



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO  
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

**DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA  
IDENTIFICACIÓN Y TRAZABILIDAD DE PLACAS ELECTRÓNICAS EN  
PROCESOS DE RECICLAJE**

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

AUTOR: Anthony Fabian Pachacama Paucar

TUTOR: Víctor Vinicio Tapia Calvopiña

Quito-Ecuador

2026

## **CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Yo, Anthony Fabián Pachacama Paucar con documento de identificación N° 1727480277 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 13 de enero del año 2026

Atentamente,



---

Anthony Fabián Pachacama Paucar

1727480277

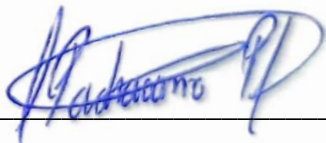
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE  
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Anthony Fabian Pachacama Paucar con documento de identificación N° 1727480277, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: Desarrollo de un sistema inteligente para la identificación y trazabilidad de placas electrónicas en procesos de reciclaje, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Quito, 13 de enero del año 2026

Atentamente,



---

Anthony Fabián Pachacama Paucar

1727480277

## **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN**

Yo, Víctor Vinicio Tapia Calvopiña con documento de identificación N° 1708547219, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN SISTEMA INTELIGENTE PARA LA IDENTIFICACIÓN Y TRAZABILIDAD DE PLACAS ELECTRÓNICAS EN PROCESOS DE RECICLAJE**, realizado por Anthony Fabián Pachacama Paucar con documento de identificación N° 1727480277, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 13 de enero del año 2026

Atentamente,



Ing. Víctor Vinicio Tapia Calvopiña Ms. C.

1708547219

## **DEDICATORIA**

A mi familia, porque gracias a su amor, apoyo incondicional y presencia constante, he podido avanzar incluso en los momentos más difíciles.

A ustedes, que con sus palabras, acciones y ejemplo me han enseñado lecciones profundas que guardaré por siempre.

A quienes han sido guías, consejeros y compañía fiel en este camino, demostrando que el lazo familiar va más allá de lo sanguíneo: es compromiso, cariño y entrega.

A quienes han asumido con amor los roles de madre, padre, hermanos, tíos y tías, y que día a día me han ofrecido su comprensión y fortaleza.

Gracias por ser mi refugio, mi impulso y mi mayor motivo para seguir adelante. Esta dedicación es para ustedes, con todo mi corazón.

## **AGRADECIMIENTO**

A mi familia, por ser el pilar que me ha sostenido en cada etapa de este camino.

Gracias por su amor constante, por el apoyo silencioso y también por las palabras oportunas que me impulsaron a seguir adelante.

Por estar presentes en los momentos de alegría y también en los de dificultad, brindándome fuerza, comprensión y compañía sincera.

Cada gesto, cada consejo, cada abrazo ha sido fundamental para llegar hasta aquí.

Este logro no es solo mío, es el reflejo del compromiso, el cariño y la unión que siempre me han brindado.

Con todo mi corazón, gracias por ser mi mayor motivación.

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1</b> .....	4
<b>ANTECEDENTES</b> .....	4
1.1 Problema.....	4
1.1.1 Antecedentes .....	4
1.1.2 Importancia y alcances .....	4
1.1.3 Delimitación .....	4
1.2 Objetivos del proyecto .....	7
1.2.1 Objetivo general .....	7
1.2.2 Objetivos específicos.....	7
<b>CAPÍTULO 2</b> .....	8
<b>FUNDAMENTOS TEÓRICOS</b> .....	8
<b>2.1 Fundamentos físicos y óptico</b> .....	19
2.1.1 Naturaleza de la luz .....	19
2.1.2 Formación de imágenes.....	20
<b>2.2 Fundamentos para el procesamiento digital de imágenes</b> .....	22
2.2.1 Adquisición de Imágenes .....	22
2.2.2 Procesamiento .....	22
<b>2.3 Fundamentos de visión artificial</b> .....	23
2.3.1 Representación digital de imágenes .....	24
2.3.2 Organización y etiquetado supervisado.....	24
2.3.3 Extracción de características.....	24
<b>2.4 Fundamentos de una red neuronal</b> .....	24
2.4.1 Neurona artificial.....	24
2.4.2 Arquitectura de capas .....	25

2.4.3	Funciones de activación .....	25
2.4.4	Parámetros .....	25
2.4.5	Función de partida .....	25
2.4.6	Algoritmo de entrenamiento y retropropagación .....	25
2.4.7	Capacidad universal de aproximación.....	25
2.5	Fundamentos de comunicación .....	26
2.5.1	Protocolo de comunicación .....	26
2.5.2	Application Programming Interface .....	26
2.5.3	Interoperabilidad .....	26
2.6	Métodos inteligentes de reciclaje que permita la gestión de residuos.....	26
2.6.1	Sistema inteligente de reciclaje Open Bot.....	27
2.6.2	Sistema de recolección optimizado mediante IoT.....	27
2.6.3	Sistema inteligente de gestión de residuos mediante IoT .....	27
2.6.4	Contenedor automatizado para residuos RAEE .....	8
 <b>CAPÍTULO 3 .....</b>		<b>19</b>
<b>MARCO METODOLÓGICO .....</b>		<b>19</b>
3.1	Propuesta de proyecto .....	19
3.2	Equipos necesarios para la implementación.....	20
3.2.1	Dispositivo para la captura de imágenes .....	20
3.2.2	Raspberry Pi 5 .....	21
3.2.3	Sensor de Distancia Sharp GP2Y0A41SK0F.....	22
3.2.4	LCD 16x2 I2C .....	23
3.3	Factores relevantes para la implementación.....	23
3.3.1	Luminosidad del entorno.....	23
3.4	Desarrollo del sistema .....	25
3.5	Conexiones del sistema .....	35

<b>CAPÍTULO 4</b> .....	38
<b>PRUEBAS Y RESULTADOS</b> .....	38
4.1 Pruebas del sistema de reconocimiento.....	38
4.2 Pruebas de ROI.....	40
4.3 Pruebas con el entrenamiento.....	41
4.4 Pruebas de manejo de información .....	41
4.5 Pruebas de la nube.....	41
<b>CONCLUSIONES</b> .....	43
<b>RECOMENDACIONES</b> .....	44

## ÍNDICE DE TABLAS

Tabla 2.1: Tabla de protocolos de comunicación .....	14
Tabla 2.2: Clasificación de las API por acceso .....	15
Tabla 2.3: Clasificación de las API por arquitectura .....	16
Tabla 2.4: Clasificación de las API por interacción .....	16
Tabla 3.1: Características de la Raspberry Pi 5 .....	22
Tabla 3.2: Tabla de ángulos de enfoque en fotografía .....	24
Tabla 4.1: Tiempo de subida de datos a la nube con la red de la UPS .....	42
Tabla 4.2: Tiempo de subida de datos a la nube con la red domestica .....	42

## ÍNDICE DE FIGURAS

Figura 3.1: Esquema general del sistema electrónico distribuido .....	20
Figura 3.2: Webcam Ins.....	21
Figura 3.3: Raspberry Pi 5 .....	22
Figura 3.4: Sensor de Sharp.....	23
Figura 3.5: Display LCD .....	23
Figura 3.6: Script para la decepción de características de las placas electrónicas .....	26
Figura 3.7: Diagrama de flujo del algoritmo de programación para la captura automática de imágenes de las placas electrónicas .....	27
Figura 3.8: Script para el entrenamiento del sistema de reconocimiento por visión artificial.....	28
Figura 3.9: Script para el entrenamiento del sistema de reconocimiento por visión artificial.....	29
Figura 3.10: Script para la Generación de Códigos QR Industriales con Registro Automatizado en Google Sheets.....	30
Figura 3.11: Diagrama de flujo del algoritmo implementado para el dispositivo central .....	31
Figura 3.12: Script para el reconocimiento automatizado de placas con control físico, registro cloud y alertas en Tiempo Real .....	33
Figura 3.13: Diagrama de flujo del sistema principal .....	34
Figura 3.14: Archivo JSON .....	35
Figura 3.15: Conexión de LEDES .....	36
Figura 3.16: Conexión del sensor Sharp.....	36
Figura 3.17: Conexión del I2C .....	37
Figura 4.1: Pruebas dentro del Laboratorio de Analógica.....	39
Figura 4.2: Pruebas dentro del Laboratorio de Robótica.....	39
Figura 4.3: Pruebas dentro del Laboratorio de Mecatrónica .....	40

## RESUMEN

Dentro del Bloque C, se ha identificado el problema de reciclaje para placas electrónicas de computadoras que fueron reportados fuera de servicio o que ya cumplieron con su ciclo de trabajo. Para este problema, si bien en la actualidad no se tiene una correcta forma de clasificar y almacenar este tipo de desperdicios electrónicos en específico, la universidad ha intentado implementar sistemas de reciclaje. Por lo que se han ejecutado varios proyectos nivel institucional en cada uno de las áreas que conforman la Universidad Politécnica Salesiana Sede Quito Campus Sur, sin obtener resultados eficaces.

Se han planteado muchas propuestas por parte de las carreras que pertenecemos al área de ingenierías para contribuir con el reciclaje a nivel institucional. El correcto manejo de los desechos como lo son las placas de computadoras que generan cada una de las carreras, sin obtener resultados positivos, debido a la falta de un sistema eficiente y óptimo que evite que los usuarios mezclen todos los desechos, teniendo así un pésimo control de los mismos, evitando ser reciclados o volver a ser a usados dentro de procesos de remanufactura. Por esta razón, el proyecto que a continuación se detalla en este documento busca proponer un sistema de reciclaje empleando redes neuronales convolucionales y técnicas de aprendizaje y predicción, para el reconocimiento de placas electrónicas de computadoras, no solo identificándolas, sino también extrayendo toda la información de trazabilidad de este componente electrónico para llevar un control detallado del recorrido que realizó dicho producto y posteriormente pase a un sistema de almacenamiento el cual las recolectara para que posteriormente la empresa encargada de procesar las placas fuera de servicio venga a retirarlos cuando el contenedor se encuentre al máximo de su capacidad de contención.

**Palabras Clave:** Visión Artificial, Reciclaje, Clasificación, Entrenamiento.

## **ABSTRACT**

Within Block C, the problem of recycling electronic computer boards that have been reported as out of service or have reached the end of their working life has been identified. Although there is currently no correct way to classify and store this specific type of electronic waste, the university has attempted to implement recycling systems. Several institutional-level projects have been carried out in each of the areas that make up the Salesian Polytechnic University, Quito South Campus, without obtaining effective results.

Many proposals have been put forward by the engineering departments to contribute to recycling at the institutional level. The proper management of waste, such as computer boards generated by each of the departments, has not yielded positive results due to the lack of an efficient and optimal system that prevents users from mixing all types of waste, resulting in poor control of the waste and preventing it from being recycled or reused in remanufacturing processes. For this reason, the project detailed below in this document seeks to propose a recycling system using convolutional neural networks and learning and prediction techniques for the recognition of computer circuit boards, not only identifying them, but also extracting all the traceability information from this electronic component to keep detailed track of the product's journey and then pass it on to a storage system which will collect them so that the company responsible for processing the out-of-service boards can come and remove them when the container is at maximum capacity.

**Keywords:** Computer Vision, Recycling, Classification, Training.

## INTRODUCCIÓN

Dentro de la Universidad Politécnica Salesiana Sede Quito Campus Sur, al tener una gran variedad de computadoras de escritorio, tanto a nivel educativo, administrativo y de mantenimiento de las instalaciones, al tener una extensa variedad de computadoras de escritorio, se ha vuelto muy común que sufran daños por el uso exhaustivo que se les dan a estos equipos, ya puede ser por factores como el tiempo que han estado operativos dentro de la institución, por golpes que hayan sufrido a lo largo del tiempo, daños a nivel de software provocados por malwares que hayan afectado de alguna forma al equipo, o incluso debido a un incorrecto mantenimiento preventivo de los mismos equipos.

Por tal motivo, dentro del Bloque C se planteó la propuesta de realizar un sistema de identificación para placas de computadoras, este sistema al identificar una placa de computadora, adquirirá y subirá todos los datos de trazabilidad a una base de datos, para posteriormente activar un sistema de contención que almacenara las placas para su posterior proceso de reciclaje o de remanufacturado, el cual tendrá un control de nivel de nivel máximo, que al sobrepasar los límites previamente establecidos enviara un mensaje al Smartphone del operario cuando el contenedor se encuentre al máximo de su capacidad de almacenamiento.¿

El desarrollo de este proyecto técnico, se presenta en 4 secciones. En la primera sección abordaremos la problemática que se solucionara, revisaremos los antecedentes que hubo anteriormente a nuestra propuesta, realizaremos un evaluó de la importancia que tendrá nuestro proyecto dentro del Bloque C, posteriormente después de una exhaustiva investigación realizaremos las delimitaciones del proyecto, para finalmente dar cumplimiento a nuestros objetivos. En la segunda sección presentara todos los fundamentos teóricos abordados para el desarrollo de este proyecto técnico. En la tercera sección, se presentará de forma detallada cada etapa del desarrollo de nuestro proyecto técnico. En la cuarta sección, expondremos todas las pruebas realizadas durante el desarrollo del proyecto y todos los resultados que fueron obtenidos conforme el proyecto fue avanzando, logrando así establecer nuestras conclusiones y recomendaciones después de haber finalizado el proyecto.

# CAPÍTULO 1

## ANTECEDENTES

### 1.1 Problema

#### 1.1.1 Antecedentes

La trazabilidad es un proceso fundamental que toda empresa manufacturera debería aplicar. Caso contrario, puede generar múltiples consecuencias negativas, como la pérdida de control sobre los procesos, dificultando la identificación de fallos o problemas en la producción, distribución o gestión de residuos. Además, puede comprometer la seguridad y calidad de productos, afectando la confianza del consumidor y la reputación de la empresa o entidad. La ausencia de trazabilidad también puede derivar en dificultades para cumplir regulaciones específicas, lo que puede resultar en sanciones o restricciones operativas

En Bloque C de la Universidad Politécnica Salesiana Sede Quito, campus Sur, la gestión de placas electrónicas constituye un problema cada vez más relevante y urgente, esto se debe a la acumulación de placas electrónicas sin un sistema que garantice su adecuado manejo, lo que dificulta su reciclaje y el posible aprovechamiento de materiales valiosos. Esta problemática se acentúa más en el ecosistema porque están compuestas por materiales peligrosos como; plomo, mercurio y cadmio, cuya manipulación indebida puede causar graves daños al medio ambiente y representar riesgos significativos para la salud humana (Sánchez Silva, Bonales Valencia, & Espinoza Torres, 2020).

Se ha propuesto implementar sistemas de trazabilidad electrónica, empleando códigos de barra en el proceso de fabricación, con el objetivo de lograr identificar los defectos y mediante esta información mejorar la calidad de nuestros procesos industriales, sin embargo este sistema presenta fallas como la limitada información que podemos llevar a través del uso de códigos de barras y la escalabilidad muy limitada que se tienen debido a que este sistema solo está enfocado en una determinada etapa de todo nuestro proceso y no podremos realizar un análisis amplio y variado a nivel general de todo el proceso (Toaquiza & Paredes, 2020, p. 243).

Dentro de las investigaciones llevadas a cabo encontramos el trabajo titulado "Desarrollo de un sistema de visión artificial para el reconocimiento, clasificación y maquinado de patrones con una tarjeta ARM", el objetivo principal es proponer solución viable y automatizada para optimizar procesos industriales integrando visión artificial y métodos de procesamiento embebido. Cabe recalcar que una desventaja presente en este sistema

es la ausencia de un modelo de arquitectura tipo REMI, lo que podría limitar las características de escalabilidad y de interoperabilidad del sistema en entornos más mucho más adversos (Andrade Fierro & Chulca Simbaña, 2018).

El artículo académico titulado "Internet de las cosas y visión artificial, funcionamiento y aplicaciones: Revisión de literatura", propone realizar un análisis a detalle sobre la convergencia entre el IoT o Internet de las Cosas y la visión artificial, resaltando su potencial para la transformación de diversos sectores. El principal objetivo es buscar un método para integrar estas tecnologías que nos permiten el manejo de imágenes y videos con características de REAL TIME, logrando aplicaciones innovadoras en diversos campos como el educativo, área de la salud, los edificios inteligentes y los sistemas inteligentes de vigilancia. Sin embargo, una característica limitante es la ausencia de una infraestructura que permita la convergencia de diferentes protocolos de comunicación y dispositivos, que puedan manejar eficientemente las fases, tanto de adquisición como de procesamiento de imágenes. Esta limitante restringe la capacidad del sistema para optimizar procesos de manera más eficaz y lograra así adaptarse rápidamente a cambios dentro de la producción, aspectos que son sumamente importantes dentro de entornos industriales modernos donde la toma de decisiones ágil es clave para la eficiencia y competitividad (Alvear-Puertas, Rosero-Montalvo, Peluffo-Ordóñez, & Pijal-Rojas, 2017).

La buscar trabajos relacionales al manejo de residuos, encontramos el proyecto titulado "Sistema de detección y alerta de residuos sólidos mediante Inteligencia Artificial", el cual nos propone una solución tecnológica viable enfocada en la mejora de la gestión de desechos dentro de espacios públicos mediante la automatización de reconocimiento de residuos como botellas, colillas, papel y metales. Empleando una red neuronal convolucional entrenada mediante imágenes reales y basada en una arquitectura MobileNet, el sistema tiene la capacidad de identificar y clasificar residuos con características de REAL TIME utilizando una cámara web. Sin embargo, este proyecto presenta la desventaja de que el solo nos permite clasificar, mas no cuenta con un mecanismo de contención que nos garantice que el usuario deposite sus desechos en el contenedor correspondiente, lo que limita su practicidad en una implementación de campo (Zambrano Oñate, 2023).

Uno de los desafíos más comunes al utilizar sistemas de visión artificial para la detección de objetos específicos surge cuando se presentan superficies reflectantes. Este tipo de materiales puede provocar que la luz emitida por el sistema de identificación regrese

directamente al sensor, generando errores en el reconocimiento debido a la distorsión causada por el reflejo luminoso (Ser & Vicios, 2006).

Al hablar de la reflexión de luz, podemos hablar de la especular que la tenemos presente en superficies lisas, donde la luz reflejada posee una dirección coherente y la reflexión difusa que sucede en objetos que tienen características rugosas y a desnivel en su superficie, teniendo como efecto la dispersión de luz en varias direcciones.

### **1.1.2 Importancia y alcances**

El desarrollo de este sistema, nos plantea una solución inteligente y eficaz para la correcta gestión y contención automatizada de las placas electrónicas de computadoras, aplicando tecnologías que actualmente exige la industria 4.0 como lo son el uso de visión artificial, trazabilidad digital y monitoreo en una base de datos dentro de la nube en tiempo real.

El tener un manejo adecuado y correcto de contención de este tipo de desechos, como lo son las placas de computadoras no solo ayuda a contribuir con el medio ambiente, además reduce el impacto ambiental negativo que estas pueden generar por los residuos de las mismas, ya que podemos extraer materiales valiosos que podrán servir a futuro para la fabricación de nuevos productos, logrando así llegar a tener una economía circular.

Implementar un sistema de trazabilidad nos ayuda a tener un control de calidad y detección de fallas, transparencia dentro de nuestra cadena de suministro, cumplimiento de las distintas normas regulatorias a las cuales estén sujetas el área en específico al que se les está realizando el seguimiento, mejora continua y eficiencia operativa, confianza del consumidor y responsabilidad ambiental.

En la actualidad, dentro del Bloque C de la Universidad Politécnica Salesiana sede Quito, se han aplicado métodos de clasificación y reciclaje rudimentarios, con nuestro sistema el cual emplea características de la industria 4.0 como; el uso de IoT, automatización, visión artificial por computadora y tecnología y protocolos de comunicación mediante internet. Podremos

### **1.1.3 Delimitación**

El sistema que va ser implementado dentro del Bloque C de la Universidad Politécnica Salesiana Sede Quito, se enfocara netamente en el reconocimiento mediante visión artificial y su posterior contención, para que un agente externo las retire y realice el manejo más adecuado de este tipo de desechos.

Este proyecto posee un sistema que adquirirá datos almacenados dentro de un código QR, y los subirán a un espacio creado en Google Sheets, el cual será utilizado para realizar una base de datos básica y en la cual netamente se subirá y escribirá la información obtenida a partir de los códigos QR, mas no se podrá realizar modificaciones en la información obtenida y no se podrán realizar ningún otro tipo de cambios dentro de la hoja de cálculos de Google Sheets.

En lo que respecta al contenedor de las placas electrónicas de computadoras, tendrá un sistema que estará constantemente monitoreando el espacio dentro del contenedor, el cual cuando llegue a una determinada altura por las placas apiladas dentro del contenedor, el cual enviará un mensaje de alerta vía Telegram para que el operario se acerque a vaciar el contenedor y reiniciar el sistema.

## **1.2 Objetivos del proyecto**

### **1.2.1 Objetivo general**

Desarrollar un sistema de gestión de reciclaje de placas electrónicas para la reducción del impacto ambiental, mediante la trazabilidad de E-waste, e implementada en el bloque C de la Universidad Politécnica Salesiana, sede Quito.

### **1.2.2 Objetivos específicos**

- Indagar sobre métodos inteligentes de reciclaje que permita la gestión de residuos, mediante base de datos y repositorios universitarios que contiene artículos académicos indexados y tesis.
- Diseñar un módulo contenedor inteligente para el reconocimiento y almacenamiento de placas electrónicas, mediante inteligencia artificial y captura de datos.
- Desarrollar una plataforma con herramientas informáticas para el almacenamiento, seguimiento y notificación automatizada, mediante Cloud Computing.
- Establecer una comunicación entre el módulo contenedor inteligente y la nube para la transmisión de la información del proceso de trazabilidad, mediante sistema Iot.
- Verificar el funcionamiento del sistema de trazabilidad de placas electrónicas reciclables para su validación mediante pruebas experimentales de campo.

## **CAPÍTULO 2**

### **FUNDAMENTOS TEÓRICOS**

#### **2.1 Fundamentos físicos y óptico**

##### **2.1.1 Naturaleza de la luz**

Para comprender la naturaleza y el comportamiento de la luz, primero se debe conocer que la luz es una forma de energía electromagnética que se propaga en forma de ondas, o explicado de otra manera podemos expresar que es el flujo de partículas conocidas como fotones. A esta dualidad se la conoce también con el nombre de la naturaleza dual de la luz (Fernandez, Gil, Moriel & Recio, 2009).

Si bien la luz posee una naturaleza dual, no quiere decir que la luz sea de forma simultánea una onda y una partícula a la vez, lo que nos quiere decir es que estas propiedades se manifiestan dependiendo del contexto en el cual se encuentra en ese momento.

##### **2.1.2 Formación de imágenes**

Se puede establecer que la óptica geométrica es un concepto con el cual mediante los rayos de luz logramos describir la trayectoria de la luz y la generación de imágenes. Esta es la base fundamental para comprender como trabajan los sistemas ópticos de las cámaras. Los principios fundamentales en los que se basan son:

- La propagación rectilínea de la luz la cual permite predecir las trayectorias de los rayos de luz y la proyección de sombras con precisión
- La ley de reflexión que nos establece que cuando un rayo de luz incide sobre una superficie con características reflectantes,
- La ley de Snell o de refracción establece que cuando la luz pasa de un medio a otro tiende a cambiar su dirección según sus índices de refracción.
- El principio de reversibilidad nos dice que la trayectoria de un rayo de luz tiende a ser la misma si invertimos su dirección, esto nos ayudan para a tratar sistemas ópticos como si fueran simétricos.

#### **2.2 Fundamentos para el procesamiento digital de imágenes**

##### **2.2.1 Adquisición de Imágenes**

Para realizar una óptima adquisición de imágenes, tenemos que en la primera fase definir el sensor más adecuado, esta decisión se tomara dependiendo de la información que deseamos obtener y el que mejor se acople a nuestro sistema,

Los sensores con tecnología RGB, son dispositivos electrónicos lo cuales tienen la capacidad de capturar imágenes a color, todo esto mediante la detección de luz visible y posteriormente dividiéndola en sus tres elementos bases que son el rojo, verde y azul, simulando el trabajo que realizan las células fotosensibles del cuerpo humano. Tienen un amplio uso en cámaras digitales, dispositivos inteligentes como los son los smartphones y sistemas que empleen visión artificial.

Los sensores infrarrojos, tienen la característica de detecta radiación infrarroja imperceptible para el ojo humano. Esta característica permite una visión en condiciones con baja cantidad de luz o para ser empleados en la detección de objetos por su firma de calor, es ampliamente utilizada en sistemas de seguridad, detectores de fugas y hasta en equipos de visión nocturna.

Los sensores con tecnología térmica, corresponden a un específico tipo de sensor infrarrojo, ampliamente utilizados para realizar capturas del calor que emiten los objetos. Estos sensores crean imágenes térmicas a partir de la detección de radiación infrarroja que un cuerpo emite en función de su temperatura. A partir de estas características se forman imágenes térmicas, las cuales se forman por la diferencia de temperatura en un entorno, incluso en un ambiente totalmente oscuro.

### **2.2.2 Procesamiento**

En la primera la fase de procesamiento tenemos la adquisición de la imagen a través de un sensor óptico, para posteriormente seguir con la conversión analoga\_digital. Esta presenta la imagen en forma de una matriz de píxeles que poseen valores de intensidad. Para esta fase los aspectos claves a ser tomados en cuenta son el sensor que se seleccionará, el digitalizador que transforma las señales en datos de tipo binario, la interfaz que se utilizará para la transferencia de la imagen y finalmente el software que controlará los parámetros como la resolución, exposición y formato entre los aspectos principales a ser tomados en cuenta.

En la segunda fase de procesamiento tenemos al preprocesamiento, el cual tiene como objetivo mejorar la calidad de la imagen antes de su análisis, para lo cual filtraremos las imperfecciones como el ruido, modificación del contraste y normalización de la iluminación. Para esta fase tenemos los filtros que ayudan a suavizar y detallar de mejor manera nuestra imagen, ecualización de histograma y reducción de ruido frecuencial.

En la tercera fase tenemos a la segmentación, encargada de dividir la imagen en regiones de mayor relevancia, lo que permite aislar objetos, zonas de baja relevancia para continuar con su análisis posterior. Para esta fase nos ayudaremos con métodos de umbralización, detección de bordes, clustering y watershed.

## **2.3 Fundamentos de la visión artificial**

### **2.3.1 Representación digital de imágenes**

Para entender como el sistema de visión artificial interpreta una imagen, primero debemos entender que cada imagen digital se la representa en forma de matriz bidimensional, entendido este principio básico, posteriormente aplicaremos una escala de grises, lo que quiere decir que le daremos un solo canal de intensidad luminosa, lo que reducirá el procesamiento computacional y eliminar información de color irrelevante para el reconocimiento de texturas dentro de nuestro sistema. Al pasar por todo este proceso pasaremos por un proceso de redimensionamiento, lo que nos ayuda a estandarizar nuestras imágenes para mantener la consistencia en las características de tamaño y aspecto, permitiendo que el algoritmo compare patrones de una manera altamente precisa y eficiente.

### **2.3.2 Organización y etiquetado supervisado**

Para un aprendizaje supervisado clásico, se jerarquiza por categorías, ubicando en subcarpetas toda la información obtenida de las placas a través de la captura de imágenes. Las cuales pueden ser nombradas por características visibles o de cualquier otro tipo, estas a su vez actúan como etiquetas visibles.

Un paso fundamental para los sistemas es la correcta asociación de etiquetas, en donde cada imagen se vincula a un número o variable que representa su clase correspondiente.

Toda esta información está contenida dentro de un Dataset estructurado, el cual es fundamental para que un sistema logre diferenciar entre clases durante la etapa de entrenamiento. Se prepara un conjunto balanceado de datos y etiquetados para la parte de reconocimiento.

### **2.3.3 Extracción de características**

Para la extracción de características de un determinado objeto, se entrena un sistema de reconocimiento en donde se aplica métodos que traducen elementos visuales que posee una imagen en datos de tipo numérico que pueda interpretar un determinado algoritmo.

El método de forma y geometría describe las características de estructura y de contorno de un determinado objeto, este método detecta bordes cerrados dentro de una imagen que esta segmentada, a continuación, devuelve una lista de todos los contornos que representan todas las formas que posee.

La técnica de momento geométrico, nos permite calcular propiedades como el área total o parcial, el centroide, en que orientación se encuentra y hasta la inercia del mismo. Estos tienen la característica de que son invariantes a datos como escala, rotación y traslación. Son usados ampliamente para el reconocimiento en patrones de formar.

El Bounding Box, es una técnica que encierra un determinado objeto en una figura geometría cuadra o rectangular, entregando datos de posiciones y proporciones para establecer sus características. Ampliamente utilizado en seguimiento y detección de objetos.

Los Descriptores de Fourier, permite analizar a una determinada forma como si fuera una señal y la transforma en el dominio de la frecuencia. Es utilizada para comparar figuras complejas de forma matemática

Aplicación de Histogramas de color, esta técnica analiza y describe la cantidad de pixeles que posee cada tono, los cuales se pueden crear por canales de tipo RGB o por espacios HSV, esta técnica tiene características de robustez frente a cambios de luminosidad siempre y cuando se usen dentro de espacios HSV o LAB.

La segmentación por umbral de color y análisis de color dominante, son técnicas que se basan en el aislamiento de regiones con un color específico, clasificando objetos por sus características de color. Su eficiencia no es óptima debido a que netamente identifica características de color y no toma en cuenta otras características relevantes.

La técnica de Local Binary Patterns o LBP, se basa en la comparación de píxeles vecinos los cuales crean un patrón binario, posteriormente generando histogramas locales por bloque. Esta técnica es robusta frente a factores de luminosidad variable.

La técnica Gray-Level Co-occurrence Matrix o GLCM, analiza los distintos pares de intensidad en una determinada dirección. Todo esto gracias a la extracción de métricas como contrastes, homogeneidad que presente el objeto y las características de correlación que posea. Empleado principalmente para distinguir superficies que posean características de rugosidad entre objetos que posean una superficie lisa.

## **2.4 Fundamentos de una red neuronal**

### **2.4.1 Neurona Artificial**

Un modelo neuronal artificial es el núcleo principal de cualquier red neuronal, esta es una unidad computacional, la cual recibe múltiples entradas de tipo numéricas, cada uno con un determinado peso que indica su relevancia. Adicionalmente a estas entradas se les suma un valor adicional que se lo conoce como sesgo, el cual direcciona la salida hacia arriba o hacia abajo. Posteriormente el resultado se lo transforma mediante la aplicación de una función de activación, esta función nos permitirá decidir si la neurona procede activarse o a desactivarse, logrando así imitar un comportamiento casi similar al de una neurona biológica, todo esto expresando de manera matemática.

### **2.4.2 Arquitecturas de capas**

Una red neuronal se encuentra constituida por múltiples capas, en donde nuestra primera capa será la encargada de recibir los datos. En la siguiente tenemos a una a varias capas ocultas, las cuales se encargan de procesar internamente toda la información adquirida. Para finalizar tenemos la capa de salida, la cual nos entrega la decisión del sistema y nos dará el resultado final después de analizar todos los datos que se le fueron entregados.

### **2.4.3 Funciones de activación**

Dentro de estos sistemas las funciones de activación son esenciales, ya que son las encargadas de introducir la no linealidad dentro de la red neuronal. Si se eliminaran estas funciones, el sistema tomaría un comportamiento lineal y no estaría en la capacidad de aprender relaciones complejas.

#### **2.4.4 Parámetros**

Dentro de los parámetros tenemos a los pesos y sesgos, los cuales son parámetros ajustables de la red neuronal, estos son responsables de transformar todos los datos de entrada y determinar el comportamiento o resultado final que nos dará la red neuronal. Estos parámetros cuando nos encontramos en el proceso de entrenamiento se modifican constantemente para eliminar errores entre la salida que nos genera la red y la salida pronosticada. El correcto ajuste de estos valores ayuda a que la red aprenda y se logre adaptar a los problemas que se le presenten.

#### **2.4.5 Función de pérdida**

A la función de pérdida o también conocida como función de costos es una medida del error entre la salida real que nos entrega el modelo y la salida pronosticada, esta herramienta es esencial debido a que nos guía durante el proceso de entrenamiento. El objetivo principal de esta función es minimizar este error aplicando procesos interactivos para lograr que el modelo aprenda de una manera efectiva.

#### **2.4.6 Algoritmo de entrenamiento y retropropagación**

El algoritmo de entrenamiento y retropropagación del error o conocido como backpropagation, es parte fundamental del proceso de aprendizaje supervisado con redes neuronales artificiales. El objetivo principal es el ajuste óptimo de los pesos sinápticos para minimizar el error entre la salida pronosticada y la salida real que me entregara el sistema.

#### **2.4.7 Capacidad universal de aproximación**

Uno de los principios teóricos fundamental dentro de estos sistemas es el teorema de aproximación universal, el cual establece que un sistema con características mínimas dentro de sus capas ocultas tiene que ser versátil, lo que quiere decir que está en la

capacidad de adaptarse a cualquier problema de predicción, clasificación y reconocimiento.

## 2.5 Fundamentos de comunicación

### 2.5.1 Protocolo de comunicación

Un protocolo de comunicación es un conjunto de lineamientos y convenciones los cuales permiten el intercambio de información entre varios dispositivos o sistemas. Definiendo como se inicia, mantiene y finaliza la comunicación, garantizando que los datos lleguen correctamente y de manera segura.

Los protocolos de comunicación poseen aspectos claves para una funcionalidad óptima. Primero el establecimiento de reglas específicas para el intercambio de datos entre sistemas heterogéneos, comenzando con la sintaxis que definirá la estructura formal del mensaje y seguidamente de una semántica que definirá un significado a cada elemento de la comunicación. En segundo lugar, tenemos a la temporización, se gestiona los tiempos y la sincronización para una transmisión eficiente. Tercero, se debe considerar su fiabilidad, esta nos asegura que los mensajes lleguen completos, de forma ordenada y sin pérdidas durante el camino. Finalmente, en el cuarto aspecto tenemos la característica de seguridad, encargada de proteger la integridad, confiabilidad y autenticación de todos los datos aplicando cifrados y control de acceso. Todas estas consideraciones mencionadas tienen como finalidad asegurar que los protocolos trabajen dentro redes robustas, sistemas basados en IoT y servicios dentro de la nube.

Tabla 2.1: Tabla de protocolos de comunicación

Protocolo	Descripción
HTTP/HTTPS	Protocolo de transferencia de hipertexto utilizado en la web. HTTPS añade cifrado SSL/TLS.
MQTT	Protocolo ligero ideal para IoT. Utiliza un modelo publicador/suscriptor.
CoAP	Protocolo para dispositivos IoT de baja potencia. Basado en REST, optimizado para redes limitadas.
WebSocket	Permite comunicación bidireccional en tiempo real sobre una sola conexión TCP.
FTP/SFTP	Protocolo para la transferencia de archivos. SFTP añade una capa de seguridad usando SSH.
TCP/IP	Conjunto de protocolos que define cómo se comunican los dispositivos en redes IP.
UDP	Protocolo rápido y sin conexión. Útil para transmisión de video y audio en tiempo real.
Modbus	Protocolo industrial usado para comunicación entre dispositivos de automatización.
CAN	Protocolo de comunicación en vehículos y sistemas embebidos. Robusto y en tiempo real.
Bluetooth	Protocolo inalámbrico de corto alcance. Ideal para dispositivos móviles y periféricos.
Zigbee	Protocolo inalámbrico para redes de sensores. Bajo consumo y bajo ancho de banda.

Protocolos de comunicación presentes en la industria para el manejo e intercambio de información, Anthony Pachacama

## 2.5.2 Application Programming Interface

Una API o Interfaz de Programación de Aplicaciones, es un conjunto de lineamientos, protocolos y herramientas las cuales permiten una interacción estructurada entre los diferentes sistemas del software. Se caracteriza por actuar como una interfaz abstracta que establece la comunicación entre los componentes, sin la necesidad de modificar o alterar su lógica interna de programación.

Características principales que presentan una Application Programming Interface:

- Encapsulamiento funcional para no revelar su implementación interna.
- Interoperabilidad para el manejo de distintos lenguajes de programación, obteniendo una comunicación estandarizada.
- Protocolos de transporte para recibir y enviar datos.
- Maneja distintos formatos de datos para una serialización y deserialización de información.

Tabla 2.2: Clasificación de las API por acceso

Tipo de API	Descripción
API pública	Disponible para cualquier desarrollador. Ejemplo: API de Google Maps. Promueve la innovación y la integración abierta.
API privada	Usada internamente en una organización. Mejora la eficiencia entre equipos sin exponer servicios al público.
API de socios	Compartida con socios estratégicos bajo acuerdos específicos. Combina control con colaboración externa.
API comuesta	Combina varias API para realizar múltiples operaciones en una sola llamada. Útil en microservicios.

Clasificación de las API según los permisos que se requieren para utilizarlas, Anthony Pachacama

Tabla 2.3: Clasificación de las API por arquitectura

Tipo de API	Descripción
REST	Usa HTTP y operaciones CRUD (GET, POST, PUT, DELETE). Es ligera, escalable y ampliamente adoptada.
SOAP	Protocolo basado en XML. Estándar en aplicaciones empresariales que requieren seguridad y transacciones complejas.
GraphQL	Permite a los clientes definir exactamente los datos que necesitan. Reduce el tráfico de red y optimiza el rendimiento.
gRPC	Basada en HTTP/2 y Protobuf. Muy eficiente en comunicación entre microservicios. Desarrollada por Google.
WebSocket API	Permite comunicación bidireccional en tiempo real. Ideal para chats, dashboards o monitoreo en vivo.
OpenAPI	Especificación para describir APIs REST. Facilita documentación y generación de código automático.

Clasificación de las API por cómo están estructuradas y organizadas técnicamente para comunicarse entre sistemas, Anthony Pachacama

Tabla 2.4: Clasificación de las API por interacción

Tipo de API	Características
Local API	Se utiliza dentro de una sola aplicación o sistema operativo. Por ejemplo, APIs de Windows o Android.
Web API	Expone funcionalidades a través de Internet, usando HTTP/HTTPS. Son la base de muchas aplicaciones móviles y web.
Hardware API	Permiten que el software controle dispositivos físicos como sensores, actuadores, cámaras, etc.
Biblioteca API (Library API)	Proveen un conjunto de funciones que se incluyen directamente en el código. Ejemplo: API de OpenCV o NumPy.

Clasificación de las API por interacción describe cómo se comunican los sistemas y cuándo ocurre el intercambio de datos, Anthony Pachacama

### 2.5.3 Interoperabilidad

Se conoce a la interoperabilidad como la capacidad estructural que tiene los sistemas digitales para comunicarse, comprender y ejecutar uno o más procesos colaborativos entre dispositivo, plataformas y servicios con características heterogéneas. Este fundamento busca estandarizar los protocolos utilizando formatos compatibles con JSON o XML, implementado modelos semánticos que no solo permiten el intercambio de información, además deben estar en la capacidad de interpretarla correctamente.

La interoperabilidad da la posibilidad de construir una arquitectura con características de modularidad, escalabilidad y resiliencia, fundamentales para que los componentes físicos

como sensores, lógicos como tarjetas controladoras o servicios dentro de la nube, logren vincularse entre sí, sin comprometer la funcionalidad del sistema general.

## **2.6 Métodos inteligentes de reciclaje que permita la gestión de residuos**

### **2.6.1 Sistema inteligente de reciclaje Open Bot**

Este proyecto planteado e implementado en la Universidad Técnica de Ambato, nos propone un contenedor inteligente para el reciclaje de envases plásticos de tipo PET, utilizando una cámara para el sistema de reconocimiento por visión artificial. El proyecto consta de una interfaz en Python, en donde se encuentra la programación general del sistema, una base de datos para el registro de sus usuarios, y un sistema que recompensa al insertar un envase plástico (Satuquinga Congacha, 2021).

### **2.6.2 Sistema de recolección optimizado mediante IoT**

Este proyecto planteado e implementado en la Universidad Técnica del Norte, nos propone el diseño de un sistema que integra sensores ultrasónicos para la detección del nivel de llenado dentro de los contenedores de residuos en sus tres posibles estados vacíos, medio y lleno. Y utilizando tecnología SigFox para el manejo de envío de datos en tiempo real a una aplicación web, la cual planificara las rutas de recolección. Este proyecto tiene un bajo costo operativo, bajo consumo de combustibles y como principal característica una baja saturación de contenedores (Obando Suárez, 2022).

### **2.6.3 Sistema inteligente de gestión de residuos mediante IoT**

Este proyecto planteado e implementado en la Escuela Superior Politécnica del Litoral, nos presenta el desarrollo de una red de contenedores inteligentes que constan de sensores que nos entregan valores de peso y proximidad, todos conectados a un mismo sistema de monitoreo remoto basado en el protocolo de comunicación MQTT y un control realizado en un entorno web. Todo esto con la característica de que se puede monitorear en tiempo real cada uno de los datos adquiridos (Ramírez Prado, Loor Castro & Cabello Sánchez, 2023).

### **2.6.4 Contenedor automatizado para residuos RAEE**

Este proyecto planteado e implementado en la Universidad San Francisco de Quito, nos presenta un contenedor especializado para residuos electrónicos y eléctricos. Consta de un sistema de apertura automática, compartimentos seguros y un espacio de materiales peligrosos. Tiene la finalidad de ser aplicado en espacio público, debido a que no tienen ningún sistema de control ni de supervisión, así que no necesita de un agente externo para su cuidado ni manteniendo (Proaño Avilés, Cali Vera, Molina López, Pástor del Salto & Sandoval Ocampo, 2020).

## CAPÍTULO 3

### MARCO METODOLÓGICO

#### 3.1 Propuesta de proyecto

Para el desarrollo e implementación del proyecto técnico que se detalla en este documento, proponemos una solución para la identificación de placas electrónicas de computadoras, utilizando visión artificial para el sistema de reconocimiento, identifica si es una placa electrónica de computadora u otros componentes que ingresen al sistema de reconocimientos, tomando en cuenta que este sistema está en la capacidad de discriminar placas electrónicas que no sean de computadoras u otros tipos de placas, debido a que otros tipos de placas no tienen los patrones y la distribución de elementos presente en una placa de computadora.

Al pasar por el sistema de reconocimiento identifica si es una placa de computadora, el sistema adquirirá los datos de trazabilidad que se encuentran dentro de un código QR. En estos se encuentran la información necesaria, que incluye características como; modelo, país fabricante, fecha de ensamblaje, fecha de compra y la fecha de regreso por daño.

Los datos adquiridos de trazabilidad, se subirán a un Google Sheets, que son hojas de cálculo web en las cuales el sistema escribirá en tiempo real todos los datos obtenidos de la visión artificial a partir de los códigos QR. Para esto se vincula el Python y Google Sheets, a través de una API, para lo cual utilizando la biblioteca gspread en conjunto con oauth2client para la autenticación a través de una cuenta que está asociada a los servicios que nos ofrece Google Cloud. Esta comunicación está fundamentada bajo los protocolos HTTP/HTTPS, con esto se logra realizar peticiones REST a nuestro servidor de Google y empleando el estándar OAuth 2.0 del protocolo de autenticación para procesar y aceptar el acceso a escritura dentro de la hoja de cálculos de Google Sheets.

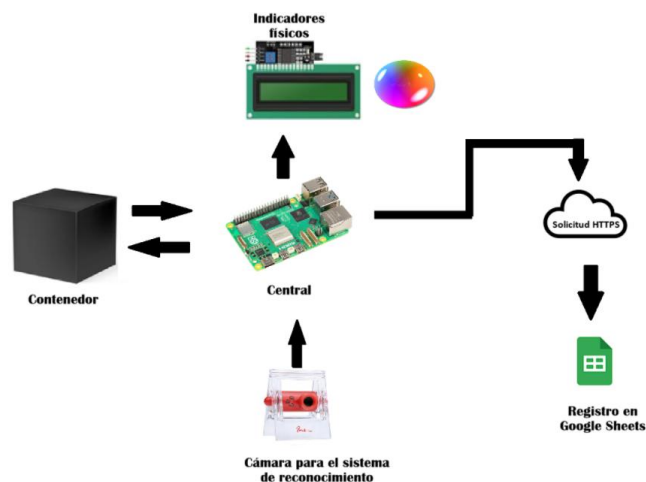
Una vez que los datos obtenidos a partir de los códigos QR sean subidos correctamente a la base datos en la hoja de cálculo web de Google Sheets, se activa un sistema de contención para ingresar la placa que anteriormente paso por un sistema de identificación y un sistema de adquisición de datos, este sistema de contención cumple la función de receptor las placas electrónicas hasta un determinado nivel.

Dentro del sistema de contención se implementa un sistema de control de nivel para notificar cuando el contenedor se encuentra a su máxima capacidad de almacenamiento, para lo cual se implementará un sensor sharp gp2y0a21yk0f y, que realizará el monitoreo en tiempo real del nivel, que, una vez llegada a una determinada altura, el programa

escrito en Python envía un mensaje al monitor serial indicando que se encuentra en su máximo nivel

El Smartphone se vincula con el sistema a través de la Aplicación Telegram, la cual es una aplicación de uso gratuito y está disponible en Android y IOS a través de TELEGRAM BOT API, este utiliza una comunicación basada en el protocolo HTTPS para intercambiar mensajes entre el sistema escrito en Python y la aplicación de Telegram. Una vez establecida la comunicación entre el sistema y el operario, al detectar que el sistema de contención está a su máxima capacidad de almacenamiento, enviara un mensaje a Telegram con una alerta que el contenedor está lleno y debe el operario acercarse.

Figura 3.1: Esquema general del sistema electrónico distribuido



Esquema general de interconexión de los elementos del sistema distribuido y sus recursos de visualización de datos, Anthony Pachacama

## 3.2 Equipos necesarios para la implementación

### 3.2.1 Dispositivo para la captura de imágenes

Tomando en cuenta todo lo anteriormente expuesto se optó por adquirir la cámara web INS M014, la cual posee una compatibilidad universal con los sistemas operativos Windows, macOS y Linux, facilitando su uso dentro de Python al tener una compatibilidad versátil y lograr probar nuestro sistema tanto a la hora de realizar pruebas de detección de imágenes en nuestra computadora personal, como a la hora de implementarlo en nuestros prototipos y finalmente en nuestro sistema final, dándonos como resultado una alta fiabilidad.

Figura 3.2: Webcam Ins



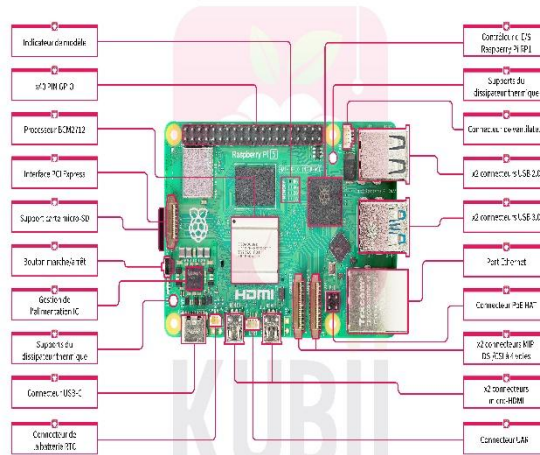
Webcam Ins M014, Usb, jack 3.5 Mm, 5mp, sin micrófono (Casa Éxito, s.f.)

### 3.2.2 Raspberry Pi 5

Para la implementación de este proyecto se optó por seleccionar el microordenador Raspberry Pi 5, e cual es un dispositivo compacto que posee todas las características de una computadora tradicional como lo son el procesador, memoria RAM, unidad de almacenamiento, puertos de entrada y salida. Todas estas características integradas dentro de una sola placa, la cual trabaja con el sistema operativo de Linux.

Esta placa de bajo costo es actualmente la versión más potente que está en el mercado, y es ampliamente usada para el desarrollo de prototipos, automatización de procesos, aplicaciones con IoT, diseños de interfaces HMI y es perfecto para aplicaciones de inteligencia artificial por sus capacidades y mejoras a comparación de sus versiones anteriores.

Figura 3.3: Raspberry Pi 5



Computadora de placa única Raspberry Pi 5 (CanaKit, s.f.)

Tabla 3.1: Características de la Raspberry Pi 5

Componente	Especificación
Procesador (SoC / CPU)	Broadcom BCM2712 con 4 núcleos ARM Cortex-A76 a 2.4 GHz, 64 bits, 512 KB de caché L2 por núcleo y 2 MB de caché L3 compartida
GPU	Broadcom VideoCore VII a 800 MHz, compatible con OpenGL ES 3.1 y Vulkan 1.2
Memoria RAM	LPDDR4X-4267; versiones disponibles de 2 GB, 4 GB, 8 GB y 16 GB
Almacenamiento	Ranura microSD UHS-I (modo SDR104, hasta 104 MB/s); compatible con SSD vía PCIe mediante adaptador externo
Puertos de Pantalla	2x micro-HDMI con soporte 4Kp60 HDR
Puertos de Cámara/Pantalla	2x conectores MIPI CSI/DSI de 4 carriles (1.5 Gbps por canal)
USB	2x USB 3.0 (5 Gbps simultáneos) + 2x USB 2.0
Ethernet	Gigabit Ethernet (1 Gbps), compatible con PoE+ mediante HAT
Conectividad inalámbrica	Wi-Fi 802.11ac de doble banda (2.4/5 GHz) + Bluetooth 5.0 / BLE
Interfaz PCIe	1x PCIe 2.0 x1 vía conector FPC, controlado por chip RP1 (requiere adaptador adicional para SSD M.2 o HAT de expansión)
GPIO (40 pines)	Compatible con estándares anteriores Raspberry Pi, soporte para SPI, UART, I <sup>2</sup> C, PWM, etc.
Chip de I/O (Southbridge)	RP1 (chip propio de Raspberry Pi): controla USB, Ethernet, CSI, DSI, GPIO, SPI, I <sup>2</sup> C, UART, PWM, I <sup>2</sup> S, QOS y más
Alimentación	USB-C con Power Delivery: 5 V / 5 A; soporte para carga rápida hasta 15 V
Reloj en Tiempo Real (RTC)	RTC con respaldo por batería externa CR1220 (no incluida)
Botón de encendido/apagado	Incluido físicamente por primera vez en una Raspberry Pi
Ventilador y UART dedicados	Pines dedicados separados del GPIO principal para ventilador y UART (liberando multiplexación)
Dimensiones	85 x 56 mm (idénticas a Raspberry Pi 4, compatible con muchas carcasas existentes)
Disponibilidad garantizada	En producción hasta al menos enero de 2035 (útil para proyectos industriales a largo plazo)

Especificaciones técnicas de una Raspberry Pi 5, Anthony Pachacama

### 3.2.3 Sensor de Distancia Sharp GP2Y0A41SK0F

El sensor de Distancia Sharp, es un sensor óptico infrarrojo que mide una distancia tomando como punto de inicio su posición actual y la siguiente la del objeto en cuestión que deseamos saber la distancia. A diferencia de otros sensores de distancia, este sensor tiene una alta precisión para distancias cortas y medias. Este sensor emite un haz de luz de tipo infrarrojo, el cual se refleja en el objeto en cuestión que detecte y regresará al sensor, esta señal al retornar ingresa en un detector de posiciones o PSD, el cual mide el ángulo de retorno y a partir de este dato calcula la distancia con un rango de error mínimo.

Figura 3.4: Sensor de Sharp

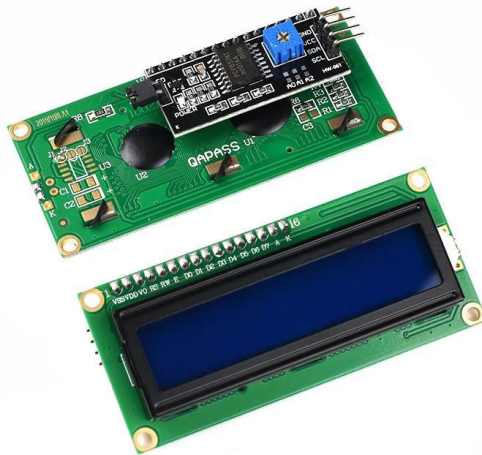


Sensor de Distancia Sharp GP2Y0A41SK0F de 4cm a 30cm, (AV Electronics, s.f.)

### 3.2.4 LCD 16x2 I2C

El LCD 16x2 con interfaz I2C, cuenta con características alfanuméricas que consta de una estructura de visualización de 16 columnas por 2 filas, cada carácter que podemos visualizar está compuesto por una matriz de 5x8, la interfaz I2C que posee tiene un chip PCF8574 para la comunicación entre la tarjeta controladora.

Figura 3.5: Display LCD



Display LCD 16x2 con interfaz I2C, (Naylamp Mechatronics, s.f.)

## 3.3 Factores relevantes para la implementación

### 3.3.1 Luminosidad del entorno

Durante el desarrollo de un sistema de reconocimiento empleando visión artificial, uno de los factores primordiales que se deben tomar en cuenta es la luminosidad del entorno,

el cual es un aspecto fundamental para tener una alta precisión en el sistema de visión artificial, ya que, al tener una correcta luminosidad para nuestro sistema, tendremos como resultado una alta calidad de las imágenes obtenidas por los sensores.

Al implementar el sistema en un entorno con una baja luminosidad, tenemos como resultado imágenes oscuras, pérdida de detalles en las imágenes y una reducción del rango dinámico, lo que conlleva en nuestro sistema a pérdidas de datos y características que optimizarían nuestro sistema de reconocimiento.

Al implementar el sistema en un entorno con una alta luminosidad, tenemos como consecuencia que los sensores se saturan, lo que lleva a los valores de pixel al máximo, lo que con lleva a un efecto de blanqueamiento o clipping, dando como resultado la pérdida de información en áreas que tengas características de reflectancia a la luminosidad.

Un factor muy importante a tomarse en cuenta al momento de hacer un procesamiento digital de imágenes, es tener un sistema libre de ruido digital, esto quiere decir no se debe tener variaciones aleatorias en la intensidad, brillo o color que posea una imagen.

### 3.3.2 Ángulos de visión

Los ángulos de visión, es el campo visual que un sistema de captura de imágenes puede tener desde su posición fija, se la expresa en grados. Por lo que es crucial seleccionar el ángulo de visión más adecuado para nuestro sistema y en base a esto tomar en cuenta factores técnico relevantes como la distancia focal, resolución del sensor y distorsión óptica.

Tabla 3.2: Tabla de ángulos de enfoque en fotografía

Tipo de Ángulo de Enfoque	Rango Aproximado de Ángulo	Características Principales	Uso Común
Ángulo estrecho	< 20°	Campo visual pequeño, alta precisión en objetos distantes	Fotografía de deportes, naturaleza, microscopios
Ángulo medio	20° - 50°	Campo de visión equilibrado, enfoque moderado	Fotografía general, retratos
Ángulo amplio	50° - 90°	Campo de visión amplio, captura paisajes o interiores	Fotografía de paisajes, arquitectura
Ángulo ultra amplio	> 90°	Máximo campo de visión, posible distorsión visible	Cámaras de acción, vigilancia, panorámicas

Rango angular que determina cuánto campo visual puede capturar un sistema óptico, Anthony Pachacama

### **3.3.3 Procesamiento y algoritmo eficiente**

Al tener un procesamiento de imagen eficiente tenemos como resultado datos libres de ruidos y correctas segmentaciones. Teniendo como resultado la obtención de características fiables como lo son sus bordes, texturas y colores para su identificación correcta y verificar que no se trate de algún otro objeto que no sea una placa de computadora,

### **3.3.4 Arquitectura eficiente de una red neuronal**

Para lograr una arquitectura eficiente dentro de nuestra red neural, se debe primero establecer la estructura base del modelo que son las capas, las cuales recibirán, procesarán y nos presentara un resultado final, logrando así determinar si tenemos un correcto flujo tanto de información como de aprendizaje. En el segundo punto tenemos la selección del tipo de red más eficiente para este proyecto, en nuestro caso utilizamos una red neuronal CNN debido a que este tipo de sistema nos ayuda a capturar datos relevantes de una imagen lo que es ideal para un sistema de identificación y clasificación. En el tercer punto tenemos a la correcta afinación de parámetros para que nuestro sistema tenga un mejor desempeño y eliminar los errores futuros.

## **3.4 Desarrollo del sistema**

### **3.4.1 Programación para la captura automática de imágenes de placas**

Se realizó un Script en Python, el cual desde una cámara web tomara las imágenes de las placas de computadora, adquiriendo con esto las características de las zonas de mayor interés, además de otras características físicas como el color y el modelo en específico el cual le presentamos. Este programa además organiza y crea de forma automática carpetas creando así una base de datos ordenada y eficiente de todos los datos que adquiriremos

Este programa al iniciar, solicita que introduzcamos el color y el modelo de la placa, posteriormente la cámara se accionara para proceder con la captura de las características. Se aplican técnicas de visión computacional como la conversión a escalas de grises, desenfoco gaussiano y detección de bordes por Canny para la identificación del contorno. El sistema cada dos segundos tomara y almacenara la imagen.

Figura 3.6: Script para la decepción de características de las placas electrónicas

```
if best_rect:
    x, y, w, h = best_rect
    margin = 20
    x = max(x - margin, 0)
    y = max(y - margin, 0)
    w = min(w + 2 * margin, frame.shape[1] - x)
    h = min(h + 2 * margin, frame.shape[0] - y)
    cv2.rectangle(display_frame, (x, y), (x + w, y + h), (0, 255, 255), 2)

cv2.putText(display_frame, f'Capturas: {count}', (10, 30),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
cv2.imshow("Captura", display_frame)

current_time = time.time()
if current_time - last_capture_time >= intervalo:
    imageName_full = f'placa_{count}_completa.jpg'
    imagePath_full = os.path.join(placaPath, imageName_full)
    cv2.imwrite(imagePath_full, frame)
    print(f'📸 Imagen completa guardada: {imageName_full}')

    if best_rect:
        roi = frame[y:y+h, x:x+w]
        if roi.size > 0:
            imageName_roi = f'placa_{count}_recorte.jpg'
            imagePath_roi = os.path.join(placaPath, imageName_roi)
            cv2.imwrite(imagePath_roi, roi)
            print(f'📸 Imagen recortada guardada: {imageName_roi}')

    count += 1
    last_capture_time = current_time

key = cv2.waitKey(1) & 0xFF
if key == 27 or key == 50: # ESC o tecla '2'
    print("🛑 Captura finalizada por usuario.")
    break
```

Script para la captura y recorta automáticamente imágenes de placas, organizándolas por color y modelo en carpetas estructuradas, Anthony

Pachacama

Figura 3.7: Diagrama de flujo del algoritmo de programación para la captura automática de imágenes de las placas electrónicas

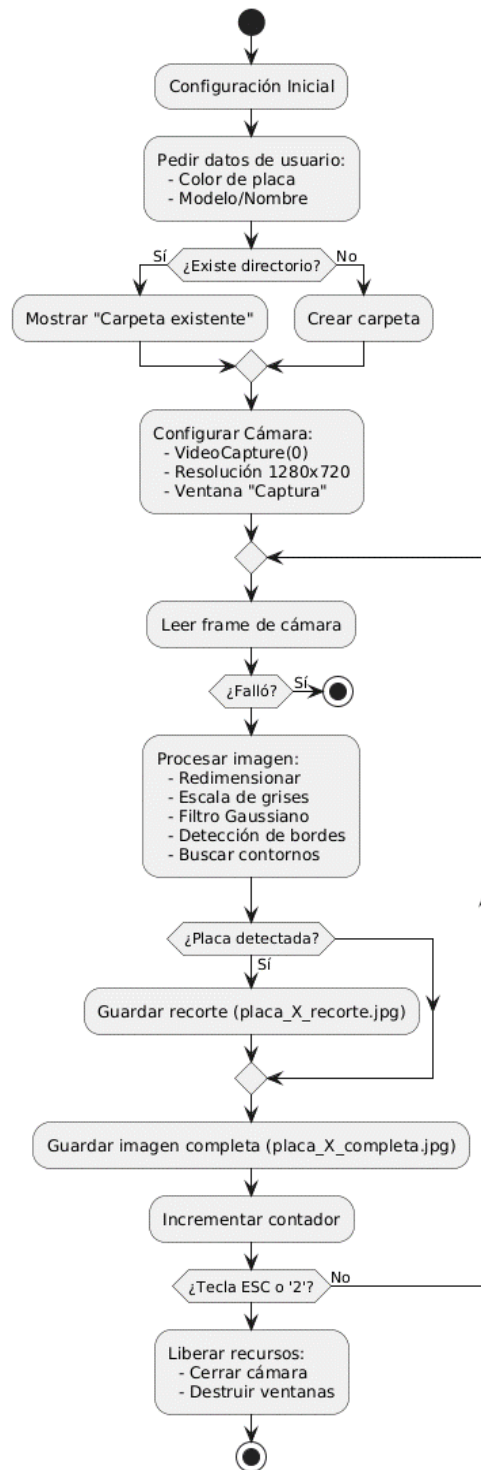


Diagrama de flujo del proceso para la captura automática de imágenes de placas las placas electrónicas de computadora, Anthony Pachacama

### 3.4.2 Programación de entrenamiento

Para nuestro sistema de entrenamiento creamos un código de reconocimiento de imágenes usando un algoritmo LBPH o por sus siglas Local Binary Patterns Histograms de OpenCV, el cual se encuentra orientado dentro de nuestro sistema para el reconocimiento de placas electrónicas de computadoras, tomando las características de color y modelo que adquirimos anteriormente. Para lo cual jerarquiza las imágenes por sus características, se procesa y se etiqueta de manera ordenada, con estos datos entrenamos nuestro modelo LBPH y finalmente el modelo se lo guarda en conjunto con las categorías de reconocimiento para su posterior uso dentro de nuestro sistema general.

Para la lectura y clasificación de imágenes, el programa recorre las carpetas en donde se almacena toda la información que se adquirió previamente de las placas, posteriormente se redimensiona a 350x150 píxeles, procede a guardar su etiqueta numerada que se almacena a partir de la categoría. Después utilizando todos los datos procesados y etiquetados, entrenamos nuestro modelo de reconocimiento, en nuestro caso reconociendo patrones y elementos físicos de relevancia. Finalmente, se exporta el modelo en un archivo en formato XML y un archivo de texto en donde se muestran las categorías.

Figura 3.8: Script para el entrenamiento del sistema de reconocimiento por visión artificial

```
# Recorrer carpetas color/modelo
for color in os.listdir(dataPath):
    colorPath = os.path.join(dataPath, color)
    if os.path.isdir(colorPath):
        for modelo in os.listdir(colorPath):
            modeloPath = os.path.join(colorPath, modelo)
            if os.path.isdir(modeloPath):
                category_name = f"{color}_{modelo}"
                categories.append(category_name)
                print(f'📁 Leyendo imágenes de: {category_name}')
                imageCount = 0
                for fileName in os.listdir(modeloPath):
                    filePath = os.path.join(modeloPath, fileName)
                    if fileName.lower().endswith(('.jpg', '.jpeg', '.png')):
                        img = cv2.imread(filePath, cv2.IMREAD_GRAYSCALE)
                        if img is not None and img.size > 0:
                            img_resized = cv2.resize(img, (350, 150)).astype('uint8')
                            imagesData.append(img_resized)
                            labels.append(label_id)
                            imageCount += 1
                print(f'✅ {imageCount} imágenes procesadas para "{category_name}")')
                label_id += 1
```

Script que procesa las imágenes en escala de grises, entrena un modelo LBPH y guarda tanto el modelo como las categorías para reconocimiento automático, Anthony

Pachacama

Figura 3.9: Script para el entrenamiento del sistema de reconocimiento por visión artificial

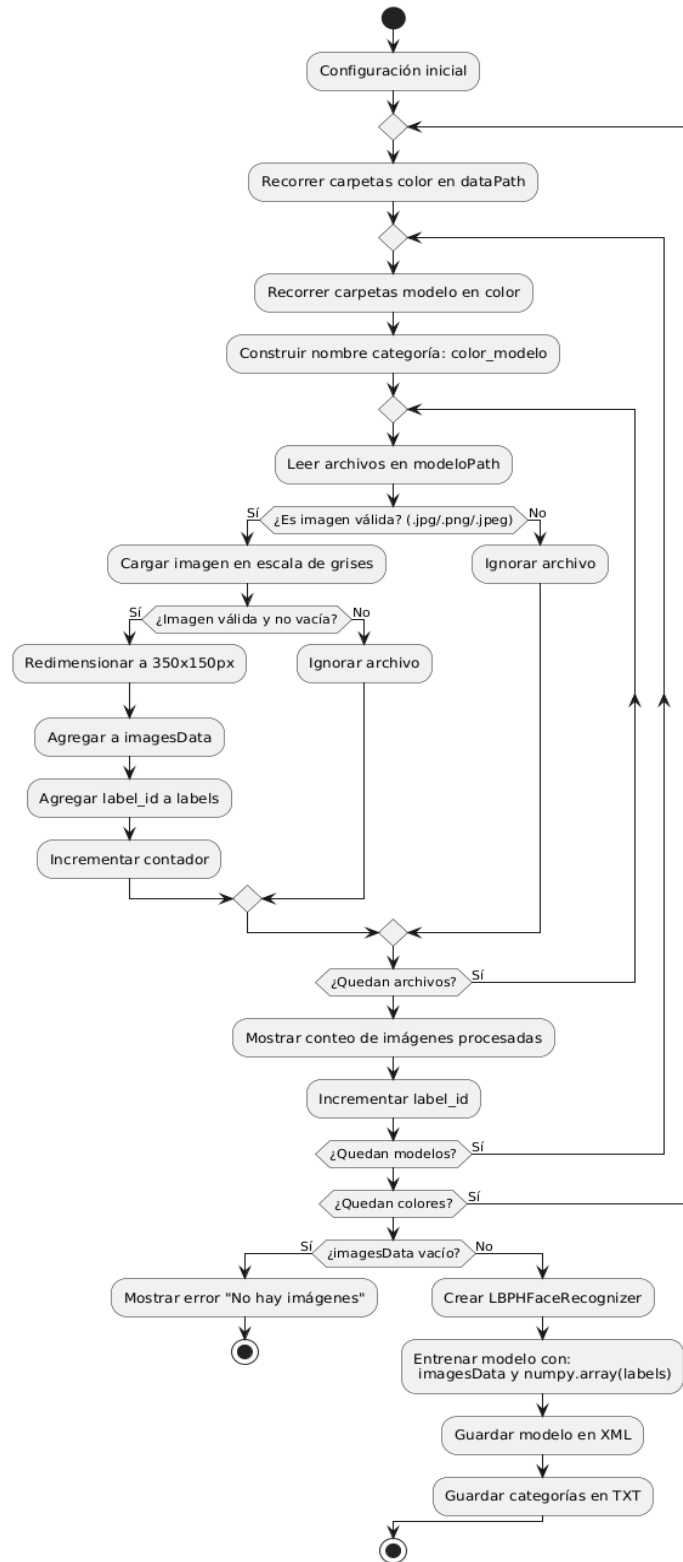


Diagrama de flujo de entrenamiento y exportación de modelo LBPH para clasificación de placas

### 3.4.3 Generación de QR

Este programa realizado en Python genera códigos QR únicos, los cuales contiene la información detallada de toda su ruta de trazabilidad y algunos detalles técnicos básicos de cada placa electrónica. Este programa convierte la información en un texto estructurado y se la codifica en un código QR de alta calidad. El resultado final es una imagen de alta calidad que contiene los QR con la información.

Figura 3.10: Script para la Generación de Códigos QR Industriales con Registro Automatizado en Google Sheets

```
# === CONFIGURACIONES ===
qr_folder = r'C:\Users\jeanr\Desktop\Proyectos Akamaru\Clasificador de placas'
os.makedirs(qr_folder, exist_ok=True)

nombre_hoja = 'InventarioPlacas'
credenciales_path = r'C:\Users\jeanr\Desktop\Proyectos Akamaru\Clasificador de placas\credenciales.json'

placas = [
    {'nombre': 'Intel_G41M_Motherboard', 'fabricante': 'Intel Corporation', 'fabricacion': 'Shenzhen, China', 'pais': ['China', 'Vietnam'], 'uso_en': ['HP El
    {'nombre': 'AMD_A320M_Chipset', 'fabricante': 'AMD Inc.', 'fabricacion': 'Pune, India', 'pais': ['India', 'Taiwán'], 'uso_en': ['ASRock A320M', 'MSI Pro
    {'nombre': 'NVIDIA_GTX1050_TI', 'fabricante': 'NVIDIA', 'fabricacion': 'Guadalajara, México', 'pais': ['México', 'Alemania'], 'uso_en': ['Dell Inspiron G
    {'nombre': 'ASUS_2370_Prime', 'fabricante': 'ASUSTeK Computer Inc.', 'fabricacion': 'Taipei, Taiwán', 'pais': ['Taiwán', 'Malasia'], 'uso_en': ['ASUS ROG
    {'nombre': 'Gigabyte_B550M_DS3H', 'fabricante': 'Gigabyte Technology', 'fabricacion': 'Bangkok, Tailandia', 'pais': ['Tailandia', 'Singapur'], 'uso_en':
    {'nombre': 'MSI_H310M_PRO', 'fabricante': 'MSI', 'fabricacion': 'Ho Chi Minh, Vietnam', 'pais': ['Vietnam', 'Corea del Sur'], 'uso_en': ['Corporate PCs',
    {'nombre': 'EVGA_GTX1660_SC', 'fabricante': 'EVGA', 'fabricacion': 'California, USA', 'pais': ['EE.UU.', 'Canadá'], 'uso_en': ['Gaming PCs', 'Mini PCs']}
    {'nombre': 'Raspberry_Pi_CM4', 'fabricante': 'Raspberry Pi Foundation', 'fabricacion': 'Cambridge, Reino Unido', 'pais': ['UK', 'Sudáfrica'], 'uso_en': [
    {'nombre': 'Intel_NUC_10i7FNH', 'fabricante': 'Intel Corporation', 'fabricacion': 'Penang, Malasia', 'pais': ['Malasia', 'Filipinas'], 'uso_en': ['Mini E
    {'nombre': 'ASRock_H470M_ITX', 'fabricante': 'ASRock Inc.', 'fabricacion': 'Taoyuan, Taiwán', 'pais': ['Taiwán', 'India'], 'uso_en': ['Pequeños servidores
]
```

Script generador de códigos QR únicos con datos industriales aleatorios y los exporta como imágenes, Anthony Pachacama

Figura 3.11: Diagrama de flujo del algoritmo implementado para el dispositivo central

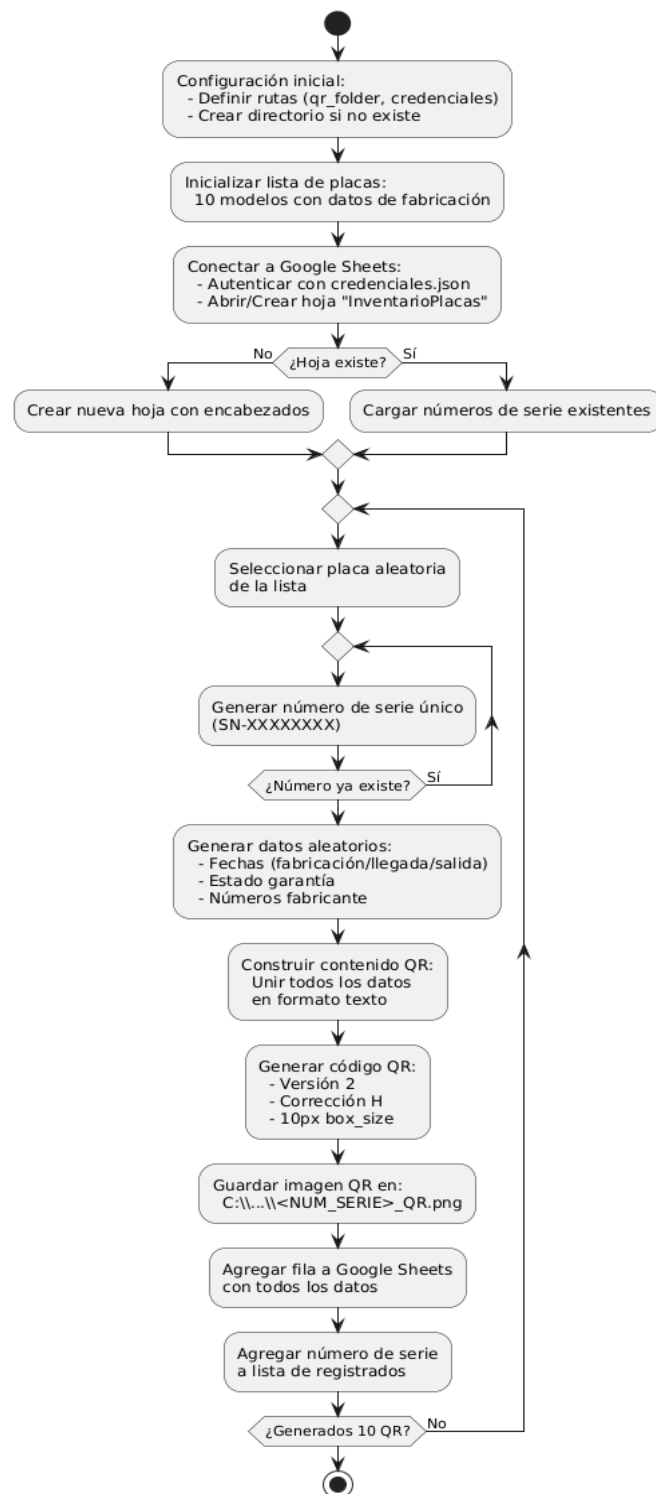


Diagrama de flujo para la Generación y Registro de Códigos QR para Componentes Industriales, Anthony Pachacama

### 3.4.4 Programación Main del sistema general

Dentro del programa Main se tiene la función principal que es el reconocimiento de placas electrónicas en tiempo real mediante la aplicación de visión artificial, una vez aplicado el reconocimiento procede a escanear los QR para adquirir la información que contienen, estos datos se verán reflejados en el display con interfaz I2C y el indicador led, posteriormente esta información se la pasará a la nube dentro de un registro dentro de una hoja de cálculos que nos ofrecen los servicios de Google Sheets. Una vez acabado el proceso de reconocimiento y adquisición de datos se acciona el servomotor que controla el acceso de las placas electrónicas al contenedor. Finalmente, cuando el contenedor se haya llenado el programa enviará un mensaje a la aplicación Telegram, que el contenedor se encuentra al máximo de su capacidad.

Para la interacción directa con el entorno físico del sistema, se aplicaron controladores GPIO, para leer el estado del sensor Sharp y activar nuestros leds que actúan como indicadores físicos, realizando una gestión de líneas de entrada y salida de una manera eficiente.

Para la fase de reconocimiento por visión artificial, ocupamos OpenCv para su uso en escalas de grises, aplicación de desenfoque y uso de umbral adaptativo. Además, tenemos la detección de contorno con filtración de área y número de vértices de los que el objeto está compuesto, el cual identifica regiones rectangulares similares a los de la placa. Ocupamos LBPH o Local Binary Patterns Histograms, para el reconocimiento de patrones que fueran entrenados previamente. Aplicando el ROI o Región de interés, clasificamos y visualizamos la predicción de confianza. Para finalizar tenemos la decodificación de los QR con la librería pyzbar, que extrae los datos almacenados.

Para la comunicación con la nube, se utilizó la librería gspread para el proceso de autenticación vía OAUTH 2.0. En donde cada entrada consta de un timestamp, modelo o color de placa, nivel de confianza en la predicción y la detección del QR. Finalmente, tenemos el API REST con la librería request para el envío de información a través de un bot de la aplicación Telegram.

En la sección de automatización inteligente se aplica lógica condicional para la activación o desactivación de acciones. Esto da como resultado que el sistema se convierte en una unidad de decisión autónoma, con la capacidad de reaccionar al entorno con acciones eficaces y precisas.

Figura 3.12: Script para el reconocimiento automatizado de placas con control físico, registro cloud y alertas en Tiempo Real

```
# === Configuración GPIO con gpiod ===
chip = gpiod.Chip('gpiochip0')
led_verde = chip.get_line(17)
led_rojo = chip.get_line(27)
led_amarillo = chip.get_line(22)
sharp_sensor = chip.get_line(18)

led_verde.request(consumer="led_control", type=gpiod.LINE_REQ_DIR_OUT, default_vals=[0])
led_rojo.request(consumer="led_control", type=gpiod.LINE_REQ_DIR_OUT, default_vals=[0])
led_amarillo.request(consumer="led_control", type=gpiod.LINE_REQ_DIR_OUT, default_vals=[0])
sharp_sensor.request(consumer="sharp_sensor", type=gpiod.LINE_REQ_DIR_IN)

# === Configuración servo con gpiozero y lgpio backend ===
factory = LGPIOFactory(chip=4) # Ajusta chip si es necesario

# Crear servo sin initial_angle para evitar movimiento inicial inesperado
servo = AngularServo(12, pin_factory=factory, min_angle=0, max_angle=180,
                    min_pulse_width=0.0005, max_pulse_width=0.0025)

def mover_servo_a(angulo):
    if 0 <= angulo <= 180:
        print(f"Moviendo servo a {angulo} grados")
        servo.angle = angulo
    else:
        print("Ángulo fuera de rango (0-180)")
```

Script que detecta y clasifica las placas, lee el QR asociado y registra toda la información en Google Sheets y controla la parte física del sistema, Anthony Pachacama

Figura 3.13: Diagrama de flujo del sistema principal



Diagrama de flujo del análisis visual, respuesta física, y trazabilidad digital, Anthony Pachacama

### 3.4.5 Vinculación con la nube

Para nuestra vinculación a la nube, en la primera fase se crea la credencial dentro de Google Cloud Console, creamos un proyecto y habilitamos la API de Google Sheets, posteriormente creamos una cuenta de servicio dentro del proyecto, asignamos los permisos y se genera un clave con formato JSON, esta contiene los datos fundamentales para la autenticación del script de Python mediante OAuth 2.0.

En la segunda fase, adecuamos las celdas de la hoja de Google Sheets, para que la información de los QR se ubique de forma correcta y en el orden con el que se les asigno al momento de codificarlos.

En la tercera fase, llamamos dentro del programa Main a las credenciales para establecer comunicación, aplicando librerías nativas de Pyton, y lógica de programación para subir los datos que adquirimos de la visión artificial.

Dentro del archivo JSON que observamos en la Figura 3.#, se encuentran las credenciales de la cuenta de servicio de Google Cloud, estas nos permitirán establecer la comunicación con Google Sheet.

Figura 3.14: Archivo JSON

```
{
  "type": "service_account",
  "project_id": "plcas-pachacama",
  "private_key_id": "4f33b48db453bf65e4a02804274cef6aa809acf0",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEVAIBADANBgkqhkiG9w0BAQEFAASCByNnggsIAGeAa0IBAQKZiJaYeYk/SRns5AspL2p1GhAc07aXFKuB+5Hy6D9E3b1fkwsc031EjE1T7xvPdbh+ABS+P+X3VnVnRGMU2h75kKkgH+PMAE325dkt0i0pAC1HhT7Hul5ZVA0J/9Fp7kRlD2lFD-QE\nVnu46jYZZZ/pRCp31x17nQ929krc4MTBkr77of5Uw6QNFra0A4s92Uqxyqk0L1VzU0k1bHTV1th7HDUQ63eZdtm9q0b8AdNwApoSND1+IunTrsTwKR3YkUd0IeJn\nVNB5G202Lcu5/q7B7rXQ98VBSj2rvQ2uHDp6Xjps3R3AFLDAnuRMTD3EBXG/vn+muZzIZAgMBAEECgeADHxeY0Ih3cI1pvHyEx5jycFdqVuxFvVtNBgwV1M231G8\nVnPutDF/vnMw6HFbQk0k0Cyc7j20542VbdmpTCLBwvHr2P1cz9yQJCdMwBwEtoG/vnG2D0cVp1kkyhJ6W5118r0+xbSH7QH8kXCV5tYJMM/P6IOSr5D2QIld2/1b9p5JH\nVnJtsa0mqT82kH90F9XhMjJr5kyt1tKZgZTmWbKerR5Jx8hpH7z0to8k52vNX9on7/MAQuHf1BvW8DnRp77kCmHjBFJum+XRFhQCRG0dUNVeh4xm1jF89scSUJG\nVnvURLE1Tuz6dax6QlV81HbnasBZozXnDooCjYga3Q0B8QPPvR8+Jhnm9LkNkVnmDensCms1Z5SuFzTKCLs5dxXV6UQ77g6W6131BpJymeJhwcbnY1p0H1yx\nVtanzu1uYEn8jK9p0dsy/nw40/a8R5mH+49ozymLEjrc10LQ2xqG10uL/RvNkyV1TnQX11/vAntp0M/6zUMK9G6C647A0ImxYV/FCTIhesP11n0qenvtZFa\nVnHX2LngvXK5zabyZ91A4xk07no1EHuZcuHgVLRd7/NEH7adtnV6P6BENNtY+ZvndbN/DoV6d6x705BNH+h1i1zA8Nhwldu83VkwLPnsTEVQ2uccucc3FBpLhSNCZ\nVnClw/621Y6K8F0c1U42w+kYtQLpyoN1YeSn1UPio8q2Skt1+5L47xFFG+vNB3ktY5rYnVs1/H7J1uBtc2335YE9scs3ou51kVtXKz0V0kQ/hHjLy1A27e6+R2\nVnH85MvtivakHP0ImUhu23XnDudfI569yq9kInU17K0s+Bk+11sX1hA0GAX11T/vnBx5Vn6w32TnJjeU0m3Rmb4sXkR7HUREC7vuoZim+RTKqvXnt/608J/uxp0B4H\nVnB5/rD831G5Wk6xWDS78e803v5j5xIT+86eF+CZFT1cgh00gEk+0o41x+nACKvN9hnAF4J3CSoyMh1hcqzGyAN5rs3YyFD5sIRKCYAZMl9G1r4eyLoNS83rHEK\nVnMwKf099aeFKQ0/nQ0C2Z3F45nTm6Qsef7Cb8aHkXr85ZuwrUppBRV725hVnR+1Gz6E+e0N3A3+Hq76yWvSeouf7L012m22IDRkhaokGrwHqou1r0L18ou+5k8\nVn-----END PRIVATE KEY-----\n",
  "client_email": "gspread-access@plcas-pachacama.iam.gserviceaccount.com",
  "client_id": "117126201026852321677",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/gspread-access%40plcas-pachacama.iam.gserviceaccount.com",
  "universe_domain": "googleapis.com"
}
```

Credenciales que contiene el archivo JSON para la vinculación con los servicios de Google, Anthony Pachacama

### 3.5 Conexiones del sistema

El hardware del proyecto consta de la parte de indicadores físicos en donde tenemos los LEDs y la pantalla LCD con módulo I2C, como se observa en la Figura 3.15 y Figura 3.17. Dentro del módulo de contención tenemos un sensor Sharp el cual esta sujeto en la parte superior y presenta la conexión como se aprecia en la Figura 3.16.

Figura 3.15: Conexión de LEDS

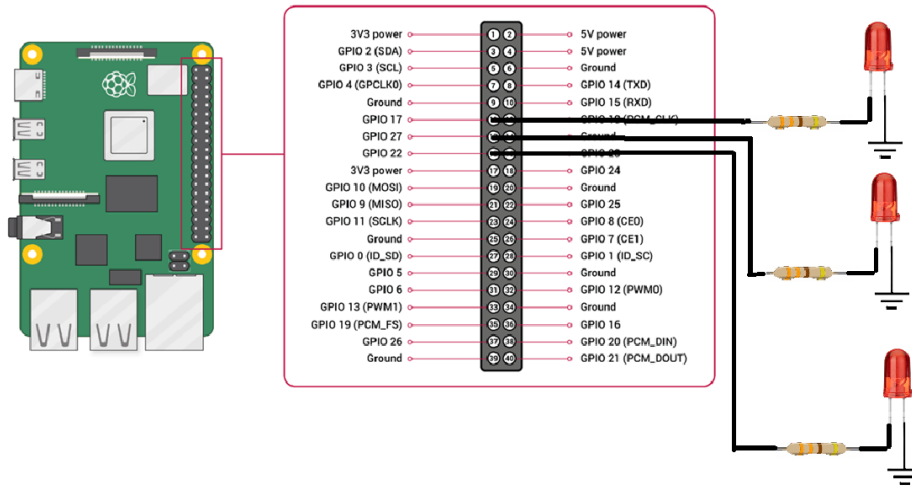


Diagrama de conexión de LEDS indicadores, Anthony Pachacama

Figura 3.16: Conexión del sensor Sharp

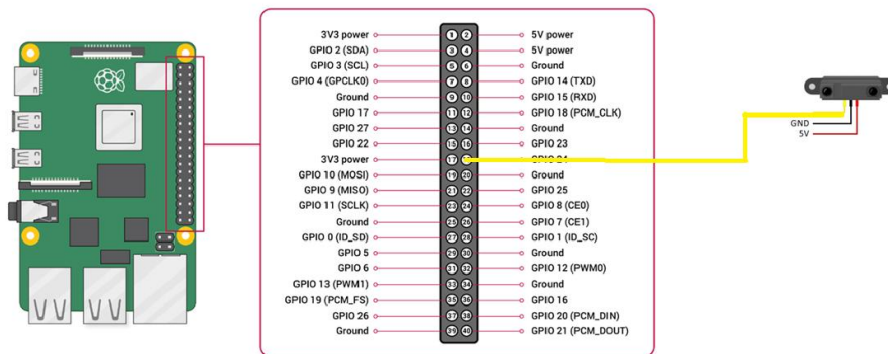


Diagrama de conexión del sensor Sharp, Anthony Pachacama

Figura 3.17: Conexión del I2C

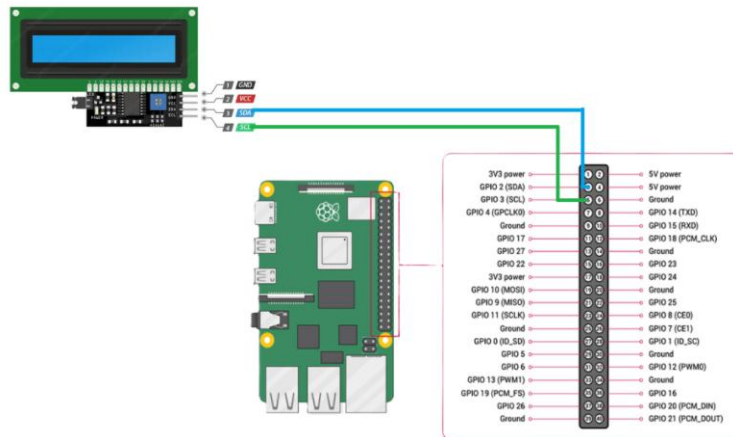


Diagrama de conexión del LCD I2C, Anthony Pachacama

## CAPÍTULO 3

### PRUEBAS Y RESULTADOS

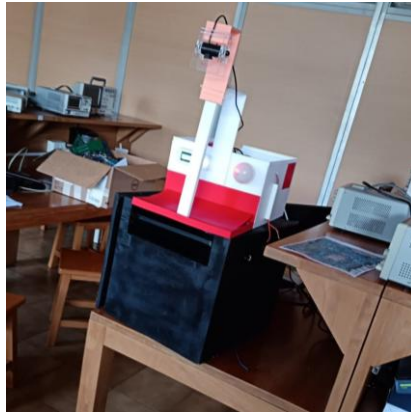
Una vez acabado e implementado en su totalidad nuestro sistema. La parte de reconocimiento por visión artificial en sus primeras etapas de desarrollo presento varios inconvenientes al darnos respuestas erróneas, hasta perfeccionar el sistema con diversos métodos y modificaciones. En la segunda parte tenemos el sistema de conexión con la nube, esta parte es fundamental para el manejo de subida de los datos, teniendo como uno de los principales retos la vinculación con los servicios Cloud de Google. En la tercera parte tenemos al contenedor de placas electrónica el cual consta de varios subsistemas de control, el cual presentaba errores debido a la actualización de librerías para nuestro modelo de Raspberry.

#### 4.1 Pruebas del sistema de reconocimiento

Durante todo el desarrollo del sistema de reconocimiento, se realizaron varias modificaciones las cuales van desde el cambio de lugar a un ambiente con mejor control de luz para la adquisición de datos. Debido a que durante el proceso de entrenamiento teníamos problemas con ciertas imágenes que poseían un mayor entorno iluminado y no se lograban capturar áreas de gran relevancia que tiene la placa, además de otras características como colores y texturas que las diversas placas poseen. Al principio se implementó el sistema con la cámara nativa de la Raspberry Pi 5, debido a su lenta respuesta para adquirir las imágenes y su baja resolución, afecto el reconocimiento y lectura de los QR, imposibilitando adquirir datos claros y rápidos, por lo que se optó por reemplazarla por un cámara Web Ins. Esta modificación de equipos nos retrasó dos semanas debido a que el sistema ya estaba adaptado y programado con librerías nativas de Python y al aplicar la nueva cámara nos vimos en la necesidad de volver a entrenar todo nuestro sistema, sin mencionar cambios en las librerías y adaptadores para poder aplicar la nueva cámara.

Una vez modificado y afinado todos los parámetros de nuestro sistema de reconocimiento, se llevaron pruebas en distintas áreas del Bloque C, como se puede observar en la Figura 4.1, Figura 4.2 y Figura 4.3. Logrando tener un óptimo desempeño en cada uno de los espacios.

Figura 4.1: Pruebas dentro del Laboratorio de Analógica



Pruebas dentro del Laboratorio de Analógica con baja luminosidad, Anthony Pachacama

Figura 4.2: Pruebas dentro del Laboratorio de Robótica



Pruebas dentro del Laboratorio de Robótica con alta luminosidad, Anthony Pachacama

Figura 4.3: Pruebas dentro del Laboratorio de Mecatrónica



Pruebas dentro del Laboratorio de Mecatrónica con luminosidad variable, Anthony Pachacama

#### **4.2 Pruebas de ROI**

Se presentaron varios problemas al realizar la detección de regiones de interés o zonas con características relevantes con el script de captura de imágenes, para esto se aplicaron correcciones de redimensionamiento dentro del código de la figura 3.6, dependiendo del modelo de placa se aumentó o disminuyó las dimensiones.

En primera instancia se optó por realizar el cálculo del tamaño del objeto aplicando una escala proporcional, pero no fue eficiente debido a que no se lograba una correcta adquisición de medidas, por ende, se presentaban errores al detectar bordes externos del objeto que realizábamos el análisis. Por tal motivo se decidió utilizar el redimensionamiento solo para iniciar el proceso de detección de las placas y más no para adquirir otro tipo de característica.

El script presenta la desventaja de falsos positivos, esto debido a que el programa está configurado para que cada 3 segundos de forma automática capture las imágenes con las características de la placa, esto hace que el sistema adquiera la información de todo tipo sin descartar un objeto ajeno al que se desee adquirir sus características.

La cámara Web M014 que utilizamos en el sistema tiene la característica que se puede mejorar la resolución y calidad aplicando medidas de resolución exactas y adecuadas dentro del script, esta mejora se la quiso aplicar con la cámara nativa de la Raspberry Pi 5 sin lograr mejoras significativas y persistiendo el problema de lenta respuesta, además

de errores al momento de intentar leer el QR, debido a la baja resolución, dando como resultado que no logre decodificar la información contenida dentro.

Se establecieron características como la detección de bordes, clasificación de formas cerradas, identificación del rectángulo con mayor área. Estas características fueron modificadas para adquirir las características como su forma general, forma de sus sócalos internos y características geométricas de alta relevancia que sean importantes cuando el programa se lleve a la fase de entrenamiento.

#### **4.3 Pruebas con el entrenamiento**

Si bien el script de adquisición de imágenes me entrega toda la información con un mínimo rango de pérdida del 9%, entre estas pérdidas están los duplicados, corruptos y carpetas mal nombradas. Al ver esta problemática se decidió optar por la validación de rutas para el manejo de todos los datos que ingresaremos al entrenamiento.

Al finalizar el entrenamiento teníamos errores como la detección de placas electrónicas que no son de computadoras, esto se debió al bajo control de formatos y consistencia de las imágenes, lo que daba como resultado que no todas las imágenes se estaban siendo procesadas correctamente ya que tiene un distinto tamaño y colores, para esto se aplicó un redimensionamiento estándar en escala de grises.

Al ingresar imágenes dañadas o vacías, el sistema presento la falla que cualquier entorno o figura era una placa, para esto se aplicó un filtrado de datos por tamaño ya que los archivos dañados en su mayoría tienen un peso significativamente más pequeño al resto de imágenes adquiridas.

#### **4.4 Pruebas de manejo de información**

Para que la información llegue con orden general de todas las placas, se diseñó una estructura de datos. En donde se conceptualiza el tipo de placa, fabricantes, salida, entrada y retorno del equipo.

La integración entre Google Sheets y los QR con la información de trazabilidad, comienza con la inserción de las credenciales en el script de Python donde se generan los datos, con esto, cada vez que generemos nuevos datos permitirá que los datos se almacenen de forma automática dentro de la nube. Para esto configuramos mediante OAuth 2.0 y estructuras de columnas trazables.

#### **4.5 Pruebas de la nube**

La conexión del entero sistema realizado dentro de las Raspberry Pi 5 con la hoja de Google Sheets, mediante el uso de una API, presento varios problemas cuando se puso a prueba sus funcionamientos para él envió de los datos almacenados dentro de los QR, uno

de estos problemas se dio debido a una mala configuración de las credenciales que realiza la vinculación con los servicios de GOOGLE, debido a que durante la manipulación de los archivos se modificaron los parámetros

Se presentaron fallas durante las pruebas realizadas dentro de la Universidad Politécnica Salesiana Sede Quito, Campus Sur. A causa de la congestión y la infraestructura de seguridad que presentaba la red estudiantil, cabe recalcar que el sistema trabaja con normalidad con la red estudiantil, el problema radica en que el tiempo de subida de los datos es un poco lenta, por lo que baja su optima respuesta a comparación con una conexión a internet domestica que tienen menos restricciones haciendo que la subida de datos sea casi al instante, como logramos apreciar en las tabla4.1 y tabla 4.2, donde observamos los tiempos que tarda en llegar la información a la nube.

Tabla 4.1: Tiempo de subida de datos a la nube con la red de la UPS

Momento del día	Congestión	Red UPS
Mañana	Muy Alta	± 7s
Mediodía	Alta	± 4s
Tarde	Moderada	± 3s

Tiempo de procesamiento y subida información a la nube con la red UPS, Anthony Pachacama

Tabla 4.2: Tiempo de subida de datos a la nube con la red domestica

Momento del día	Congestión	Red Domestica
Mañana	Moderada	± 3s
Mediodía	Baja	± 2s
Tarde	Baja	±2s

Tiempo de procesamiento y subida información a la nube con la red Doméstica, Anthony Pachacama

## CONCLUSIONES

De los métodos de reciclajes indagados, este proyecto trabaja con un sistema híbrido entre una red neural convolucional y OpenCV, teniendo como resultado un sistema de reconocimiento fiable a comparación a todos los sistemas anteriormente expuestos. Además, posee un sistema que adquiere los datos de trazabilidad y los almacenados dentro una base de datos, logrando así tener un registro de históricos para la evaluación de los mismos. Todas estas características nos dan como resultado una amplia gama de posibilidades para aplicarlo en otras áreas a nivel institucional.

Una vez implementado el reconocimiento de placas electrónicas de computadora y el área de contención, se obtuvo como resultado un sistema automatizado con características de modularidad y escalabilidad, lo que permite implementarlo en entornos reales. El sistema fue puesto en funcionamiento en diferentes áreas del Bloque C como se puede apreciar en las figuras, dando como resultado un correcto desempeño de todos los subsistemas que compone el proyecto.

Como resultado, al momento de aplicar herramientas informáticas para el sistema, se estructuró una arquitectura en la nube que se destaca por su escalabilidad, utilizando los servicios de Google, lo que permite almacenar y consultar datos en tiempo real. Además, al utilizar una Interfaz de Programación de Aplicaciones que automatiza, integra y extiende las funcionalidades de Telegram al enviar notificaciones automáticas a un agente externo para que el sistema trabaje de manera constante.

Para establecer la comunicación entre el módulo contenedor inteligente y la nube, se aplicó un protocolo Hypertext Transfer Protocol Secure para la transmisión de datos la cual emplea una autenticación OAuth 2.0 que protege los recursos del usuario sin la necesidad de compartir credenciales directamente, dando como resultado un acceso granular y una forma eficiente, rápida y segura de manejar la información de trazabilidad obtenida a través de los QR de las placas de computadora.

## **RECOMENDACIONES**

La red universitaria al poseer muchas restricciones dentro de su infraestructura, además de un elevado tráfico de información que maneja la misma debido a la gran cantidad de estudiantes y personal colaborador de la institución, nos da como resultado una lenta respuesta al subir información a la nube. Por esto, se recomienda para futuros trabajos crear una conexión a internet libre, a la cual tengan acceso solo estudiantes que van a desarrollar proyectos donde se maneje una rápida subida y bajada de datos.

Actualmente la Raspberry Pi 5 es una de las placas de desarrollo más actuales y potentes que existe en el mercado, si bien es un equipo más robusto y eficiente a diferencia de sus antecesores, presento muchas fallas al no estar sus librerías totalmente actualizadas y optimizadas. Para lo cual, se recomienda trabajar con librerías más actuales o que no sean nativas de este sistema. Al aplicar esta solución eliminamos problemas como la lenta respuesta del sistema de control físico y problemas a nivel de la programación como la vinculación de la cámara al sistema de adquisición de imágenes y de visión artificial.

Dentro de los espacios en donde se hicieron las pruebas para medir la eficacia del sistema de reconocimiento de placas electrónicas, el principal problema que se presento era la variación de luz en cada uno de los entornos. Por esta razón, se recomienda para futuros proyectos desarrollar un entorno de luminosidad controlado que trabaje en función de los datos que obtenga la cámara, para aumentar o reducir la cantidad de luz presente en el ambiente, logrando que la cámara recopile toda la información relevante y tengamos un sistema con un nivel de eficiencia más alto.

## REFERENCIAS BIBLIOGRÁFICAS

- Alvear-Puertas, V., Rosero-Montalvo, P., Peluffo-Ordóñez, D., & Pijal-Rojas, J. (2017). Internet de las cosas y visión artificial, funcionamiento y aplicaciones: Revisión de literatura. *Enfoque UTE*, 8(1), 244–256. <https://doi.org/10.29019/enfoqueute.v8n1.121>
- Andrade Fierro, L. G., & Chulca Simbaña, L. A. (2018). Desarrollo de un sistema de visión artificial para el reconocimiento, clasificación y maquinado de patrones con una tarjeta ARM (Tesis de grado). Universidad Politécnica Salesiana. Recuperado de <https://dspace.ups.edu.ec/handle/123456789/15284>
- AV Electronics. (s.f.). Sharp GP2Y0A21YK0F Analog Distance Sensor 10-80cm. AV Electronics. Recuperado el 18 de julio de 2025, de <https://avelectronics.cc/producto/sharp-gp2y0a21yk0f-analog-distance-sensor-10-80cm/>
- Bejarano Sotomonte, J. C., Chaux Colorado, H. F., & Corrales Torres, C. T. (2022). Visión artificial para las redes neuronales
- Caballero Perdomo, A. L. (2024). Diseño e implementación de políticas de asignación de pedidos en plataformas digitales buscando el bienestar de todos los actores.
- Fernandez, M., Gil, Y., Moriel, A., & Recio, J. (2009). La Luz. Links of Science for Kids. Recuperado de [http://www.quimicaweb.net/grupo\\_trabajo\\_ccnn\\_2/tema5/index.htm](http://www.quimicaweb.net/grupo_trabajo_ccnn_2/tema5/index.htm)
- Horna, W. A. N., & Manzano, E. A. (2021). Sistema de visión artificial basado en redes neuronales convolucionales para la selección de arándanos según estándares de exportación. *Revista Campus*, 26(32).
- León-Duarte, J. A., De La Re-Iñiguez, B. M., & Romero-Dessens, L. F. (2020). Ventajas del uso de sistemas de trazabilidad electrónica en procesos de manufactura. *Información Tecnológica*, 31(1), 237–244. <https://doi.org/10.4067/S071807642020000100237>

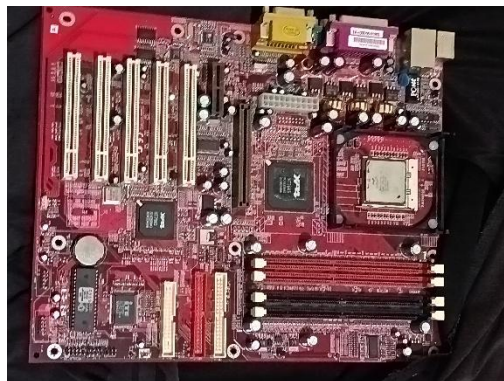
- Murazzo, M. A., Millán, I. F., Rodríguez, N. R., Segura, D., & Villafañe, D. A. (2010). Desarrollo de aplicaciones para Cloud Computing. In XVI Congreso Argentino de Ciencias de la Computación.
- Obando Suárez, J. P. (2022, 28 de octubre). Sistema de recolección de desechos optimizado, mediante IoT en el sector El Olivo de la ciudad de Ibarra, provincia de Imbabura (Tesis de pregrado en Ingeniería en Electrónica y Redes de Comunicación). Universidad Técnica del Norte. Recuperado de <https://repositorio.utn.edu.ec/handle/123456789/13118>
- Proaño Avilés, J. S., Cali Vera, C. D., Molina López, I. S., Pástor del Salto, M., & Sandoval Ocampo, N. B. (2020). *Contenedor automatizado para reciclaje de residuos de aparatos eléctricos y electrónicos (RAEE)* (Tesis de ingeniería). Universidad San Francisco de Quito. Recuperado de <https://repositorio.usfq.edu.ec/handle/23000/9258>
- Ramírez Prado, M. J., Loor Castro, A. J., & Cabello Sánchez, J. G. (2023). Sistema inteligente de gestión de residuos mediante Internet de las cosas (Proyecto integrador). Escuela Superior Politécnica del Litoral (ESPOL). Recuperado de <https://dspace.espol.edu.ec/handle/123456789/58243>
- Sánchez Silva, M., Bonales Valencia, J., & Espinoza Torres, R. (2020). Contaminación del medio ambiente en la región oriente del estado de Michoacán por desechos electrónicos de equipo de cómputo obsoleto.
- Satuquinga Congacha, C. I. (2021). Sistema inteligente de reciclaje Open Bot (Tesis de pregrado). Universidad Técnica de Ambato, Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Recuperado de <https://repositorio.uta.edu.ec/handle/123456789/32306>
- Ser, C. Y., & Vicios, A. (2006). *Visión artificial: Análisis teórico del tratamiento digital de imágenes para su aplicación en la identificación de objetos*. Universidad Austral de Chile.
- Toaquiza, A., & Paredes, J. (2020). Sistema de trazabilidad electrónica para mejorar la calidad en la fabricación de arneses automotrices. *Revista Ingeniare*, 28(1), 237–

248. [https://www.scielo.cl/scielo.php?pid=S0718-07642020000100237&script=sci\\_arttext](https://www.scielo.cl/scielo.php?pid=S0718-07642020000100237&script=sci_arttext)

Zambrano Oñate, M. B. (2023). Sistema de detección y alerta de residuos sólidos mediante Inteligencia Artificial [Tesis de pregrado, Universidad Central del Ecuador]. Repositorio UCE. <https://www.dspace.uce.edu.ec/handle/25000/32546>

## ANEXOS

### ANEXO 1: Modelos de placas más comunes



## ANEXO 2: Sistema finalizado e implementado



## ANEXO 3: Diseño en software CAD

