

UNIVERSIDAD TECNOLÓGICA
DE PANAMÁ

FACULTAD DE INGENIERÍA DE
SISTEMAS COMPUTACIONALES

CENTRO REGIONAL DE VERAGUAS

PROTOTIPO DE CUBO DE BASURA
INTELIGENTE CON DETECCIÓN DE
MATERIAL Y MONITOREO DE
LLENADO BASADO EN ARDUINO Y
ESP32 PARA LA UNIVERSIDAD
TECNOLÓGICA DE PANAMÁ

PRESENTA:

BARRERA, ARLAND

ORTEGA, PRISCILA

PUGA, ELBIN

TUTOR

Dr. Cristian Pinzón Trejos

2024



1	Capítulo 1: Generalidades del Proyecto	6
1.1	Antecedentes	6
1.1.1	Sistema de clasificación inteligente de residuos	6
1.1.2	Sistema de gestión de residuos inteligente basado en IoT	6
1.1.3	Sistema de contenedores de basura basado en IoT que utiliza un microcontrolador NodeMCUu ESP32	7
1.2	Identificación del problema	7
1.3	Objetivo general	8
1.4	Objetivos específicos	8
1.5	Justificación	8
1.6	Delimitaciones	9
2	Capítulo 2: Marco Teórico	10
2.1	Área de Investigación	10
2.1.1	Inteligencia artificial en el reciclaje	10
2.1.2	Sostenibilidad	10
2.1.3	Microcontroladores	11
2.1.4	Uso de Arduino y ESP32	12
2.1.4.1	Pines analógicos	12
2.1.4.2	Pines digitales	12
2.1.5	Sistemas de clasificación automática	14
2.1.6	Aplicaciones de IoT	15
2.1.7	Campo de aplicación	17
2.1.7.1	Entornos Urbanos	17
2.1.7.2	Espacios Comerciales	18
2.1.7.3	Industria Hotelera	18
2.1.7.4	Instalaciones Sanitarias	19
2.1.7.5	Instituciones Educativas	19
2.1.8	Futuro del tema de investigación	19
2.2	Proyectos similares	19
2.2.1	Diseño de un Contenedor Municipal Automático	19
2.2.2	Sistema de Clasificación Inteligente de Residuos	20
2.2.3	Contenedor de Reciclaje Automatizado de Bajo Costo	20
2.3	Tecnologías De Desarrollo	21
2.3.1	Arduino UNO R3	21
2.3.2	IDE Arduino	21

2.3.3	Micropython	22
2.3.4	CNN	23
2.3.5	Visión Artificial	23
2.3.6	Open CV	24
2.3.6.1	Numpy	24
2.3.7	XIAO ESP32 CAM	25
2.3.8	Sensor ultrasónico (HC-SR04)	26
2.3.9	Cables jumper, resistencias, protoboard	27
2.3.10	Servomotor (SG90)	27
3	Capítulo 3: Propuesta del Proyecto	30
3.1	Metodo-Experimentación	30
3.1.1	Tipo y alcance de la investigación	30
3.1.2	Observación	30
3.1.3	Planteamiento de Hipótesis	31
3.1.4	Experimentación	31
3.1.4.1	Desarrollo del Prototipo	31
3.1.4.2	Integración Electrónica:	31
3.1.4.3	Programación del Sistema	32
3.1.4.4	Interacción del Usuario	33
3.1.5	Análisis de Datos	33
3.1.6	Análisis y conclusiones	34
3.2	Escenario De Aplicación	34
3.3	Cronograma de Actividades	35
3.4	Presupuesto Real	38
3.5	Diseño conceptual	39
3.5.1	Descripción del diseño conceptual planteado	41
	Bibliografía	46

Lista de figuras

2.1	Elementos de un microcontrolador	12
2.2	Arduino UNO R3	13
2.3	ESP32	13
2.4	XIAO ESP32 CAM	14
2.5	Margen m maximizado	15
2.6	Sistema de monitoreo de acuicultura - Tecnologia Humanizada . . .	16
2.7	IoT - contenedor de basura inteligente	17
2.8	Basurero inteligente	18
2.9	ARDUINO UNO R3	21
2.10	IDE Arduino	22
2.11	Interfaz de MicroPython	22
2.12	Arquitectura de una red CNN	23
2.13	Logo de OpenCV	24
2.14	Logo de Numpy	24
2.15	XIAO ESP32 CAM	25
2.16	Sensor ultrasónico HC-SR04	27
2.17	Jumpers	28
2.18	Resistores	28
2.19	ProtoBoard	28
2.20	Mini Sevo motor	29
3.1	ESP32 CAM y Machine Learning	42

Lista de tablas

2.1	DataSheet de la XIAO ESP32 CAM	26
2.2	Especificaciones del HC-SR04	27
2.3	Especificaciones del SG90	29
3.1	Cronograma de Actividades	35
3.2	Presupuesto Real del Proyecto	38

Capítulo 1: Generalidades del Proyecto

1.1 Antecedentes

Revisando en la literatura trabajos similares que se han realizado antes:

1.1.1 Sistema de clasificación inteligente de residuos

"INTELLIGENT WASTE SORTING SYSTEM: LEVERAGING ARDUINO FOR AUTOMATED TRASH IDENTIFICATION AND CATEGORIZATION". El artículo presenta un sistema inteligente para la clasificación automática de residuos, utilizando modelos de detección de objetos basados en YOLOv5x y tecnología Arduino. La propuesta surge debido al aumento de residuos como consecuencia de la urbanización, lo que genera un desafío para el medio ambiente y la gestión de desechos. Se busca optimizar el proceso de detección y clasificación de residuos con el fin de mejorar su reutilización y reciclaje.

Para lograrlo, se entrena un modelo YOLOv5x utilizando un conjunto de datos personalizados que contiene imágenes de siete tipos de residuos, como plástico, vidrio, papel y metales. El sistema utiliza un microcontrolador Arduino para interactuar con el entorno físico, permitiendo que el sistema realice acciones como la clasificación automática de residuos. Los resultados muestran que la combinación de YOLO y Arduino ofrece una solución eficiente para la clasificación de basura, con el potencial de ser implementada en diversos contextos, desde espacios públicos hasta plantas de tratamiento de residuos. [1]

1.1.2 Sistema de gestión de residuos inteligente basado en IoT

"An Internet of Things based Smart Waste System". Presenta un sistema de gestión de residuos basado en la tecnología de Internet de las Cosas (IoT) utilizando el microcontrolador ESP-32 Wi-Fi. El objetivo del sistema es evitar la acumulación de basura en las calles, reducir la carga laboral de los recolectores

y automatizar el proceso de vaciado de los contenedores. Los contenedores inteligentes incorporan sensores ultrasónicos para medir el nivel de residuos, sensores DHT-22 para monitorear la humedad y temperatura, y motores servo para abrir y vaciar los contenedores automáticamente. Los datos se muestran en una pantalla LCD y se transmiten a una aplicación móvil a través de IoT, permitiendo la supervisión remota del sistema. El sistema es eficiente y económico, lo que lo convierte en una solución viable para mantener las ciudades limpias y libres de contaminación. [2]

1.1.3 Sistema de contenedores de basura basado en IoT que utiliza un microcontrolador NodeMCUu ESP32

"IOT-BASED GARBAGE CONTAINER SYSTEM USING NODEMCU ESP32 MICROCONTROLLER". Presenta un sistema de contenedores de basura inteligentes basado en la tecnología de Internet de las Cosas (IoT) y el microcontrolador NodeMCU ESP32. Este sistema automatiza la apertura y cierre de la tapa de los contenedores de basura, y permite detectar cuando están llenos. Los usuarios pueden monitorear en tiempo real el nivel de llenado a través de una página web y recibir notificaciones mediante Telegram. Los resultados de las pruebas indican que el sistema mejora la comodidad y cumplimiento de los usuarios al disponer de basura, y facilita la gestión de residuos al informar a los trabajadores cuándo vaciar los contenedores. El uso de IoT para este fin optimiza el manejo de los residuos, evita desbordes y mantiene el entorno limpio. [3]

1.2 Identificación del problema

De acuerdo con el Programa de Naciones Unidas para el Desarrollo (PNUD) en su documento “Un país pequeño y un reto enorme: La gestión integral de residuos sólidos en Panamá”, en el país con 4,500,000 millones de habitantes cada persona genera 1.2 kilogramos de residuos sólidos por persona al día de los cuales solo el 65 por ciento termina en sitios de disposición final. El resto va a vertederos ilegales, ríos, quebradas, el mar y otros sitios no destinados a la disposición de desechos. [4]

Uno de los principales problemas radica en que los materiales reciclables a menudo no se separan adecuadamente desde el origen, dificultando su reciclaje y aumentando la contaminación. Los sistemas tradicionales no detectan cuando los contenedores están llenos, lo que puede causar desbordamientos y acumulación de desechos.

Esta propuesta del prototipo consiste en un sistema autónomo basado en Arduino

y ESP32 para detectar y clasificar materiales de desecho en un cubo de basura inteligente. El cubo está equipado con un módulo de cámara ESP32-CAM para identificar el tipo de material, y un sensor de que monitorea el nivel de llenado.

El sistema procesará los datos en tiempo real mediante la placa Arduino, determinando el tipo de basura a depositar y el estado de llenado del cubo. Cuando se detecte que el cubo está lleno o un material específico, se enviarán notificaciones a través de la ESP32

1.3 Objetivo general

Desarrollar un sistema de contenedor de basura inteligente que utilice tecnologías basadas en Arduino y ESP32 para detectar automáticamente el tipo de material desechado y monitorear el nivel de llenado en el Centro Regional de Veraguas.

1.4 Objetivos específicos

- **Investigar y revisar literatura científica rigurosa:** buscar información en artículos, revistas y sitios web de rigurosidad científica para desarrollar el proyecto.
- **Diseñar el modelo conceptual:** diseñar el flujo de comportamiento del sistema.
- **Desarrollar el prototipo con los materiales y herramientas propuestos:** tras detectar en la literatura los materiales necesarios para desarrollar de manera física el diseño conceptual, se procede a crear un prototipo.
- **Presentar los resultados:** dar a conocer los resultados del proyecto.

1.5 Justificación

Actualmente, una gran cantidad de residuos reciclables se pierde debido a la falta de segregación en la fuente, y los sistemas tradicionales de recolección de basura no siempre son eficientes lo que genera desbordamientos y acumulación de desechos en la UTP.

Este proyecto de un cubo de basura inteligente, basado en Arduino y ESP32, está diseñado para abordar estos problemas al proporcionar una solución automatizada y conectada. La capacidad de detectar y clasificar diferentes tipos de materiales

permite una mejor separación de residuos, lo que facilita el reciclaje y reduce la contaminación.

Esto ayuda a reducir el costo manual que conlleva encargarse de los residuos. Además, el monitoreo en tiempo real del nivel de llenado garantiza que los contenedores sean vaciados oportunamente, evitando desbordamientos y mejorando la eficiencia en la recolección de basura.

1.6 Delimitaciones

El objetivo del proyecto es desarrollar un sistema de clasificación de residuos sólidos reciclables, enfocado específicamente en tres tipos de materiales: papel, cartón y plástico. Si bien esta selección permite abordar una parte importante de los residuos generados, limita la aplicabilidad del sistema a otros tipos de desechos, como metales, vidrios u orgánicos, restringiendo su capacidad para ser utilizado en escenarios de mayor diversidad de residuos.

La implementación tecnológica del sistema se sustenta en la integración de componentes de hardware, como ESP32-CAM y Arduino, lo que ofrece ventajas en términos de programación y control. Sin embargo, esta elección puede plantear desafíos en cuanto al costo, especialmente si se busca replicar el proyecto a gran escala. Además, el éxito del sistema está condicionado por las características del entorno donde se instale, lo que puede limitar su utilización en áreas con restricciones presupuestarias o de infraestructura tecnológica.

Se plantea un enfoque particular en el Centro Regional de Veraguas de la Universidad Tecnológica de Panamá (UTP), lo que implica que su aplicación se encuentra restringida a un entorno académico específico. Si bien este espacio ofrece condiciones controladas para la prueba y validación del sistema, la escala de implementación es limitada, lo que sugiere la necesidad de futuras adaptaciones para extender su uso a otras instituciones o contextos urbanos más amplios.

Capítulo 2: Marco Teórico

2.1 Área de Investigación

2.1.1 Inteligencia artificial en el reciclaje

Otra tecnología emergente con la que se está experimentando en el campo del reciclaje es la utilización de inteligencia artificial o IA. Los algoritmos de la IA pueden analizar grandes cantidades de datos y aprender de ellos para mejorar la eficiencia del proceso de reciclaje. Por ejemplo, se pueden utilizar sistemas de aprendizaje automático para identificar materiales específicos en los residuos y separarlos con mayor precisión.

Además, la IA también podría ayudar a predecir cuánto material se puede reciclar de un lote específico de residuos, lo que permitiría una mejor planificación y gestión del proceso de reciclaje. Aunque esta tecnología todavía está en una etapa temprana de implementación [5].

2.1.2 Sostenibilidad

Cada vez más, la tecnología cobra un papel más importante a la hora de dar respuesta a los desafíos a nivel medioambiental, social y económico a los que nos enfrentamos actualmente. Los avances tecnológicos han permitido avanzar hacia un desarrollo global sostenible.

Las tecnologías sostenibles se conocen como aquellas que están enfocadas en los principios de sostenibilidad. Son aquellas que a través de la reutilización, el reciclaje, la conservación de recursos naturales y de la eficiencia energética, reducen la contaminación. Por lo tanto, minimizan el impacto ambiental.

Ciertamente, las tecnologías sostenibles tienen muy presentes las diferentes necesidades de la sociedad. Además, estas tecnologías implicadas con el desarrollo sostenible, emplean menos energía para realizar los procesos, emplean una cantidad menor de recursos limitados. Por consecuencia, reducen la utilización de los recursos naturales en todas sus etapas (creación, puesta en marcha o utilización) [6].

2.1.3 Microcontroladores

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador. Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

El microcontrolador es, en definitiva, un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso, el controlador recibe el nombre de controlador empotrado (embedded controller) [7].

El microcontrolador nace cuando las técnicas de integración han progresado lo bastante para permitir su fabricación; pero también porque, muy a menudo, tanto en las aplicaciones domésticas como industriales, se tiene la necesidad de sistemas “inteligentes” o, al menos, programables. Un ejemplo muy simple es el programador de una lavadora, el cual debe controlar una cierta cantidad de elementos con ciclos y cadencias perfectamente definidas, pero variables en función del programa seleccionado. Otras aplicaciones más técnicas tienen, igualmente, necesidad de sistemas programables. Por ejemplo, una fotocopidora debe controlar permanentemente un gran número de elementos y de funciones. Gracias a la llegada de los microcontroladores, tarjetas que contenían varias decenas de circuitos lógicos clásicos se han visto reducidas a dos o tres microcontroladores.

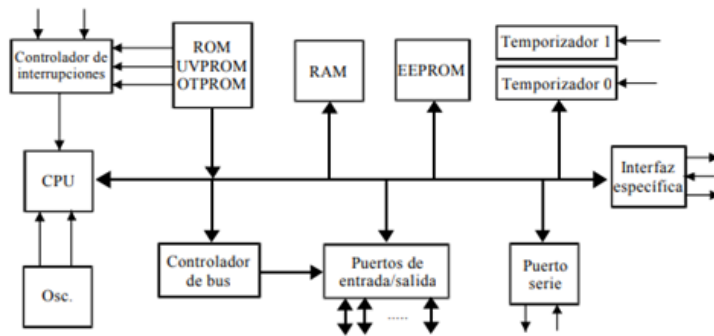


Figura 2.1: Elementos de un microcontrolador
Fuente: Adaptado de [7]

2.1.4 Uso de Arduino y ESP32

Arduino es una placa de desarrollo de hardware libre, que puede ser utilizada tanto por aficionados como por fabricantes para diseñar y construir dispositivos que interactúen con el mundo real, a través de una gran cantidad de sensores y otros elementos electrónicos que están disponibles en el mercado. Aunque utilizamos el término Arduino para referirnos a un tipo específico de placa de desarrollo, también se usa para hablar de la empresa que fabrica estas placas o para describir a la comunidad en torno a las diferentes placas compatibles [8].

Clasificación Dse Pines

2.1.4.1 Pines analógicos

Los pines analógicos pueden leer un amplio rango de valores, por lo que son útiles para realizar un control más detallado de las lecturas obtenidas. En general se encuentran 6 pines y mediante estos pines es posible obtener datos de sensores en forma de variaciones continuas de un voltaje.

2.1.4.2 Pines digitales

Los pines digitales son capaces de leer y escribir un solo estado, es decir, encendido o apagado. En la mayoría de las placas Arduino hay 14 pines [8].

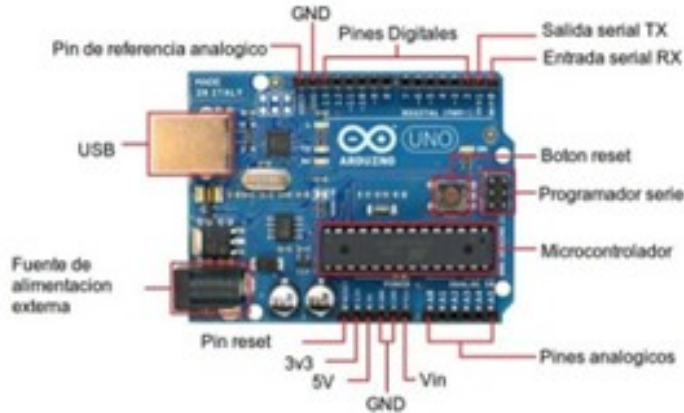


Figura 2.2: Arduino UNO R3
Fuente: Adaptado de [9]

ESP32

Plataforma de hardware abierto, basada en un sistema en chip (SoC, del inglés System on a Chip) con hardware específico para comunicaciones inalámbricas, entre otros recursos. Este permite el desarrollo de aplicaciones en diferentes lenguajes de programación, frameworks, bibliotecas y recursos diversos.

El SoC ESP32 de Espressif Systems es la evolución del ESP8266, diseñado para superar a su antecesor en capacidad de procesamiento y conectividad; integra un potente microcontrolador con arquitectura de 32 bits, conectividad wifi y Bluetooth. El sistema en el módulo (SoM, del inglés System on Module) ESP-32S fabricado por Ai-Thinker integra en un módulo el SoC ESP32, memoria FLASH, cristal oscilador y antena wifi en PCB.



Figura 2.3: ESP32
Fuente: Adaptado de [10]

Se destaca el módulo ESP32-CAM, que incluye una cámara OV2640 y varios GPIO para conectar periféricos usando un ESP32. El módulo también cuenta

con una ranura para tarjeta microSD, que permite almacenar imágenes tomadas de la cámara o almacenar archivos y viene con el módulo de cámara de 2MP. Las funcionalidades de este módulo y su capacidad de realizar diversas tareas lo han convertido en un componente eficaz para la seguridad. Este módulo se puede utilizar en videovigilancia para transmitir imágenes o funcionar como sensores en un sistema de visión [11].

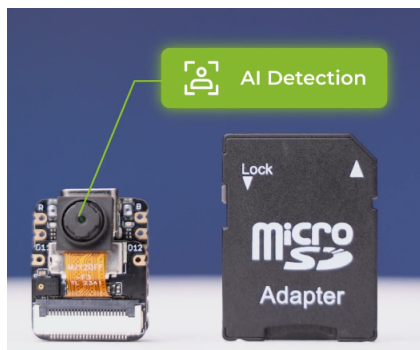


Figura 2.4: XIAO ESP32 CAM
Fuente: Adaptado de [12]

2.1.5 Sistemas de clasificación automática

Machine Learning es un área de la inteligencia artificial que engloba un conjunto de técnicas que hacen posible el aprendizaje automático a través del entrenamiento con grandes volúmenes de datos. Hoy en día, existen diferentes modelos que utilizan esta técnica y consiguen una precisión incluso superior a la de los humanos en las mismas tareas, por ejemplo, en el reconocimiento de objetos en una imagen. La construcción de modelos de Machine Learning requiere adaptaciones propias debido a la naturaleza de los datos o a la problemática a la que se aplica. Así, surge la necesidad de investigar las diferentes técnicas que permitan obtener resultados precisos y confiables en un tiempo razonable [13].

Dentro de esta clasificación, podemos además encontrar un gran número de algoritmos específicos con diferentes características para el tratamiento de los datos. Entre los más relevantes encontramos:

- **Deep Learning:** consiste en la utilización de algoritmos para hacer representaciones abstractas de la información y facilitar el aprendizaje automático.
- **Active Learning:** es un caso especial de aprendizaje semi-supervisado donde el algoritmo de aprendizaje puede interactuar con un usuario u otra fuente de

información para obtener los resultados deseados.

- **Support Vector Machines:** busca la maximización de la distancia entre la recta o el plano y las muestras que se encuentran a un lado u otro. En el caso de que las muestras no sean linealmente separables, se utiliza una transformación llamada kernel.

La teoría de las Máquinas de Soporte Vectorial (SVM, por su nombre en inglés, Support Vector Machines) es una nueva técnica de clasificación y ha tomado mucha atención en años recientes. La teoría de la SVM está basada en la idea de minimización de riesgo estructural (SRM). En muchas aplicaciones, las SVM han mostrado tener un gran desempeño más que las máquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación.

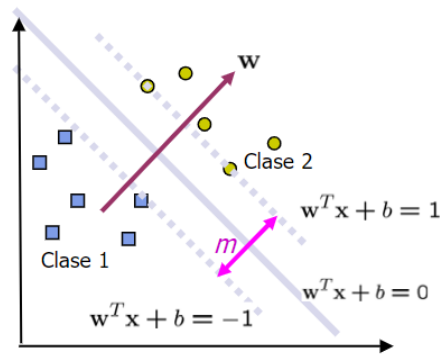


Figura 2.5: Margen m maximizado
Fuente: Adaptado de [14]

Una Máquina de Soporte Vectorial (SVM) aprende la superficie de decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera [14].

2.1.6 Aplicaciones de IoT

Comunicación y monitoreo

El IoT tiene una amplia gama de aplicaciones en diversos sectores como se muestra en la ilustración No. 1. En el ámbito de la salud, el IoT se utiliza para monitorear pacientes y proporcionar atención médica remota. En el transporte,

el IoT permite la gestión de flotas de vehículos y el seguimiento en tiempo real de la ubicación de mercancías. En la agricultura, el IoT se aplica para el monitoreo y control de cultivos, optimizando el riego y la fertilización. Además, el IoT ha encontrado aplicaciones en ciudades inteligentes, hogares inteligentes, industria manufacturera, entre otros campos.

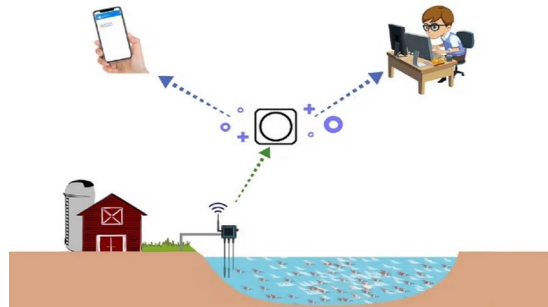


Figura 2.6: Sistema de monitoreo de acuicultura - Tecnología Humanizada
Fuente: Adaptado de [15]

Sin embargo, la implementación exitosa de IoT en la recolección de residuos sólidos requiere de un enfoque integral que abarque aspectos técnicos, operativos y de privacidad de los datos. En la literatura científica existen algunos trabajos con propósitos similares que utilizan metodologías, arquitecturas y equipos diferentes, a continuación, se detallan.

Tarandeep Singh, en su trabajo denominado *IOT based Smart Waste Management Framework (Singh)*, proporciona una solución para mejorar la gestión de residuos sólidos mediante la monitorización remota de los parámetros de los contenedores de basura y la optimización de las rutas de recolección. La metodología utiliza la tecnología X-bee y GSM para monitorear de forma remota los parámetros de los contenedores, como el llenado del contenedor, la temperatura, la humedad y el volumen de CO₂ dentro del contenedor. Los datos recopilados se transmiten a una estación base que los envía a una estación de control para su análisis y optimización de rutas de recolección de basura. Las conclusiones del estudio sugieren que el uso de tecnología IoT en la gestión de residuos sólidos puede ser una solución efectiva y rentable para mejorar la eficiencia de la recolección de basura y reducir los costos operativos y las emisiones de gases de efecto invernadero.

Mientras tanto, Raúl Hernández y su equipo analizan la gestión y el proceso de recolección de residuos sólidos urbanos (RSU) en el municipio de Nezahualcóyotl, Estado de México, donde identifican las principales características de la gestión y recolección de RSU en el municipio y aplican la idea de la Ciudad Inteligente (CI)

para mejorar el servicio de recolección de basura [16].

La aplicación del Internet de las Cosas (IoT) en sistemas de clasificación de basura es una tendencia innovadora para mejorar la gestión de residuos.

Sistemas de clasificación automática basados en cámaras y sensores ópticos conectados a la red IoT pueden identificar diferentes tipos de residuos (plásticos, metales, vidrio, etc.). Esto mejora la eficiencia del reciclaje al separar automáticamente los residuos según su tipo.



Figura 2.7: IoT - contenedor de basura inteligente
Fuente: Adaptado de [17]

Las plataformas IoT permiten a los gestores de residuos monitorear remotamente el estado de los contenedores y otros equipos en tiempo real, lo que mejora la eficiencia operativa y reduce el tiempo de inactividad.

2.1.7 Campo de aplicación

Contenedores de Basura Inteligentes

Los contenedores de basura inteligentes encuentran aplicaciones en varios sectores:

2.1.7.1 Entornos Urbanos

En ciudades y municipios, los contenedores de basura inteligentes están ayudando a gestionar grandes volúmenes de residuos de manera eficiente. Son particularmente útiles en áreas de mucho tráfico, ya que reducen la carga de trabajo de los equipos de recolección de residuos [18].

El Internet de las Cosas está teniendo un impacto tanto en la industria como en

la vida cotidiana de las personas. Actualmente existen contenedores de basura inteligentes, que han funcionado exitosamente en ciudades como Sevilla. A través del proyecto europeo LIFE EWAS, se ha implementado una solución de TICS que permite monitorizar el volumen de llenado de los contenedores. Se implementan sensores con sistemas de medidas volumétricas basados en parámetros como la distancia, es decir, el trecho entre el nivel de residuos y el sensor. Esta información permite conocer la capacidad de un contenedor. Al dotar de inteligencia al contenedor, se puede entregar información a un gestor de servicios, optimizando así la recolección de los residuos [19].



Figura 2.8: Basurero inteligente
Fuente: Adaptado de [20]

2.1.7.2 Espacios Comerciales

En entornos comerciales, como centros comerciales y aeropuertos, los contenedores inteligentes mejoran la gestión de residuos, asegurando que se vacíen rápidamente y reduciendo las molestias del desbordamiento de basura [18].

2.1.7.3 Industria Hotelera

Los hoteles y restaurantes están utilizando contenedores inteligentes para mantener la limpieza y la higiene, brindando una mejor experiencia a los huéspedes [18]. Los botes de basura inteligentes pueden mejorar la experiencia del cliente proporcionando una gestión automática y sin contacto, aumentando la comodidad y reduciendo la necesidad de mantenimiento frecuente. Pueden notificar cuando están llenos, asegurando que el personal de limpieza los vacíe de manera oportuna, lo cual es una buena manera de asegurar que los clientes no se disgusten y se sientan bien atendidos.

2.1.7.4 Instalaciones Sanitarias

En entornos sanitarios, donde la higiene es primordial, los contenedores de basura inteligentes desempeñan un papel crucial en la gestión eficiente de residuos y el control de infecciones [18].

2.1.7.5 Instituciones Educativas

Las escuelas y universidades están utilizando contenedores inteligentes para enseñar a los estudiantes sobre la eliminación responsable de residuos y, al mismo tiempo, mantener limpios sus campus. Este es uno de los campos de aplicación que motivan este proyecto, ya que son situaciones comunes en las instituciones educativas en el día a día.

2.1.8 Futuro del tema de investigación

Acceso biométrico: Se pueden incorporar funciones biométricas como el reconocimiento de huellas dactilares en los contenedores inteligentes para evitar el uso no autorizado y mejorar la seguridad.

Conectividad mejorada: Los contenedores inteligentes pueden convertirse en parte de redes más grandes de ciudades inteligentes, permitiendo la comunicación en tiempo real con otros sistemas municipales, como la gestión del tráfico.

2.2 Proyectos similares

Existen proyectos similares que se han realizado anteriormente:

2.2.1 Diseño de un Contenedor Municipal Automático

“DISEÑO DE UN CONTENEDOR MUNICIPAL AUTOMÁTICO MEDIANTE RED DE SENSORES CON MONITOREO INALÁMBRICO PARA LA MEJORA DEL PROCESO DE SELECCIÓN DE RESIDUOS SÓLIDOS, CALLAO 2022”. Este trabajo de investigación tiene como fin plantear una solución a una realidad que afecta a la sociedad actual, específicamente en el entorno urbano, donde la población citadina crece año tras año debido al centralismo. Este crecimiento poblacional conlleva un aumento en la producción de residuos de diversas categorías, entre ellos, los residuos sólidos.

Como objetivo general, tiene diseñar un contenedor municipal automático mediante red de sensores y monitoreo inalámbrico, capaz de mejorar el proceso de selección de residuos sólidos. El sistema utiliza sensores y un microcontrolador

para distinguir entre diferentes tipos de desechos, como plástico, vidrio, cartón y papel. El resultado final de este proyecto es un prototipo de un contenedor de residuos sólidos inteligente, que tiene instalado un módulo GPS para controlar la ubicación y un sensor ultrasónico que mide el nivel de llenado. En la Figura 90 se muestra el diseño del dispositivo que será ubicado dentro de un contenedor de residuos sólidos. Este dispositivo se compone de un sensor ultrasónico, un módulo GPS, un módulo ESP32 con su batería recargable, un transceptor LoRa RFM95 con una antena helicoidal y un panel solar para cargar las baterías de litio [21].

2.2.2 Sistema de Clasificación Inteligente de Residuos

“INTELLIGENT WASTE SORTING SYSTEM: LEVERAGING ARDUINO FOR AUTOMATED TRASH IDENTIFICATION AND CATEGORIZATION”. En este trabajo de investigación, se utiliza un conjunto de datos personalizado, una colección completa de imágenes de categorías de desechos divididas en siete clases, con modelos de detección de objetos como YOLO para automatizar la detección de basura y la clasificación precisa, mejorando así la precisión en la clasificación de desechos. El objetivo principal es aprovechar los atributos distintivos de los modelos YOLO combinados con Arduino para desarrollar un sistema de clasificación de basura más eficaz. Los procesos tradicionales de gestión de basura son laboriosos, lo que los hace lentos, costosos y propensos a errores humanos. Para superar estas deficiencias, se propone un modelo basado en el aprendizaje profundo que pueda detectar y clasificar rápidamente varios tipos de desechos en tiempo real. YOLO, un algoritmo de detección de objetos de última generación, sirve como columna vertebral de este modelo, asegurando una identificación rápida y precisa de los desechos. La integración de un microcontrolador Arduino agrega una dimensión práctica e interactiva, facilitando la comunicación entre el modelo de aprendizaje profundo y el mundo físico, permitiendo que el sistema active acciones como la clasificación, el reciclaje o la alerta a las autoridades [22].

2.2.3 Contenedor de Reciclaje Automatizado de Bajo Costo

“A LOW-COST AUTOMATED SORTING RECYCLE BIN POWERED BY ARDUINO MICROCONTROLLER”. Este artículo presenta el desarrollo de un contenedor de reciclaje de bajo costo que clasifica automáticamente diferentes tipos de desechos reciclables utilizando un microcontrolador Arduino. Los objetivos de este trabajo son construir un prototipo de contenedor de reciclaje con un mecanismo de detección que pueda clasificar los desechos reciclables (como metal, papel y plástico) y asignar automáticamente los desechos a particiones específicas del contenedor según sus tipos. Este artículo también presenta el análisis de sensibilidad

del prototipo de contenedor de reciclaje para clasificar los desechos correctamente. La construcción del prototipo se divide en dos partes: i) detección y ii) mecánica. La parte de detección identifica el tipo de material de desecho, utilizando un sensor de proximidad inductivo para detectar metales y un diodo emisor de luz (LED) junto con una resistencia dependiente de la luz (LDR) para papel y plástico. La parte mecánica incorpora un servomotor junto con el microcontrolador para clasificar el tipo de desecho. Los resultados muestran que el prototipo es capaz de clasificar los residuos con éxito, especialmente los plásticos, aunque se necesita mejorar la sensibilidad a los residuos de papel y metal para lograr una segregación eficaz [23].

2.3 Tecnologías De Desarrollo

2.3.1 Arduino UNO R3

Esta tarjeta se encarga de recibir y almacenar el orden de cada señal entregada [24].

Es el microcontrolador principal que controla todas las funciones básicas del sistema, como leer datos de los sensores (nivel de llenado, detección de movimiento) y activar los actuadores (abrir/cerrar la tapa del cubo). También es el que recibe las señales de los sensores y ejecuta las acciones correspondientes, como encender un buzzer o enviar alertas.



Figura 2.9: ARDUINO UNO R3

Fuente: Adaptado de [9]

2.3.2 IDE Arduino

El entorno de programación de la placa de Arduino se denomina Integrated Development Environment (IDE) el cual permite llevar a cabo la escritura de las sentencias para el funcionamiento de los elementos físicos de la placa de Arduino. Este software tiene por sí solo un conjunto de herramientas que permite, editar el código, compilar y depurar todo a través de una interfaz gráfica, así mismo, nos

da la oportunidad de interactuar con el microcontrolador almacenando los programas realizados en su memoria interna para poner en marcha todo el hardware. El lenguaje que utiliza el IDE de Arduino está basado en C/C++ de una manera más simplificada, ofreciendo la oportunidad de cargar librerías que sean necesarias para el buen funcionamiento de nuestros proyectos [25].

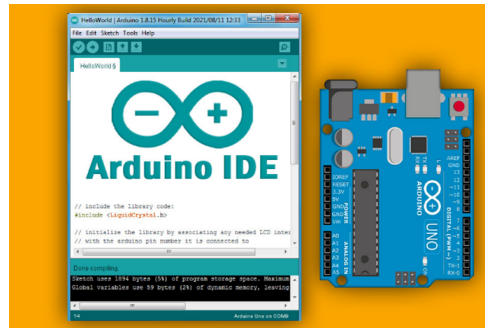


Figura 2.10: IDE Arduino
Fuente: Adaptado de [25]

2.3.3 Micropython

MicroPython es una reimplementación del lenguaje de programación Python que apunta a microcontroladores y sistemas integrados.

Los microcontroladores son computadoras compactadas en un solo chip muy pequeño. Los sistemas integrados son computadoras que funcionan dentro de un sistema mecánico o eléctrico más grande [26].

Los sistemas integrados a menudo utilizan microcontroladores.



Figura 2.11: Interfaz de MicroPython
Fuente: Adaptado de [27]

2.3.4 CNN

Una red neuronal convolucional es un tipo de red neuronal artificial donde las neuronas artificiales, corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria (V1) de un cerebro biológico. Este tipo de red es una variación de un perceptron multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones [28].

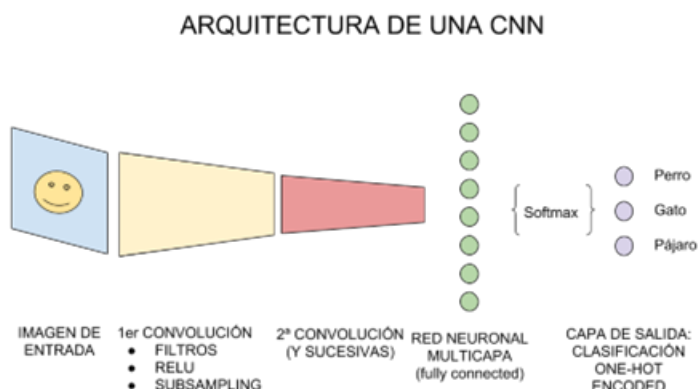


Figura 2.12: Arquitectura de una red CNN

Fuente: Adaptado de [28]

2.3.5 Visión Artificial

Es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un ordenador. Tal y como los seres humanos usamos nuestros ojos y cerebros para comprender el mundo que nos rodea, la visión informática trata de producir el mismo efecto para que los ordenadores puedan percibir y comprender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación. Esta comprensión se consigue gracias a distintos campos como la geometría, la estadística, la física y otras disciplinas. La adquisición de los datos se consigue por varios medios como secuencias de imágenes, vistas desde varias cámaras de video o datos multidimensionales desde un escáner médico [29].

2.3.6 Open CV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta). OpenCV ofrece soporte para varios sistemas operativos y varias arquitecturas de hardware, pero también ofrece el código fuente para que cualquier desarrollador lo compile en cualquier sistema operativo y arquitectura particular [30].

OpenCV y ESP32 se pueden utilizar juntos para crear sistemas de visión integrados, en los que se pueden ejecutar tareas de visión por computadora simples en un microcontrolador ESP32.



Figura 2.13: Logo de OpenCV
Fuente: Adaptado de [30]

2.3.6.1 Numpy

NumPy es una biblioteca para el lenguaje de programación Python, que agrega soporte para matrices y arreglos multidimensionales grandes, junto con una gran colección de funciones matemáticas de alto nivel para operar en estos arreglos [31].



Figura 2.14: Logo de Numpy
Fuente: Adaptado de [31]

2.3.7 XIAO ESP32 CAM

El ESP32-CAM es un módulo de cámara muy pequeño y económico que cuenta con el chip ESP32-S. Además de la cámara OV2640 y varios GPIO para conectar periféricos, también cuenta con una ranura para tarjeta microSD que puede ser útil para almacenar imágenes tomadas con la cámara o para almacenar archivos para servir a los clientes

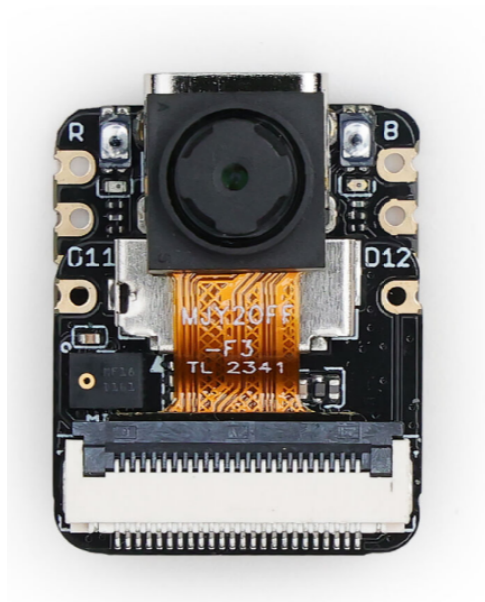


Figura 2.15: XIAO ESP32 CAM
Fuente: Adaptado de [12]

A continuación, se presenta el DataSheet encontrado.

Ítem	Descripción
Procesador	ESP32-S3R8, Xtensa LX7 de doble núcleo, 32 bits, hasta 240 MHz
Inalámbrico	Subsistema Wi-Fi completo de 2.4GHz, BLE: Bluetooth 5.0, Bluetooth mesh
Sensores Integrados	Sensor de cámara OV2640 (1600*1200), Micrófono digital
Memoria	8M PSRAM y 8MB Flash en chip, Ranura para tarjeta SD (hasta 32GB FAT)
Interfaz	1x UART, 1x IIC, 1x IIS, 1x SPI, 11x GPIOs (PWM), 9x ADC, 1x LED de usuario, 1x LED de carga 1x conector B2B (2 GPIOs adicionales), 1x botón de reinicio, 1x botón de arranque
Dimensiones	21 x 17.8 x 15 mm (con placa de expansión)
Alimentación	Voltaje de entrada (Type-C): 5V, Voltaje de entrada (BAT): 4.2V Voltaje de operación (Type-C): 5V@38.3mA, Voltaje de operación (BAT): 3.8V@43.2mA (con expansión)
Modelo de Bajo Consumo	Sueño de módem: 44mA, Sueño ligero: 5mA, Sueño profundo: 3mA
Consumo de Potencia (Wi-Fi)	Modelo activo: 110 mA (con expansión)
Consumo de Potencia (BLE)	Modelo activo: 102 mA (con expansión)
Temperatura de Funcionamiento	-40°C 65°C

Tabla 2.1: DataSheet de la XIAO ESP32 CAM

Fuente: [32]

2.3.8 Sensor ultrasónico (HC-SR04)

Este sensor mide la distancia entre el sensor y el nivel de basura dentro del cubo. Con esta información, se puede calcular el porcentaje de llenado. Funciona enviando pulsos ultrasónicos y midiendo el tiempo que tarda en regresar el eco, lo

que indica la distancia al objeto más cercano (en este caso, la basura).

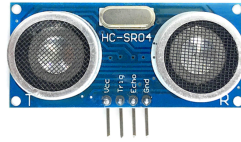


Figura 2.16: Sensor ultrasónico HC-SR04
Fuente: Adaptado de [33]

Especificación	Descripción
Voltaje de funcionamiento	DC 5V
Corriente de funcionamiento	15mA
Frecuencia de funcionamiento	40kHz
Alcance máximo	4 m
Alcance mínimo	2 cm
Precisión de medición	3 mm
Ángulo de medición	15 grados
Señal de entrada del disparador	Pulso TTL de 10 μ s
Dimensiones	45 x 20 x 15 mm

Tabla 2.2: Especificaciones del HC-SR04
Fuente: Adaptado de [33]

2.3.9 Cables jumper, resistencias, protoboard

Estos componentes se utilizan para conectar todos los sensores, microcontroladores y actuadores de manera temporal o durante la fase de pruebas. Los cables jumper permiten hacer conexiones rápidas entre pines, y la protoboard facilita las conexiones sin necesidad de soldar.

2.3.10 Servomotor (SG90)

El SG90 es un servo miniatura de gran calidad y diminutas dimensiones, además es bastante económico. Funciona con la mayoría de las tarjetas electrónicas de control con microcontroladores y además con la mayoría de los sistemas de radio control comerciales. Funciona especialmente bien en aeronaves dadas sus características de torque, tamaño y peso.

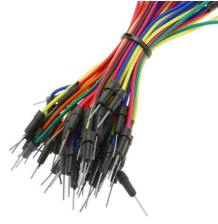


Figura 2.17: Jumpers
Fuente: Propia



Figura 2.18: Resistores
Fuente: Propia

El servo SG90 tiene un conector universal tipo “S” que encaja perfectamente en la mayoría de los receptores de radio control incluyendo los Futaba, JR, GWS, Cirrus, Hitec y otros. Los cables en el conector están distribuidos de la siguiente forma: Rojo = Alimentación (+), Café = Alimentación (-) o tierra, Naranja = Señal PWM.

Este tipo de servo es ideal para las primeras experiencias de aprendizaje y prácticas con servos, ya que sus requerimientos de energía son bastante bajos y se permite alimentarlo con la misma fuente de alimentación que el circuito de control. Por ejemplo, si se conecta a una tarjeta arduino, se puede alimentar durante las pruebas desde el puerto USB de la PC sin mayor problema [34].

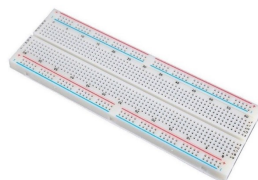


Figura 2.19: ProtoBoard
Fuente: Propia



Figura 2.20: Mini Sevo motor
Fuente: Adaptado de [34]

A continuación, se muestra el DataSheet encontrado:

Especificación	Descripción
Dimensiones	22mm x 11,5mm x 27mm
Peso	9 gramos
Peso con cable y conector	10.6 gramos
Torque a 4.8 volts	1.2 kg/cm
Voltaje de operación	4.0 a 7.2 volts
Velocidad de giro a 4.8 volts	120 ms / 60 °
Conector	Universal para la mayoría de los receptores de radio control
Compatibilidad	Compatible con tarjetas como Arduino y microcontroladores que funcionan a 5 volts.

Tabla 2.3: Especificaciones del SG90
Fuente: Adaptado de [34]

Capítulo 3: Propuesta del Proyecto

3.1 Metodo-Experimentación

3.1.1 Tipo y alcance de la investigación

La investigación es de tipo experimental con el diseño de un prototipo funcional, que emplea tecnologías de reconocimiento de imágenes y sensores ultrasónicos para clasificar de manera automática los residuos. Esto permite validar la hipótesis mediante pruebas en condiciones reales, evaluando la efectividad del sistema en clasificar diferentes tipos de materiales y contribuyendo a la reducción de residuos no clasificados correctamente.

La elección de un enfoque experimental obedece a la necesidad de observar y medir la precisión del sistema en un entorno controlado antes de una posible implementación a gran escala. Este tipo de investigación también permite hacer ajustes y mejoras al prototipo en función de los resultados obtenidos durante las pruebas.

El alcance del proyecto es de prueba de concepto, es decir, se busca construir un prototipo funcional que demuestre la viabilidad y eficacia de una solución automatizada para la clasificación de residuos. Al limitar el alcance a una prueba de concepto, se facilita el desarrollo inicial, y se enfoca en comprobar si la tecnología y el diseño propuestos cumplen con los objetivos de clasificación eficiente y mejora en la gestión de residuos.

Este prototipo servirá como base para evaluaciones futuras, pudiendo ser optimizado para aplicaciones más amplias en el campus o en otros contextos similares. Al finalizar el proyecto, se espera contar con datos que validen la hipótesis y ofrezcan una visión clara de los beneficios potenciales de la tecnología aplicada.

3.1.2 Observación

Contexto Universitario:

En centro regional de Veraguas de la Universidad Tecnológica de Panamá, se ha identificado un deficit creciente por la gestión de residuos, particularmente en áreas como la cafetería, pasillos y espacios comunes donde los desechos se mezclan, dificultando la separación de reciclables.

La alta concentración de personas genera un volumen significativo de residuos diarios, incluidos papel, cartón y plástico principalmente.

A menudo, las personas depositan residuos en cualquier basurero disponible sin tener en cuenta si son reciclables o no. Esto se traduce en una alta tasa de contaminación en los materiales reciclables, dificultando su procesamiento en las plantas de reciclaje.

3.1.3 Planteamiento de Hipótesis

Con la observación presentada anteriormente, nos hacemos la siguiente pregunta: **¿Cómo eliminamos la mezcla de desechos que dificulta la separación para reciclables?**

Nuestra hipótesis radica en la implementación de un sistema automatizado de clasificación de residuos que utilice tecnología de reconocimiento de imágenes (ESP32 CAM) aumentará la tasa de reciclaje en el centro regional de Veraguas al facilitar la separación de materiales reciclables y sensibilizar a la comunidad universitaria sobre la importancia del reciclaje.

3.1.4 Experimentación

3.1.4.1 Desarrollo del Prototipo

Estructura Física:

- **Diseño de la estructura:** Nos proponemos diseñar una estructura robusta que soporte los contenedores y los componentes electrónicos. La estructura debe ser de fácil acceso para los usuarios, permitiendo que los materiales sean presentados a la cámara.
- **Ubicación de la Cámara:** Esta parte se trata de instalar el ESP32 CAM en una posición elevada, asegurando que tenga un campo de visión claro para capturar el material que la persona va a presentar.

3.1.4.2 Integración Electrónica:

- **Conexiones:** Conectamos la ESP32 CAM a una fuente de alimentación adecuada y a los servomotores con la principal tarea de abrir las compuertas.

- **Sensores ultrasonicos:** Usamos sensores ultrasónicos para detectar cuando el usuario presenta un material. Esto nos ayuda a que la cámara esté activa y lista para capturar la imagen del material.

3.1.4.3 Programación del Sistema

Desarrollo del Algoritmo de Reconocimiento:

- **Recopilación de Datos:**
 - Se trata de recopilar un DataSet de Kaggle donde se tengan imágenes de cartón, papel y plástico en diversas condiciones de iluminación y ángulos. Esto nos da la oportunidad de que el modelo sea robusto y pueda manejar variaciones en el entorno real.
 - Luego, etiquetaremos las imágenes para crear un conjunto de datos que se utilizará en el entrenamiento del modelo de reconocimiento.
- **Utilizar Transfer Learning:**
 - Tomaremos una cnn pre-entrenada y aplicaremos técnicas de visión por computadora con el conjunto de datos recopilado. Este proceso puede incluir la validación cruzada para asegurar que el modelo no esté sobreajustado.
 - Realizaremos pruebas de precisión del modelo con un conjunto de imágenes no utilizadas durante el entrenamiento para evaluar su rendimiento y comportamiento.
- **Control del Hardware:**
 - **Lógica de Control:**
 - * Se trata de programar la lógica que controlará la activación del reconocimiento de imágenes cuando el sensor ultrasonico detecte que un usuario está presentando un material.
 - * Una vez que se identifica el material, se activa la compuerta correspondiente, utilizando los servomotores.
 - **Sistema de Retroalimentación:**
 - * Implementaremos un sistema de retroalimentación visual (por ejemplo, luces LED) que indique al usuario si la clasificación fue correcta o incorrecta. Si el material es reconocido correctamente, se abrirá la compuerta; si no, se podrá mostrar un mensaje en una pantalla o con luces que indiquen que el material no fue clasificado.

3.1.4.4 Interacción del Usuario

Proceso de Uso:

- El usuario se acercará al basurero inteligente y presentará el material al módulo ESP32 CAM.
- El sistema activará la cámara para tomar una imagen del material.
- El algoritmo analizará la imagen y, tras identificar correctamente el material, la compuerta correspondiente se abrirá para que el usuario deposite el residuo.
- En caso de que el material no sea reconocido, se proporcionará retroalimentación al usuario sobre la incorrecta clasificación con el uso del Buzzer.

3.1.5 Análisis de Datos

Con el prototipo desarrollado esperamos obtener los siguientes datos:

1. **Cantidad Total de Residuos:** Registrar el volumen total de residuos que se depositan en el basurero inteligente.
2. **Cantidad de Residuos Clasificados Correctamente:** Contar cuántos residuos fueron identificados y clasificados correctamente por el sistema en comparación con los incorrectamente clasificados.

Generar gráficos para la precisión del algoritmo de clasificación

Esta recolección de data nos permite los siguientes puntos:

1. **Visualizar de forma clara los datos**
 - (a) **La Comprensión:** Los gráficos transforman datos numéricos en visualizaciones fáciles de interpretar, lo que permite comprender rápidamente el rendimiento del algoritmo sin necesidad de analizar tablas complejas.
 - (b) **Identificar tendencias:** A través de gráficos de líneas podremos observar cómo la precisión ha cambiado a lo largo del tiempo, lo que puede indicar si el sistema está mejorando con el uso.
2. **Comparar**

- (a) **Comparación entre categorías:** Con gráficos de barras, podemos comparar la precisión entre diferentes categorías de materiales, lo que ayuda a identificar cuál material se clasifica mejor y cuál necesita mejoras.
- (b) **Evaluación de resultados:** Permiten comparar resultados de diferentes períodos de prueba o diferentes versiones del algoritmo, facilitando la toma de decisiones sobre ajustes y mejoras.

3. Identificar problemas

- (a) **Detectar anomalías:** Un gráfico que muestra una caída considerable o preocupante para nosotros en la precisión puede alertar sobre problemas en el algoritmo o cambios en el comportamiento del usuario.
- (b) **Clasificación Incorrecta:** Un gráfico de pastel puede mostrar la proporción de clasificaciones correctas frente a incorrectas puede ayudar a identificar el alcance del problema de clasificación.

4. Datos para historial de mejoras

- (a) **Cronograma de Mejoras:** Utilizar gráficos para establecer un cronograma de versiones del algoritmo permite visualizar cuándo se implementarán mejoras y ajustes basados en los datos recolectados.
- (b) **Documentación de Cambios:** Cada versión puede ir acompañada de gráficos que muestren la evolución de la precisión, lo que facilita la comunicación sobre el impacto de cada actualización en el rendimiento del sistema.

3.1.6 Análisis y conclusiones

Los datos obtenidos durante la experimentación se analizarán para medir la precisión, velocidad y eficiencia en la clasificación de materiales. Se espera observar una adecuada precisión en la clasificación de materiales y una mejora en la disposición de residuos reciclables.

3.2 Escenario De Aplicación

El prototipo se puede aplicar en el centro regional de Veraguas de la Universidad Tecnológica de Panamá, específicamente en áreas de alta concurrencia como la cafetería y pasillos. Este entorno es ideal debido al volumen de desechos generados diariamente, y se espera que el prototipo contribuya a mejorar la tasa de reciclaje.

El impacto esperado incluye la reducción de residuos mezclados y una mayor concientización sobre la importancia de la correcta clasificación de desechos en el campus universitario. Además, se espera que el sistema genere datos útiles sobre los tipos de residuos más comunes en estas áreas.

3.3 Cronograma de Actividades

Actividad	Duración Est.	Fecha Inicio	Fecha Fin
Investigación inicial y planificación	2 semanas	11/9/2024	24/9/2024
Diseño conceptual del prototipo	2 semanas	25/9/2024	8/10/2024
Adquisición de componentes	1 semana	9/10/2024	15/10/2024
Desarrollo y programación del sistema	1 semana	16/10/2024	22/10/2024
Ensamblaje del prototipo	2 semanas	23/10/2024	5/11/2024
Pruebas y ajustes del prototipo	1 semana	6/11/2024	12/11/2024
Análisis de resultados	2 días	13/11/2024	14/11/2024

Tabla 3.1: Cronograma de Actividades

- **Investigación inicial y planificación**

- **Duración:** 2 semanas (11/9/2024 - 24/9/2024)
- **Descripción:** Esta fase la consideramos primordial porque nos apoyamos de hombros de gigantes e implica la recopilación de información relevante sobre el proyecto y la identificación de los objetivos y metas. Se realiza una revisión de la literatura existente y se elabora un plan de acción que guiará el desarrollo del prototipo. Con esto podemos avanzar a las etapas de diseño y ejecución.

- **Diseño conceptual del prototipo**

- **Duración:** 2 semanas (25/9/2024 - 8/10/2024)
- **Descripción:** En esta etapa, se desarrollan diferentes versiones del diseño inicial del prototipo. Esto es como van interactuando los componentes en el prototipo y cual es la salida o solución brindada al usuario. Se busca definir cómo interactuarán los distintos elementos del sistema, asegurando que cumplan con los requisitos establecidos durante la fase de investigación.

- **Adquisición de componentes**

- **Duración:** 1 semana (9/10/2024 - 15/10/2024)
 - **Descripción:** Durante esta fase, se compran y obtienen todos los componentes necesarios para la construcción del prototipo. Esto incluye el hardware que se verá en la otra sección (sensores, microcontroladores, etc.) como el software necesario para el desarrollo. Es crucial asegurarse de que todos los elementos estén disponibles para la fase de desarrollo.
- **Desarrollo y programación del sistema**
 - **Duración:** 1 semana (16/10/2024 - 22/10/2024)
 - **Descripción:** Esta etapa implica la codificación del software y la integración de los diferentes componentes del sistema. Se desarrollan las funciones necesarias y se realizan pruebas unitarias para asegurar que cada parte del sistema funcione correctamente. Es una fase crítica, ya que una buena programación facilitará la posterior integración y pruebas del prototipo.

- **Ensamblaje del prototipo**

- **Duración:** 2 semanas (23/10/2024 - 5/11/2024)
- **Descripción:** En esta fase, se ensamblan físicamente todos los componentes del prototipo según el diseño conceptual. Se presta atención a como van a estar los elementos, la conexión de cables y la implementación del sistema en un formato compacto. Este paso es esencial para garantizar que el sistema esté listo para las pruebas.

- **Pruebas y ajustes del prototipo**

- **Duración:** 1 semana (6/11/2024 - 12/11/2024)
- **Descripción:** Durante esta etapa, se realizan muchas pruebas para identificar fallos o áreas que vamos a tener que mejorar en el prototipo. Se ajustan parámetros, se corrigen errores y se optimiza el rendimiento del sistema. Las pruebas son vitales para asegurarnos que el prototipo cumpla con lo que queremos.

- **Análisis de resultados**

- **Duración:** 2 días (13/11/2024 - 14/11/2024)
- **Descripción:** En esta última fase, se evalúan los resultados obtenidos de las pruebas. Se analizan los datos recopilados y se determinan las conclusiones sobre el desempeño del prototipo. Este análisis es crucial para decidir si se requiere una revisión adicional del diseño o si el prototipo está listo para su presentación o implementación.

3.4 Presupuesto Real

Recurso	Cantidad	Costo Est(dls)
Recursos Humanos		
Ingeniero en electrónica (diseño y programación)	1	500
Técnico en ensamblaje	1	100
Recursos Económicos		
Placa Arduino UNO R3	1	16.99
XIAO ESP32 S3 Sense	1	23.99
Módulo de cámara OV2640	1	8.99
Sensor ultrasonico	3	6.99
mini-Servomotores	3	6.99
Deposito de basura	1	8.99
Fuente de alimentación	1	5.00
Cables y conectores,LEDS	Varios	9.99
ProtoBoard	2	6.96
Buzzer	1	0.70
Pantalla LED	1	3.99
Recursos Tecnológicos		
Computadora para desarrollo y programación	1	(ya disponible)
Software de diseño y programación	2	0 (de uso libre)
Total		699.58

Tabla 3.2: Presupuesto Real del Proyecto
Precios referencia: [35, 39, 38, 36, 37]

- **Recursos Humanos** Los dos roles incluidos, un ingeniero en electrónica para el diseño y programación, y un técnico en ensamblaje son sumamente necesarios ya que el proyecto requiere habilidades especializadas para integrar tanto el hardware como el software. El ingeniero es fundamental en esta fase, pues diseñará el sistema de control y programación de los sensores y actuadores. El técnico de ensamblaje asegurará que los componentes se monten correctamente, lo cual es crucial para pruebas y un funcionamiento efectivo.
- **Recursos Económicos** Los componentes electrónicos detallados, como la placa Arduino UNO R3, el microcontrolador XIAO ESP32 S3 Sense y el módulo de cámara OV2640, apuntan a un sistema que necesita procesar imágenes y posiblemente tomar decisiones automáticas en tiempo real. Este conjunto

de piezas sugiere un sistema de clasificación de objetos, tal vez para separar materiales reciclables, como se deduce del "Depósito de basura".

- **Sensores y Actuadores:** La presencia de sensores ultrasónicos indica que el sistema debe detectar proximidad, activando la cámara o algún mecanismo solo cuando un usuario presenta un objeto.
- **Servomotores y Fuente de alimentación:** Los servomotores se utilizan para abrir compuertas o mover componentes en función de lo detectado.
- **Conectores y Cables:** Estos aseguran que todos los componentes se comuniquen correctamente y tengan la energía necesaria.

El hecho de incluir un buzzer y una pantalla LED se trata de un mecanismo de retroalimentación para el usuario, posiblemente para indicar si un material ha sido clasificado correctamente o no.

- **Recursos Tecnológicos** La necesidad de una computadora y software libre para el desarrollo y programación refleja un esfuerzo por minimizar costos al aprovechar recursos disponibles. La computadora es necesaria para desarrollar y cargar el código al microcontrolador, mientras que el software libre como Arduino IDE o plataformas de Machine Learning de código abierto permiten reducir los costos sin limitar las capacidades del prototipo.
- **Total del Presupuesto** El monto final de 699.58 USD es razonable para un prototipo experimental con múltiples componentes y tecnología de detección, clasificación, y retroalimentación visual/auditiva. Al tratarse de un presupuesto tentativo, refleja el gasto en una configuración mínima viable para validar la funcionalidad del sistema y probar su eficacia.

Además, es un monto que cualquier empresa privada podría comprar o incluso patrocinar.

3.5 Diseño conceptual



INCORPORA
ESP32 CAM

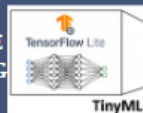


INCORPORADO
EN BASURERO



ARDUINO UNO
PROTOBOARD

INCLUYE
MACHINE
LEARNING



CLASIFICAR
MATERIALES



ABREN
COMPUER
TAS



MATERIALES
RECICLADOS



SISTEMA
DE
NOTIFICACIÓN



HC-SR04
MONITOREO
CONTINUO

3.5.1 Descripción del diseño conceptual planteado

1. Arduino UNO R3

El Arduino Uno es el corazón del sistema, encargado de recibir señales de los distintos componentes (cámara, sensores, etc.) y tomar decisiones en función de los datos recibidos.

Interacción: El Arduino coordina todo el sistema. Recibe las imágenes desde la cámara (ESP32 CAM), los datos de clasificación del modelo de machine learning (TinyML), y las lecturas del sensor ultrasónico (HC-SR04). Luego, controla la apertura de las compuertas y activa el sistema de notificación según sea necesario.

Acción: Procesa todas las señales y ejecuta comandos para manejar los actuadores y sensores.

2. Incorporación de ESP32 CAM (Cámara)

La ESP32 CAM está instalada dentro del basurero. Su principal tarea es capturar imágenes de los residuos que se introducen.

Interacción: Cada vez que se coloca un objeto en el basurero, se activa la cámara para tomar una foto. Esta imagen es esencial para el siguiente paso, que es la clasificación mediante machine learning.

3. Procesamiento con Machine Learning (TinyML)

Una vez que la cámara toma la imagen del residuo, esta se envía al sistema de machine learning, específicamente utilizando TensorFlow Lite, una versión ligera diseñada para dispositivos como la ESP32.

Interacción: El sistema de machine learning procesa la imagen en tiempo real y clasifica el material como plástico, papel, vidrio, etc. Esto se basa en el entrenamiento previo de un modelo que ha aprendido a identificar diferentes tipos de residuos.

Decisión: Dependiendo de la clasificación del material, el sistema decide en qué compartimiento debe depositarse el residuo.

A continuación planteamos el siguiente diseño de como el modulo ESP32 CAM está involucrado de cierto modo con el proceso de machine Learning:

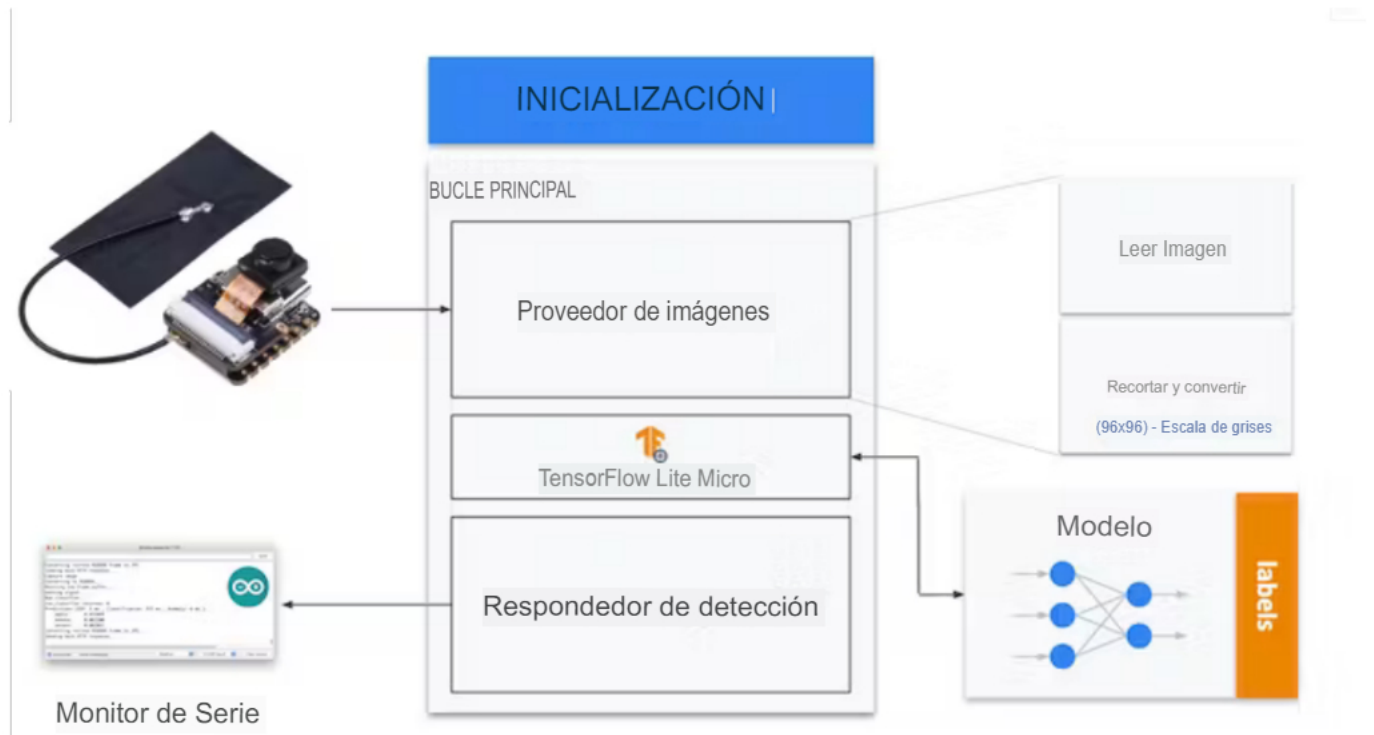


Figura 3.1: ESP32 CAM y Machine Learning
Fuente: Adaptado de [40]

Descripción

Inicialización

Basicamente es configurar el sistema, inicializando todos los componentes del hardware y el software necesarios para realizar el procesamiento de imágenes y la clasificación.

Luego se inicializa la XIAO ESP32 CAM y el modelo de TensorFlow Lite Micro se carga en la memoria.

Proveedor de Imágenes

Función: La XIAO ESP32 CAM captura la imagen del objeto o material que se desea clasificar.

Proceso:

Cuando se le da una imagen a la cámara, su principal tarea es capturar la imagen y enviarla para ser procesada.

Leer Imagen: Se obtiene la imagen capturada en bruto.

Recortar y Convertir: Esta imagen es procesada, recortada y convertida a escala de grises para reducir la cantidad de datos y permitir una clasificación más eficiente. Aquí se menciona que la imagen se ajusta a una resolución de 96x96 píxeles.

TensorFlow Lite Micro

Función: Una vez que la imagen ha sido preprocesada, es enviada al módulo de TensorFlow Lite Micro, que está ejecutándose en la ESP32. Este módulo es responsable de aplicar el modelo de machine learning a la imagen procesada.

Proceso: La imagen recortada y convertida es alimentada al modelo entrenado que reside en el sistema. Este modelo ha sido previamente entrenado para identificar ciertos patrones en las imágenes (clasificación de materiales, por ejemplo).

Modelo: El modelo de machine learning es una red neuronal simple, optimizada para sistemas embebidos con bajos recursos como la ESP32. El modelo toma como entrada la imagen convertida y devuelve una clasificación, basada en etiquetas predefinidas (labels) que corresponden a los materiales o categorías que el sistema debe reconocer.

Respondedor de Detección

Función: Una vez que el modelo ha procesado la imagen y ha generado una salida, el respondedor de detección se encarga de interpretar esta salida y tomar una acción en función de la predicción.

Proceso: El sistema compara la salida del modelo con las etiquetas posibles (plástico, papel y cartón).

La respuesta de la detección (la etiqueta del material identificado) se envía al monitor serie para que podamos visualizarla. Esto lo vamos a hacer por motivos de ver como se va comportando el modelo de detección y en base a eso, hacer ajustes o aplicar técnicas de optimizar.

El Arduino cuando lo conectemos puede utilizar esta salida para realizar una acción, como activar un motor para mover el residuo al compartimiento correspondiente o encender una señal de alerta.

Monitor de Serie (Interfaz de Usuario)

Función: El monitor de serie muestra las salidas del sistema, permitiendonos ver la clasificación en tiempo real.

Proceso: A través de la consola de monitor serial, se pueden ver las etiquetas o categorías de los materiales clasificados.

Este componente puede también actuar como una herramienta de depuración, ayudando a entender cómo está funcionando el proceso en general.

4. Apertura de compuertas según la clasificación

Una vez que el machine learning ha clasificado el residuo, el sistema de control que viene siendo Arduino recibe la instrucción de abrir la compuerta correspondiente.

Interacción: El Arduino Uno controla motores o actuadores que abren las compuertas del basurero según el tipo de material identificado. Por ejemplo, si se detecta plástico, se abre la compuerta destinada para plásticos.

Acción: El residuo es depositado en el compartimiento adecuado para su recolección posterior.

5. Monitoreo continuo del nivel de llenado (HC-SR04)

Mientras los residuos se van acumulando, el sensor ultrasónico HC-SR04 monitorea continuamente el nivel de llenado de cada compartimiento.

Interacción: Este sensor mide la distancia entre el residuo más alto y el sensor, que está colocado en la parte superior del compartimiento. A medida que los residuos se acumulan, el sensor detecta que la distancia se reduce, lo que indica que el compartimiento se está llenando.

Decisión que va a tomar Si el compartimiento alcanza un cierto nivel como "lleno", el sistema activará una alerta para indicar que se necesita vaciar el compartimiento(s).

6. Sistema de notificación

El sistema incluye un mecanismo de notificación visual y/o sonora (buzzer y LEDs). Este se activa en dos situaciones principales:

Llenado del compartimiento: Cuando el sensor HC-SR04 detecta que un compartimiento está lleno, activa una señal para notificar que el basurero necesita ser vaciado.

Error o evento inusual: También puede notificar si se detecta algún material que no puede clasificarse, o si hay algún error en el proceso.

Interacción: El Arduino Uno controla las luces LED o el buzzer, emitiendo una alerta visual (como encender una luz roja) o sonora (emitir un pitido) cuando es necesario.

7. Materiales reciclados y gestión de residuos

Los residuos clasificados correctamente son depositados en diferentes compartimientos dentro del basurero.

Interacción: Una vez que el material se clasifica y se deposita, el sistema de monitoreo continúa supervisando los niveles de llenado. Los materiales reciclados se almacenan hasta que un operador recoja los residuos para su tratamiento en una planta de reciclaje.

- [1] Mohd Mahboob Ali, Harshavardhan CHVS, Gundumalla Sai Teja, B. Veera Jyothi, L. Suresh Kumar. (2024). *Intelligent waste sorting system: leveraging arduino for automated trash identification and categorization*. Disponible en: <https://cspub-ijcisim.org/index.php/ijcisim/article/view/733>
- [2] Jasim, A. M., Qasim, H. H., Jasem, E. H., Saihood, R. H. (2021). *An internet of things based smart waste system*. *International Journal of Electrical and Computer Engineering (IJECE)*, 11(3), 2577-2585.
- [3] Anggrawan, A., Hadi, S., Satria, C. (2022). *IoT-Based garbage container system using NodeMCU ESP32 microcontroller*. *Journal of Advances in Information Technology Vol*, 13(6).
- [4] G, C. a. G. (2024, July 24). *PANAMÁ: DISPOSICIÓN y RECOLECCIÓN DE DESECHOS. RETOS DE UN NUEVO GOBIERNO. OPRA*. Disponible en: <https://observapanama.com/panama-disposicion-y-recoleccion-de-desechos-retos-de-un-nuevo-gobierno/>
- [5] Ecoembes, «La tecnología del reciclaje, ¿ayuda al medioambiente? | Ecoembes», *Ecoembes Reduce Reutiliza y Recicla*. Accedido: 10 de octubre de 2024. [En línea]. Disponible en: <https://reducereutilizarecicla.org/tecnologia-del-reciclaje/>
- [6] G. Caputo, «Tecnología Sostenible ¿Qué es y cómo contribuye al desarrollo sostenible?», *DoGood People*. Accedido: 10 de octubre de 2024. [En línea]. Disponible en: <https://www.dogoodpeople.com/es/tendencias-rsc/tecnologia-sostenible/tecnologia-sostenible-su-contribucion-desarrollo-sostenible/>
- [7] «Introduccion A Los Microcontroladores Pic», *dokumen.pub*. Accedido: 10 de octubre de 2024. [En línea]. Disponible en: <https://dokumen.pub/introduccion-a-los-microcontroladores-pic.html>
- [8] C. Peña, *Introducción a Arduino*. RedUsers, 2020.
- [9] “Arduino ¿Que es, como funciona? y sus partes”. *Ingeniería Mecafenix*. Accedido el 13 de octubre de 2024. <https://www.ingmecafenix.com/electronica/programacion/arduino/>

- [10] *ESP32 NodeMCU Módulo WLAN WiFi Junta de Desarrollo* <https://www.az-delivery.de/es/products/esp32-developmentboard>
- [11] R. A. Salvador y C. B. Prieto, «Sistema de seguridad con reconocimiento facial en módulo ESP32», *Mare Ingenii*, vol. 4, n.o 1, Art. n.o 1, oct. 2022, doi: 10.52948/mare.v4i1.684.
- [12] *Seeed Studio XIAO ESP32S3 Sense* Accedido el 10 de octubre de 2024 <https://www.seeedstudio.com/XIAO-ESP32S3-Sense-p-5639.html>
- [13] C. Russo, H. Ramón, N. Alonso, B. Cicerchia, L. Esnaola, y J. P. Tessore, *Tratamiento masivo de datos utilizando técnicas de machine learning. Universidad Nacional de Entre Ríos*, 2016. Accedido: 8 de octubre de 2024. [En línea]. Disponible en: <http://repositorio.unnoba.edu.ar/xmlui/handle/23601/107>
- [14] G. A. Betancourt, «LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs)», *Sci. Tech.*, vol. 1, n.o 27, Art. n.o 27, abr. 2005. Accedido: 8 de octubre de 2024. [En línea]. Disponible en: <https://revistas.utp.edu.co/index.php/revistaciencia/article/view/6895>
- [15] *Sistema de monitoreo de acuicultura* Accedido el 10 de octubre de 2024. <https://humanizationoftechnology.com/sistema-de-monitoreo-de-acuicultura/revista/iot/05/2022/>.
- [16] I. F. Sinaluisa Lozano, C. A. Gallardo Naula, A. J. Morales Domínguez, y C. W. Montufar Paguay, «Internet de las Cosas aplicada a la movilidad y recolección inteligentes de residuos sólidos municipales», *Polo Conoc. Rev. Científico - Prof.*, vol. 8, n.o 7 (JULIO 2023), pp. 944-961, 2023.
- [17] *Empresa Logix5 Smart Solutions S.L* Accedido el 10 de octubre de 2024 <https://roboticaeducativa.logix5.com/>
- [18] M. Buying, «Contenedores de basura inteligentes: pioneros en una revolución en la gestión sostenible de residuos», *Contenedores De Basura México*. Accedido: 8 de octubre de 2024. [En línea]. Disponible en: <https://contenedoresdebasura.com/contenedores-de-basura-inteligentes-pioneros-en-una-revolucion-en-la-gesti>
- [19] M. Quimbita y M. Andrés, «Evaluación de pasarela LoRa/LoRaWAN en entornos urbanos», oct. 2018. Accedido: 8 de octubre de 2024. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/109791>

- [20] *primer basurero “inteligente” de Chile en el centro de Santiago* Accedido el 10 de octubre de 2024 <https://www.plataformaurbana.cl/archive/2015/11/12/instalan-el-primer-basurero-inteligente-de-chile-en-el-centro-de-santiago/>.
- [21] *H. M. Manrique, «UNIVERSIDAD NACIONAL DEL CALLAO».*
- [22] *M. M. Ali, H. Chvs, G. S. Teja, B. V. Jyothi, y L. S. Kumar, «Intelligent Waste Sorting System: Leveraging Arduino for Automated Trash Identification and Categorization», Int. J. Comput. Inf. Syst. Ind. Manag. Appl., vol. 16, n.o 3, Art. n.o 3, jul. 2024.*
- [23] *H. Hassan, F. Saad, y M. S. Mohd Raklan, «A Low-Cost Automated Sorting Recycle Bin powered by Arduino Microcontroller», en 2018 IEEE Conference on Systems, Process and Control (ICSPC), dic. 2018, pp. 182-186. doi: 10.1109/SPC.2018.8704146.*
- [24] *D. A. Buitrago Aponte, «Sistema de clasificación de alimentos a través de inteligencia artificial: Estudio del tomate», may 2023. Accedido: 8 de octubre de 2024. [En línea]. Disponible en: <https://hdl.handle.net/1992/73814>*
- [25] *I. H. Pérez-Tavera, «Arduino IDE», Vida Científica Bol. Científico Esc. Prep. No 4, vol. 11, n.o 21, Art. n.o 21, ene. 2023.*
- [26] *N. H. Tollervey, Programming with MicroPython: Embedded Programming with Microcontrollers and Python. O’Reilly Media, Inc., 2017.*
- [27] *ESP32 MPY-Jama un nuevo entorno de desarrollo para Micropython / profe Tolocka* Accedido el 10 de octubre de 2024. <https://www.profetolocka.com.ar/2023/02/02/esp32-mpy-jama-un-nuevo-entorno-de-desarrollo-para-micropython/>
- [28] *«Red neuronal convolucional», Wikipedia, la enciclopedia libre. 24 de mayo de 2024. Accedido: 9 de octubre de 2024. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Red_neuronal_convolutional&oldid=160312860*
- [29] *«Visión artificial», Wikipedia, la enciclopedia libre. 21 de julio de 2024. Accedido: 9 de octubre de 2024. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Visi%C3%B3n_artificial&oldid=161428067*

- [30] «*OpenCV*», *Wikipedia, la enciclopedia libre*. 14 de febrero de 2024. Accedido: 9 de octubre de 2024. [En línea]. Disponible en: <https://es.wikipedia.org/w/index.php?title=OpenCV&oldid=158183931>
- [31] «*NumPy*», *Wikipedia*. 30 de septiembre de 2024. Accedido: 9 de octubre de 2024. [En línea]. Disponible en: <https://en.wikipedia.org/w/index.php?title=NumPy&oldid=1248522317>
- [32] *DataSheet de XIAO ESP32 CAM* Disponible en: https://www.seeedstudio.com/XIAO-ESP32S3-Sense-p-5639.html?queryID=be6f53c7ae317f6cbbc28f0668fb03df&objectID=5639&indexName=bazaar_retailer_products
- [33] *How the HC-SR04 Sensor Works and Example Programs with Arduino», NN Digital | Learn Arduino, ESP8266 / NodeMCU, STM32, Raspberry Pi, Microcontroller and Other Information Technology*. Accedido el 9 de octubre de 2024 <https://www.nn-digital.com/en/blog/2019/08/07/how-the-hc-sr04-sensor-works-and-example-programs-with-arduino/>
- [34] «*Micro Servo Motor SG90 9g | Arduino.cl - Compra tu Arduino en Línea*». Accedido: 9 de octubre de 2024. [En línea]. Disponible en: <https://arduino.cl/producto/micro-servo-motor-sg90-9g/>
- [35] *SIPYTOPF Paquete de Servomotores*. Accedido el 10 de octubre de 2024. https://www.amazon.com/-/es/SIPYTOPF-Paquete-servomotores-Helic%C3%B3ptero-Caminar/dp/B09185SC1W/ref=sr_1_32?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&sr=8-32.
- [36] *BOJACK Conectores y Cables para Protoboard*. Accedido el 10 de octubre de 2024. https://www.amazon.com/BOJACK-Solderless-Flexible-Breadboard-Connecting/dp/B08YRGVYPV/ref=mp_s_a_1_3?sr=8-3.
- [37] *MCIGICM Protoboard sin Soldadura*. Accedido el 10 de octubre de 2024. https://www.amazon.com/MCIGICM-soldadura-bloques-conexi%C3%B3n-distribuci%C3%B3n/dp/B08115P2T4/ref=sr_1_2_sspa?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=KKEKRCG6EY1X&dib=eyJ2IjoMSJ9.n_ouMOH1-4U1TfT6NnyzOP27agXFvwpdpQDA9kKxVzYLIAPmmt34gPN84r30HZayHjnz7fQt8U3TrfMGmwHs2DcVvZUrXjoNGETKxSpBpJic-OuEVggA_

1Fd7uPpaGgGgm20KoLvqq30nabWY1ExK5I87qrsEqVxgn4BwJps4tvDKiF9VqoFEjmbZ8p_
EcKd_mvLJ7fmupow2SDy6tBmHZKkMvwW4xAo.SCKCgXJxXbckWRCMhoJUNEoTPZyXpXKILz-1F
dib_tag=se&keywords=2+protoboards&qid=1728768709&
sprefix=2+protoboard%2Caps%2C139&sr=8-2-spons&sp_csd=
d2lkZ2V0TmFtZT1zcF9hdGY&psc=1.

[38] *Buzzer Piezoeléctrico para Proyectos Electrónicos*. Accedido el 10 de octubre de 2024. https://www.amazon.com/piezoel%C3%A9ctricos-electr%C3%B3nico-computadoras-componentes-electr%C3%B3nicos/dp/B07VK1GJ9X/ref=sxin_16_pa_sp_search_thematic_sspa?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&cv_ct_cx=buzzer&sbo=RZvfv%2F%2FHxDF%2B05021pAnSA%3D%3D&sr=1-5-6024b2a3-78e4-4fed-8fed-e1613be3bcce-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9zZWYyY2hfdGhlbWFOaWM&psc=1.

[39] *Placa Elegoo ATmega328P Arduino UNO*. Accedido el 10 de octubre de 2024. https://www.amazon.com/Elegoo-Placa-ATmega328P-ATMEGA16U2-Arduino/dp/B01EW0E0UU/ref=sxin_16_pa_sp_search_thematic_sspa?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&cv_ct_cx=arduino+uno&s=electronics&sbo=RZvfv%2F%2FHxDF%2B05021pAnSA%3D%3D&sr=1-1-6024b2a3-78e4-4fed-8fed-e1613be3bcce-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9zZWYyY2hfdGhlbWFOaWM&psc=1.

[40] *MJRoBot (Marcelo Rovai). "TinyML Made Easy: Image Classification w/ XIAO ESP32S3 Sense"*. Accedido el 10 de octubre de 2024. <https://www.hackster.io/mjrobot/tinyml-made-easy-image-classification-w-xiao-esp32s3-sense-cb42ae>