2025-05-23	Bolas de papel	Ninguna	Si	Aprovechable	Aprovechable
2025-05-23	Retazos de cartón	Ninguna	No	Orgánico	Aprovechable
2025-05-23	Gaseosa en Lata Oro Kola 355 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-05-26	Vaso de poliestireno blanco pequeño	Ninguna	Si	No aprovechable	No aprovechable
2025-05-26	Envoltura de galleta chips ahoy 37 g	Ninguna	No	Orgánico	No aprovechable
2025-05-26	Tabletas de pastillas	Ninguna	Si	Peligroso	Peligroso
2025-05-26	Vaso de poliestireno con diseño	Ninguna	No	Peligroso	No aprovechable
2025-05-26	Envase de agua oxigenada 120 mL	Ninguna	Si	Peligroso	Peligroso
2025-05-27	Vaso de plástico transparente	Ninguna	Si	No aprovechable	No aprovechable
2025-05-27	Cascara de huevo partida a la mitad	Ninguna	No	Peligroso	Orgánico
2025-05-27	Restos de manzana	Ninguna	Si	Orgánico	Orgánico

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-05-27	Sobres de anís	Ninguna	Si	Orgánico	Orgánico
2025-05-27	Cascara de platano	Ninguna	Si	Orgánico	Orgánico
2025-05-28	Pan integral	Ninguna	Si	Orgánico	Orgánico
2025-05-28	Vaso de plástico transparente	Ninguna	No	Orgánico	No aprovechable
2025-05-28	Vaso de poliestireno blanco pequeño	Ninguna	Si	No aprovechable	No aprovechable
2025-05-28	Restos de manzana	Ninguna	Si	Orgánico	Orgánico
2025-05-28	Sobres de anís	Ninguna	Si	Orgánico	Orgánico
2025-05-29	Envoltura panadol antigripal verde	Ninguna	No	Orgánico	Peligroso
2025-05-29	Envase de agua oxigenada 120 mL	Ninguna	Si	Peligroso	Peligroso
2025-05-29	Botella plástica Inca Kola 500 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-05-29	Late de leche bonle defensis 390 g	Ninguna	Si	Aprovechable	Aprovechable
2025-05-29	Bolas de papel	Ninguna	No	Peligroso	Aprovechable
2025-05-30	Retazos de cartón	Ninguna	Si	Aprovechable	Aprovechable
2025-05-30	Gaseosa en Lata Oro Kola 355 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-05-30	Vaso de poliestireno blanco pequeño	Ninguna	No	Peligroso	No aprovechable
2025-05-30	Envoltura de galleta chips ahoy 37 g	Ninguna	Si	No aprovechable	No aprovechable
2025-05-30	Vaso de poliestireno con diseño	Ninguna	Si	No aprovechable	No aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-06-02	Envoltura de galleta ritz 52.5 g	Ninguna	No	Peligroso	No aprovechable
2025-06-02	Vaso de plástico transparente	Ninguna	Si	No aprovechable	No aprovechable
2025-06-02	Vaso de poliestireno blanco pequeño	Ninguna	Si	No aprovechable	No aprovechable
2025-06-02	Restos de manzana	Ninguna	Si	Orgánico	Orgánico
2025-06-02	Sobres de anís	Ninguna	Si	Orgánico	Orgánico
2025-06-03	Cascara de platano	Ninguna	No	Peligroso	Orgánico
2025-06-03	Pan integral	Ninguna	Si	Orgánico	Orgánico
2025-06-03	Late de leche sin etiqueta	Ninguna	No	Orgánico	Aprovechable
2025-06-03	Botella plástica Inca Kola 500 mL	Ninguna	Si	Aprovechable	Aprovechable
2025-06-03	Botella plástica Inca Kola 500 mL	Ninguna	No	Peligroso	Aprovechable
2025-06-03	Late de leche bonle defensis 390 g	Ninguna	Si	Aprovechable	Aprovechable
2025-06-03	Bolas de papel	Ninguna	Si	Aprovechable	Aprovechable
2025-06-03	Vaso de poliestireno con diseño	Ninguna	No	Peligroso	No aprovechable
2025-06-03	Envoltura de galleta ritz 52.5 g	Ninguna	Si	No aprovechable	No aprovechable
2025-06-03	Vaso de plástico transparente	Ninguna	No	Peligroso	No aprovechable
2025-06-04	Cascara de huevo partida a la mitad	Ninguna	Si	Orgánico	Orgánico
2025-06-04	Restos de manzana	Ninguna	Si	Orgánico	Orgánico
2025-06-04	Retazos de cartón	Ninguna	Si	Aprovechable	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-06-04	Gaseosa en Lata Oro Kola 355 mL	Ninguna	No	Peligroso	Aprovechable
2025-06-04	Vaso de poliestireno blanco pequeño	Ninguna	Si	No aprovechable	No aprovechable
2025-06-05	Envoltura de galleta chips ahoy 37 g	Ninguna	Si	No aprovechable	No aprovechable
2025-06-05	Pan integral	Ninguna	No	Peligroso	Orgánico
2025-06-05	Late de leche sin etiqueta	Ninguna	Si	Aprovechable	Aprovechable
2025-06-05	Botella plástica Inca Kola 500 mL	Ninguna	No	Orgánico	Aprovechable
2025-06-05	Late de leche bonle defensis 390 g	Ninguna	Si	Aprovechable	Aprovechable
2025-06-06	Bolas de papel	Ninguna	Si	Aprovechable	Aprovechable
2025-06-06	Vaso de poliestireno blanco pequeño	Ninguna	No	Orgánico	No aprovechable
2025-06-06	Envoltura de galleta chips ahoy 37 g	Ninguna	Si	No aprovechable	No aprovechable

Una vez concluido los 20 días de recolección de datos de los eventos de segregación con el proceso manual, se continuó con la siguiente recolección de datos de los eventos de segregación con el uso del prototipo durante 20 días desde el 9 de junio hasta el 4 de julio, en esta etapa los cuatro contenedores correspondientes a las categorías de residuos se encontraban en el interior del prototipo. El prototipo fue trasladado al aula indicada en la institución educativa el día 9 de junio con el permiso previo de la Señora Directora Victoria Huaylinos Gonzales, se registró todos los eventos de segregación en las columnas de la segunda ficha durante el turno mañana de los alumnos en todo momento bajo mi

supervisión. La columna de “Predicción” fue completada en base al registro automático que realiza el prototipo en un documento excel donde indica la clase del residuo que se ha inferido y cuyo nombre finalmente esta relacionada al nombre exacto como lo indica la tabla 4 que es la correspondencia entre clases y nombres de los residuos. También es importante recalcar que antes de comenzar con la recolección de los datos se instruyo a cada estudiante en el uso del prototipo como se observa en la imagen siguiente:

Tabla 4

Correspondencia entre clases y nombres de residuos

Clases de residuos	Nombre de residuos
Aguaoxygenada	Envase de agua oxygenada 120 mL
Carton	Retazos de cartón
Cascara	Cascara de huevo partida a la mitad
Foco	Foco LED
Galleta1	Envoltura de galleta ritz 52.5 g
Galleta2	Envoltura de galleta chips ahoy 37 g
Gaseosalata	Gaseosa en Lata Oro Kola 355 mL
Incakola	Botella plástica Inca Kola 500 mL
Jeringa	Jeringa
Lata	Lata de leche sin etiqueta
Lataleche	Lata de leche bonle defensis 390 g

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

Manzana	Restos de manzana
Medicina1	Tabletas de pastillas
Medicina3	Envoltura panadol antigripal verde
Pan	Pan integral
Papel	Bolas de papel
Pila	Pila panasonic
Platano	Cascara de platano
Te	Sobres de anís
Vasoplastico	Vaso de plástico transparente
Vasopoliblanco	Vaso de poliestireno blanco pequeño
Vasopolicolor	Vaso de poliestireno con diseño

Figura 65

Charla y pruebas sobre el uso correcto del prototipo



DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

A continuación se presenta la segunda ficha de recolección de datos:

Tabla 5

Recolección de datos de segregación mediante el uso del prototipo por parte de los estudiantes de 5to año "A"

Fecha	Tipo de residuo observado	Predicción	¿Fue correctamente segregado? (Sí/No)	Contenedor al que fue dirigido	Contenedor esperado
2025-06-09	Vaso de poliestireno con diseño	Vaso de poliestireno con diseño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-09	Vaso de poliestireno con diseño	Vaso de poliestireno con diseño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-09	Envoltura de galleta ritz 52.5 g	Envoltura de galleta ritz 52.5 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-09	Vaso de plástico transparente	Vaso de plástico transparente	<i>Si</i>	No aprovechable	No aprovechable
2025-06-10	Restos de manzana	Envoltura panadol antigripal verde	<i>No</i>	Peligroso	Organico
2025-06-10	Botella plástica Inca Kola 500 mL	Botella plástica Inca Kola 500 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-10	Cascara de huevo partida a la mitad	Tabletas de pastillas	<i>No</i>	Peligroso	Organico
2025-06-11	Vaso de poliestireno con diseño	Vaso de poliestireno con diseño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-11	Pan integral	Pan integral	<i>Si</i>	Organico	Organico
2025-06-11	Foco LED	Foco LED	<i>Si</i>	Peligroso	Peligroso
2025-06-11	Lata de leche sin etiqueta	Lata de leche sin etiqueta	<i>Si</i>	Aprovechable	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-06-11	Gaseosa en Lata Oro Kola 355 mL	Gaseosa en Lata Oro Kola 355 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-12	Cascara de huevo partida a la mitad	Cascara de huevo partida a la mitad	<i>Si</i>	Organico	Organico
2025-06-12	Vaso de poliestireno con diseño	Vaso de poliestireno con diseño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-12	Vaso de poliestireno con diseño	Vaso de poliestireno con diseño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-12	Lata de leche sin etiqueta	Lata de leche sin etiqueta	<i>Si</i>	Aprovechable	Aprovechable
2025-06-12	Pan integral	Pan integral	<i>Si</i>	Organico	Organico
2025-06-12	Vaso de poliestireno blanco pequeño	Vaso de poliestireno blanco pequeño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-13	Vaso de plástico transparente	Vaso de plástico transparente	<i>Si</i>	No aprovechable	No aprovechable
2025-06-13	Bolas de papel	Bolas de papel	<i>Si</i>	Aprovechable	Aprovechable
2025-06-13	Gaseosa en Lata Oro Kola 355 mL	Gaseosa en Lata Oro Kola 355 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-13	Vaso de poliestireno blanco pequeño	Vaso de poliestireno blanco pequeño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-13	Cascara de platano	Cascara de platano	<i>Si</i>	Organico	Organico
2025-06-13	Gaseosa en Lata Oro Kola 355 mL	Gaseosa en Lata Oro Kola 355 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-16	Vaso de plástico transparente	Vaso de plástico transparente	<i>Si</i>	No aprovechable	No aprovechable
2025-06-16	Tabletas de pastillas	Sobres de anís	No	Organico	Peligroso
2025-06-16	Envoltura de galleta chips ahoy 37 g	Envoltura de galleta chips ahoy 37 g	<i>Si</i>	No aprovechable	No aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-06-16	Lata de leche bonle defensis 390 g	Lata de leche bonle defensis 390 g	<i>Si</i>	Aprovechable	Aprovechable
2025-06-17	Sobres de anís	Tabletas de pastillas	No	Peligroso	Organico
2025-06-17	Envoltura de galleta ritz 52.5 g	Envoltura de galleta ritz 52.5 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-17	Envoltura de galleta ritz 52.5 g	Envoltura de galleta ritz 52.5 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-17	Bolas de papel	Bolas de papel	<i>Si</i>	Aprovechable	Aprovechable
2025-06-17	Envoltura de galleta chips ahoy 37 g	Envoltura de galleta chips ahoy 37 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-17	Lata de leche bonle defensis 390 g	Lata de leche bonle defensis 390 g	<i>Si</i>	Aprovechable	Aprovechable
2025-06-18	Lata de leche sin etiqueta	Lata de leche sin etiqueta	<i>Si</i>	Aprovechable	Aprovechable
2025-06-18	Lata de leche bonle defensis 390 g	Lata de leche bonle defensis 390 g	<i>Si</i>	Aprovechable	Aprovechable
2025-06-18	Pan integral	Pan integral	<i>Si</i>	Organico	Organico
2025-06-18	Vaso de poliestireno blanco pequeño	Vaso de poliestireno blanco pequeño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-18	Bolas de papel	Bolas de papel	<i>Si</i>	Aprovechable	Aprovechable
2025-06-18	Retazos de cartón	Retazos de cartón	<i>Si</i>	Aprovechable	Aprovechable
2025-06-19	Retazos de cartón	Retazos de cartón	<i>Si</i>	Aprovechable	Aprovechable
2025-06-19	Sobres de anís	Sobres de anís	<i>Si</i>	Organico	Organico
2025-06-19	Pan integral	Pan integral	<i>Si</i>	Organico	Organico
2025-06-19	Retazos de cartón	Retazos de cartón	<i>Si</i>	Aprovechable	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-06-20	Envoltura de galleta chips ahoy 37 g	Envoltura de galleta chips ahoy 37 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-20	Envoltura de galleta ritz 52.5 g	Envoltura de galleta ritz 52.5 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-20	Restos de manzana	Restos de manzana	<i>Si</i>	Organico	Organico
2025-06-20	Lata de leche sin etiqueta	Lata de leche sin etiqueta	<i>Si</i>	Aprovechable	Aprovechable
2025-06-20	Retazos de cartón	Retazos de cartón	<i>Si</i>	Aprovechable	Aprovechable
2025-06-20	Cascara de huevo partida a la mitad	Cascara de huevo partida a la mitad	<i>Si</i>	Organico	Organico
2025-06-23	Botella plástica Inca Kola 500 mL	Botella plástica Inca Kola 500 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-23	Cascara de platano	Cascara de platano	<i>Si</i>	Organico	Organico
2025-06-23	Lata de leche sin etiqueta	Lata de leche sin etiqueta	<i>Si</i>	Aprovechable	Aprovechable
2025-06-23	Vaso de poliestireno con diseño	Vaso de poliestireno con diseño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-23	Restos de manzana	Restos de manzana	<i>Si</i>	Organico	Organico
2025-06-24	Cascara de platano	Cascara de platano	<i>Si</i>	Organico	Organico
2025-06-24	Vaso de poliestireno blanco pequeño	Vaso de poliestireno blanco pequeño	<i>Si</i>	No aprovechable	No aprovechable
2025-06-25	Vaso de plástico transparente	Vaso de plástico transparente	<i>Si</i>	No aprovechable	No aprovechable
2025-06-25	Botella plástica Inca Kola 500 mL	Botella plástica Inca Kola 500 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-25	Gaseosa en Lata Oro Kola 355 mL	Gaseosa en Lata Oro Kola 355 mL	<i>Si</i>	Aprovechable	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-06-25	Cascara de huevo partida a la mitad	Cascara de huevo partida a la mitad	<i>Si</i>	Organico	Organico
2025-06-26	Pan integral	Pan integral	<i>Si</i>	Organico	Organico
2025-06-26	Envoltura de galleta chips ahoy 37 g	Envoltura de galleta chips ahoy 37 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-26	Botella plástica Inca Kola 500 mL	Botella plástica Inca Kola 500 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-26	Cascara de platano	Cascara de platano	<i>Si</i>	Organico	Organico
2025-06-26	Restos de manzana	Restos de manzana	<i>Si</i>	Organico	Organico
2025-06-27	Vaso de plástico transparente	Vaso de plástico transparente	<i>Si</i>	No aprovechable	No aprovechable
2025-06-27	Restos de manzana	Restos de manzana	<i>Si</i>	Organico	Organico
2025-06-27	Vaso de plástico transparente	Vaso de plástico transparente	<i>Si</i>	No aprovechable	No aprovechable
2025-06-30	Gaseosa en Lata Oro Kola 355 mL	Gaseosa en Lata Oro Kola 355 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-06-30	Cascara de platano	Cascara de platano	<i>Si</i>	Organico	Organico
2025-06-30	Envoltura de galleta chips ahoy 37 g	Envoltura de galleta chips ahoy 37 g	<i>Si</i>	No aprovechable	No aprovechable
2025-06-30	Bolas de papel	Bolas de papel	<i>Si</i>	Aprovechable	Aprovechable
2025-06-30	Pan integral	Pan integral	<i>Si</i>	Organico	Organico
2025-06-30	Vaso de poliestireno blanco pequeño	Vaso de poliestireno blanco pequeño	<i>Si</i>	No aprovechable	No aprovechable
2025-07-01	Sobres de anís	Sobres de anís	<i>Si</i>	Organico	Organico
2025-07-01	Lata de leche bonle defensis 390 g	Lata de leche bonle defensis 390 g	<i>Si</i>	Aprovechable	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

2025-07-01	Sobres de anís	Sobres de anís	<i>Si</i>	Organico	Organico
2025-07-01	Envoltura de galleta chips ahoy 37 g	Envoltura de galleta chips ahoy 37 g	<i>Si</i>	No aprovechable	No aprovechable
2025-07-02	Sobres de anís	Sobres de anís	<i>Si</i>	Organico	Organico
2025-07-02	Envoltura de galleta ritz 52.5 g	Envoltura de galleta ritz 52.5 g	<i>Si</i>	No aprovechable	No aprovechable
2025-07-02	Envoltura de galleta ritz 52.5 g	Envoltura de galleta ritz 52.5 g	<i>Si</i>	No aprovechable	No aprovechable
2025-07-03	Vaso de poliestireno blanco pequeño	Vaso de poliestireno blanco pequeño	<i>Si</i>	No aprovechable	No aprovechable
2025-07-03	Lata de leche bonle defensis 390 g	Lata de leche bonle defensis 390 g	<i>Si</i>	Aprovechable	Aprovechable
2025-07-03	Restos de manzana	Restos de manzana	<i>Si</i>	Organico	Organico
2025-07-03	Cascara de huevo partida a la mitad	Cascara de huevo partida a la mitad	<i>Si</i>	Organico	Organico
2025-07-03	Cascara de platano	Cascara de platano	<i>Si</i>	Organico	Organico
2025-07-03	Botella plástica Inca Kola 500 mL	Botella plástica Inca Kola 500 mL	<i>Si</i>	Aprovechable	Aprovechable
2025-07-04	Retazos de cartón	Retazos de cartón	<i>Si</i>	Aprovechable	Aprovechable
2025-07-04	Sobres de anís	Sobres de anís	<i>Si</i>	Organico	Organico
2025-07-04	Cascara de huevo partida a la mitad	Cascara de huevo partida a la mitad	<i>Si</i>	Organico	Organico
2025-07-04	Bolas de papel	Bolas de papel	<i>Si</i>	Aprovechable	Aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

Figura 66

Registro automático del prototipo parte 1

	A	B	C	D
1	Fecha	Hora	Objeto	Categoría
2	2025-06-09	09:42:08	Vasopolicolor	No aprovechable
3	2025-06-09	09:57:01	Vasopolicolor	No aprovechable
4	2025-06-09	10:46:40	Galleta1	No aprovechable
5	2025-06-09	11:01:59	Vasoplastico	No aprovechable
6	2025-06-10	09:02:03	Medicina3	Peligroso
7	2025-06-10	09:03:22	Incakola	Aprovechable
8	2025-06-10	11:52:33	Medicina1	Peligroso
9	2025-06-11	08:10:35	Vasopolicolor	No aprovechable
10	2025-06-11	08:16:57	Pan	Organico
11	2025-06-11	09:30:26	Foco	Peligroso
12	2025-06-11	09:40:55	Lata	Aprovechable
13	2025-06-11	11:47:55	Gaseosalata	Aprovechable
14	2025-06-12	08:54:16	Cascara	Organico
15	2025-06-12	10:17:55	Vasopolicolor	No aprovechable
16	2025-06-12	10:49:28	Vasopolicolor	No aprovechable
17	2025-06-12	11:22:22	Lata	Aprovechable
18	2025-06-12	11:33:43	Pan	Organico
19	2025-06-12	11:50:22	Vasopoliblanco	No aprovechable
20	2025-06-13	09:31:02	Vasoplastico	No aprovechable
21	2025-06-13	11:01:45	Papel	Aprovechable
22	2025-06-13	11:59:29	Gaseosalata	Aprovechable
23	2025-06-13	12:14:42	Vasopoliblanco	No aprovechable
24	2025-06-13	12:16:41	Platano	Organico
25	2025-06-13	12:38:53	Gaseosalata	Aprovechable
26	2025-06-16	08:28:03	Vasoplastico	No aprovechable
27	2025-06-16	10:56:15	Te	Organico
28	2025-06-16	12:32:49	Galleta2	No aprovechable
29	2025-06-16	12:53:33	Lataleche	Aprovechable
30	2025-06-17	08:26:05	Medicina1	Peligroso
31	2025-06-17	09:13:26	Galleta1	No aprovechable
32	2025-06-17	09:42:04	Galleta1	No aprovechable
33	2025-06-17	09:46:42	Papel	Aprovechable
34	2025-06-17	11:17:30	Galleta2	No aprovechable
35	2025-06-17	11:51:53	Lataleche	Aprovechable
36	2025-06-18	08:59:11	Lata	Aprovechable
37	2025-06-18	09:09:32	Lataleche	Aprovechable
38	2025-06-18	09:38:03	Pan	Organico
39	2025-06-18	09:51:31	Vasopoliblanco	No aprovechable
40	2025-06-18	11:04:41	Papel	Aprovechable
41	2025-06-18	11:40:33	Carton	Aprovechable
42	2025-06-19	08:52:18	Carton	Aprovechable
43	2025-06-19	09:31:37	Te	Organico
44	2025-06-19	10:08:55	Pan	Organico
45	2025-06-19	11:02:23	Carton	Aprovechable
46	2025-06-20	08:09:15	Galleta2	No aprovechable
47	2025-06-20	09:27:48	Galleta1	No aprovechable
48	2025-06-20	10:01:27	Manzana	Organico
49	2025-06-20	10:49:54	Lata	Aprovechable
50	2025-06-20	11:43:40	Carton	Aprovechable
51	2025-06-20	12:51:16	Cascara	Organico
52	2025-06-23	08:13:13	Incakola	Aprovechable
53	2025-06-23	09:11:24	Platano	Organico
54	2025-06-23	09:58:23	Lata	Aprovechable
55	2025-06-23	10:34:29	Vasopolicolor	No aprovechable
56	2025-06-23	12:04:38	Manzana	Organico
57	2025-06-24	08:16:32	Platano	Organico
58	2025-06-24	08:53:50	Vasopoliblanco	No aprovechable
59	2025-06-25	08:25:01	Vasoplastico	No aprovechable
60	2025-06-25	09:34:55	Incakola	Aprovechable
61	2025-06-25	10:02:01	Gaseosalata	Aprovechable
62	2025-06-25	12:55:34	Cascara	Organico
63	2025-06-26	08:47:27	Pan	Organico
64	2025-06-26	10:06:53	Galleta2	No aprovechable
65	2025-06-26	10:11:51	Incakola	Aprovechable
66	2025-06-26	11:09:57	Platano	Organico
67	2025-06-26	12:52:53	Manzana	Organico
68	2025-06-27	08:22:51	Vasoplastico	No aprovechable

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

Figura 67

Registro automático del prototipo parte 2

69	2025-06-27 11:27:44	Manzana	Organico
70	2025-06-27 12:02:52	Vasoplastico	No aprovechable
71	2025-06-30 08:33:21	Gaseosalata	Aprovechable
72	2025-06-30 08:46:53	Platano	Organico
73	2025-06-30 11:11:53	Galleta2	No aprovechable
74	2025-06-30 11:34:54	Papel	Aprovechable
75	2025-06-30 11:38:15	Pan	Organico
76	2025-06-30 12:58:43	Vasopoliblanco	No aprovechable
77	2025-07-01 09:13:28	Te	Organico
78	2025-07-01 10:00:26	Lataleche	Aprovechable
79	2025-07-01 12:12:07	Te	Organico
80	2025-07-01 12:44:41	Galleta2	No aprovechable
81	2025-07-02 08:27:28	Te	Organico
82	2025-07-02 11:28:28	Galleta1	No aprovechable
83	2025-07-02 12:37:10	Galleta1	No aprovechable
84	2025-07-03 09:03:06	Vasopoliblanco	No aprovechable
85	2025-07-03 10:13:08	Lataleche	Aprovechable
86	2025-07-03 11:12:38	Manzana	Organico
87	2025-07-03 12:19:11	Cascara	Organico
88	2025-07-03 12:20:30	Platano	Organico
89	2025-07-03 12:28:37	Incakola	Aprovechable
90	2025-07-04 08:00:00	Carton	Aprovechable
91	2025-07-04 08:51:53	Te	Organico
92	2025-07-04 11:27:10	Cascara	Organico
93	2025-07-04 12:37:10	Papel	Aprovechable

También se muestra los gráficos de barras generados automáticamente el último día de prueba, tanto semanalmente como mensualmente, teniendo que mensualmente a dicho día se tuvo un número de 30 residuos no aprovechables, 30 residuos aprovechables, 28 residuos orgánicos y 4 residuos peligrosos mientras que semanalmente a ese día se tuvo 11 residuos organicos, 8 residuos no aprovechables, 7 residuos aprovechables y 0 residuos peligrosos.

Figura 68

Gráfico de segregación de residuos mensual 2025-06-04 a 2025-07-04

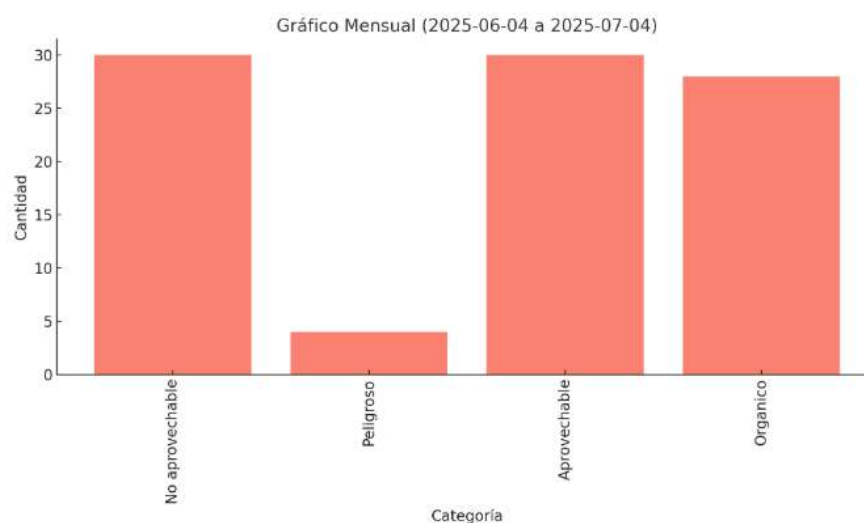
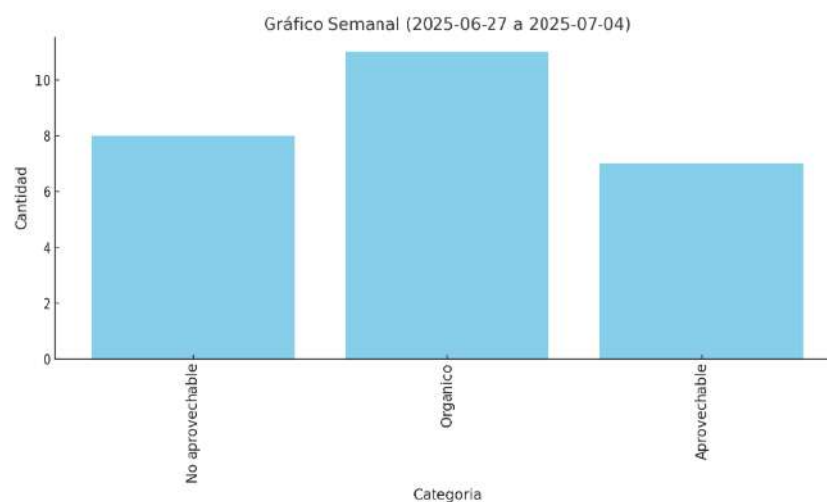


Figura 69

Gráfico de segregación de residuos semanal 2025-06-27 a 2025-07-04



3.2 Análisis descriptivo

En esta sección se presentan los resultados descriptivos de la variable dependiente: Calidad de la Segregación de Residuos. El objetivo es resumir y comparar el desempeño del proceso de segregación antes de la intervención (pre-test, método manual) y después de la implementación del prototipo "EcoSplit" (post-test).

3.2.1 Indicador: Porcentaje de segregación efectiva

Los resultados descriptivos se resumen en la tabla 6, la cual muestra los recuentos y porcentajes de residuos clasificados de manera correcta e incorrecta para cada método. A simple vista, se observa una mejora en la efectividad de la segregación al utilizar el prototipo, pasando de un 40.4% de aciertos en la fase manual a un 95.7% en la fase con el prototipo.

Tabla 6

Estadísticos descriptivos de la Calidad de la Segregación de Residuos antes y después de implementar el prototipo

Tabla cruzada Método de Segregación*Segregación de residuo

			Segregación de residuo		Total
			Correcto	Incorrecto	
Método de Segregación	Manual	Recuento	40	59	99
		% dentro de Método de Segregación	40.4%	59.6%	100.0%
	Prototipo	Recuento	88	4	92
		% dentro de Método de Segregación	95.7%	4.3%	100.0%
Total	Recuento	128	63	191	
	% dentro de Método de Segregación	67.0%	33.0%	100.0%	

Partiendo de los resultados previamente descritos en la tabla 3 y 5, y aplicando la fórmula del Porcentaje de Segregación Efectiva definida en la metodología, se puede evidenciar un cambio significativo. Antes de la implementación del prototipo, el porcentaje de segregación efectiva con el proceso manual tuvo un valor del 40.40%, obtenido al aplicar la fórmula a los datos observados:

$$\text{Porcentaje de segregación efectiva pre – test} = \frac{40 \text{ residuos correctos}}{99 \text{ residuos totales}} \times 100 = 40.40\%$$

Posteriormente, con el uso del prototipo, este indicador subió a un valor de 95.62%, calculado de la siguiente manera:

$$\text{Porcentaje de segregación efectiva post - test} = \frac{88 \text{ residuos correctos}}{92 \text{ residuos totales}} \times 100 = 95.62\%$$

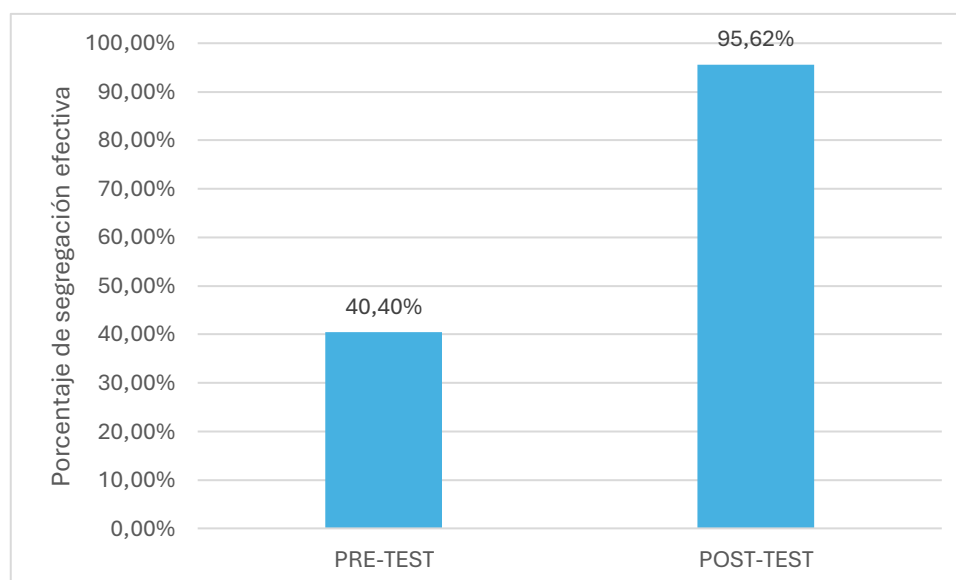
Esto representa una mejora neta del 55.22 puntos porcentuales en la efectividad del proceso de segregación con el uso del prototipo.

$$\text{Mejora Neta del Porcentaje de Segregación Efectiva con el prototipo} = 95.62\% - 40.40\% = 55.22\%$$

Este cambio también se refleja en la drástica reducción de errores, que pasaron de constituir el 59.6% de los casos en el pre-test (59 de 99 residuos mal segregados) a solo un 4.3% en el post-test (únicamente 4 de 92 residuos mal segregados). Dicha variación se aprecia de manera más clara en la figura 70.

Figura 70

Resultado del Porcentaje de Segregación Efectiva antes y después del uso del prototipo



3.3 Análisis inferencial

El propósito del análisis inferencial es proporcionar la evidencia estadística para contrastar la Hipótesis Específica 4, la cual postula que: "Si se utiliza el prototipo en la I.E. 3049 Imperio del Tahuantinsuyo, entonces el indicador porcentaje de segregación efectiva será estadísticamente superior al obtenido mediante el proceso manual." Se busca determinar si la mejora observada en el análisis descriptivo es estadísticamente significativa y no producto del azar.

3.3.1 Justificación y selección de la prueba estadística

Para realizar la contrastación de la hipótesis, el primer paso fue seleccionar la prueba estadística adecuada. A diferencia de los análisis que comparan medias de datos numéricos (los cuales requerirían una prueba de normalidad como Shapiro-Wilk), este estudio trabaja con variables de naturaleza categórica y nominal:

- Variable Independiente: Método de Segregación (con categorías: Manual, Prototipo).
- Variable Dependiente: Calidad de la Segregación (con categorías: Correcto, Incorrecto).

Adicionalmente, los datos provienen de muestras independientes, ya que el grupo de 99 residuos del pre-test no está relacionado de forma pareada con el grupo de 92 residuos del post-test.

Dadas estas condiciones, se determinó que la prueba más idónea y robusta es la Prueba de Chi-cuadrado (χ^2) de independencia. Esta prueba no paramétrica está diseñada específicamente para evaluar si existe una asociación significativa

entre dos variables categóricas, comparando las frecuencias observadas con las que se esperarían si no existiera relación alguna entre ellas.

3.3.2 Contrastación de hipótesis

Para someter a prueba la Hipótesis Específica 4, se plantearon las siguientes hipótesis estadísticas, estableciendo un nivel de significancia (alfa) de 0.05:

- H_0 (Hipótesis Nula): No existe una asociación estadísticamente significativa entre el Método de Segregación y la Calidad de la Segregación. El indicador "porcentaje de segregación efectiva" es independiente del método utilizado.
- H_1 (Hipótesis Alternativa): Existe una asociación estadísticamente significativa entre el Método de Segregación y la Calidad de la Segregación, indicando que el indicador de efectividad varía dependiendo del método utilizado.

Los resultados de la prueba, ejecutada mediante el software IBM SPSS Statistics, se presentan en la tabla 7.

Tabla 7

Resultados de la Prueba de Chi-Cuadrado de Independencia

Pruebas de chi-cuadrado					
	Valor	gl	Significación asintótica (bilateral)	Significación exacta (bilateral)	Significación exacta (unilateral)
Chi-cuadrado de Pearson	65.848 ^a	1	<.001		
Corrección de continuidad ^b	63.372	1	<.001		
Razón de verosimilitud	75.732	1	<.001		
Prueba exacta de Fisher				<.001	<.001
Asociación lineal por lineal	65.503	1	<.001		
N de casos válidos	191				

a. 0 casillas (.0%) han esperado un recuento menor que 5. El recuento mínimo esperado es 30.35.

b. Sólo se ha calculado para una tabla 2x2

El análisis arroja un valor para el estadístico Chi-cuadrado de Pearson de $\chi^2(1, N = 191) = 65.848$, con una significación (valor p) de $p < .001$. La nota al pie de la tabla confirma la validez del análisis, indicando que se cumplen los supuestos de la prueba.

3.3.2 Decisión y Conclusión sobre la Hipótesis

Dado que el valor p ($< .001$) es menor que el nivel de significancia establecido ($\alpha = 0.05$), se rechaza la hipótesis nula (H_0). Este resultado indica que existe una fuerte evidencia estadística para afirmar que la mejora en el "porcentaje de segregación efectiva" no es una casualidad. Se acepta, por tanto, la hipótesis alternativa (H_1), lo que permite comprobar de manera contundente la Hipótesis Específica 4. Se concluye que el uso del prototipo "EcoSplit" mejora de manera estadísticamente significativa el indicador de "porcentaje de segregación efectiva" en comparación con el proceso manual. Este hallazgo, a su vez, proporciona el soporte empírico principal para validar la Hipótesis General de la investigación.

3.4 Contrastación de las hipótesis de la investigación

Una vez presentados los resultados descriptivos e inferenciales, se procede a la contrastación formal de cada una de las hipótesis específicas planteadas en el estudio.

- Comprobación de Hipótesis 1:

Hipótesis 1: Si se identifican correctamente los requerimientos funcionales y técnicos, se podrá establecer una base adecuada para el diseño.

Esta hipótesis se comprueba mediante la demostración de que cada decisión de diseño tomada en el proyecto es una respuesta directa y necesaria

a los requerimientos establecidos. Para ilustrar esta conexión causal de manera clara, se presenta la siguiente matriz de trazabilidad que mapea los requerimientos clave con sus correspondientes decisiones en el diseño del prototipo "EcoSplit".

Tabla 8

Matriz de trazabilidad de requerimientos para el diseño del Prototipo "EcoSplit"

Requerimiento funcional (RF)/Requerimiento técnico (RT)	Decisión de diseño	Referencia/Evidencia
<p>RF: Tener 4 compartimientos internos para las 4 categorías de residuos.</p> <p>RT: Dimensiones cercanas a la del tacho Reyplast usado en la institución por los alumnos de 53x51x84 cm.</p> <p>RT: Debido a que se necesita poder supervisar los procesos del prototipo para verificar su correcto funcionamiento y diagnóstico de problemas, seleccionar un material translúcido.</p>	<ul style="list-style-type: none"> • Diseño y modelado 3D en Autodesk Inventor de una estructura con divisiones internas para 4 contenedores, definiendo las dimensiones a 46x46x100 cm. • Para la estructura principal, los contenedores y los paneles de soporte, se seleccionó acrílico transparente de 3 mm de espesor para tener visibilidad para la observación y diagnóstico de los procesos del prototipo en funcionamiento y por la facilidad y precisión de fabricación. Además, que el acrílico ofrece una rigidez estructural suficiente para la escala y las cargas ligeras del prototipo a un costo asequible. 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - Sección 3.1.3.2.7 Diseño de la estructura - Sección 3.1.3.3 Implementación (Selección del material acrílico 3 mm) - Figuras 54, 55, 56 y 57
<p>RT: Diseñar el prototipo con tal de que se puedan retirar los residuos fácilmente.</p> <p>RT: Disponer al prototipo de elementos que le permitan ser fácil movilización.</p>	<ul style="list-style-type: none"> • Se dividió el prototipo en dos partes separando la parte inferior donde se encuentran los compartimientos para su fácil retiro y pueda ser nuevamente acoplado con todos los componentes del prototipo en la parte superior mediante un encaje. • Se dispuso al prototipo de 4 ruedas locas, las cuales dos de ellas poseen un mecanismo de frenado para que posicionado en un solo lugar. 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - Sección 3.1.3.2.7 Diseño de la estructura - Sección 3.1.3.3 Implementación (Selección del material acrílico 3 mm) - Figuras 54, 55, 56 y 57
<p>RF: Recibir y procesar un residuo a la vez.</p> <p>RT:</p>	<ul style="list-style-type: none"> • Diseño de un mecanismo Iris de 1 grado de libertad para la entrada y salida de los residuos que aprovecha el giro permitido de 180° por los servomotores MG90S, cuyo tamaño de abertura fue dimensionada en base al tamaño de los residuos elegidos asegurando el ingreso 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - 3.1.3.2.5 Diseño del subsistema de actuación - Sección 3.1.3.2.7 Diseño de la estructura (Sobre el espacio de presentación)

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

<ul style="list-style-type: none"> - Diseñar mecanismos adecuados de apertura/cierre. - Contar con un espacio cuyo volumen sea suficiente para la etapa de presentación de los residuos seleccionados a la cámara 	<p>individual de los residuos al espacio de presentación.</p> <ul style="list-style-type: none"> • Para el accionamiento de los mecanismos Iris y re-dirigidor, se seleccionaron los servomotores modelo MG90S. La elección de este componente específico se basó en el giro angular permitido, relación costo-beneficio, durabilidad y fiabilidad de los engranajes metálicos, suficiencia de torque para la aplicación, facilidad de integración y control, cada uno de estos puntos se describe con mayor detalle en la sección 3.1.3.2.5 Diseño del subsistema de actuación. • El volumen del espacio de presentación a la cámara tiene dimensiones de 10x10x24 cm en base a los tamaños de los residuos seleccionados. 	<ul style="list-style-type: none"> - Figura 47
<p>RF: Depositar el residuo en el compartimiento correcto.</p> <p>RT: Diseñar un mecanismo de redirección funcional.</p>	<ul style="list-style-type: none"> • Diseño de un mecanismo re-dirigidor espacial, accionado por dos servomotores, que orienta una rampa hacia el contenedor correspondiente. 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - 3.1.3.2.5 Diseño del subsistema de actuación (Sobre el mecanismo re-dirigidor) - Figura 49
<p>RF: Realizar la identificación de forma automática.</p> <p>RT: Usar cámara de color y definir el modelo convolucional YOLO adecuado.</p>	<ul style="list-style-type: none"> • Implementación del subsistema de percepción (Con la cámara web USB de color RT-A200) y diseño del subsistema de procesamiento y decisión donde la laptop mencionada ejecuta el modelo convolucional YOLOV11 re-entrenado small, el cual fue elegido debido a que ofrece un equilibrio entre precisión y tiempo de inferencia por cada imagen con un valor de 47.5 COCO mAP_{Val}^{50-95} que representa el promedio de 10 evaluaciones con umbrales de "Intersection over Union" (IoU) crecientemente estrictos, desde una superposición del 50% hasta una casi perfecta del 95%, acompañado con latencia de inferencia de aproximadamente 2.5 milisegundos por imagen, adecuada para la rápida inferencia de los residuos. • Se estableció criterios de verificación por tiempo en la programación en python donde el modelo tenía que identificar un mismo objeto durante 10 segundos continuos para dar el resultado final de la inferencia. 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - 3.1.3.2.2 Diseño del subsistema de percepción - 3.1.3.2.4 Diseño del subsistema de procesamiento y decisión - Figura 6, 23, 24, 37 y 38
<p>RF: Ser de fácil interacción (un solo botón).</p>	<ul style="list-style-type: none"> • Diseño del subsistema de señalización/activación compuesto por un interruptor de presión para iniciar un ciclo de segregación y dos LED's 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p>

DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

<p>RT: Proveer retroalimentación visual (LED's).</p>	<p>(rojo/verde) para indicar el estado del prototipo.</p>	<p>- 3.1.3.2.3 Diseño del subsistema de señalización/activación - Figura 9</p>
<p>RT: Emplear una arquitectura de hardware de bajo costo basada en la laptop para el computo de la visión artificial y la placa Arduino Uno para el control de los actuadores.</p>	<ul style="list-style-type: none"> • Adopción de una arquitectura distribuida: el Arduino Uno gestiona el control de servomotores, y LED's, verifica si el interruptor ha sido pulsado, mientras la laptop se encarga del procesamiento de la visión artificial. 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - 3.1.3.2.3 Diseño del subsistema de señalización/activación - 3.1.3.2.4 Diseño del subsistema de procesamiento y decisión - Figura 3, 8, 37, 38, 50 y 51
<p>RF: Registrar cada evento de segregación. RT: Seleccionar fuente de alimentación adecuada a los actuadores seleccionados.</p>	<ul style="list-style-type: none"> • Implementación de una rutina en Python (pandas y openpyxl) para registrar cada detección en un archivo Excel. • Selección de una fuente de 24V-5A y un convertidor DC-DC XL4015 para alimentar los cuatro servomotores MG90S de acuerdo al consumo de estos. 	<p>Las evidencias que muestran las decisiones tomadas en el diseño se muestran en:</p> <ul style="list-style-type: none"> - 3.1.3.2.4 Diseño del subsistema de procesamiento y decisión - 3.1.3.2.6 Diseño del subsistema de alimentación - Figura 37, 38 y 53

La existencia de esta correspondencia directa demuestra que la identificación correcta de los requerimientos permitió establecer una base de diseño completa y adecuada, confirmando así la primera hipótesis.

- **Comprobación de Hipótesis 2**

Hipótesis 2: Si se diseñan adecuadamente los subsistemas del prototipo, entonces se logrará la construcción de un prototipo funcional e integrado. La comprobación de esta hipótesis se realiza mediante la demostración empírica de que el diseño, detallado en la metodología, fue factible basandose en los requerimientos, lógica del proceso y cálculos de ingeniería. También se observar en la tabla 8, matriz de trazabilidad, las decisiones de diseño tomadas que responden a los requerimientos. Posterior a esto se materializó en un

prototipo físico funcional y eficiente. La evidencia se presenta a continuación, validando tanto la factibilidad del diseño como la funcionalidad del prototipo.

1. Evidencia de la factibilidad del diseño de los sub-sistemas

La factibilidad de los diseños de los sub-sistemas se confirma a través de los entregables de ingeniería documentados en la sección 3.1 Instrumentos, materiales y métodos. La correcta aplicación de principios de diseño mecatrónico permitió la creación de subsistemas coherentes y viables:

Tabla 9

Evidencia de factibilidad del diseño de estructura del prototipo “EcoSplit”

Diseño de la estructura	<p>1. Las dimensiones del ancho, largo y alto se basaron a los tachos de 100 L Reyplast (53x51x84 cm) que utilizaba la institución educativa pues dichos tachos ya eran de fácil manipulación por los estudiantes y por el personal de limpieza, y se adecuaba al espacio disponible en el aula, por ello las dimensiones tomadas para el prototipo se aproximaron a estas siendo modificadas lo suficiente para tener suficiente espacio en otras zonas, como el espacio de identificación o el espacio necesario para los mecanismos resultando en dimensiones de 46x46x100 cm.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - 3.1.3.2.7 Diseño de la estructura - Figura 57
	<p>2. De acuerdo al tamaño de los residuos elegidos para la identificación se determino un espacio para la etapa de presentación de residuos de 10x10x24 cm.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - Figura 55
	<p>3. Se dividió el espacio para el depósito final de los residuos en cuatro partes iguales para que todos los compartimentos tengan la misma capacidad de guardar los residuos.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - Figura 54

	<p>4. Se dividió el espacio total del prototipo en dos partes con tal de que los residuos puedan ser retirados del prototipo fácilmente, mientras que la parte donde se encuentren todos los componentes físicos no puedan verse afectados por esa acción para luego se pueda volver a acoplar la parte superior mediante un encaje.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - Figura 54 y 55
	<p>5. Para la estructura principal, los contenedores y los paneles de soporte, se seleccionó acrílico transparente de 3 mm de espesor. Esta elección se sustenta en dos ventajas metodológicas clave para este proyecto:</p> <ul style="list-style-type: none"> • Visibilidad para la Observación y Diagnóstico: El objetivo principal de este prototipo es también validar un proceso. La naturaleza transparente del acrílico fue un requisito de diseño fundamental, ya que permite la observación directa y sin obstrucciones de todos los procesos internos en tiempo real. Esto facilitó enormemente las fases de prueba, y validación, permitiendo verificar visualmente la cinemática de los mecanismos, el flujo de los residuos y el funcionamiento de los componentes electrónicos. Esta visibilidad es crucial para el análisis de fallos y la identificación de futuras mejoras. • Facilidad y Precisión de Fabricación: El acrílico es un material ideal para el corte por láser, una tecnología de fabricación rápida y de alta precisión. Al basar el diseño estructural en este material, se garantizó que todas las piezas se fabricaran con las dimensiones exactas especificadas en los modelos CAD de Autodesk Inventor. Esto aseguró un ensamblaje correcto, con una alineación correcta de todos los orificios para pernos y soportes de componentes, reduciendo significativamente el tiempo de construcción y eliminando errores de fabricación manual. • Además, el acrílico ofrece una rigidez estructural suficiente para la escala y las cargas ligeras del prototipo, a un costo asequible. <p>Evidencia de factibilidad del diseño:</p>

	<ul style="list-style-type: none"> - 3.1.3.3 Implementación - Figura 60 y 64
	<p>6. Se dispuso en el diseño de también utilizar ruedas para el fácil traslado del prototipo por la institución hasta el aula correspondiente, ahorrando el trabajo de tener que cargarlo al movilizarlo, dos de las cuales poseen un freno para posicionar el prototipo en un solo lugar.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - Figura 60 y 64

Tabla 10

Evidencia de factibilidad del diseño del sub-sistema de procesamiento y decisión del prototipo "EcoSplit"

<p>Diseño del sub-sistema de procesamiento y decisión</p>	<p>1. El uso del modelo convolucional YOLOV11 permitió la detección de objetos en tiempo real para establecer criterios de verificación de residuos por tiempo. El modelo "11s" de YOLOV11 representa una solución de vanguardia que alcanza un equilibrio excepcional entre precisión y velocidad. Su puntuación de 47.5 COCO mAP_{Val}^{50-95} es una métrica de rendimiento integral, no una simple exactitud. Corresponde a la mAP (mean Average Precision), una puntuación que evalúa la capacidad del modelo para minimizar tanto los falsos positivos como los falsos negativos dentro del exigente y complejo dataset COCO en su conjunto de validación (Val). El aspecto más crítico de esta métrica es el 50-95, que indica que la puntuación final es el promedio de 10 evaluaciones con umbrales de "Intersection over Union" (IoU) crecientemente estrictos, desde una superposición del 50% hasta una casi perfecta del 95%. Debido a que este método penaliza severamente cualquier imprecisión en la localización del objeto, un valor de 47.5 demuestra un rendimiento de alta fiabilidad en</p>
--	--

	<p>un amplio espectro de exigencia. Esta robusta precisión se complementa con una latencia de inferencia de aproximadamente 2.5 milisegundos por imagen, una velocidad extrema medida en un entorno de producción optimizado, lo que confirma su idoneidad para la tarea de detección de residuos en tiempo real que demandan tanto una respuesta instantánea como una correcta detección.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - 3.1.3.2.4 Diseño del subsistema de procesamiento y decisión - Figura 24
	<p>2. Se estableció criterios de verificación por tiempo en la programación en python donde el modelo tenía que identificar un mismo objeto durante 10 segundos continuos para dar el resultado final de la inferencia.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - Figura 37 y 38
	<p>3. Se implementaron bloques de programación de una rutina en Python (pandas y openpyxl) para registrar cada detección en un archivo Excel.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - Figura 37 y 38
	<p>4. Se comprobó que la arquitectura de hardware distribuida entre la laptop para el cómputo de la visión artificial y la placa Arduino Uno para el control de los actuadores mediante la programación en python y el código cargado en la placa Arduino usando comunicación serial funcionaba correctamente.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - 3.1.3.2.4 Diseño del subsistema de procesamiento y decisión - Figura 3 y 8
	<p>5. También durante la etapa de validación del modelo YOLOV11 se obtuvieron las métricas sobre el desempeño del modelo sobre el conjunto</p>

	<p>de imágenes de validación cuya interpretación y comprobación de idoneidad se encuentra en la figura 36.</p> <p>Evidencia de factibilidad del diseño:</p> <ul style="list-style-type: none"> - 3.1.3.2.4 Diseño del subsistema de procesamiento y decisión - Figura 36
--	--

Tabla 11

Evidencia de factibilidad del diseño del sub-sistema de actuación del prototipo “EcoSplit”

Diseño del sub-sistema de actuación	<p>1. En el modelado del prototipo CAD en el software Autodesk Inventor se acopló y designó los espacios adecuados para los mecanismos Iris y de redirección como se observa en:</p> <ul style="list-style-type: none"> - 3.1.3.2.5 Diseño del subsistema de actuación - 3.1.3.2.7 Diseño de la estructura - Figura 46, 49 y 56
	<p>2. El diseño del mecanismo Iris y su cinemática se validó en el software Autodesk Inventor, comprobando que el giro angular máximo producido por el servomotor sobre el engranaje de entrada producía que se adoptaran las configuraciones específicas de apertura y cierre en 180° y 0° respectivamente como se observa en la figura 46 y 47. De la misma forma las configuraciones útiles que podía tomar el mecanismo de re-dirección se válido también en el software inventor como se ve en la figura 49, 76 y 77.</p>

	<p>3. Para el accionamiento de los mecanismos Iris y re-dirigidor, se seleccionaron los servomotores modelo MG90S. La elección de este componente específico se basó en una evaluación de los requisitos de la aplicación, funcionamiento de los mecanismos, priorizando la fiabilidad, la suficiencia de torque y la facilidad de integración dentro del contexto de un prototipo funcional. La justificación se detalla en los siguientes puntos:</p> <ul style="list-style-type: none">• Giro angular permitido: El servomotor MG90S fue seleccionado debido a que con el giro angular que se les permite se pueden diseñar mecanismos eficientes que aprovechen esa característica para tener las configuraciones útiles, a su vez también teniendo en cuenta la relación con el costo, como se ve en el siguiente punto.• Relación Costo-Beneficio: En el contexto de un prototipo de bajo costo, el MG90S ofrece un excelente equilibrio entre un precio asequible y las ventajas significativas en giro angular permitido, durabilidad que aportan sus engranajes metálicos, representando una mejora sustancial en fiabilidad por una diferencia de coste marginal en comparación con las alternativas de plástico, facilidad de integración para su control.• Durabilidad y Fiabilidad (Engranajes Metálicos): A diferencia de otros servomotores de tamaño similar, como el popular SG90 que utiliza engranajes de nylon, el MG90S cuenta con un tren de engranajes metálicos. Esta característica fue un factor decisivo, ya que proporciona una durabilidad y resistencia al desgaste significativamente mayores. Considerando que el prototipo operará de forma repetitiva, los engranajes metálicos
--	---

	<p>previenen fallos por rotura o barrido de dientes, un riesgo común en actuadores con engranajes de plástico bajo cargas cíclicas o bloqueos inesperados, garantizando así una mayor vida útil y fiabilidad para el prototipo.</p> <ul style="list-style-type: none">• Suficiencia de Torque para la Aplicación (Estimación de Carga): Se realizó una estimación cualitativa de las cargas de trabajo. Las tareas de los servomotores consisten en mover los componentes de los mecanismos, fabricados en material ligero (PLA impreso en 3D), y redireccionar residuos individuales cuyo peso es mínimo (ej. una botella de plástico vacía, una cáscara de fruta). El torque nominal del MG90S (aproximadamente 2.2 kg-cm a 6V) excede con un amplio margen de seguridad las bajas demandas de torque requeridas por la aplicación. Por lo tanto, se determinó que no era necesario un análisis dinámico complejo para validar su suficiencia, siendo una elección adecuada y eficiente para la escala del prototipo.• Facilidad de Integración y Control: El MG90S es un servomotor estándar que opera en un rango de voltaje (4.8V a 6V) compatible con el subsistema de alimentación diseñado (regulador DC-DC ajustado a 6V). Además, es totalmente compatible con la biblioteca Servo.h de Arduino, lo que simplificó enormemente el desarrollo del software de control y permitió una rápida implementación y ajuste de los movimientos de los mecanismos.
--	---

Tabla 12

Evidencia de factibilidad del diseño del sub-sistema de alimentación del prototipo

“EcoSplit”

Diseño del sub-sistema de alimentación	<p>1. Dado que se esta usando una arquitectura de hardware repartida entre la laptop y la placa Arduino Uno, la laptop era capaz de proveer a los siguientes componentes:</p> <ul style="list-style-type: none">• La cámara USB utilizada era directamente alimentada por la laptop.• La placa Arduino Uno conectada a la laptop era alimentada por esta, a su vez esta placa podía alimentar los dos LED's rojo y verde de 5 mm usados, mandar señales PWM a los servomotores MG90S para su control, verificar el estado del interruptor de presión. <p>Como se puede evidenciar en:</p> <ul style="list-style-type: none">- 3.1.3.2.6 Diseño del subsistema de alimentación- Figura 53
---	--

	<p>2. Análisis de la Corriente de Consumo de los Servomotores:</p> <p>La selección de una fuente de alimentación externa se fundamenta en la comparación entre la demanda de corriente de los servomotores y la capacidad de suministro de la placa Arduino Uno.</p> <ul style="list-style-type: none">• Corriente de Consumo del Servomotor MG90S: Un solo servomotor MG90S presenta un consumo de corriente variable dependiendo de su estado operativo:<ul style="list-style-type: none">> Corriente en Reposo (Stall Current): Aproximadamente 10-20 mA.> Corriente de Operación (Sin Carga): Puede variar entre 100 mA y 250 mA mientras el motor se desplaza.> Corriente de Pico o de Arranque (Stall/Peak Current): Al iniciar un movimiento, al cambiar de dirección bruscamente o si se encuentra con una resistencia mecánica (bloqueo), el motor puede demandar una corriente de pico que puede alcanzar hasta 1 A (1000 mA) por un breve instante.• Capacidad de Suministro del Arduino Uno: La placa Arduino Uno, cuando es alimentada por USB o por su conector de barril, tiene límites estrictos de suministro de corriente:<ul style="list-style-type: none">> Suministro Total del Regulador de 5V: El regulador de voltaje interno de la placa puede suministrar un máximo de aproximadamente 500 mA a 800 mA en total antes de sobrecalentarse y fallar.> Límite por Pin de E/S: Cada pin digital o de 5V individual no debe exceder los 40 mA de forma continua. <p>3. Justificación Técnica de la Fuente Externa:</p> <p>Al considerar el peor escenario operativo, donde los cuatro servomotores podrían activarse simultáneamente o experimentar picos de corriente, la demanda total del sistema de actuación podría teóricamente acercarse a 4 A (4 x 1 A).</p> <p style="text-align: center;">Demanda de Corriente Máxima (Teórica) \approx 4000 mA</p> <p>Esta demanda de 4000 mA es drásticamente superior a la capacidad máxima de suministro del regulador del Arduino Uno (\approx 500 mA). Intentar alimentar los servos desde la placa Arduino habría provocado los siguientes fallos catastróficos:</p>
--	---

	<ul style="list-style-type: none">• Caída de Tensión (Brownout): La alta demanda de corriente habría causado una caída de tensión en el riel de 5V, provocando que el microcontrolador ATmega328P se reiniciara constantemente (condición de "brownout"), haciendo el sistema inoperable.• Daño Físico a la Placa: Forzar al regulador de voltaje a suministrar una corriente tan elevada habría causado su sobrecalentamiento y daño permanente, inutilizando la placa Arduino. <p>Por estas razones, se diseñó un subsistema de alimentación robusto y desacoplado, compuesto por una fuente de alimentación externa de 24V y 5A y un convertidor DC-DC reductor (buck converter) modelo XL4015. Este diseño fue adecuado porque:</p> <ul style="list-style-type: none">• Suministra Corriente Suficiente: La fuente de 5A provee la corriente necesaria para cubrir incluso los picos de consumo simultáneos de los cuatro servos, garantizando un funcionamiento estable y sin caídas de tensión.• Proporciona el Voltaje Correcto: El convertidor XL4015 se configuró para reducir el voltaje de 24V a 6V, el voltaje óptimo de operación para los servomotores MG90S, asegurando que entreguen su máximo torque y rendimiento. <p>Esta separación de las fuentes de alimentación garantiza la estabilidad del sistema, protege los componentes de control y asegura que los actuadores dispongan de la potencia necesaria para operar de manera fiable.</p>
--	---

Tabla 13

Evidencia de factibilidad del diseño del sub-sistema de percepción del prototipo

“EcoSplit”

<p>Diseño del sub-sistema de percepción</p>	<p>1. El uso de la cámara web RT-A200 es adecuado porque la alta resolución (1080p) asegura que las imágenes capturadas contengan un alto nivel de detalle (texturas, bordes finos, colores precisos). Esto es un requisito fundamental para un modelo de red neuronal convolucional como YOLO, que necesita de estas características visuales para diferenciar objetos complejos y a veces similares. Una cámara de menor resolución podría haber fallado en capturar detalles cruciales, llevando a una menor precisión de clasificación. La tasa de 30 fps es estándar para video en tiempo real, garantizando una captura fluida sin perder información. La elección de la cámara web RT-A200 con resolución Full HD (1920x1080) y 30 fps se puede observar en la figura 4 y 6.</p>
	<p>2. Este diseño es adecuado porque aborda un problema crítico en la visión artificial: la consistencia y la reducción de ruido en la entrada de datos. Al fijar la cámara en una posición y ángulo constantes, se asegura de que todos los residuos sean vistos desde una perspectiva similar, lo que simplifica la tarea del modelo de IA. Al pintar el fondo de negro, se minimizó el "ruido visual", evitando que el modelo se confundiera con objetos o texturas irrelevantes en el fondo. Esta decisión de diseño incrementa directamente la robustez y la precisión del sistema de detección.</p> <p>Como se puede evidenciar en:</p> <ul style="list-style-type: none"> - Figura 5
	<p>3. La afirmación de que se utilizó la misma cámara para la captura del dataset de entrenamiento y para la operación del prototipo es un punto metodológico muy fuerte. Un diseño es adecuado si no introduce</p>

	<p>variables que confundan al sistema. Al usar la misma cámara, me aseguro de que las características de la imagen (distorsión de la lente, perfil de color, etc.) sean idénticas entre los datos con los que el modelo "aprendió" y los datos que "ve" en tiempo real. Esto elimina una fuente potencial de error.</p>
--	---

Tabla 14

Evidencia de factibilidad del diseño del sub-sistema de señalización/activación del prototipo "EcoSplit"

<p>Diseño del sub-sistema de señalización/activación</p>	<p>1. La elección de un único interruptor de presión como método de activación para todo el proceso es adecuado porque cumple con el principio de facilidad de uso, un requerimiento clave para un dispositivo que será utilizado por estudiantes en un entorno no técnico. En lugar de un panel de control complejo, un solo botón con una única función ("iniciar proceso") elimina cualquier ambigüedad, reduce la posibilidad de error del usuario y hace que la interacción sea inmediata e intuitiva.</p> <p>Como se puede evidenciar en:</p> <ul style="list-style-type: none"> - 3.1.3.2.3 Diseño del subsistema de señalización/activación - Figura 9
	<p>2. El uso de dos LEDs de colores distintos (rojo y verde) para comunicar los dos estados principales del sistema es adecuado porque utiliza una convención de colores universalmente reconocida: Rojo = Detenido/En espera/Alerta y Verde = Activo/Funcionando/OK. Esta retroalimentación visual es instantánea y no requiere que el usuario lea texto o interprete señales complejas. Informa claramente al usuario si la máquina está lista para recibir un residuo o si está ocupada en un ciclo de procesamiento,</p>

	<p>gestionando las expectativas y guiando la interacción.</p> <p>Como se puede evidenciar en:</p> <ul style="list-style-type: none"> - Figura 9
	<p>3. La implementación del pin del botón en modo INPUT_PULLUP en el código de Arduino y el cálculo explícito de las resistencias de 220Ω para los LED'S evita estados flotantes en el pin del botón, previniendo activaciones falsas y asegurando que la señal de activación sea siempre fiable. Calcular las resistencias limitadoras de corriente para los LEDs demuestra que se protegió activamente a los componentes (los LEDs y los pines del Arduino) de un exceso de corriente, garantizando la seguridad y la longevidad del circuito.</p> <p>Como se puede evidenciar en la figura:</p> <ul style="list-style-type: none"> - Figura 9, 50 y 51

2. Evidencia de la Construcción y Funcionalidad del Prototipo

La consecuencia de un diseño adecuado es la capacidad de construir y operar un prototipo funcional. La evidencia de este logro se materializa en el prototipo "EcoSplit":

- Evidencia de Construcción (Viabilidad Física): Los diseños se llevaron a la realidad convirtiéndose en un ensamblaje físico, como se documenta visualmente en la sección 3.1.3.3 Implementación (Figuras 58-64, 71-79). Las imágenes del prototipo armado demuestran que los subsistemas se integraron correctamente, confirmando que el diseño era físicamente construible.
- Evidencia de Funcionalidad (Viabilidad Operativa): El prototipo construido demostró ser plenamente funcional como se observa en las figuras 71 a 80 donde se muestra el ingreso de los residuos y su

posterior depósito por el prototipo en los correspondientes compartimentos. La siguiente imagen muestra las categorías de residuos que le corresponden a cada compartimento, donde el punto 1 es para los residuos aprovechables, el punto 2 es para los residuos no aprovechables, el punto 3 es para los residuos orgánicos, el punto 4 es para los residuos peligrosos.

Figura 71

Categoría de residuos correspondientes a cada compartimento



Durante la fase de experimentación, la máquina ejecutó el ciclo completo de segregación. Las pruebas de funcionamiento, como las registradas en las tablas 3 y 5, evidencian que el prototipo:

- Recibió correctamente los residuos a través del mecanismo Iris, el cual tardó menos de 1 segundo en completar la apertura de este según las pruebas grabadas.

Figura 72

Recepción de gaseosa incakola



Figura 73

Recepción de lata vacía



- Identificó el residuo, y su categoría mediante el subsistema de procesamiento y decisión.

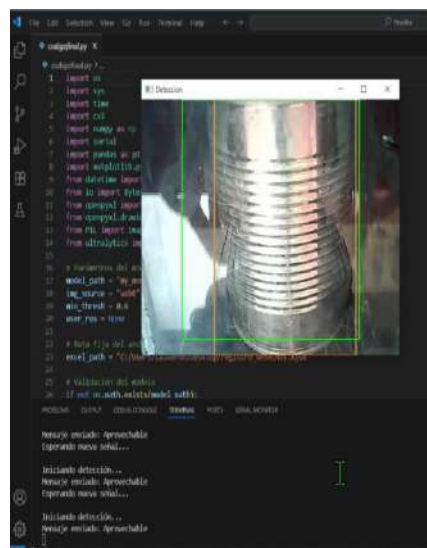
Figura 74

Identificación del residuo botella incakola y su categoría



Figura 75

Identificación del residuo lata y su categoría



- Accionó el mecanismo re-dirigidor para depositarlo en el compartimento correcto, el cual tardo menos de un segundo en adoptar la configuración indicada.

Figura 76

Mecanismo re-dirigidor orientado al compartimento de residuos aprovechables para depositar la botella incakola



Figura 77

Mecanismo re-dirigidor orientado al compartimento de residuos aprovechables para depositar la lata vacía



DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

- Depositó el residuo en el compartimento correspondiente a la categoría del resultado de la inferencia.

Figura 78

Depósito de la botella incakola



Figura 79

Depósito de la lata vacía



Figura 80

Registro de los residuos automáticamente

Fecha	Hora	Objeto	Categoría
2025-08-0	15:04:05	Inca kola	Aprovechable
2025-08-0	15:05:36	Gaseosala	Aprovechable
2025-08-0	15:06:40	Lata	Aprovechable

El análisis descriptivo e inferencial de los datos indico un porcentaje de segregación efectiva del 95.62% que certifica que el prototipo es funcional y que dicho valor porcentual no se debió al azar sino que es debido al uso del prototipo, cumpliendo su objetivo principal con eficacia.

Al haberse demostrado que el diseño adecuado de los subsistemas (evidencia de diseño) permitió la construcción de un prototipo físico (evidencia de construcción) que opera con una alta tasa de éxito (evidencia de funcionalidad), se comprueba y valida la segunda hipótesis.

- Comprobación de Hipótesis 3:

Hipótesis 3: Si se prototipa en físico adecuadamente la máquina automática, integrando sus componentes, entonces será capaz de ejecutar la segregación de residuos.

La comprobación de esta hipótesis se basa en la evaluación del rendimiento operativo del prototipo "EcoSplit". Un "prototipado adecuado" implica no solo el ensamblaje físico, sino también el funcionamiento integrado de todos los subsistemas. La evidencia de que este prototipado adecuado conduce a una "ejecución correcta" de la tarea de segregación se demuestra a través de los resultados cualitativos y de los resultados del análisis descriptivo e inferencial de los datos cuantitativos obtenidos durante la fase experimental.

1. Evidencia Cualitativa de la Ejecución Correcta del Proceso

Durante las pruebas del prototipo, se observó que este ejecutaba correctamente la secuencia completa del proceso de segregación para cada residuo introducido como se había diseñado en el diagrama de flujo de la figura 2. La evidencia cualitativa de una ejecución correcta se basa en el funcionamiento correcto de cada etapa del ciclo:

- **Recepción y Posicionamiento:** El mecanismo Iris se abrió y cerró de manera fiable en menos de 1 segundo, permitiendo el paso de un solo residuo al área de presentación como se observa en la figura 72 y 73.
- **Identificación Visual:** El sub-sistema de percepción capturó las imágenes y el modelo YOLOV11 las procesó, identificando la clase de residuo, como se documenta en las predicciones registradas en las tablas 3 y 5, y en las figuras 74 y 75.
- **Actuación y Redirección:** Tras la identificación, el subsistema de actuación respondió correctamente a la señal del subsistema de procesamiento y decisión. Los servomotores movieron el mecanismo re-dirigidor a la posición angular precisa correspondiente a la categoría del residuo (Aprovechable, Orgánico, etc.) como se observa en las figuras 76 y 77.
- **Depósito Final:** El mecanismo de apertura inferior se activó luego de que mecanismo re-dirigidor adoptara la inclinación correspondiente al depósito donde se determinó a donde iría el residuo por parte del sub-sistema de procesamiento y decisión. Esto se evidenció en las figuras 78 y 79.

El flujo de operación observado fue fluido, secuencial y sin fallos mecánicos o de software evidentes, lo que constituye la primera prueba de que el prototipo puede ejecutar la tarea correctamente.

2. Evidencia Cuantitativa de la Ejecución Correcta (Efectividad)

La prueba más contundente de una "ejecución correcta" es la medición de su efectividad. La ejecución no es correcta si el resultado final es incorrecto. Los datos cuantitativos, recolectados durante los 20 días del experimento con el prototipo, demuestran un alto nivel de precisión mediante el porcentaje de segregación efectiva:

- De un total de 92 residuos procesados por la máquina, se registró que 88 fueron segregados correctamente en su contenedor correspondiente. Esto se traduce en un "Porcentaje de Segregación Efectiva" de 95.62% mediante la fórmula mencionada en la sección de metodología y validado en la sección 3.2 Análisis descriptivo y 3.3 Análisis inferencial.

El mínimo de errores presentados se debió a que en dichos días donde se produjeron estos eventos hubo una iluminación ambiental mucho mayor a la de cuando se entreno el modelo produciendo que ciertos objetos reflejaran demasiado dicha iluminación, causando la confusión del modelo con otros residuos como se observa en la tabla 5, se tiene entonces un 4.38% de tasa de error. Este análisis descriptivo de la muestra en la sección 3.2 posteriormente fue acompañado de un análisis inferencia en la sección 3.3 para determinar que la mejora en la calidad de segregación de residuos no fue resultado del azar sino que esta relacionado directamente al uso del prototipo.

La combinación de la evidencia cualitativa (la observación de ciclos de segregación) y la evidencia cuantitativa (una efectividad de segregación del 95.62% y el análisis descriptivo e inferencial de los datos en la sección 3.2 y 3.3) demuestra de manera concluyente que el prototipado adecuado de la máquina permitió que esta ejecutara correctamente su función principal. Por lo tanto, la Hipótesis 3 se considera validada.

- Comprobación de la Hipótesis 4:

Hipótesis 4: El uso del prototipo mejorará el indicador de "porcentaje de segregación efectiva" en comparación con el método manual.

Esta hipótesis se confirma de manera contundente mediante la evidencia presentada en las secciones anteriores:

- Descriptivamente (sección 3.2): Se observó una mejora neta de 55.22 puntos porcentuales en la efectividad (del 40.40% manual al 95.62% con el prototipo).
- Inferencialmente (sección 3.3): La prueba de Chi-cuadrado demostró que esta mejora es estadísticamente significativa ($p < .001$), descartando que el resultado sea producto del azar.

CAPÍTULO IV: DISCUSIÓN Y CONCLUSIONES

4.1 Discusión:

La implementación y validación del prototipo "EcoSplit" ha culminado con un porcentaje de segregación efectiva del 95.62%, demostrando una mejora estadísticamente significativa sobre el proceso manual preexistente (40.40%), dicha mejora fue validada estadísticamente mediante la prueba de Chi-Cuadrado de independencia ($p < 0.001$), confirmando la cuarta hipótesis de que si se utiliza el prototipo en la I.E. 3049 Imperio del Tahuantinsuyo, entonces el indicador porcentaje de segregación efectiva será estadísticamente superior al obtenido mediante el proceso manual. Es importante también recordar que se eligió actuar sobre la primera etapa donde se generará el residuo en vez de una etapa intermedia o posterior debido a que en esos casos la complejidad del problema aumenta al haber una gran variedad de residuos de diferentes materiales y al estar muy mezclados entre sí, eso lleva hoy en día a ser complicados de manejar con una alta eficiencia de clasificación por las máquinas industriales para esa función, por eso la etapa inicial es adecuada para que el prototipo desde un inicio separe correctamente los residuos ahí, evitando que problema se vuelva inmanejable si se esperaría tratar en etapas posteriores. Para contextualizar y valorar críticamente estos hallazgos, esta sección los contrasta con los estudios previos analizados en el marco teórico, abordando las decisiones clave en el diseño del prototipo, la elección de la tecnología de visión artificial y el alcance de la validación.

Uno de los diferenciadores fundamentales de "EcoSplit" radica en su paradigma de detección basado en inteligencia artificial, en contraste con los sistemas de automatización clásica. Trabajos como el de Belji et al. (2021) proponen soluciones robustas para la industria basadas en Controladores Lógicos Programables (PLC) y

sensores ópticos dedicados para la clasificación de botellas PET por color. Si bien este enfoque es altamente eficiente para una tarea única y definida, su rigidez lo hace inviable para el problema heterogéneo de la segregación de residuos múltiples. Un sensor de color, por ejemplo, no podría distinguir entre una botella de plástico verde y una manzana de color verde. De manera similar, el sistema de Llerena Buenaño & Salazar Villamar (2022), aunque utiliza una cámara, se basa en la segmentación por color en el espacio HSV (Tono, Saturación, Valor), una técnica de visión artificial tradicional que presenta las mismas limitaciones. Si bien este método es eficaz para tareas didácticas con objetos de colores uniformes y bien diferenciados, presenta limitaciones críticas que lo hacen inviable para la complejidad de la segregación real de residuos. El funcionamiento de la segmentación HSV se basa en definir rangos numéricos fijos para aislar píxeles de un color específico. Por ejemplo, se le instruye al sistema que identifique todo lo que se encuentre dentro de un rango de "rojo". El problema fundamental es que esta técnica identifica colores, no objetos. Esta limitación se manifiesta de varias maneras críticas en el contexto de los residuos. Primero, es incapaz de diferenciar entre materiales distintos que comparten el mismo color. Un sistema basado en HSV no podría distinguir entre una botella de plástico roja, una lata de un envase de medicina roja o una envoltura de algún producto no aprovechable de color rojo; para el algoritmo, todos son simplemente "píxeles rojos" y los clasificaría erróneamente en la misma categoría. Segundo, es extremadamente sensible a las condiciones de iluminación. Una simple sombra puede alterar drásticamente el componente de 'Valor' (V) y los reflejos especulares pueden desaturar el 'Color' (S), provocando que los píxeles del objeto caigan fuera de los umbrales predefinidos y, por lo tanto, no sean detectados en absoluto. Finalmente, carece de la capacidad de comprender el contexto y la forma del objeto. Una botella de plástico con una etiqueta de papel multicolor sería un objeto casi imposible de clasificar para un

sistema HSV, ya que no puede inferir que la forma subyacente es la de una "botella" a pesar de la variedad de colores en su superficie. En esencia, la segmentación por HSV es un método rígido que depende de condiciones visuales casi perfectas y de una diferenciación cromática simple. "EcoSplit", al emplear un modelo de aprendizaje profundo, supera esta barrera, ya que su capacidad de clasificación se basa en un conjunto complejo de características visuales (forma, textura, patrones) aprendidas de un extenso set de datos, permitiéndole identificar la naturaleza del objeto más allá de su color.

Dentro del dominio de la inteligencia artificial, la elección de una arquitectura de Cadetección de objetos como YOLOv11 se revela como un factor metodológico crítico y una ventaja sustancial para el éxito del sistema. Esto se evidencia de manera clara al comparar el enfoque de "EcoSplit" con el de Liseth (2025), cuyo prototipo emplea un modelo de clasificación de imágenes basado en una Red Neuronal Convolutiva (CNN) general implementada en TensorFlow. A un nivel fundamental, la diferencia entre ambos enfoques es cómo procesan la información visual. Un clasificador de imágenes, por su diseño, analiza la imagen capturada en su totalidad como una única entidad; su objetivo es asignar una sola etiqueta (p. ej., "plástico", "vidrio") a la totalidad de los píxeles de entrada. En la práctica, esto significa que el modelo no distingue entre el objeto de interés y el fondo, lo que lo hace vulnerable a errores si el residuo no está perfectamente centrado o si existen elementos distractores en el entorno.

En contraste, el modelo YOLOv11 de "EcoSplit" opera bajo un paradigma de dos etapas lógicas: primero, localiza el residuo dentro de la imagen dibujando un cuadro delimitador (bounding box) a su alrededor y, segundo, clasifica únicamente las características visuales contenidas dentro de ese cuadro. Esta capacidad de aislamiento de características confiere a "EcoSplit" una robustez significativamente mayor frente a

las variaciones del mundo real. Por ejemplo, si un residuo se desplaza ligeramente del centro del área de presentación, un clasificador de imágenes podría fallar al no encontrar el patrón en la posición esperada; YOLO, en cambio, simplemente ajusta las coordenadas de su cuadro delimitador. De igual manera, pequeños reflejos, sombras o texturas irrelevantes en el fondo, que podrían confundir a un clasificador general, son eficazmente ignorados por YOLO, ya que quedan fuera del área de interés que el propio modelo ha delimitado.

Esta superioridad conceptual en conjunto con las decisiones de diseño que se tomaron como priorizar un ingreso individual de residuos permitido por no haber una alta tasa de segregación de residuos simultáneamente, entrenar el modelo con imágenes cuyo ambiente fuera similar al de las pruebas (Fondo de pintura negra, operación del prototipo con una misma iluminación debido a un horario constante de funcionamiento) por maximizar el desempeño, se tradujo directamente en un rendimiento empírico más elevado, explicando la diferencia en precisión obtenida (95.62% en "EcoSplit" frente al >90% reportado por Liseth (2025)). Además, al situar nuestro trabajo en el contexto de la evolución de la propia familia YOLO, se observa que el rendimiento de "EcoSplit" supera incluso a los de estudios que utilizan versiones anteriores como YOLOv5s (Alejandro et al., 2024) o YOLOv8 (Arishi, 2025). Es particularmente notable que nuestra precisión sea superior a la obtenida por Arishi (mAP 89.5%), quien necesitó incorporar complejos mecanismos de atención algorítmica para mejorar su modelo base.

Es particularmente notable que la precisión de "EcoSplit" sea superior a la obtenida por Arishi (2025) (mAP 89.5%), quien para mejorar el rendimiento de su modelo base de YOLOv8 necesitó incorporar complejos mecanismos de atención algorítmica. Estos mecanismos son componentes de software adicionales que se integran en la

arquitectura de la red neuronal para imitar la capacidad de atención selectiva del sistema visual humano. Cuando un ser humano observa una escena compleja, se enfoca instintivamente en las regiones y características más relevantes mientras ignora el ruido de fondo. De manera análoga, un mecanismo de atención, como el CBAM (Convolutional Block Attention Module) utilizado por Arishi, aprende a asignar dinámicamente "pesos" o "puntuaciones de importancia" a diferentes partes de la información visual que procesa la red.

En la práctica, esto significa que el modelo de Arishi fue modificado para que aprendiera a "prestar más atención" a las características visuales más discriminatorias de los residuos. Por ejemplo, ante una imagen con varios objetos y un fondo complejo, el mecanismo de atención amplificaría la importancia de las texturas de un trozo de cartón o los bordes de una botella de plástico, mientras suprime activamente la influencia de sombras irrelevantes o patrones en la cinta transportadora. Esta es una solución algorítmica para hacer el modelo más robusto y preciso, especialmente cuando se enfrenta a un dataset grande y variado con 17 clases, donde las diferencias entre categorías pueden ser muy sutiles. La relevancia de este punto para mi discusión es que "EcoSplit" alcanza una precisión superior sin la necesidad de recurrir a esta capa adicional de complejidad algorítmica. Nuestro enfoque se basó en una "atención por diseño": el sistema físico, a través de su entorno controlado y su proceso de alimentación individual, presenta al modelo de IA una imagen ya "limpia" y enfocada en un único objeto de interés. En lugar de forzar al software a aprender a ignorar el ruido, nuestro hardware lo elimina desde el principio. Esto sugiere que mi estrategia de ingeniería mecatrónica integrada no solo es más simple, sino también más eficaz para esta aplicación específica en la institución

educativa, logrando un rendimiento superior sin el costo computacional y la complejidad de entrenamiento asociados a los mecanismos de atención.

Otro aspecto que confiere una notable validez a los resultados de este proyecto es su culminación en un prototipo físico funcional, superando el alcance de estudios que se mantienen en la fase de simulación o diseño conceptual. El trabajo de Colmenares Velasquez & Chavez Vasquez (2024), por ejemplo, valida exitosamente un sistema complejo con un robot delta en el entorno de simulación CoppeliaSim, alcanzando una precisión del 92%. Sin embargo, "EcoSplit" fue construido y probado en un entorno real, enfrentando y superando los desafíos inherentes a la interacción con objetos físicos, como las variaciones de iluminación y las imperfecciones mecánicas. La obtención de una efectividad del 95.62% en estas condiciones representa un avance significativo, validando no solo el algoritmo, sino el prototipo del sistema mecatrónico en los aspectos de funcionalidad integrada del hardware y visión artificial. De igual manera, el estudio de Romero (2023), aunque implementa su modelo en una maqueta física, esta solo simula la clasificación encendiendo un LED, sin realizar una actuación mecánica sobre el residuo, lo que posiciona a nuestro prototipo en un nivel de madurez tecnológica superior al completar el ciclo de percepción, procesamiento y decisión y actuación.

Finalmente, el diseño mecatrónico de "EcoSplit" demuestra una clara adecuación de la tecnología al problema. Mientras que antecedentes como el de Aydin Adnan & Rehman (2022) o Belji et al. (2021) proponen soluciones de escala industrial con componentes de alto costo y rendimiento (actuadores neumáticos, PLC), y otros utilizan robots de alta complejidad como el brazo de 6 ejes de Llerena Buenaño & Salazar Villamar (2022) o el robot delta de Colmenares Velasquez & Chavez Vasquez (2024), "EcoSplit" resuelve el problema con la mínima complejidad necesaria sin perder la

eficacia necesaria para este caso. El diseño de mecanismos a medida, como el mecanismo Iris y el mecanismo re-dirigido accionado por servomotores de bajo costo, valida que para aplicaciones de baja cadencia y en entornos no industriales, no es necesario recurrir a soluciones sobredimensionadas. Este enfoque de ingeniería frugal y optimizada es crucial para el desarrollo de soluciones tecnológicas accesibles y de alto impacto social y educativo.

A pesar de los resultados positivos, es fundamental reconocer las limitaciones del presente estudio. El prototipo fue diseñado y validado para el procesamiento de un residuo a la vez en un entorno controlado, habiéndose imposibilitado la segregación simultánea de varios residuos, pues es debido al estudio de Alejandro et al. (2024) quien indica que el rendimiento de los sistemas de visión disminuye al aumentar la densidad de objetos, iteraciones futuras a partir de "EcoSplit" que busquen la segregación múltiple de varios residuos al mismo tiempo, tendrían el requisito de modificar el espacio de presentación ante la cámara, esto causando que se necesite modificar otros muchos aspectos del prototipo como el de re-ubicar la posición de la cámara, re-evaluar el diseño del mecanismo Iris de salida y de el mecanismo de separación, seleccionar nuevamente los actuadores para los nuevos mecanismos en caso se modifiquen, modificar la programación python y Arduino Uno del sub-sistema de procesamiento y decisión y actuación. Asimismo, la metodología de Arishi (2025) sobre el uso de mecanismos de atención y aumento de datos ofrece una vía clara para mejorar la robustez del modelo si se deseara expandir el número de categorías o la variabilidad de los residuos en futuras iteraciones del prototipo. Otra de las limitaciones presentes en este trabajo es el horario de funcionamiento del prototipo, debido a que el data set de imágenes fue creado en las mismas condiciones ambientales de operación (Iluminación natural entre los meses de

Mayo y Julio durante el turno mañana de clases) se pudo obtener un alto rendimiento en ese caso, sin embargo el prototipo no podría llevar a cabo su función en el turno tarde ya que la falta de iluminación natural impediría que el modelo con el entrenamiento actual en este trabajo, no reconozca los residuos, para dicho caso teniendo que explorar en futuras iteraciones el uso de herramientas como data augmentation que tomaría el conjunto del data set y a partir de cada imagen real, crearía múltiples imágenes en base a esa, modificadas con variaciones de brillo, contraste, color, ángulo, tamaño o incluso con sombras y reflejos simulados, logrando mayor robustez del prototipo. Otra opción para tratar esta limitación podría ser la creación de una “caja oscura”, es decir en una futura iteración del prototipo que busque tratar aspectos como el material final de la estructura, ya no requeriría el uso de un material translúcido pues ya se espera que prototipos previos hayan validado la funcionalidad integrada del hardware y en gran parte el de la visión artificial, es entonces que si es que se busca que las condiciones ambientales externas no afecten en nada al proceso, se podría crear un espacio cerrado dentro del prototipo donde la iluminación ambiental no tenga nada de influencia sobre lo que ve la cámara, sino que dentro de dicho espacio se tenga una iluminación propia del prototipo con algún elemento que lo produzca como un LED brillante, donde se podría crear el data set en ese mismo entorno ya controlado que será el mismo donde se operará, garantizando que las condiciones ambientales naturales no afectarán en nada al proceso.

Otra de las implicaciones de este estudio es el impacto que puede llegar a tener en las estrategias de gestión de residuos en instituciones como colegios, hospitales, centros comerciales, instituciones tanto de servicios públicos y privados, ya que el modelo YOLOV11 en realidad puede ser re-entrenado para el conjunto de residuos que se presenten en dichos ambientes, presentando la flexibilidad de adaptarse en dicho aspecto.

Otro punto de la gestión de residuos sobre la que tiene implicancias este estudio, es el hecho en como se motiva a los estudiantes a participar en la actividad del reciclaje correcto, una futura iteración del proyecto podría agregar un sub-sistema dedicado a registrar la participación de cada alumno midiendo el aporte de residuos en la máquina, y en coordinación con las autoridades de las instituciones construir una economía circular a partir del incremento del correcto reciclaje y su valorización, pudiendo recompensar a los usuarios economicante con parte de dicha economía circular, impulsando así la mejora de la calidad de segregación de residuos en instituciones y la re-construcción de una cultura del reciclaje al aprender observando las decisiones de la máquina al segregar residuos. Las opciones para llegar a escalar este prototipo e implementar más unidades en una institución pueden llegar a un debate muy profundo que requiere un próximo estudio más detallado, sin embargo partiendo de puntos generales como el uso de un único servidor general al cual distintas unidades repartidas por la institución se conecten inalámbricamente, este conteniendo el del sub-sistema de procesamiento y decisión general de todas las unidades, recibiría las imágenes capturas por cada unidad cuando sea activada, y regresando como respuesta a cada unidad y su sub-sistema de actuación la señal necesaria para mover los actuadores de los mecanismos definidos para segregar correctamente el residuo presente.

La discusión comparativa posiciona a "EcoSplit" como una solución mecatrónica integral y validada para el caso particular expuesto, que no solo se alinea con el estado del arte en el uso de arquitecturas de IA, sino que avanza al demostrar una implementación física de bajo costo con una eficacia superior a la de alternativas simuladas o basadas en tecnologías de automatización tradicional. El principal aporte de este trabajo reside en la demostración empírica de que la correcta integración de un

modelo de IA avanzado con un diseño mecánico y accesible permite crear soluciones de alto impacto, cerrando la brecha entre la prueba de concepto algorítmica y la funcionalidad mecatrónica en el mundo real.

4.2 Conclusiones

A partir de la investigación, el diseño, la construcción y la validación del prototipo "EcoSplit", se llega a las siguientes conclusiones, las cuales responden directamente a los objetivos específicos planteados:

- En respuesta al primer objetivo, se concluye que el análisis de los requerimientos funcionales y técnicos fue fundamental para establecer un diseño enfocado y eficaz. La investigación determinó que la principal deficiencia del proceso manual era la baja precisión, no el volumen de residuos. Esto condujo a la definición de requerimientos clave como el procesamiento individual de residuos y el uso de un sistema de visión capaz de diferenciar objetos por características complejas, no solo por color. Esta etapa inicial fue crítica para orientar el proyecto hacia una solución de alta precisión en lugar de uno de alta tasa de procesamiento de residuos simultáneos, asegurando que el diseño final abordara directamente la raíz del problema en la institución educativa.
- En respuesta al segundo objetivo, se concluye que el diseño de los cinco subsistemas del prototipo (percepción, procesamiento y decisión, actuación, señalización/activación y alimentación) demostró ser una solución mecatrónica integrada y viable. Se logró un diseño que combina exitosamente un subsistema de percepción de alta calidad (cámara web) con un subsistema de procesamiento de IA avanzado (modelo YOLOv11) y un subsistema de actuación de bajo costo pero alta efectividad (mecanismos a medida con servomotores). El diseño validó que no es

necesario recurrir a componentes industriales costosos para lograr una automatización precisa, estableciendo un modelo de ingeniería frugal y adecuado para el contexto de aplicación.

- En respuesta al tercer objetivo, se concluye que el prototipado físico de "EcoSplit" fue exitoso y demostró la factibilidad de la construcción del sistema. La integración de los componentes estructurales de acrílico, los mecanismos impresos en 3D y los subsistemas electrónicos se realizó sin contratiempos significativos, lo que confirma que el diseño conceptual era robusto y estaba bien planificado. La construcción del prototipo no solo materializó la solución propuesta, sino que también validó la elección de materiales y componentes de bajo costo como suficientes para crear un prototipo de máquina funcional para la tarea especificada dentro de un contexto específico.
- En respuesta al cuarto y último objetivo, se concluye que el prototipo "EcoSplit" es altamente efectivo en la mejora de la calidad de la segregación de residuos. La evaluación cuantitativa arrojó un incremento del "porcentaje de segregación efectiva" de un 40.40% en el proceso manual a un 95.62% con el uso del prototipo. Esta mejora neta de 55.22 puntos porcentuales, validada como estadísticamente significativa ($p < 0.001$), responde de manera contundente al problema de investigación. Este hallazgo no solo demuestra que el prototipo funciona, sino que prueba su superioridad sobre el método manual actual, confirmando que la implementación de una solución mecatrónica basada en inteligencia artificial es una intervención exitosa y de alto impacto para mejorar la gestión de residuos en la fuente.

REFERENCIAS

- Agudelo, N., Tano, G., Eccí, C. V.-U., & 2020, undefined. (n.d.). Historia de la Automatización. *Ingenierovizcaino.Com*. Retrieved January 5, 2025, from <https://ingenierovizcaino.com/ecci/aut1/corte1/articulos/Historia%20de%20la%20Automatizacion.pdf>
- Alejandro, L., José, B., Leonardo, F., Rentería, B., & Riobamba, E. (2024). *Sistema de clasificación de botellas mediante el reconocimiento de imágenes para el tratamiento de basura*. <http://dspace.unach.edu.ec/handle/51000/12206>
- Arishi, A. (2025). Real-Time Household Waste Detection and Classification for Sustainable Recycling: A Deep Learning Approach. *Sustainability 2025, Vol. 17, Page 1902, 17(5)*, 1902. <https://doi.org/10.3390/SU17051902>
- Aydin Adnan, T., & Rehman, U. (2022). *Development of optical sorting machine*. <http://adudspace.adu.edu.tr:8080/xmlui/handle/11607/4824>
- Belji, T., Naupay, F., Cristian, F., & Lopez, Q. (2021). Diseño de una máquina clasificadora de botellas pet por color mediante un control automático en Huancayo - Junín - 2021. *Universidad Continental*. <https://repositorio.continental.edu.pe/handle/20.500.12394/10543>
- Cáceres Tello, J., & Cáceres Tello, J. (2006). La visión artificial y las operaciones morfológicas en imágenes binarias. *Una Perspectiva de La Inteligencia Artificial En Su 50 Aniversario: Campus Multidisciplinar En Percepción e Inteligencia, CMPI 2006, Albacete, España, 10-14 de Julio Del 2006 : Actas, 2*, 570–575. <http://dialnet.unirioja.es/servlet/articulo?codigo=4455328>
- Cardona, E., & Gratacós, J. R. (2023). *Automatización robótica de procesos con inteligencia artificial*. <https://upcommons.upc.edu/handle/2117/384518>
- Colmenares Velasquez, I., & Chavez Vasquez, L. P. (2024). Diseño y simulación de máquina clasificadora de residuos reciclables (cartón, metal, plástico y vidrio) usando inteligencia artificial y robot delta en la provincia de Cajamarca. *Universidad Peruana de Ciencias Aplicadas (UPC)*.

<https://repositorioacademico.upc.edu.pe/handle/10757/674489>

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*.

<https://doi.org/10.4258/HIR>

Gutiérrez Rodríguez, E. (2018). Segregación de residuos sólidos domiciliarios y su incidencia en la inadecuada recolección y eliminación en la protección ambiental, distrito Villa María del Triunfo, Lima – 2016. *Universidad Nacional Federico Villarreal*. <https://repositorio.unfv.edu.pe/handle/20.500.13084/2105>

Hernández, Roberto., Fernández, Carlos., & Baptista, Pilar. (2014). Metodología de la investigación. Capítulos ampliados. *McGraw-Hill Education*, 1–600.

https://books.google.com/books/about/Metodolog%C3%ADa_de_la_investigaci%C3%B3n.html?hl=es&id=oLbjoQEACAAJ

Hernández Sampieri, R., Fernández Collado, C., María del Pilar Baptista Lucio, D., & Méndez Valencia Christian Paulina Mendoza Torres, S. (n.d.). *Con la colaboración de*.

Instituciones, E. N., Privadas, E., Distrito, D., San, D. E., & De, J. (2023). Gestión de residuos sólidos domiciliarios y educación ambiental en instituciones educativas privadas del distrito de San Juan de Lurigancho – Lima, 2019. *Repositorio Institucional - USS*. <https://repositorio.uss.edu.pe/handle/20.500.12802/11811>

Liseth, C. J. D. (2025). *Desarrollo de prototipo de modelo de machine Learning para clasificación de desechos con visión artificial y componentes de bajo costo*. Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Ingeniería en Teleinformática. <https://repositorio.ug.edu.ec/handle/redug/78579>

Llerena Buenaño, A. S., & Salazar Villamar, M. J. (2022). *Automatización de un sistema identificador y posicionador de objetos a través de un Brazo robótico mediante Visión Artificial con Lenguaje Python*. <http://dspace.ups.edu.ec/handle/123456789/22832>

Loayza Maturrano, E. F. (2021). El fichaje de investigación como estrategia para la formación de competencias investigativas. *Aacademica.Org*.

<https://doi.org/10.35383/educare.v9i1.594>

Maquinas y mecanismos. (2011).

Meza, A. J. C., Haro, N. A. L., & Flores-Castañeda, R. O. (2025). Artificial Intelligence Based System for Sorting and Detection of Organic and Inorganic Waste. *International Journal of Advanced Computer Science and Applications*, 16(5), 87–94. <https://doi.org/10.14569/IJACSA.2025.0160509>

Minam. (2020). *Indicadores RRSS Año-2019.*

<https://app.powerbi.com/view?r=eyJrIjoiNDU4MjJkZTgtOTIwMy00OTRiLTgxNzctZDlhMzZkYzk1ZjE4IiwidCI6IjBlMmFiZjRILWExZjUtNDFlZi1iOWE0LWM5YWE2ZGQ1NTE4MCI9&pageName=ReportSection>

Ministerio Del Ambiente. (2020). *Resolución Ministerial N° 070-2020-MINAM: Guía para la implementación de acciones para el manejo adecuado de residuos sólidos en instituciones educativas de educación básica regular.*

<https://www.gob.pe/institucion/minam/normas-legales/454914-070-2020-minam>

Mitchell, T. (1997). *Machine Learning (McGraw-Hill International Editions Computer Science Series): Tom M. Mitchell: 9780071154673.* 414.

Robledo Mérida, C. (2006). *Técnicas y proceso de investigación científica.* Editora Educativa,.

Romero, S. M. (2023). *Sistema de clasificación de residuos mediante redes convolucionales.* <https://upcommons.upc.edu/handle/2117/392008>

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach: A Modern Approach.* Pearson, 3, 1151.
<https://thuvienso.hoasen.edu.vn/handle/123456789/8967>

Shigley, J. Edward., Uicker, J. Joseph., Contin, H. C. de., & Pérez Castellanos, J. H. . (1996). *Teoría de máquinas y mecanismos.* 613.

Sullca Estrada, B. L. (2023). Importancia de la educación ambiental en el manejo de la segregación de residuos sólidos de los estudiantes del 4to grado de primaria de la


DISEÑO Y PROTOTIPO DE UNA MÁQUINA AUTOMÁTICA MEDIANTE VISIÓN ARTIFICIAL
PARA MEJORAR LA CALIDAD DE SEGREGACIÓN DE RESIDUOS EN LA I.E. 3049 IMPERIO DEL
TAHUANTINSUYO, DISTRITO INDEPENDENCIA, LIMA

I.E. N.º 50580 Mahuaypampa del distrito de Maras, provincia de Urubamba -
Cusco. *Universidad Continental*.


<https://repositorio.continental.edu.pe/handle/20.500.12394/14074>

ANEXOS

ANEXO N° 1. Permiso institucional para realizar la recolección de datos



"Año de la recuperación y consolidación de la economía peruana"



Solicito: Autorización para poner en funcionamiento mi proyecto de titulación en Ingeniería Mecatrónica de la Universidad Privada del Norte. "Prototipo de máquina automática clasificadora de residuos basada en visión artificial" en la I.E. 3049 Imperio del Tahuantinsuyo. En colaboración con los docentes del área de CT y los estudiantes del 5to grado "A" de Secundaria.

Señor Director(a): VICTORIA HUAYLINOS GONZALES de la institución educativa 3049 Imperio del Tahuantinsuyo, Independencia.

Yo, JOSEPH MILTON CALDERON ESPEJO identificado con DNI N° 70368776, egresado de la carrera de Ingeniería Mecatrónica de la Universidad Privada del Norte - Lima Norte me dirijo a usted con el debido respeto que merece, para exponer lo siguiente.

Que, en mi condición de egresado y próximo a obtener el título profesional en la UNIVERSIDAD PRIVADA DEL NORTE - LIMA NORTE, tengo el interés de implementar y poner en funcionamiento mi proyecto de titulación, el cual consiste en un prototipo de una máquina automática clasificadora de residuos basada en visión artificial, dentro de las instalaciones de su prestigiosa institución educativa.


El objetivo de esta implementación es llevar a cabo pruebas funcionales del prototipo en un entorno real y recabar datos relevantes para la elaboración de mi trabajo de tesis. Para ello solicito la autorización para utilizar un espacio dentro de su institución por un periodo de tiempo definido de cuarenta días hábiles, comprendido entre el 12 de mayo de 2025 al 4 de julio de 2025.

Asimismo, me gustaría contar con la colaboración de los estudiantes del grado de 5to grado de secundaria de la sección "A" del turno mañana y los docentes del área de CT. Mediante su participación activa en la interacción y uso del prototipo. Considero que esta colaboración no solo enriquecerá mi investigación, sino que también ofrecerá a los estudiantes una valiosa experiencia práctica y la oportunidad de familiarizarse con conceptos de ingeniería y tecnología aplicada a la gestión de residuos.

Agradezco de antemano su atención y favorable consideración a la presente solicitud. Estaré a su disposición para proporcionar cualquier información adicional que estimen pertinente y coordinar los detalles de la implementación, asegurando en todo momento el cumplimiento de las normativas de la institución.

Por todo lo expuesto sirva Ud. acceder a mi petición por ser de justicia.

Lima, 5 de Mayo del 2025


JOSEPH MILTON CALDERON ESPEJO
DNI N° 70368776