

	GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS		CÓDIGO	FO-GS-15
			VERSIÓN	02
	ESQUEMA HOJA DE RESUMEN		FECHA	03/04/2017
			PÁGINA	1 de 1
ELABORÓ		REVISÓ		APROBÓ
Jefe División de Biblioteca		Equipo Operativo de Calidad		Líder de Calidad

RESUMEN TRABAJO DE GRADO

AUTOR(ES): NOMBRES Y APELLIDOS COMPLETOS

NOMBRE(S): PAULA LINNED

APELLIDOS: SUESCUN VERA

NOMBRE(S): PAOLA ROCÍO

APELLIDOS: NOCUA ENCISO

FACULTAD: INGENIERÍA

PLAN DE ESTUDIOS: INGENIERÍA ELECTRÓNICA

DIRECTOR:

NOMBRE(S): MARIO JOAQUÍN

APELLIDOS: ILLERA BUSTOS

NOMBRE(S): GEINER GIOVANNY

APELLIDOS: BARBOSA CASANOVA

TÍTULO DEL TRABAJO (TESIS): SISTEMA DE CONTENEDORES CONTROLADO POR VOZ PARA LA CORRECTA CLASIFICACIÓN DE LOS RESIDUOS SÓLIDOS

RESUMEN: El reciclaje es una estrategia que permite mejorar el trato de los residuos sólidos que pueden ser útiles en otros aspectos, pero su correcta clasificación no siempre resulta satisfactoria. El problema se presenta cuando se desconoce cómo deshacerse de los desperdicios que ya no representan un valor útil, esta falta de conocimiento y de la poca participación en actividades de reciclaje conlleva al aumento de la contaminación ambiental. Por esta razón, en este trabajo se presenta un prototipo de un sistema de contenedores controlado por voz con capacidad de 12 litros como herramienta de apoyo en el proceso de clasificación de los residuos sólidos, el cual fue implementado sobre el sistema embebido Raspberry Pi 3B+ y codificado en lenguaje Python. Dicha codificación basó su funcionamiento en la herramienta de voz 'Vosk' en su modelo de idioma español para el reconocimiento de voz. En sus pruebas de funcionamiento se obtuvo un porcentaje de aciertos del 92.64% en el reconocimiento de los comandos de voz, también, se obtuvo una Tasa de Error de Palabra (WER) del 5.41% para el modelo utilizado en el procesamiento digital de voz.

PALABRAS CLAVES: Reciclaje, Herramienta Vosk, Residuos Sólidos, Redes Neuronales, Reconocimiento de Voz.

CARACTERÍSTICAS:

PÁGINAS: 87

PLANOS: 0

CD ROOM: 0

ILUSTRACIONES: 15

**SISTEMA DE CONTENEDORES CONTROLADO POR VOZ PARA LA CORRECTA
CLASIFICACIÓN DE LOS RESIDUOS SÓLIDOS**

**PAULA LINED SUECUN VERA – 1161512
PAOLA ROCIO NOCUA ENCISO – 1161540**

**UNIVERSIDAD FRANCISCO DE PAULA SANTANDER
FACULTAD DE INGENIERÍA
PROGRAMA ACADÉMICO DE INGENIERÍA ELECTRÓNICA
SAN JOSÉ DE CÚCUTA**

2023

**SISTEMA DE CONTENEDORES CONTROLADO POR VOZ PARA LA CORRECTA
CLASIFICACIÓN DE LOS RESIDUOS SÓLIDOS**

**Proyecto de grado para optar por el título de
INGENIERO ELECTRÓNICO**

**Presentado por
PAULA LINED SUECUN VERA – 1161512
PAOLA ROCIO NOCUA ENCISO – 1161540**

**Director
MSc. MARIO JOAQUÍN ILLERA BUSTOS
Ingeniero Electrónico**

**Codirector
MSc. GEINER GIOVANNY BARBOSA CASANOVA
Ingeniero Electrónico**

**UNIVERSIDAD FRANCISCO DE PAULA SANTANDER
FACULTAD DE INGENIERÍA
PROGRAMA ACADÉMICO DE INGENIERÍA ELECTRÓNICA
SAN JOSÉ DE CÚCUTA**

2023

ACTA DE SUSTENTACIÓN DE UN TRABAJO DE GRADO

Fecha: CÚCUTA, 03 DE MARZO DE 2023

Hora: 14:00

Lugar: SALON SB 302

Plan de Estudios: INGENIERÍA ELECTRÓNICA

Título del trabajo de grado: "SISTEMA DE CONTENEDORES CONTROLADO POR VOZ PARA LA CORRECTA CLASIFICACIÓN DE LOS RESIDUOS SÓLIDOS".

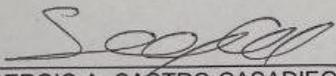
Jurados: IE MSc. SERGIO A. CASTRO CASADIEGO
IE MSc. JULIAN O. TARAZONA ANTELIZ

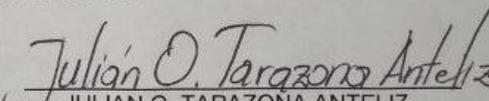
Director: IE. MSc. MARIO JOAQUIN ILLERA BUSTOS

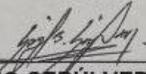
Codirector: IE. MSc. GEINER GIOVANNY BARBOSA CASANOVA

Nombre del Estudiante:	Código:	Calificación:	
		Número	Letra
PAULA LINNED SUESCUN VERA	1161512	4,2	Cuatro,dos

APROBADA


SERGIO A. CASTRO CASADIEGO


JULIAN O. TARAZONA ANTELIZ


SERGIO SÉPULVEDA MORA
Coordinador Comité Curricular
Ingeniería Electrónica

ACTA DE SUSTENTACIÓN DE UN TRABAJO DE GRADO

Fecha: CÚCUTA, 03 DE MARZO DE 2023

Hora: 14:00

Lugar: SALON SB 302

Plan de Estudios: INGENIERÍA ELECTRÓNICA

Título del trabajo de grado: "SISTEMA DE CONTENEDORES CONTROLADO POR VOZ PARA LA CORRECTA CLASIFICACIÓN DE LOS RESIDUOS SÓLIDOS".

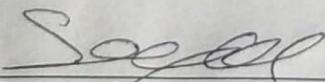
Jurados: IE MSc. SERGIO A. CASTRO CASADIEGO
IE MSc. JULIAN O. TARAZONA ANTELIZ

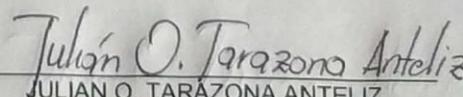
Director: IE. MSc. MARIO JOAQUIN ILLERA BUSTOS

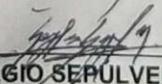
Codirector: IE. MSc. GEINER GIOVANNY BARBOSA CASANOVA

Nombre del Estudiante:	Código:	Calificación:	
		Número	Letra
PAOLA ROCIO NOCUA ENCISO	1161540	4,2	Cuatro, dos

APROBADA


SERGIO A. CASTRO CASADIEGO


JULIAN O. TARAZONA ANTELIZ


SERGIO SEPÚLVEDA MORA
Coordinador Comité Curricular
Ingeniería Electrónica

CONTENIDO

1. DESCRIPCIÓN DEL PROBLEMA	13
1.1. Planteamiento del problema.....	13
1.2. Justificación	14
1.2.1. Beneficios Tecnológicos	15
1.2.2. Beneficios Económicos	15
1.2.3. Beneficios Sociales	16
1.2.4. Beneficios Institucionales	16
1.3. Alcances	16
1.3.1. Tipo de proyecto.....	16
1.4. Delimitaciones	17
1.5. Objetivos	17
1.5.1. Objetivo General	17
1.5.2. Objetivos Específicos	17
2. MARCO TEÓRICO.....	18
2.1. Generación de residuos sólidos.....	18
2.2. Procesamiento digital de señales	19
2.3. Software de procesamiento de voz	20
2.4. Contenedores autónomos	20
2.5. Sistemas de control	21
2.6. Sistemas embebidos	21
2.7. Sensores	22
2.8. Actuadores	23
2.9. Aprendizaje automático	23
2.10. Técnicas de procesamiento digital de voz	24

2.10.1. Programación dinámica.....	24
2.10.2. Modelos ocultos de Markov.....	25
2.10.3. Redes neuronales.....	25
2.11. Redes Neuronales Profundas (DNN).....	26
2.12. Vosk.....	26
3. ESTADO DEL ARTE.....	27
3.1. Software.....	27
3.2. Hardware.....	30
4. SELECCIÓN DE COMPONENTES ELECTRÓNICOS Y SOFTWARE.....	36
4.1. Elección de los componentes electrónicos.....	36
4.2. Elección del software.....	38
5. ALGORITMO Y CODIFICACIÓN DEL SISTEMA.....	41
5.1. Diagrama de flujo.....	41
5.2. Encuesta.....	41
5.3. Codificación del sistema.....	44
6. PRUEBAS DE FUNCIONAMIENTO.....	48
6.1. Prueba 1.....	48
6.2. Prueba 2.....	52
6.3. Prueba 3.....	54
6.4. Prueba 4.....	57
7. CONCLUSIONES.....	59
8. TRABAJOS FUTUROS.....	61
REFERENCIAS.....	62
ANEXOS.....	71

Lista de Figuras

Figura 1. Etapas del procesamiento digital de señales	19
Figura 2. Integración interna de un sensor	22
Figura 3. Diagrama de flujo del sistema	42
Figura 4. Proceso para la codificación del sistema de contenedores	44
Figura 5. Configuración de las entradas y salidas de sonido de la Raspberry Pi 3B+	45
Figura 6. Voluntario hombre para la prueba N°1	49
Figura 7. Voluntario mujer para la prueba N°1	49
Figura 8. Código de programación para los servomotores, prueba N°2	53
Figura 9. Conexión del servomotor y la Raspberry Pi 3b+, prueba N°2	53
Figura 10. Diagrama circuital para un servomotor y la Raspberry Pi 3B+, prueba N°2	54
Figura 11. Conexión de los servomotores y la Raspberry Pi 3b+, prueba N°3.....	55
Figura 12. Diagrama circuital de los 3 servomotores y la Raspberry Pi 3B+, prueba N°3.....	56
Figura 13. Montaje del sistema de contenedores controlados por voz, prueba N°4.	57
Figura 14. Vista trasera del sistema de contenedores	57
Figura 15. Apertura de la tapa de uno de los contenedores de basura	58

Lista de Tablas

Tabla 1. Resumen del estado del arte de software.	31
Tabla 2. Resumen del estado del arte de hardware.	35
Tabla 3. Matriz de ponderación de parámetros para los sistemas embebidos.....	36
Tabla 4. Matriz de comparación de los sistemas embebidos.	37
Tabla 5. Matriz de selección del sistema embebido.....	38
Tabla 6. Matriz de ponderación de parámetros para las herramientas.	38
Tabla 7. Matriz de comparación de las herramientas de procesamiento de voz.	39
Tabla 8. Matriz de selección la herramienta de procesamiento de voz.....	40
Tabla 9. Determinación del valor PxQ.....	43
Tabla 10. Cálculo de WER para la frase 1.	50
Tabla 11. Cálculo de WER para la frase 2.	50
Tabla 12. Cálculo de WER para la frase 3.	51
Tabla 13. Resultados prueba N°4.....	58

Lista de anexos

Anexo 1. Encuesta realizada a los estudiantes del Programa de Ingeniería Electrónica.	72
Anexo 2. Carta de Veolia, con los desechos que se puede reciclar y los que no.	76
Anexo 3. Codificación para el sistema de contenedores.	78
Anexo 4. Consentimiento informado, prueba N°1.	81
Anexo 5. Registro de los resultados de la prueba N°4.	83
Anexo 6. Manual de usuario para el sistema de contenedores.	85
Anexo 7. Manual e implementación técnica para el sistema de contenedores.	86

AGRADECIMIENTOS

Agradecemos primeramente a Dios, quien nos ha dado la fortaleza y la sabiduría para continuar con este propósito. Agradecemos a nuestras familias, por haber sido parte de este proceso, por apoyarnos en cada decisión tomada, por creer en nuestra capacidad de superarnos, por desearnos siempre lo mejor en este camino y por siempre estar para nosotras con mucha comprensión y amor. Ya para finalizar, agradecemos a los directores del proyecto, por tomar el riesgo de orientarnos y aconsejarnos cuando los necesitábamos, además, por tomarse el tiempo para compartir sus conocimientos y poder completar el desarrollo de esta investigación.

Dedicado a

Mis padres, Gloria y Leonardo.

Y a mi ahijado Emiliano.

Linned Suescun.

Mis padres, Marisol y Darío.

Y a mis hermanos Joan y Andrea.

Paola Nocua.

1. DESCRIPCIÓN DEL PROBLEMA

1.1. Planteamiento del problema

Los residuos sólidos son una constante en la vida de los seres humanos, cada vez que se termina con el uso de algún elemento para satisfacer alguna necesidad inmediatamente se busca desecharlo, debido a que pierde su utilidad. El problema se presenta cuando no se conoce cómo deshacerse de esto que se consideran desperdicios o restos que ya no representan un valor útil, esta falta de conocimiento conlleva al aumento de la contaminación ambiental. En el año 2020 se estimó que los residuos sólidos a nivel mundial se aproximaban a una cifra de 1.3 billones de toneladas anuales, estimando un aumento de 0.9 billones de toneladas para el año 2025, esperando un total de 2.2 billones de toneladas de residuos sólidos [1]. A partir de estas cifras se observa la necesidad de recurrir a estrategias donde las personas puedan aportar de manera significativa a esta situación ambiental actual. En Latinoamérica y el Caribe la cultura del reciclaje se hace presente solo en un 20% de los desechos, lo cual quiere decir que el resto terminan en disposición final, donde son confinados para evitar la contaminación, ya que se generan diariamente 541.000 toneladas de desechos en los sectores urbanos y se espera un aumento del 25% para el año 2050 [2].

El proceso de separación de residuos sólidos por medio de contenedores de basura es reconocido a nivel nacional. Sin embargo, esta actividad no siempre cumple con el objetivo de lograr que los residuos sean depositados en el contenedor adecuado debido al desinterés, la falta de conocimiento y de integración de la cultura del reciclaje, generando el hábito de mezclar los residuos entre sí, lo que puede provocar malos olores y la acumulación de desechos minimizando el proceso de reutilización. Estos factores impiden que exista un progreso más notorio de la educación del reciclaje en la sociedad. Para el caso específico de la ciudad de San José de

Cúcuta, donde se desarrolla la actual investigación, en la última década, se ha reportado que existe una participación del 30,5% en la labor de reciclaje [3].

Desde la ingeniería electrónica se han incluido diferentes soluciones de manera innovadora y dinámica para la correcta clasificación de residuos sólidos. Mediante las TIC, se han presentado diferentes aplicaciones móviles que permiten encontrar el punto ecológico más cercano y a su vez ampliando el conocimiento sobre la clasificación de residuos e identificación de los respectivos lugares para cada desperdicio, buscando interactuar con el usuario [4]. También se han propuesto diferentes plantas de procesamiento que reutilizan y convierten los desechos en algo nuevo. Existen prototipos con tecnologías aplicadas para clasificar residuos sólidos, los cuales usan sensores que detectan proximidad y llenado de cada contenedor, alertando al personal encargado y al usuario. Así mismo, el diseño de contenedores inteligentes que ayudan con la labor de sensibilización a los usuarios que pertenecen a un entorno estudiantil u otros entornos sociales sobre el cuidado del ambiente [5]. Teniendo en cuenta lo anteriormente descrito surge la siguiente pregunta de investigación:

¿Cómo desarrollar un sistema de contenedores controlado por voz que ayude en la separación de residuos sólidos?

1.2. Justificación

Según la corporación Punto Azul, la cual es una entidad que representa a industrias dedicadas al cuidado del medio ambiente en Colombia, en el país de 12 millones de toneladas de residuos generados al año solo se recicla el 17%, el resto terminan en los rellenos sanitarios, en los mares o en los ríos aumentando la problemática ambiental [6]. El 92.1% de los residuos sólidos que se generan en Norte de Santander terminan en depósitos sanitarios, y el 7.9% son reciclados por personas que participan en actividades convocadas por entidades como Arenorte y Corponor [7].

Es importante seguir fomentando estas técnicas y mejorar el porcentaje de participación ciudadana, a través de la motivación y la curiosidad, mitigando la cantidad de residuos sólidos que terminan en los rellenos sanitarios.

Con la construcción del sistema de contenedores propuesto se buscaba contribuir a la clasificación de residuos sólidos a partir de una estrategia tecnológica e innovadora. Teniendo en cuenta los Objetivos de Desarrollo Sostenible (ODS), el N°12 trata sobre la producción y el consumo responsable. Este objetivo tiene metas como: reducir la generación de residuos sólidos a través de estrategias como el reciclaje y la reutilización. También, racionalizar los residuos para evitar su liberación en el agua y en el suelo, minimizando los efectos adversos en el medio ambiente y la salud humana [8].

Este proyecto sirve como base para la comunidad científica en general y de la Universidad Francisco de Paula Santander para realizar futuras investigaciones. Además, el sistema puede ser modificado para utilizarse en diferentes aplicaciones específicas.

1.2.1. Beneficios Tecnológicos

Con la realización de este trabajo se espera contribuir al desarrollo de herramientas tecnológicas que permitan optimizar el uso de contenedores de basura de los puntos ecológicos tradicionales, aplicando el reconocimiento digital de voz. Así mismo, utilizar modelos basados en redes neuronales que identifiquen la palabra pronunciada por la persona que está haciendo uso del sistema de contenedores.

1.2.2. Beneficios Económicos

Con la realización de este proyecto se espera que la correcta clasificación de los residuos conlleve a que diferentes entidades los conviertan en nuevos productos, lo cual permite el ahorro de recursos.

1.2.3. Beneficios Sociales

- Enseñar a las personas los desechos que son reutilizables y los que no, contribuyendo a la cultura del reciclaje con el uso del prototipo de sistema de contenedores, evitando el incremento de residuos sólidos y los malos olores que pueden ser generados por mezclarlos, ya que no comparten el mismo tiempo de descomposición.
- Aportar una solución innovadora que hará que los diferentes lugares de la Universidad Francisco de Paula Santander donde sea aplicado el prototipo tengan visibilidad por querer estar a la vanguardia de nuevas estrategias para la clasificación de residuos.

1.2.4. Beneficios Institucionales

Aportar a la labor que se tiene como sociedad, a través de los conceptos y tecnologías aprendidas en el programa académico de Ingeniería Electrónica, brindando una asociación entre la responsabilidad social y los conocimientos adquiridos durante la carrera en la Universidad Francisco de Paula Santander.

1.3. Alcances

Como resultado del desarrollo de este trabajo de grado se obtuvo un sistema automatizado de contenedores para garantizar la correcta clasificación de los residuos sólidos. El diseño se realizó utilizando diferentes componentes electrónicos como sensores, actuadores y tarjetas microcontroladoras para el procesamiento digital de señales.

1.3.1. Tipo de proyecto

El trabajo de grado realizado corresponde a una investigación exploratoria basada en un diseño experimental que empleó el método científico y conocimientos adquiridos durante los cursos de formación profesional del programa académico de Ingeniería Electrónica.

1.4. Delimitaciones

- Las mediciones, y así mismo, la construcción del sistema de contenedores se realizó en los laboratorios del Departamento de Electricidad y Electrónica de la Universidad Francisco de Paula Santander, usando elementos tales como multímetros digitales y fuentes de alimentación.
- El desarrollo de este trabajo de grado comprometió un prototipo de un sistema de contenedores de residuos sólidos controlados por voz. Los contenedores tendrían una capacidad de llenado de 12 Litros. Además, incluiría elementos electrónicos como sensores, actuadores y tarjetas microcontroladoras. Por otra parte, en este trabajo de grado no se creó, ni se mejoraron bases de datos o sistemas de gestión de información.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un sistema automatizado de contenedores controlado por voz que ayude a la correcta clasificación de los residuos sólidos.

1.5.2. Objetivos Específicos

- Realizar una búsqueda del estado del arte en temas relacionados con el procesamiento digital de señales, contenedores autónomos y herramientas para el procesamiento digital de voz.
- Seleccionar los componentes electrónicos para el diseño del sistema de contenedores, teniendo en cuenta la compatibilidad entre ellos y con el software.
- Diseñar el algoritmo y la codificación del sistema.
- Aplicar pruebas de funcionamiento al sistema de contenedores para identificar posibles errores.

2. MARCO TEÓRICO

2.1. Generación de residuos sólidos

Los residuos sólidos generados diariamente van creciendo cada vez más con un ritmo acelerado, lo que conlleva a que se presenten problemas ambientales mundiales como la incorrecta separación de los residuos sólidos y que más del 90% de los residuos reutilizables terminen en vertederos o en quemas a cielo abierto. Los países más afectados son aquellos en vías de desarrollo, y dentro de ellos, las familias de bajos ingresos que viven cerca de los vertederos son más propensas a sufrir enfermedades porque tienen contacto directo con todo tipo de residuos, o inhalan de cerca el humo de las quemas indiscriminadas [9], [10].

La ineficiente separación de los residuos sólidos afectará también al planeta de manera económica al no darle una oportunidad a los desechos reciclables. Según cifras del informe “*What a Waste 2.0*” del Banco Mundial en el año 2020 se generaron 2010 millones de toneladas de residuos y de estos solo el 33% fueron desechados sin afectar al medio ambiente. Si se sigue minimizando el problema, se espera que para el año 2100 los desechos generados aumenten a 12 millones de toneladas diarias, ya que la tasa de reciclaje es de 16% en 91 países estudiados a nivel mundial [9], [10], [11].

El banco mundial está asociado con países de todo el mundo para apoyar y dar soluciones financieras y eficaces en programas para mejorar la gestión de residuos sólidos. Desde el punto de vista económico, vale la pena invertir en este aspecto para fomentar un medio ambiente sostenible a partir de sistemas óptimos y apropiados para dicha tarea, ya que son menos costosos que manejar el impacto negativo de seguir con una mala administración de los desechos [11].

2.2. Procesamiento digital de señales

El procesamiento digital de señales es un área de la ingeniería y la ciencia que permite la manipulación de señales. Comprende el uso de algoritmos, matemáticas y todas las técnicas utilizadas para el tratamiento de señales digitales [12]. Ha presentado un desarrollo rápido en los últimos tiempos debido a los avances tecnológicos en el campo de la informática como lo son los circuitos integrados, por ejemplo, los MSI, LSI y VLSI que son escalas de integración (media, larga y muy larga) permitiendo un avance en los softwares y hardwares digitales con el objetivo de construir sistemas digitales [13].

Hoy en día el procesamiento digital de señales está involucrado en diferentes aplicaciones muy importantes en áreas como telecomunicaciones, procesamiento del habla o voz, electrónica de consumo, sistemas de biomedicina, procesamiento de imágenes, electrónica militar, aplicaciones industriales, electrónica automotriz y aeroespacial. Dentro de la electrónica de consumo se pueden encontrar dispositivos de uso personal y no comercial como televisores, celulares, computadores, CD-DVD [14]. El procesamiento digital está compuesto por varias etapas como se muestra en la Figura 1.



Figura 1. Etapas del procesamiento digital de señales. Elaboración propia.

El procesamiento de voz es una de las aplicaciones del tratamiento o procesamiento digital de una señal. También conocido como procesamiento del habla, incluye aplicaciones como la traducción de texto a voz, el reconocimiento, la síntesis, y el dictado de voz [14].

En el reconocimiento de voz se utilizan señales acústicas donde están las palabras, las sílabas y los fonemas, los cuales son representaciones en valores discretos en un tiempo determinado.

Los reconocedores automáticos de voz enfocan su técnica en un análisis de comparación de las señales con patrones que han sido almacenados con el fin de identificar la información dentro de la señal analizada [15].

2.3. Software de procesamiento de voz

El procesamiento de voz se puede aplicar en diferentes herramientas como puede ser MATLAB, el cual contiene comandos para soluciones complejas, así mismo, tiene herramientas aplicadas al procesamiento de audio, análisis de voz y medición acústica, una de estas herramientas es el Audio Toolbox [16]. Además, tiene funciones de procesamiento digital de señales de audio como son las transformadas de Fourier, funciones para la manipulación de un audio, ya sea la lectura o reproducción de este, y su manipulación en diferentes frecuencias [17].

Otro software conocido para el procesamiento digital de señales es *Python*, el cual cuenta con diferentes herramientas para el procesamiento de voz, algunas de estas son: Apiai, Assemblyai, Google Cloud Speech, Pocketsphinx, SpeechRecognition, Wit y Watson Developer Cloud. Este software también trabaja reconociendo frases u oraciones que contienen ruido, trabajan con acceso a micrófono con librerías PyAudio [18].

2.4. Contenedores autónomos

Los contenedores o depósitos de reciclaje son los puntos estratégicos donde se deposita el residuo sólido dependiendo de su clasificación. Sin embargo, la creación o los modelos de estos contenedores trabajan de manera autónoma, es decir, por medio de sensores, microcontroladores, inteligencia artificial, entre otros. Estos contenedores autónomos han sido implementados en diferentes países. Por ejemplo, en Perú desarrollaron un prototipo que interactúa con la persona que deposita el residuo. Este modelo trabaja con inteligencia artificial teniendo como objetivo la simplificación del reciclaje [19].

2.5. Sistemas de control

Los sistemas de control son una agrupación de diferentes dispositivos conectados entre sí, que cumplen con la función de mantener y/o regular un valor para una variable de salida manteniendo una secuencia de ejecución sin ningún error y sin importar los cambios en su señal de entrada [20]. Es importante resaltar que los tipos de sistemas de control dependen del tipo de respuesta, y pueden ser de lazo abierto o de lazo cerrado. El sistema de lazo abierto no depende de un valor a la entrada para poder cumplir su función, lo cual lo hace más estable; mientras que uno de lazo cerrado si necesita un valor de referencia para poder satisfacer el valor de salida [21].

También es posible clasificar los sistemas de control dependiendo del tipo de señales que se utilicen, como sistema de control analógico o sistema de control digital. Los sistemas analógicos operan con señales analógicas las cuales toman valores continuos que pueden variar en el tiempo, por ejemplo, una señal de voz [22]. Por otro lado, los sistemas digitales están diseñados para trabajar con combinaciones de señales de tiempo continuo y tiempo discreto. Las señales de entrada son convertidas, mediante un proceso de muestro, en señales en tiempo discreto, y con estas señales opera el sistema [23], [24].

2.6. Sistemas embebidos

Un sistema embebido es un circuito electrónico que está diseñado para cumplir con una tarea específica dentro de un dispositivo en tiempo real [25]. Pueden cumplir las tareas de un sistema computacional por contener un microprocesador en el cual guardan algoritmos y secuencias de su programación; pero propiamente no lo son, ya que cada uno está destinado para un único propósito y tienen recursos limitados [25], [26].

Dentro de sus características más importantes está el bajo consumo de energía, debido a su tamaño compacto y su portabilidad, pueden estar incluidos en tareas sencillas que funcionen con

baterías. Por otra parte, su microprocesador interno debe cumplir con una prioridad en sus instrucciones, para poder ser ejecutadas en este orden repetidas veces, por lo tanto, el lenguaje más usado es el Ensamblador o el C++ [26], [27].

El uso de los sistemas embebidos cada vez es más común, pues todo lo que hoy en día usamos cuenta con uno. Las aplicaciones son mayormente en todos los dispositivos que cuenten con electricidad, entre ellos están los electrodomésticos, sistemas de seguridad y protección, la industria automotriz, dispositivos móviles o de uso personal, entre otros [25], [27]. También es importante resaltar que el sistema embebido puede ser de uso externo e ir interconectado con otros sistemas empleados para la adquisición de datos, tales como sensores.

2.7. Sensores

Un sensor es un dispositivo electrónico con la función de recibir y transmitir información, es decir, convierte las variables físicas que recibe en su entrada en señales eléctricas que transmite en su salida. Actualmente estos elementos son muy importantes para los sistemas de medición [28]. En la figura 2 se muestra la integración interna del sensor, el cual contiene todos los procesos en un solo integrado y luego la información es llevada al dispositivo final para visualizarse o para realizar una acción programada [29].

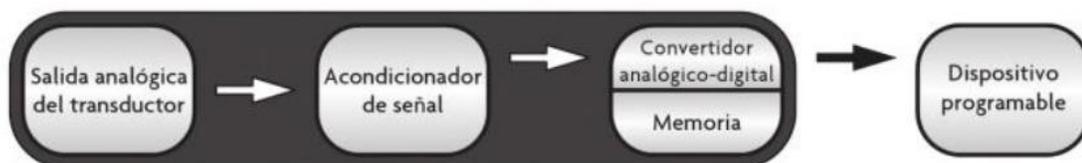


Figura 2. Integración interna de un sensor. Tomado de [29].

Los sensores son solo dispositivos de entrada y funcionan con la ayuda de transductores. Su principal diferencia es que los transductores solo cambian el dominio de la variable, y los sensores son los intermediarios entre el mundo real y el instrumento, interpretando la señal y

funcionando como sistema de control o para visualizar datos de manera analógica o digital [28], [29].

Dentro de los procesos de automatización y control existen una gran variedad de sensores dependiendo de las aplicaciones y del tipo de variable que se desea medir. Para poder seleccionar el que más se adecue a los requerimientos del sistema se necesita tener en cuenta factores como la sensibilidad, el rango de medida, el tiempo de respuesta, la exactitud, entre otros [29].

2.8. Actuadores

Los actuadores son dispositivos que convierten la señal eléctrica recibida por los sensores en otra forma de energía, esto con el fin de ejecutar una acción determinada e interactuar con el entorno [30]. Dependiendo de la función de cada elemento, estos pueden convertir su señal de entrada en una magnitud de fuerza, aceleración, cambio de posición o velocidad de algún otro objeto final al que esté conectado [29].

Se pueden considerar como el paso intermedio entre el procesamiento de la señal que ofrece el sensor y el proceso o acción del objeto [30]. Dependiendo del tipo de energía que usan se pueden clasificar en actuadores, neumáticos, hidráulicos y eléctricos [29].

Dentro de los dispositivos actuadores este trabajo de investigación se centrará en los motores eléctricos de corriente directa (DC). Estos elementos convierten la energía eléctrica en energía mecánica para poder hacer un movimiento predeterminado en el rotor, dependiendo de la cantidad de voltaje DC que se le suministra puede variar la velocidad en su movimiento rotacional [29].

2.9. Aprendizaje automático

El aprendizaje automático (Machine Learning) es una rama de la Inteligencia Artificial (IA) y, además, es una ciencia de la computación que tiene como objetivo programar ordenadores para

que simulen el aprendizaje humano a través de diferentes técnicas. Los ordenadores logran el aprendizaje mediante la observación de múltiples conjuntos de datos, creando modelos a partir de estos, los cuales pueden predecir o clasificar un evento [31], [32]. El aprendizaje supervisado es cuando los datos de entrenamiento corresponden a pares de entrada y salida, pues la salida que debe predecirse o clasificarse se basa en las características que se asignen en la entrada, es decir, siempre que tenga ese conjunto de datos en la entrada el modelo entrega dicha salida [33]. El entrenamiento no supervisado por el contrario no necesita conocimientos previos para aprender, ya que se dedica a revisar patrones permitiendo sacar sus propias agrupaciones [34].

El ML se ha aplicado a diferentes disciplinas como aplicaciones médicas y biomédicas, economía, juegos, detecciones de fraude, robótica, reconocimiento del habla, entre otros [35]. Siendo esta última aplicación la que se desarrolla en este trabajo de grado, permitiendo identificar las palabras dichas por el usuario.

2.10. Técnicas de procesamiento digital de voz

En los estudios sobre el reconocimiento de voz se pueden encontrar diferentes técnicas que ayudan a la clasificación de patrones de voz y el investigador es libre de elegir la que mejor le convenga para el desarrollo de sus proyectos. Dentro de las más conocidas y con mejores resultados se encuentra, la programación dinámica, los modelos ocultos de Markov y las redes neuronales [36].

2.10.1. Programación dinámica

En la programación dinámica se comparan patrones (o plantillas de entrada) con los patrones que están almacenados en el sistema. Para llevar a cabo el procedimiento, se parametriza la señal recibida y se analiza convirtiéndola en coeficientes espectrales, los cuales se comparan con los patrones establecidos. Esta técnica es utilizada en reconocimiento del

habla aislada y continua, pero puede presentar problemas en coincidir con la plantilla almacenada, debido a la duración de palabras y el ritmo de la pronunciación [37].

2.10.2. Modelos ocultos de Markov

Esta técnica es conocida como un modelo de procesos estocásticos, es decir, generan secuencias aleatorias de estados dependiendo de probabilidades. Funciona determinando parámetros ocultos a partir de parámetros observados, que son los tipos de estados del modelo [38]. El primer estado, es una cadena de Markov caracterizada por probabilidades de transición y el segundo estado, dependiendo de la distribución de probabilidades del estado se crean emisiones observables [39].

2.10.3. Redes neuronales

Las redes neuronales son un conjunto interconectado de elementos, unidades o nodos, los cuales basan su funcionalidad en la neurona animal. Estas redes son usadas para análisis estadístico y el modelado de datos, los cuales se aplican en problemas de clasificación o pronóstico, por ejemplo, en reconocimiento de voz, de imágenes, de caracteres textuales y de temas de práctica humana como: ayudas diagnósticas, estudio geológico de petróleo y predicciones de mercado financiero [40].

Las redes neuronales artificiales (ANN, sus siglas en inglés) son ideales para trabajar el aprendizaje automático de complejidad, como potenciar servicios de reconocimiento de voz y recomendar videos a usuarios en plataformas de streaming. Algunas de sus primeras arquitecturas son: el perceptrón, compuesta por una capa de Unidad Lógica de Umbral (TLU, por sus siglas en inglés); y el perceptrón multicapa y retro propagación, compuesta por una capa de entrada (de paso), de una o más capas ocultas de TLU y una capa de salida. Otra arquitectura de red neuronal es la convolucional (CNN, por sus siglas en inglés) la cual surge

del estudio de la corteza visual del cerebro, está compuesta por una estructura jerárquica, y es utilizada principalmente para el reconocimiento de imágenes. También, están las Redes Neuronales Profundas (DNN, por sus siglas en inglés) las cuales están compuestas por una pila profunda de capas ocultas [31].

2.11. Redes Neuronales Profundas (DNN)

Las Redes Neuronales Profundas (DNN, por sus siglas en inglés) son un tipo de Red Neuronal con una arquitectura de varias capas ocultas, que tienen como objetivo analizar los conjuntos de datos grandes, que resultan complejos y admiten modificaciones no lineales en forma matricial o tensorial. A medida que la cantidad de datos crece el modelo basado en DNN puede procesar la información de manera más eficaz. Dentro de las aplicaciones más destacadas de este tipo de red están el procesamiento de imágenes y de voz, la generación de recomendaciones según lo visto o escuchado recientemente, los asistentes digitales, el piloto automático y los bots [41], [42].

2.12. Vosk

Vosk es una herramienta utilizada para el reconocimiento de voz, con 20 idiomas admitidos entre ellos: inglés, francés, alemán, italiano, filipino y ruso. *Vosk* tiene modelos pequeños para tareas que pueden ejecutarse en equipos con capacidades de procesamiento limitadas como: Raspberry Pi y teléfonos inteligentes, para este trabajo se utilizó el modelo pequeño en español. También, tiene modelos grandes los cuales se ejecutan en servidores y se utilizan en aplicaciones de alta precisión [43]. La arquitectura del modelo consta de una combinación de una Red Neuronal Factorizada de Retardo de Tiempo (TDNN-F, por sus siglas en inglés), y una Red de Memoria a Corto y Largo Plazo (LSTM, por sus siglas en inglés) [44], [45].

3. ESTADO DEL ARTE

En esta sección se presenta una revisión del estado del arte de artículos y trabajos de grado relacionados con la implementación de redes neuronales con fines de clasificación. En la primera parte se presentan los resúmenes de diferentes tipos de redes neuronales, pero solo evaluadas a nivel computacional, mostrando sus datos más relevantes y los resultados obtenidos en la Tabla 1, permitiendo hacer una comparación entre los clasificadores y el software usado. En la segunda parte se presentan los resúmenes de redes neuronales implementadas en diferentes sistemas embebidos, sus resultados obtenidos y los datos relevantes se encuentran en la Tabla 2, logrando hacer una comparación entre las diferentes placas de desarrollo y el software usado.

3.1. Software

Para poder clasificar audios en [46] hacen uso de una Red Neuronal Convolutiva (CNN, por sus siglas en inglés) para extraer las características necesarias e introducir etiquetas de árbol. Estas características fueron 3 conjuntos de imágenes, coeficientes Cepstrales de Gammatone, coeficientes de Mel y filtros logarítmicos; pero cada uno tenía presencia y ausencia de ruido, así la red se pudo entrenar descubriendo patrones invariantes con imágenes en dos dimensiones. Hicieron varias pruebas de entrenamiento donde fusionaron diferentes características, pero la que mejores resultados arrojó es la CNN-Fusion6, estas 6 características se convirtieron en la entrada de la Máquina de Vectores de Soporte (SVM, por sus siglas en inglés), la cual realizaba la clasificación. Para el entrenamiento utilizaron un dataset con 3026 archivos el cual permitió clasificar 19 sonidos de lugares específicos de la ciudad. En la verificación del sistema obtuvieron un accuracy del 96.60% para la mejor combinación de características.

En [47] se desarrolló una herramienta computacional basada en SVM para la identificación biométrica de 16 sujetos usando los sonidos captados de la respiración con y sin habla. Los

autores recolectaron los datos de los usuarios usando un estetoscopio ubicado en el cuello para así obtener una onda de sonido óptima, en estado neutral o después de alguna actividad física. Para los entrenamientos tuvieron en cuenta las características en el dominio del tiempo y la frecuencia de los 3360 audios, usando el algoritmo de YIN y los Coeficientes Cepstrales de la Frecuencia de Mel (MFCC, por sus siglas en inglés) respectivamente. Los autores compararon su modelo con otros en los que implementaban CNN, y observaron que obtienen resultados prometedores considerando los dos escenarios de respiración (natural o profunda) y habla, logrando un 98.75% de accuracy en la clasificación de los sujetos de prueba.

Con el fin de clasificar la correcta pronunciación de las 29 letras del alfabeto árabe en [48] se propuso usar una versión modificada de la CNN denominada AlexNet. Este tipo de red neuronal extrae las características de manera automática convirtiendo los audios originales en espectrogramas, filtrando las características relevantes para la clasificación de los audios. Para definir la correcta pronunciación, realizaron una clasificación binaria con audios de hombres, mujeres y niños, nativos y no nativos, recolectando así un total de 8120 audios para los experimentos incluyendo los de aumento de datos. Esta clasificación la llevaron a cabo de dos maneras, con y sin aumento en los datos, donde verificaron que arrojaba mejores resultados con el aumento de datos y con el algoritmo Random Split, logrando un accuracy de 99.14%.

En [49] presentaron un estudio acerca del uso de dos clasificadores con redes neuronales: uno simple y otro binario. Primero definieron un método para la adquisición de muestras de voz, eligiendo un micrófono de garganta. Posteriormente, crearon los comandos de voz aislados en portugués (Brasil), los cuales constaban de: dígitos del 0 al 9, y dos palabras (Aceptar y Cancelar). Como método de extracción de características utilizaron la Predicción Lineal Perceptual (PLP). Luego, para los clasificadores: simple (una red neuronal) y binario (doce redes

neuronales binarias) utilizaron la red neuronal perceptrón multicapa (MLP, por sus siglas en inglés). Observaron que la red neuronal simple siempre tuvo un menor desempeño con respecto a las redes neuronales binarias, dejando en claro qué sistema fue más eficiente, finalmente obtuvieron un Hit rate del 95,42% para el clasificador con redes neuronales binarias, y uno del 91,88% para el clasificador de red neuronal simple.

Con la intención de detectar la dificultad respiratoria, en [50] aplicaron el procesamiento digital de señales y las redes neuronales implementando un sistema que se compone de: Adquisición de señales de voz, Método computacional, Red neuronal, Arquitectura sistema cliente servidor y un sistema web, captura de señales y transmisión. En la adquisición realizaron la grabación de voz con dos sonidos vocálicos ‘a’ y ‘o’, en condiciones sanas y patológicas. Utilizaron la aplicación móvil ‘Sonómetro’ para verificar que el ruido ambiente estuviera por debajo de 40 dB. Emplearon el software Matlab para realizar los análisis de las grabaciones, haciendo una normalización, posteriormente, un ventaneo y cálculos como: Parámetros acústicos, Parámetros espectrales, Parámetros temporales y Parámetros estadísticos. En la red neuronal usaron la topología MLP, con entrenamiento supervisado, el cual funcionaba realizando una comparación de la salida obtenida por la red con la salida deseada. Después presentaron la arquitectura de la red que constaba de cinco entradas, una capa intermedia (con diez neuronas) y una capa de salida, consiguiendo una salida binaria que representa si había dificultad respiratoria o no. Luego, para la arquitectura del sistema cliente servidor utilizaron un dispositivo con navegador web para el cliente y el software Matlab para el servidor. Adicionalmente, en el sistema agregaron dos servidores más, uno ejecutaba la interfaz web, la grabación de voz y la visualización de los resultados para el cliente, y el otro servidor se encargaba de almacenar datos y señales de voz. En los resultados determinaron elegir la vocal ‘a’ en mujeres y la vocal ‘o’ en

hombres, después de un análisis en gráficos de dispersión, obtuvieron un accuracy del 97,62% para la vocal 'a' y un 96,7% para la vocal 'o'.

En [51] presentaron el reconocimiento de fonemas por medio de redes CNN. En la investigación se discuten dos enfoques: enfoque directo y un enfoque propuesto. En el enfoque directo se hace una sola fase de clasificación, y se obtienen los fonemas directamente desde la entrada. En el enfoque propuesto, utilizan varias fases de clasificación, tomando en cuenta vocales, semivocales, explosivas, etc. En la primera fase clasifican los fonemas vocálicos (señales periódicas y semi periódicas) y los fonemas constantes (señales no periódicas). Luego, en la segunda fase estos fonemas se clasifican en vocales, semivocales y diptongos, para fonemas vocálicos; y en fricativas, nasales y oclusivas para los fonemas constantes. En la última fase, las clases de la segunda fase se alinean a los fonemas correspondientes. La base de datos utilizada fue la TIMIT, la cual consta de 39 fonemas que han sido agrupados y reducidos. Estos fonemas son convertidos en espectrograma de Mel y son utilizados para entrenamiento y prueba de la red. En las pruebas realizadas por los autores obtuvieron una accuracy del 57% para el enfoque directo y una del 61% para el enfoque propuesto.

3.2. Hardware

En [52] propusieron una red de sensores inalámbricos con el fin de clasificar y monitorear sonidos urbanos por medio de placas electrónicas Raspberry pi 3B, las cuales se encargaban de grabar el audio, procesarlo y clasificarlo para el posterior envío de notificación si se detectaba un sonido de fondo no cotidiano. Para reducir el consumo de potencia constante se creó una etapa de pre-detección, con el fin de que se envíe la alerta sólo cuando detecte un ruido diferente del tráfico habitual. Los entrenamientos los realizaron grabando sus propios audios para para las categorías: tráfico, sirenas, bocinas fuertes y vehículos ruidosos. El sistema estaba funcionando

Tabla 1. Resumen del estado del arte de software.

Autores	Ref.	Año	Clasificador	Técnica de extracción de característica	Dataset	Cantidad de muestras	Clasificación	Lenguaje de programación o plataforma	Resultados registrados
Phan et. al	[46]	2017	SVM	CNN	LITIS Rouen	3026	19 sonidos	Matlab	Accuracy 96.60%
Tran y Tsai	[47]	2019	SVM	Algoritmo de YIN, MFCC.	Propio	3360	16 sujetos	Python	Accuracy 98.75%
Ziafat et. al	[48]	2021	AlexNet	-	Arabic Quick Quran Teacher Language Hub Propio	8120	29 letras del alfabeto árabe	Matlab	Accuracy 99.14%
Ribeiro et. al	[49]	2018	MLP	PLP	Propio	1800	12 comandos de voz	-	Hit Rate 95.40%
Fernández y Ramos	[50]	2021	MLP	Cálculo de los parámetros de perturbación jitter y shimmer, espectrales, temporales y estadísticos.	Propio	1040	2 sonidos vocálicos	Matlab	Accuracy 97.20%
Smit y Al-Assimi	[51]	2020	CNN	-	TMIT	6300	39 fonemas	Python	Accuracy 61%

todo el tiempo, pero solo se activaba el clasificador cuando consideraba que ese sonido era diferente del tráfico, de esta manera redujeron el consumo de potencia de 3.70 W a 1.75 W, dándole más tiempo de vida útil a la batería.

Por medio de sistemas embebidos en [53] se quería vigilar y monitorizar los entornos urbanos en tiempo real, para ello propusieron una red de sensores inalámbricos entrenados para poder clasificar y detectar 10 sonidos urbanos. Por cada fotograma de audio de 1 segundo se hizo la extracción de las características dentro de las que estaban, los primeros 12 MFCC, el contraste, el centroide, el ancho de banda y el roll off espectral, la raíz media cuadrada (RMS) y el cruce por cero. Estas características fueron extraídas mediante el software Python y formaban un vector que ingresa al clasificador de Vecinos más Cercanos (K-NN, por sus siglas en inglés). El modelo ya entrenado se probó en el sistema embebido Raspberry Pi 3B+ para conocer sus resultados y el tiempo que tarda en realizar la extracción y la clasificación del sonido ingresado, obteniendo así un accuracy de 73,50% con un tiempo de ejecución total de 9.54 segundos.

Con el fin de identificar 4 tipos de notas musicales provenientes del piano en [54] se propuso implementar una CNN modificada, que se denomina Red Neuronal convolucional de Picos (CSN, por sus siglas en inglés) que sea capaz de simular las señales recibidas por la cóclea biológica. Para el entrenamiento de la red neuronal se incluyeron 50 audios de cada una de las notas musicales en vivo teniendo en cuenta su frecuencia fundamental, ya que esta red neuronal busca patrones de picos en las bandas de frecuencias de las señales de audios para poder hacer su clasificación. La información de los audios era transmitida a la FPGA ML507 para procesar los datos y finalmente se enviaban a un ordenador para el entrenamiento de la red, donde primeramente notaron que los resultados no eran muy buenos y decidieron incluir una capa

Winner-Take-All (WTA) entrenada a la salida de la capa convolucional y con esto obtuvieron un Hit Rate de 97.50%.

En [55] presentaron la implementación de un sistema de reconocimiento de palabras habladas aisladas, utilizando una red neuronal en una placa integrada. Para la base de datos usaron las muestras que contienen los dígitos del 0 al 9 de TIDIGIT. Posteriormente, en el entrenamiento usaron el algoritmo de Red Neuronal Retro Propagación (BPNN, por sus siglas en inglés) utilizando el software Matlab, de la red entrenada obtuvieron vectores de peso y el sesgo. Luego, llevaron a cabo la prueba en la Raspberry Pi usando el software Octave, donde exportaron los vectores de peso para ser usados en una red tipo Feed Forward. Finalmente, obtuvieron un accuracy del 100% teniendo en cuenta que es reconocimiento de voz dependiente del hablante.

En [56] implementaron un método de software y hardware de bajo costo para dirigir una silla de ruedas robótica con el propósito de ofrecer autodependencia y libertad a las personas que las usan. El sistema propuesto constaba de dos etapas: dispositivos de hardware y el desarrollo del software. En la primera etapa, se enfocaron en los dispositivos que utilizaron para el control de los movimientos de la silla de ruedas, dentro de estos incluyeron los siguientes: teléfono inteligente, motores eléctricos CC, baterías, modelos relé y el microcontrolador Raspberry Pi 4. En la segunda etapa trataron el diseño y la implementación del software para el control de la silla de ruedas con cinco comandos de voz Yes, No, Left, Right y Stop. En esa última etapa tuvieron en cuenta: el desarrollo de una aplicación móvil, en el software Flutter; El desarrollo del modelo de reconocimiento de voz, donde se realizó el proceso de extracción de características con el método MFCC; El desarrollo del modelo CNN, utilizando la estructura de red en red (NIN, sus siglas en inglés). En los resultados obtuvieron un accuracy del 87,2% para el comando Yes,

83,6% para el comando Stop, 83% para el comando No, 82,8% para el comando Left y para el comando Right 77%.

En [57] presentan una silla de ruedas inteligente con control de voz utilizando CNN. En este proyecto emplearon los comandos: Stop, Go, Left y Right con la intención de ayudar a personas con movilidad reducida. Los datos fueron seleccionados y descargados desde Google sin procesamiento y en formato WAV con duración de 1 segundo. El sistema funcionaba de la siguiente forma: primero, el usuario graba el comando de voz con un micrófono; segundo, la voz se procesa dentro de una placa microcontroladora Raspberry Pi 3B+ utilizando el método MFCC para la extracción de características y luego, realizaba la fase de clasificación con la CNN. En los resultados obtuvieron un accuracy del 95,30% siendo considerado fiable y eficiente.

Tabla 2. Resumen del estado del arte de hardware.

Autores	Ref.	Año	Clasificador	Técnica de extracción de características	Lenguaje de programación o plataforma	Sistema embebido	Adquisición de datos	Clasificación	Resultados registrados
Arce et. al	[52]	2021	CNN	-	Python	Raspberry Pi 3B +	Dataset Propio	4 sonidos urbanos	Consumo 1.75 W
Da Silva et. al	[53]	2019	K-NN	MFCC	Python	Raspberry Pi 3B+	Dataset ESC-10	10 sonidos urbanos	Accuracy 73.50%
Cerezuela et. al	[54]	2015	CSN	-	Matlab	FPGA ML507	Generados por Matlab	4 notas musicales de piano eléctrico	Hit Rate 97.50%
Pantage y Alex	[55]	2017	MLP	MFCC	Octave	Raspberry Pi 2	Dataset TIDIGITIS	Palabras aisladas	Accuracy 100%
Bakouri et. al	[56]	2022	CNN	-	Flutter	Raspberry Pi 4	Dataset	5 comandos de voz	Accuracy 87.20%
Sharifuddin et. al	[57]	2019	CNN	MFCC	Python	Raspberry Pi 3B+	Google	4 comandos de voz	Accuracy 95.30%

4. SELECCIÓN DE COMPONENTES ELECTRÓNICOS Y SOFTWARE

Con el fin de desarrollar el segundo objetivo propuesto, se seleccionan los componentes electrónicos óptimos para el desarrollo del diseño y que se ajusten a los requerimientos que se plantearon teniendo en cuenta sus características técnicas. Además, se seleccionó el software en el que se trabajó la codificación del sistema, teniendo en cuenta la compatibilidad con los dispositivos electrónicos.

4.1. Elección de los componentes electrónicos

Se debe seleccionar el sistema embebido encargado de ejecutar las tareas de control programadas. Para lo anterior se tienen en cuenta características como: microcontrolador (A), consumo de corriente en los pines (B), lenguajes soportados (C), memoria RAM (D) y precio (E). El rango de importancia se establece como: 10 mucho más importante, 5 más importante, 1 igual de importante, 1/5 menos importante y 1/10 muchos menos importante. La ponderación de estos parámetros se puede observar en la Tabla 3, obteniendo como resultado que el microcontrolador y el consumo de corriente tienen la mayor importancia en los criterios de selección. El Peso del Parámetro (PP) es el resultado de dividir cada dato sobre la suma de todos los datos y será usado posteriormente.

Tabla 3. Matriz de ponderación de parámetros para los sistemas embebidos.

	A	B	C	D	E	SUMA	PP	RES
A	x	5	1	1	10	17	0.359	1
B	1/5	x	5	5	5	15.2	0.321	2
C	1	1/5	x	1/5	5	6.4	0.135	4
D	1	1/5	5	x	1	7.2	0.152	3
E	1/10	1/5	1/5	1	x	1.5	0.032	5
Total						47.3		

Seguidamente, se realiza la matriz de comparación entre las opciones de sistemas embebidos que se están considerando, como lo son: Raspberry Pi 3B+, Arduino Mega y Pocket Beagle. La

matriz de comparación se puede observar en la Tabla 4, donde cada placa es confrontada dependiendo de cada parámetro impuesto y el rango de importancia es el mismo de la Tabla 3.

Tabla 4. Matriz de comparación de los sistemas embebidos.

Microcontrolador (A)					
	Raspberry Pi 3B+	Arduino Mega	Pocket Beagle	SUMA	PO
Raspberry Pi 3B+	x	10	5	15	0.591
Arduino Mega	1/10	x	1/10	0.2	0.008
Pocket Beagle	1/5	10	x	10.2	0.402
Total				25.4	
Consumo (B)					
	Raspberry Pi 3B+	Arduino Mega	Pocket Beagle	SUMA	PO
Raspberry Pi 3B+	x	1/10	1/5	0.3	0.015
Arduino Mega	10	x	5	15	0.732
Pocket Beagle	5	1/5	x	5.2	0.254
Total				20.5	
Memoria RAM (C)					
	Raspberry Pi 3B+	Arduino Mega	Pocket Beagle	SUMA	PO
Raspberry Pi 3B+	x	10	5	15	0.591
Arduino Mega	1/10	x	1/10	0.2	0.008
Pocket Beagle	1/5	10	x	10.2	0.402
Total				25.4	
Lenguajes soportados (D)					
	Raspberry Pi 3B+	Arduino Mega	Pocket Beagle	SUMA	PO
Raspberry Pi 3B+	x	10	5	15	0.591
Arduino Mega	1/10	x	1/10	0.2	0.008
Pocket Beagle	1/5	10	x	10.2	0.402
Total				25.4	
Precio (E)					
	Raspberry Pi 3B+	Arduino Mega	Pocket Beagle	SUMA	PO
Raspberry Pi 3B+	x	1/5	1	1.2	0.097
Arduino Mega	5	x	5	10	0.806
Pocket Beagle	1	1/5	x	1.2	0.097
Total				12.4	

Con lo anterior, se obtiene como resultado el coeficiente Peso de la Opción (PO) que finalmente será usado junto con los resultados del PP de la Tabla 3 para seleccionar el sistema embebido que obtenga el mayor Factor de Priorización (FP). Lo anterior se puede observar en la

Tabla 5, presentando como resultado que la Raspberry Pi 3B+ es la más indicada para los requerimientos propuestos del sistema teniendo en cuenta los parámetros mencionados anteriormente con un 39%.

Tabla 5. Matriz de selección del sistema embebido.

	A	B	C	D	E	FP
	PP*PO	PP*PO	PP*PO	PP*PO	PP*PO	
Raspberry Pi 3B+	0.212	0.005	0.080	0.090	0.003	0.390
Arduino Mega	0.003	0.235	0.001	0.001	0.026	0.266
Pocket Beagle	0.144	0.082	0.054	0.061	0.003	0.344

4.2. Elección del software

Se debe seleccionar la herramienta encargada de llevar a cabo el procesamiento de las señales de voz. Para esto se tienen en cuenta características como: idioma español (A), código abierto (B), función en tiempo real (C), lenguajes soportados (D) y compatibilidad con la Raspberry Pi 3B+ (E). El rango de importancia se establece como en la elección de los componentes electrónicos. La ponderación de estos parámetros se puede observar en la Tabla 6, obteniendo como resultado que el idioma español y la función en tiempo real tienen la mayor importancia en los criterios de selección. El Peso del Parámetro (PP) es el resultado de dividir cada dato sobre la suma de todos los datos y será usado posteriormente.

Tabla 6. Matriz de ponderación de parámetros para las herramientas.

	A	B	C	D	E	SUMA	PP	RES
A	x	10	5	5	1	21	0.513	1
B	1/10	x	1	5	1	7.1	0.174	3
C	1/5	1	x	5	1	7.2	0.176	2
D	1/5	1/5	1/5	x	1	1.6	0.039	5
E	1	1	1	1	x	4	0.098	4
Total						40.9		

Seguidamente, se realiza la matriz de comparación entre las opciones de herramientas transcripción de voz a texto que se están considerando, como lo son: Vosk, DeepSpeech y Pico

Voice. La matriz de comparación se puede observar en la Tabla 7, donde cada herramienta es confrontada dependiendo de cada parámetro impuesto.

Tabla 7. Matriz de comparación de las herramientas de procesamiento de voz.

Idioma español (A)					
	Vosk	DeepSpeech	Pico Voice	SUMA	PO
Vosk	x	10	10	20	0.900
DeepSpeech	1/10	x	1	1.1	0.050
Pico Voice	1/10	1	x	1.1	0.050
Total				22.2	
Código abierto (B)					
	Vosk	DeepSpeech	Pico Voice	SUMA	PO
Vosk	x	1	5	6	0.484
DeepSpeech	1	x	5	6	0.484
Pico Voice	1/5	1/5	x	0.4	0.032
Total				12.4	
Función en tiempo real (C)					
	Vosk	DeepSpeech	Pico Voice	SUMA	PO
Vosk	x	1	1	2	0.333
DeepSpeech	1	x	1	2	0.333
Pico Voice	1	1	x	2	0.333
Total				6	
Lenguajes soportados (D)					
	Vosk	DeepSpeech	Pico Voice	SUMA	PO
Vosk	x	1	1	2	0.333
DeepSpeech	1	x	1	2	0.333
Pico Voice	1	1	x	2	0.333
Total				6	
Compatibilidad con Raspberry Pi 3B+ (E)					
	Vosk	DeepSpeech	Pico Voice	SUMA	PO
Vosk	x	10	1	11	0.495
DeepSpeech	1/10	x	1/10	0.2	0.009
Pico Voice	1	10	x	11	0.495
Total				22.2	

Con lo anterior, se obtiene como resultado el coeficiente Peso de la Opción (PO) que finalmente será usado junto con los resultados del PP de la Tabla 6 para seleccionar la herramienta que obtenga el mayor Factor de Priorización (FP). Lo anterior se puede observar en

la Tabla 8, presentando como resultado que es la herramienta Vosk es la más indicada para los requerimientos propuestos del sistema teniendo en cuenta los parámetros mencionados anteriormente con un 66,7%.

Tabla 8. Matriz de selección la herramienta de procesamiento de voz.

	A	B	C	D	E	FP
	PP*PO	PP*PO	PP*PO	PP*PO	PP*PO	
Vosk	0.462	0.084	0.059	0.013	0.049	0.667
DeepSpeech	0.026	0.084	0.059	0.013	0.000	0.182
Pico Voice	0.026	0.006	0.059	0.013	0.049	0.153

Para la realización del código del sistema se eligió el software *Spyder* el cual es un entorno de desarrollo que permite la creación, edición, análisis, visualización y ejecución interactiva de scripts de *Python* [58]. La elección de *Spyder* se basó en su compatibilidad con el hardware Raspberry Pi 3B+ que ejecuta los archivos *.py* con los que trabaja la herramienta *Vosk* utilizada para el reconocimiento de voz. Además, se tuvo en cuenta la información recopilada de los trabajos, estudios y artículos del capítulo 3, donde *Python* es el lenguaje con mayor implementación.

5. ALGORITMO Y CODIFICACIÓN DEL SISTEMA

En esta sección se presenta el diseño del algoritmo y la codificación del sistema, los cuales se desarrollaron en 3 fases. La primera es la realización de un diagrama de flujo donde se exponen los principales procesos que ejecuta el sistema y sus posibles opciones de funcionamiento. La segunda es la realización de una encuesta para seleccionar las palabras que dan la apertura de cada uno de los 3 contenedores. Y finalmente, la tercera es la descripción de las partes más importantes del código de programación.

5.1. Diagrama de flujo

En la Figura 3 se describe el funcionamiento del sistema de contenedores. Este inicia escuchando al usuario, si el micrófono capta la señal de voz, se procesa, se transcribe y se guarda en una lista de *Python*; en caso de no ser así el programa sigue escuchando. Las palabras guardadas en estas listas se comparan con otras listas predefinidas para cada uno de los contenedores, si existe igualdad, el contenedor correspondiente se abre y se cierra después de 5 segundos. Si no existe igualdad se compara con la palabra ‘detener’ para finalizar el programa, o de lo contrario el sistema sigue escuchando los comandos.

5.2. Encuesta

Con el fin de seleccionar las listas de palabras para los 3 contenedores, se llevó a cabo una encuesta, para conocer como los estudiantes de ingeniería electrónica relacionaban los residuos sólidos al momento de desecharlos, teniendo en cuenta la Resolución No. 2184 de 2019 que establece los colores verde (residuos orgánicos aprovechables), blanco (residuos reciclables) y negro (residuos no aprovechables) para los contenedores de basura [59].

Se utilizó la ecuación (1) para determinar el tamaño de la muestra (n) en una población finita. En este caso, el tamaño de población (N) fue representado por los 600 estudiantes que pertenecen

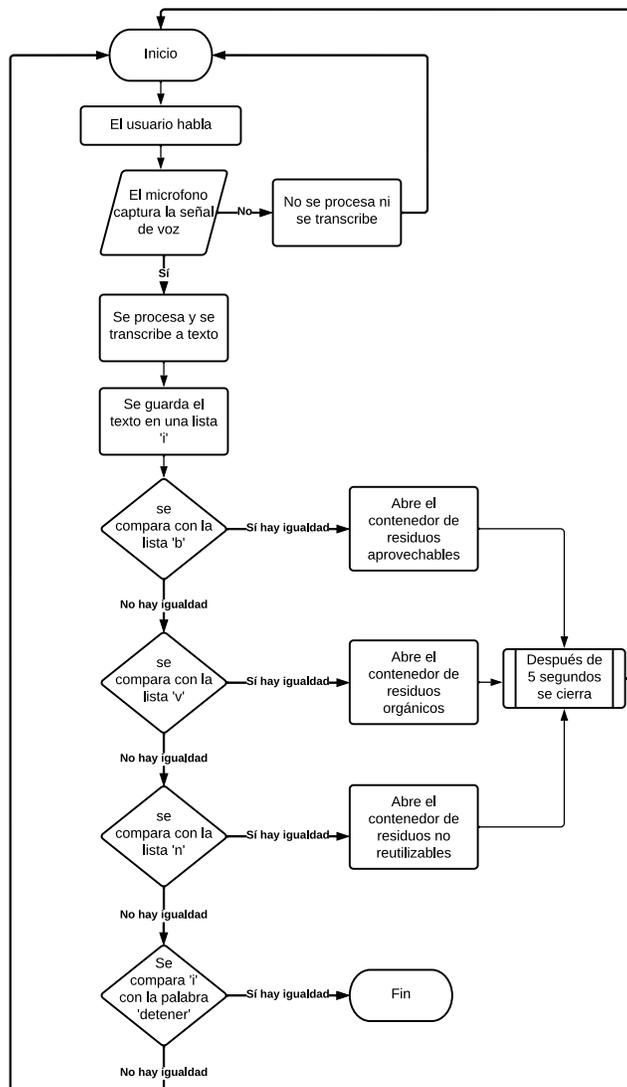


Figura 3. Diagrama de flujo del sistema. Elaboración propia.

al programa de ingeniería electrónica de la Universidad Francisco de Paula Santander. Para los parámetros de probabilidad a favor (P) y probabilidad en contra (Q) se utilizó una probabilidad del 50% es decir, la máxima incertidumbre como se muestra en la Tabla 9. El nivel de confianza es del 95%, por lo tanto, $z = 1,96$ y el margen de error (e) es del 5%. Estos valores se han establecido debido a que la encuesta no presenta antecedentes [60].

$$n = \frac{z^2 * N * P * Q}{e^2 * (N - 1) + z^2 * P * Q} \quad (1)$$

Reemplazando los datos en la ecuación (1) se obtiene:

$$n = \frac{1.96^2 * (600) * (0.50) * (0.50)}{(0.05)^2 * (600 - 1) + (1.96)^2 * (0.50) * (0.50)}$$

$$n = 234$$

Para la encuesta se esperaba que, de 600 estudiantes, al menos 234 la respondieran para tener resultados satisfactorios. Finalmente, se obtuvo 250 respuestas.

Tabla 9. Determinación del valor PxQ. Tomado de [60].

P	Q	PxQ
0.00	1.00	0.0000
0.05	0.95	0.0475
0.10	0.90	0.0900
0.15	0.85	0.1275
0.20	0.80	0.1600
0.25	0.75	0.1875
0.30	0.70	0.2100
0.35	0.65	0.2275
0.40	0.60	0.2400
0.45	0.55	0.2475
0.50	0.50	0.2500
0.55	0.45	0.2475
0.60	0.40	0.2400
0.65	0.35	0.2275
0.70	0.30	0.2100
0.75	0.25	0.1875
0.80	0.20	0.1600
0.85	0.15	0.1275
0.90	0.10	0.0900
0.95	0.05	0.0475
1.00	0.00	0.0000

Se realizaron 11 preguntas en la encuesta y los resultados arrojaron los siguientes porcentajes para las palabras seleccionadas: para el contenedor blanco, envases (56,8%), papeles (92%), latas (90%), tapas (99,6%), y residuos reciclables (80%); para el contenedor verde, desperdicios (54,8%), cultivo (48,4%) y residuos orgánicos (67,6%); y para el contenedor negro, papeles

usados (41,2%), recipientes sucios (46,8%) y residuos no reutilizables (59,6%). Los resultados y preguntas de la encuesta se pueden observar más a detalle en el Anexo 1, con las demás opciones de respuesta y sus porcentajes.

Además, de estas palabras en el sistema también se incluyeron palabras propias de los materiales que se reciclan y de los que no, basadas en la información que suministró la empresa de recolección de residuos Veolia (ver Anexo 2).

Finalmente, las listas para cada uno de los contenedores quedaron de la siguiente manera: para el contenedor blanco: envases, latas, tapas, papeles, plástico, vidrio, cartón, madera y residuos reciclables. Para el contenedor verde: desperdicios, residuos cultivo, y residuos orgánicos. Y para el contenedor negro: recipientes sucios, papeles usados, espejos, cerámica y residuos no reutilizables.

5.3. Codificación del sistema

En la Figura 4 se presenta el proceso de las partes más importantes de la codificaciones realizada para el control del sistema de contenedores, y seguidamente se describen cada una de ellas.

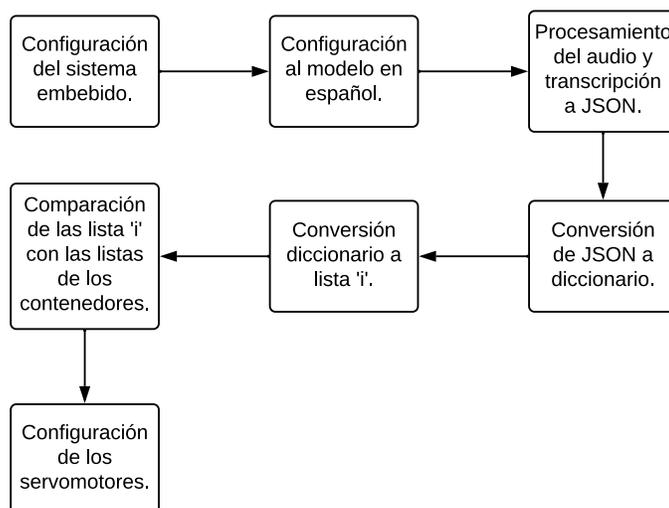


Figura 4. Proceso para la codificación del sistema de contenedores. Elaboración Propia.

Inicialmente se hace la configuración del sistema embebido para la captura de las señales de voz. Primero, se conecta un micrófono de puerto Jack a la entrada USB de la Raspberry, por medio de un adaptador. Segundo, fue necesario actualizar los repositorios en la consola utilizando el comando “*sudo apt update*” e instalando la librería “*pulseaudio*”. Tercero, se reinicia la Raspberry y se proceden a elegir los perfiles de dispositivo, donde se estableció el puerto USB como entrada (ver Figura 5).

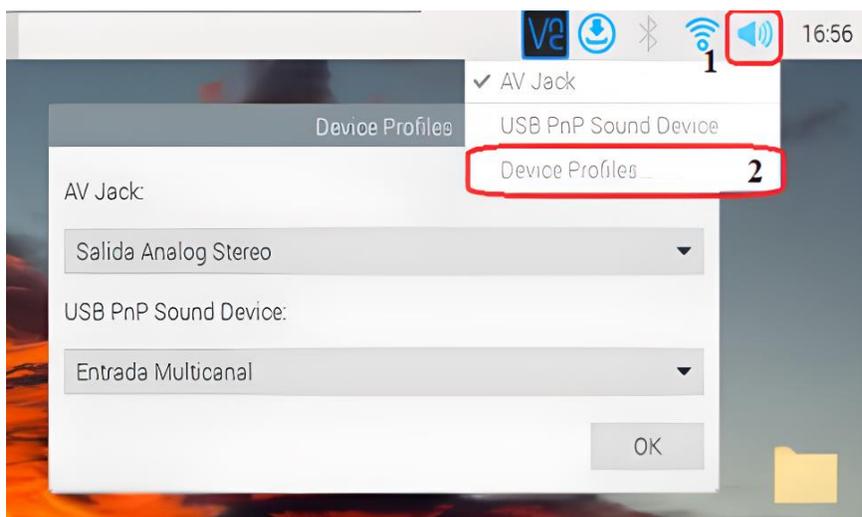


Figura 5. Configuración de las entradas y salidas de sonido de la Raspberry Pi 3B+. Elaboración propia.

Para el código del sistema (ver Anexo 3), se tomó como base el *script* de ejemplo ‘*test_microphone.py*’ de la herramienta *Vosk* el cual se modificó y se configuró con el fin de utilizar el modelo en español de 39 MB para el reconocimiento de voz. En el código con el método *RawInputStream* se procesó el audio utilizando una interfaz sin bloqueo [61]. El módulo *KaldiRecognizer* se encargó de transcribir el audio a texto, en un orden de instrucciones, en las cuales: Primero, recuperó los datos capturados por el micrófono utilizando una estructura de datos; segundo, utiliza el método *AcceptWaveform* que tiene como argumento los datos recuperados y transcribe la información que se capta; y tercero, se imprime un resultado parcial y un resultado final [62].

La salida del sistema es un archivo Java Script Object Notation (JSON). Este formato intercambia datos entre varios lenguajes de programación, su estructura puede estar definida como: una lista ordenada de valores (vectores o listas), y pares (clave/valor) [63]. Teniendo en cuenta lo presentado en [64], se utilizó el método *loads* para convertir en diccionarios de *Python* los pares generados por el JSON. Así mismo, los valores de los diccionarios se guardan en una lista. Posteriormente, fueron creadas tres listas con las palabras que representan cada uno de los contenedores, las cuales fueron comparadas con la lista de los valores del diccionario.

Se eligieron los servomotores MG946R para la apertura de la tapa de los contenedores de basura, teniendo en cuenta las especificaciones técnicas de la fuerza de torque, donde es capaz de soportar 10,5 kg/cm con una voltaje de 4.8 V, lo que es suficiente para levantar las tapas. Su rango de movimiento es de 0° a 180° y trabajan con un periodo de pulso de 20 ms. Se utilizó la Modulación por Ancho de Pulso (PWM, sus siglas en inglés) de la Raspberry Pi 3B+ para el control del ángulo de los servomotores por medio de la librería *GPIO*. Según la información del servomotor para la posición de 0° el pulso debe ser de 0.5 ms, para la posición de 90° de 1.5 ms y para la posición de 180° de 2.5 ms [65]. En el proyecto se utilizaron las posiciones de 0° (apertura) y de 108° (cierre) como barrido de ángulo para el alcance entre la hélice y la tapa de los contenedores, por lo tanto, en el código se utilizó el método *ChangeDutyCycle* cuyo argumento se define con la ecuación (2) y la ecuación (3) [66]. En la ecuación (2) da como resultado el valor del pulso (x), el cual se aplicó en la ecuación (3) para obtener el valor del ciclo de trabajo (DC) que se usó en el código. De esta manera, se utilizaron los valores redondeados al entero inferior más próximo, siendo 2 para 0° y 8 para 108° siguiendo la relación lineal entre el ciclo de trabajo y el ángulo (α), lo cual se definió a ensayo y error.

$$x = \frac{2}{180} * \alpha + 0.5 \quad (2)$$

$$x_1 = \frac{2}{180} * 0^\circ + 0.5 = 0.5 \text{ ms}$$

$$x_2 = \frac{2}{180} * 108^\circ + 0.5 = 1.7 \text{ ms}$$

$$DC = x * \frac{\text{frecuencia}}{10} \quad (3)$$

$$DC_1 = 0.5 * \frac{50}{10} = 2.4$$

$$DC_2 = 1.7 * \frac{50}{10} = 8.4$$

6. PRUEBAS DE FUNCIONAMIENTO

Para el cumplimiento del último objetivo se propusieron 4 pruebas de funcionamiento con el fin de verificar el comportamiento de cada una de las partes que componen el sistema de contenedores, por separado y completo, como se describe a continuación. En la primera, se probó la herramienta *Vosk* en el lenguaje español. En la segunda, se verificó que los servomotores funcionaran de manera individual con un código corto. En la tercera, se probaron los 3 servomotores y el código de programación con el control de los contenedores. En la cuarta prueba se hizo el montaje y se evaluó todo el sistema de contenedores para identificar posibles fallas.

6.1. Prueba 1

En la prueba N°1 se evaluó la herramienta *Vosk* en cuanto al reconocimiento de palabras. Para definir la cantidad de personas y de audios se tuvo en cuenta lo utilizado en [67] para evaluar el modelo acústico entrenado previamente. Por lo tanto, se contó con la colaboración de veinte voluntarios (10 mujeres y 10 hombres) entre los 20 y 40 años, los cuales autorizaron su participación en los experimentos por medio de la firma de un consentimiento informado (ver Anexo 4). Las pruebas se realizaron en los laboratorios del Departamento de Electricidad y Electrónica y durante ellas debían pronunciar 3 frases de referencia, consiguiendo 60 señales de audio para procesar. Las evidencias de esta prueba se pueden observar en la Figura 6 y Figura 7.

La métrica seleccionada para evaluar la precisión del modelo fue la Tasa de Error de Palabra (WER, por sus siglas en inglés), la cual ha sido utilizada en la evaluación de sistemas de reconocimiento automático de voz como: Microsoft Speech Api, Google Speech Api y Sphinx-4. También, se aplica para hacer comparaciones de rendimiento entre sistemas como Kaldi y la API de Google [68]. WER se encarga de verificar el número incorrecto de palabras en una



Figura 6. Voluntario hombre para la prueba N°1. Elaboración propia.



Figura 7. Voluntario mujer para la prueba N°1. Elaboración propia.

transcripción usando tres factores: Inserción (I), son las palabras agregadas; Eliminación (D), son las palabras no detectadas o pérdidas; y Sustitución (S), son las palabras reemplazadas. La ecuación (4) define como calcular la métrica dividiendo la suma de los tres parámetros anteriores con el número total de palabras (N) de la frase de referencia [69].

$$WER = \frac{S + D + I}{N} \quad (4)$$

La frase 1 era: “en el contenedor de residuos reciclables podemos depositar: tapas, latas, papeles, envases y plástico”, es decir, N= 14. Se tabularon los resultados de cada uno de los 20 participantes para las inserciones, eliminaciones y sustituciones. Usando (4) se obtuvo un WER

por persona y se hizo el promedio para calcular el WER de la frase 1, el cual fue de 8.21% como se puede observar en la Tabla 10.

Tabla 10. Cálculo de WER para la frase 1.

N°	I	D	S	WER_Persona (%)
1	0	0	1	7,142857143
2	0	0	3	21,42857143
3	0	0	1	7,142857143
4	0	0	1	7,142857143
5	1	2	0	21,42857143
6	0	0	2	14,28571429
7	1	0	2	21,42857143
8	0	0	1	7,142857143
9	0	0	1	7,142857143
10	0	0	1	7,142857143
11	0	0	1	7,142857143
12	0	0	1	7,142857143
13	0	0	0	0
14	0	0	0	0
15	0	0	1	7,142857143
16	0	0	0	0
17	0	0	1	7,142857143
18	0	0	0	0
19	0	0	1	7,142857143
20	0	0	1	7,142857143
WER_Frase 1 (%)				8,2142857

La frase 2 era: “en el contenedor de residuos orgánicos podemos depositar residuos de cultivo y desperdicios”, con N = 13. Se tabularon los resultados por persona con los que se calculó el promedio del WER para la frase 2, que fue de 1.15% como se muestra en la Tabla 11.

Tabla 11. Cálculo de WER para la frase 2.

N°	I	D	S	WER_Persona (%)
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	1	0	7,692307692
10	0	0	0	0
11	0	0	0	0
12	0	0	1	7,692307692
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	1	7,692307692
20	0	0	0	0
WER_Frase 2 (%)	1,153846154			

Por último, la frase 3 era: “en el contenedor de residuos no reutilizables podemos depositar recipientes sucios, papeles usados, espejos y cerámica”, es decir, con N = 16. Al igual que en las frases anteriores se calculó el promedio de WER para la frase 3, el cual fue de 6.87% como se muestra en la Tabla 12.

Tabla 12. Cálculo de WER para la frase 3.

N°	I	D	S	WER_Persona (%)
1	0	0	2	12,5
2	0	0	1	6,25
3	0	0	2	12,5
4	0	0	1	6,25
5	0	0	2	12,5
6	0	0	1	6,25
7	0	0	1	6,25
8	1	0	1	12,5
9	0	0	2	12,5
10	0	0	1	6,25
11	0	0	2	12,5
12	0	0	0	0

13	1	0	1	12,5
14	0	0	1	6,25
15	0	0	0	0
16	0	0	0	0
17	0	0	1	6,25
18	0	0	0	0
19	0	0	1	6,25
20	0	0	0	0
WER_Frase 3 (%)				6,875

Finalmente, con el cálculo de los errores de las 3 frases, se hace el promedio para obtener el WER de *Vosk* en español, el cual fue de 5.41%. Esto indica que es una herramienta de buena calidad y es apta para usarse, ya que se encuentra entre el 5% y 10% [70].

En uno de los participantes (participante 13) el modelo se probó en condiciones de ruido molesto, pero mostró mejores resultados que en otros participantes que no presentaron dicha interferencia. Esto se debe a la vocalización y al volumen de voz de cada persona, pudiendo afectar o no al modelo. Además, no se tuvo en cuenta las muletillas que tenían los voluntarios al leer, ya que el modelo si las identificaba, pero no eran parte del estudio.

6.2. Prueba 2

Se probaron individualmente los servomotores MG946R para comprobar su funcionamiento, compatibilidad con la Raspberry y que cumplieran con las especificaciones que se requerían para el sistema de contenedores. Esta prueba se realizó con un código corto y sencillo en el software *Spyder* de la Raspberry Pi 3B+ (ver Figura 8), el cual cambia el estado del servomotor de 0° a 108° con el ciclo de trabajo, como se explicó en la codificación del sistema, de manera automática cada 5 segundos.

```

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM) #Modo BCM
GPIO.setup(5,GPIO.OUT) #El pin 5 configurado como salida
servo = GPIO.PWM(5,50) #Declaracion de variable que albergara el pin (servo) (50)
                        #y la velocidad de pulsos por segundo que se enviaran
servo.start(0)
time.sleep(2)
print("iniciando")
try:
    while True:
        servo.ChangeDutyCycle(2)
        time.sleep(5)
        servo.ChangeDutyCycle(0)
        time.sleep(0.5)
        print("Contenedor Abierto")
        servo.ChangeDutyCycle(8)
        print("Contenedor Cerrado")
        time.sleep(5)
        servo.ChangeDutyCycle(0)
        time.sleep(0.5)
        servo.ChangeDutyCycle(0)
except KeyboardInterrupt: #interrupcion de teclado para limpiar los pines
    servo.stop() #detiene el servomotor
    GPIO.cleanup()#limpia los pines

```

Figura 8. Código de programación para los servomotores, prueba N°2. Elaboración propia.

La Figura 9 presenta la conexión de los servomotores, en ella se puede observar la Raspberry Pi 3B+ conectada a su cable de alimentación, al pin de comunicación GPIO 5 y con un cable HDMI a la pantalla para visualizar el código. El servomotor fue alimentado con una fuente de poder de 5 V, además, se utilizó una protoboard para conectar los pines de tierra (GND) del servomotor, la Raspberry y la fuente de poder. El diagrama circuital se puede observar en la Figura 10.

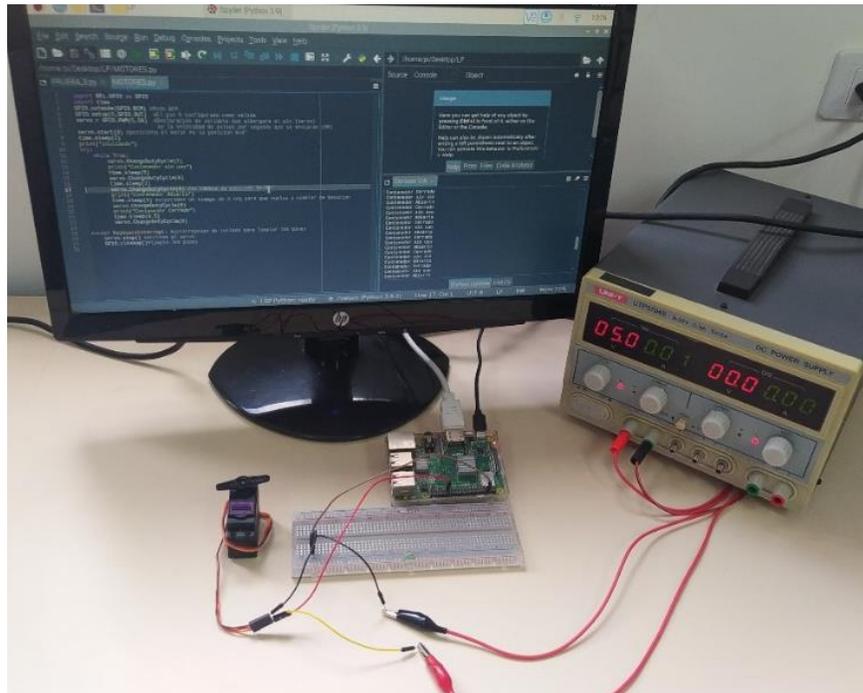


Figura 9. Conexión del servomotor y la Raspberry Pi 3b+, prueba N°2. Elaboración propia.

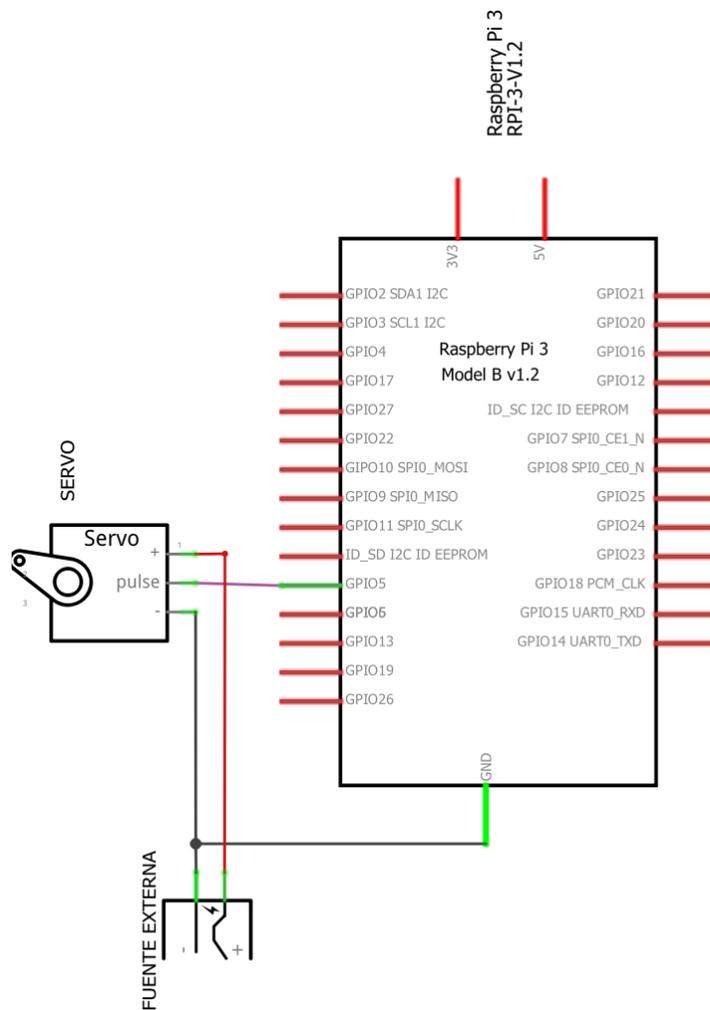


Figura 10. Diagrama circuital para un servomotor y la Raspberry Pi 3B+, prueba N°2. Elaboración propia.

Con esta prueba se pudo calcular el consumo de los servomotores, tomando en cuenta las especificaciones de su fabricante [65]. Se utilizó una fuente externa con voltaje fijo de 5 V y corriente variable, de esta manera se observó que dependiendo de la posición podía variar entre 0.01 A y 0.70 A.

6.3. Prueba 3

Se probó el código de programación que controla los servomotores, es decir, con las listas para cada uno de los contenedores. Para ello, se hace la conexión de los 3 servomotores a la Raspberry Pi 3B+ (ver figura 11). Al igual que en la prueba N°2 la Raspberry Pi 3B+ es

conectada a su cable de alimentación y al cable HDMI para visualizar el código. Para la conexión de comunicación se usaron los pines GPIO 17, GPIO 27 y GPIO 22 que corresponden a los contenedores de residuos reciclables, contenedor de residuos orgánicos y contenedor de residuos no reutilizables respectivamente. Los servomotores fueron alimentados con una fuente de poder de 5 V, además, nuevamente se utilizó la conexión de los pines de tierra (GND) de la prueba N°2. El diagrama circuital se puede observar en la Figura 12.

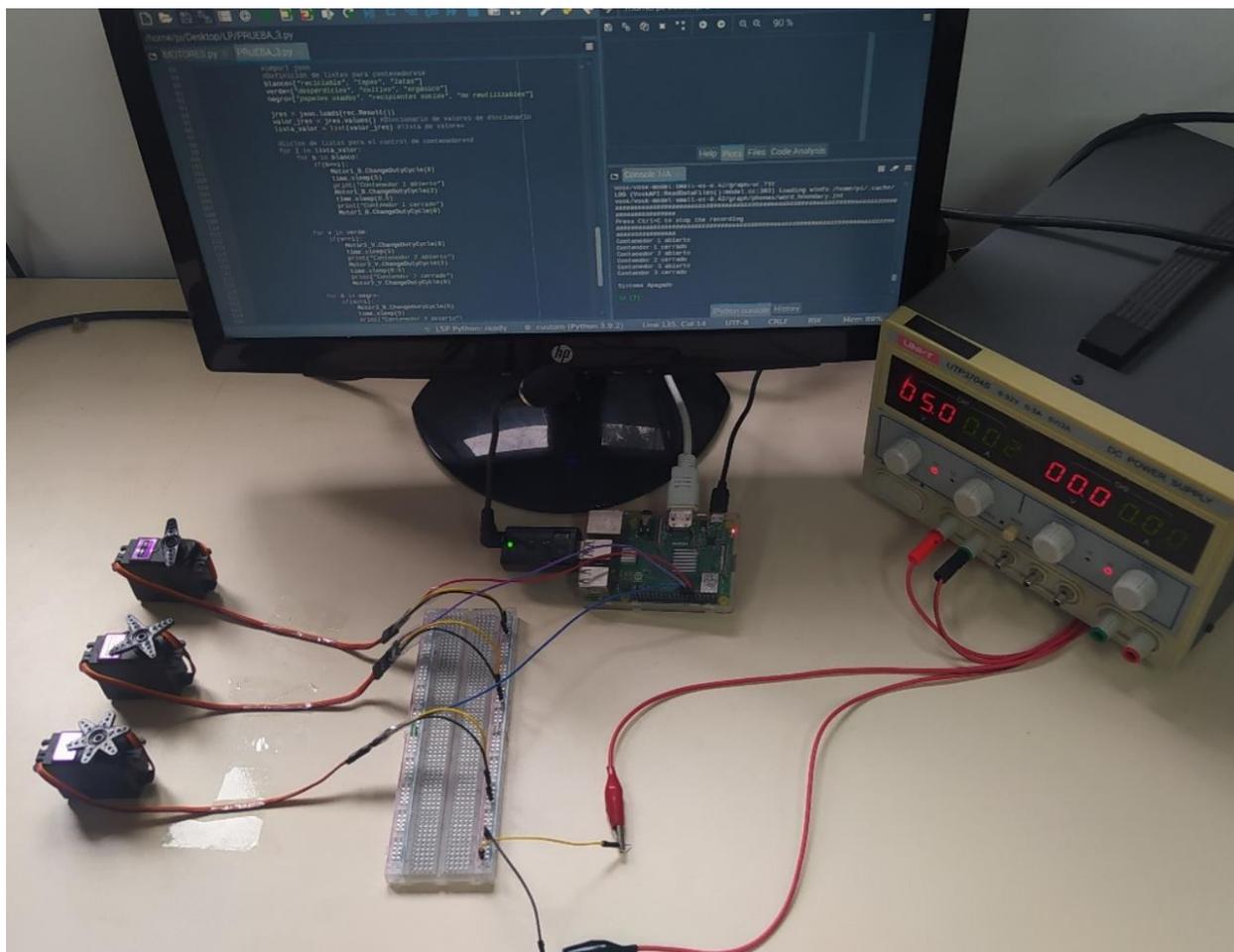


Figura 11. Conexión de los servomotores y la Raspberry Pi 3b+, prueba N°3. Elaboración propia.

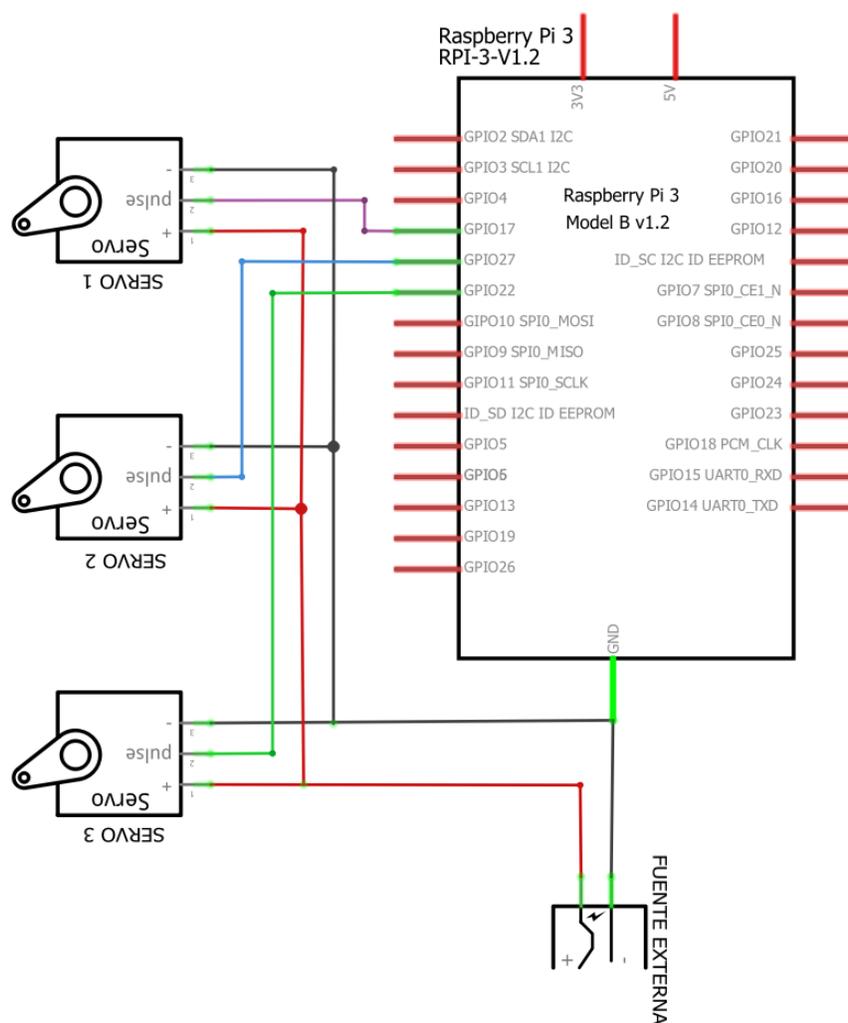


Figura 12. Diagrama circuital de los 3 servomotores y la Raspberry Pi 3B+, prueba N°3. Elaboración propia.

Con esta prueba se pudo calcular el consumo de los servomotores juntos. Aclarando que 2 servomotores no se abren al mismo tiempo, ya que así se ha hecho la programación del sistema. Se utilizó una fuente externa con voltaje fijo de 5 V y corriente variable, de esta manera se observó que dependiendo de la posición de abierto o cerrado podía variar entre 0.02 A y 0.70 A.

En las pruebas N°2 y N°3 se pudo observar que cuando las tapas de los contenedores se encuentran cerradas, no se presentan cambios en la corriente de los servomotores, por su parte, durante el proceso de apertura de las tapas, la corriente varía en los rangos mencionados previamente y esto es debido al trabajo que realizan los servomotores.

6.4. Prueba 4

En la prueba N°4 se evaluó todo el sistema de contenedores, para ello se construyó el prototipo con capacidad de llenado de 12 Litros (ver Figura 13). Primero, los servomotores fueron adheridos en la parte trasera de los contenedores justo debajo de la tapa (ver figura 14), y se modificó la hélice que traían estos, aumentando su tamaño a 6 cm, para mayor alcance en la apertura de la tapa (ver Figura 15). Segundo, se realizaron las conexiones en paralelo para los servomotores haciendo punto común con la Raspberry Pi 3B+ y la fuente externa para los pines de alimentación positiva y tierra (GND), de esta manera todos tenían el mismo voltaje y la corriente era variable según la posición como se observó en la prueba N°3. El código utilizado para esta prueba se puede observar en el Anexo 3. Para más información sobre su uso y su implementación ver el Anexo 6 y Anexo 7.



Figura 13. Montaje del sistema de contenedores controlados por voz, prueba N°4. Elaboración propia.



Figura 14. Vista trasera del sistema de contenedores. Elaboración propia.



Figura 15. Apertura de la tapa de uno de los contenedores de basura. Elaboración propia.

Para evaluar el sistema completo se realizó una prueba la cual consistió en dos partes. En la primera de ellas, las 2 investigadoras pronunciaron dos veces cada palabra de cada lista de los contenedores, esperando el tiempo de apertura y cierre de estos, y una vez se completaba una lista se pasaba a la siguiente. Por otro lado, en la segunda parte se generaron 2 listas nuevas a partir de la combinación aleatoria de la totalidad de palabras de las 3 listas de los contenedores, y las investigadoras pronunciaron estas palabras una única vez. Los resultados de estas pruebas se registraron de la siguiente forma: cuando se abría el contenedor indicado se marcaba con una ‘x’ en la opción detectó (D), de lo contrario se señalaba la opción no detectó (ND) (ver Anexo 5). Finalmente, se calcula la proporción del número de aciertos contra el total de palabras y se obtiene un porcentaje de aciertos de 92.64% como se observa en la Tabla 13.

Tabla 13. Resultados prueba N°4.

Palabras detectadas (D)	Palabras no detectadas (ND)	Total, de palabras	Porcentaje de aciertos
126	10	136	92.64%

7. CONCLUSIONES

En este trabajo se presenta un sistema de contenedores controlado por voz con capacidad de llenado de 12 Litros, el cual fue construido empleando una Raspberry Pi 3B+ y utilizando el lenguaje de programación *Python* V.3.9.2. Para seleccionar este sistema embebido, se aplicó una matriz de selección con el fin de determinar, a partir de diferentes parámetros, qué dispositivo se ajustaba mejor al diseño del sistema de contenedores y al cumplimiento de los objetivos. De esta manera se seleccionó la Raspberry Pi 3B+ con un Factor de Priorización del 39%, la cual se encarga de recibir la señal de voz, procesarla y enviar la orden a los servomotores. Así mismo, se optó por usar el lenguaje de programación *Python* V.3.9.2 en el entorno de desarrollo de *Spyder*, ya que, a partir de la revisión del estado del arte de diferentes publicaciones y trabajos de grado relacionados con Redes Neuronales utilizadas para la clasificación de audio, se pudo observar que *Python* es un lenguaje de programación popular en estas aplicaciones.

El *Script* que controla el sistema basa su funcionamiento en la herramienta *Vosk* y en el uso de librerías que permiten convertir las señales de voz en listas de *Python*. Así mismo, este controla la apertura y el cierre de los contenedores por medio de una comparación de términos entre las señales de voz y las listas preestablecidas de los contenedores. Para la elaboración de estas listas, se consultó con la empresa Veolia, además, se aplicó una encuesta para determinar las palabras más populares utilizadas por los estudiantes del Programa de Ingeniería Electrónica al momento de desechar ciertos residuos sólidos. Por otro lado, teniendo en cuenta la versatilidad de la herramienta *Vosk*, el código presentado en este trabajo puede servir de base para el desarrollo de otras aplicaciones en las que se emplee el reconocimiento de voz.

Para verificar el correcto funcionamiento de la herramienta *Vosk*, se realizaron pruebas en las cuales la métrica utilizada fue WER. En dichas pruebas el modelo en español de la herramienta

obtuvo un porcentaje del 5.41%, lo cual demuestra que este es óptimo para su implementación en el sistema de contenedores. Además de esto, se debe tener en cuenta la vocalización de los usuarios, ya que si estos presentan problemas con algunos fonemas las transcripciones no serán las adecuadas.

Con la finalidad de identificar posibles errores en el funcionamiento del sistema, se realizaron pruebas en las cuales se registraba el número de palabras detectadas y no detectadas. Se obtuvo un porcentaje de aciertos del 92.64%, lo que indica que el prototipo tiene una alta probabilidad de detectar correctamente el contenedor que se ha indicado con los comandos de voz. Se debe tener en cuenta que el porcentaje de error no indica que se abra otro contenedor, si no que el sistema no reconoció correctamente el comando, y esto puede deberse al volumen de voz o a la pronunciación del usuario.

8. TRABAJOS FUTUROS

Realizar una segunda versión del prototipo de sistema contenedores donde se implemente con una alimentación directa sin necesidad de usar la fuente de los laboratorios, pudiendo ser de baterías, paneles solares o del toma corriente con un adaptador. Además, se puede agregar un botón de encendido y apagado, de esta manera se hace el sistema más autónomo.

Realizar una prueba de concepto de uso del prototipo en su segunda versión en el Departamento de Electricidad y Electrónica que es donde se concentra la mayor parte de los estudiantes del Programa de Ingeniería Electrónica. Con ello se busca evaluar la aceptación del prototipo para su implementación comercial y que tenga una capacidad de llenado superior a los 12 Litros.

Agregar funciones adicionales para verificar el llenado de los contenedores, utilizando sensores de distancia. Así mismo, hacer un envío de mensaje SMS o email a la empresa de recolección de residuos reciclables cuando se detecte su máxima capacidad de llenado. Además, se puede desarrollar una aplicación para llevar el seguimiento por un dispositivo aplicando el concepto de IoT.

Adaptar el código base para aplicaciones que usen el reconocimiento de voz en funciones de domótica o automatización, como control de luces o electrodomésticos, apertura de puertas y control de temperatura.

REFERENCIAS

- [1] Á. Segura, L. Rojas, and Y. Pulido, “Sistemas de gestión de residuos sólidos,” *Espacios*, vol. 41, no. 17, p. 22, 2020.
- [2] A. Savino, G. Solórzano, C. Quispe, and M. Correal, “Perspectiva de la Gestión de Residuos en América Latina y el Caribe,” Oct. 2018.
<https://wedocs.unep.org/handle/20.500.11822/26448> (accessed Jul. 08, 2021).
- [3] J. C. Sayago Ortega, G. A. Carrillo Soto, and J. P. Rojas Suárez, “Management of urban solid waste in the two first decades of the XXI century in the city of San José de Cúcuta, Colombia,” *J. Phys. Conf. Ser.*, vol. 1388, no. 1, Nov. 2019, doi: 10.1088/1742-6596/1388/1/012010.
- [4] J. A. Astaiza and M. S. Cerón, “Estrategia didáctica basada en TIC para la descripción del proceso de la separación de residuos sólidos aprovechables en la fuente,” Uniautónoma del Cauca. Facultad de Ingeniería. Programa de Ingeniería de Sistemas Informáticos, Cauca, 2020.
- [5] L. Ortega, D. Borrero, and C. Pérez, “Diseño, ensamble y puesta en funcionamiento de depósito colector de residuos sólidos automatizado,” Universidad Piloto de Colombia, Bogotá D.C, 2020.
- [6] Corporación Punto Azul, “Colombia solo recicla 17% de sus residuos y la meta es 40% a 2030,” Sep. 29, 2020. <https://www.puntoazul.com.co/noticias/colombia-recicla-17-de-sus-residuos-y-la-meta-es-llegar-al-40-a-2030/> (accessed Aug. 17, 2021).
- [7] Corponor, “Residuos sólidos promovemos el manejo integral,” *Territ. Verde*, no. 3, May 2018, Accessed: Aug. 17, 2021. [Online]. Available:
https://corponor.gov.co/e/images/Tercera_Edición_Gaceta.pdf.

- [8] ONU, “Objetivo 12: Garantizar modalidades de consumo y producción sostenible,” Sep. 25, 2015. <https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/> (accessed Mar. 06, 2023).
- [9] S. Hernández Flechas and L. R. Corredor González, “Reflexiones sobre la importancia económica y ambiental del manejo de residuos en el siglo XXI,” *Rev. Tecnol.*, vol. 15, no. 1, pp. 57–76, 2016, Accessed: Oct. 29, 2021. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=6041529&info=resumen&idioma=SPA>.
- [10] Banco Mundial, “Los desechos: un análisis actualizado del futuro de la gestión de los desechos sólidos,” Sep. 20, 2018. <https://www.bancomundial.org/es/news/immersive-story/2018/09/20/what-a-waste-an-updated-look-into-the-future-of-solid-waste-management> (accessed Oct. 29, 2021).
- [11] S. Kaza, L. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*, 1st ed. Washington: World Bank, 2018.
- [12] S. Smith, “The Breadth and Depth of DSP,” in *The Scientist and Engineer’s Guide to Digital Signal Processing*, 1st ed., San Diego: California Technical Publishing, 1997, pp. 1–10.
- [13] J. G. Proakis and D. G. Manolakis, “THE DISCRETE-TIME FOURIER ANALYSIS,” in *Tratamiento digital de señales*, 4th ed., Pearson Educacion S.A, Ed. Madrid: Pearson Educacion S.A, 2007, pp. 80–97.
- [14] B. A. Sheno, *Introduction to digital signal processing and filter design*, 1st ed. New Jersey: John Wiley & Sons, 2006.
- [15] G. Pajares, *Análisis y reconocimiento de voz*, 1st ed. RC Libros, 2017.
- [16] MATLAB & Simulink, “Audio Toolbox ,” 2017.

- <https://la.mathworks.com/products/audio.html> (accessed Nov. 07, 2021).
- [17] P. . R. Hill, *AUDIO AND SPEECH PROCESSING WITH MATLAB*, 1st ed. Taylor & Francis Group, 2019.
- [18] D. Amos, “The Ultimate Guide To Speech Recognition With Python,” 2018. Accessed: Nov. 07, 2021. [Online]. Available: <https://realpython.com/python-speech-recognition/>.
- [19] M. Ángeles, “IRBin: el robot que educa sobre reciclaje,” 2019. <https://departamento.pucp.edu.pe/ingenieria/2020/02/17/irbin-el-robot-que-educa-sobre-reciclaje/> (accessed Nov. 07, 2021).
- [20] C. Smith and A. Corripio, “Introducción - Un sistema de control de procesos,” in *Control automático de procesos*, 2nd ed., LIMUSA S.A, Ed. México: LIMUSA S.A, 2014, pp. 1–3.
- [21] K. Ogata, “Introducción a los sistemas de control,” in *Ingeniería de control moderna*, 5th ed., PEARSON EDUCACIÓN S.A., Ed. Madrid: PEARSON EDUCACIÓN S.A., 2010, pp. 7–8.
- [22] A. de señales análogas y digitales Ambardar, “Señales Analógicas,” in *Procesamiento de señales análogas y digitales*, 2nd ed., THOMSON EDITORES S.A, Ed. México: EDAMSA S.A, 2002, p. 8.
- [23] D. M. Auslander, “Digital Controllers,” *Encycl. Phys. Sci. Technol.*, pp. 441–460, Jan. 2003, doi: 10.1016/B0-12-227410-5/00175-7.
- [24] K. Ogata, “Introducción a los sistemas de control discreto,” in *Sistemas de control tiempo discreto*, 2nd ed., Prentice Hall, Ed. Juárez: Prentice Hall, 1996, pp. 1–4.
- [25] D. Pérez, “Sistemas Embebidos y Sistemas Operativos Embebidos,” *Lect. en Ciencias la Comput.*, pp. 4–10, Oct. 2009.

- [26] S. Salas Arriarán, “Conceptos básicos: Sistemas embebidos,” in *Todo sobre sistemas embebidos*, 1st ed., D. Félix, Ed. Lima: Universidad Peruana de Ciencias Aplicadas, 2015, pp. 38–39.
- [27] G. Galeano, “Introducción- ¿Que es un sistema embebido?,” in *SISTEMAS EMBEBIDOS en C*, 1st ed., México: Alfaomega, 2009, pp. 8–10.
- [28] F. Ebel and S. Nestel, “Notas generales - La importancia de la tecnología de los sensores,” in *Sensores para la técnica de procesos y manipulación.*, 1st ed., Esslingen: Festo Didactic KG, 1993, pp. 12–13.
- [29] L. Corona, G. Abarca, and J. Mares, *Sensores y Actuadores. Aplicaciones con Arduino*, 1st ed. Azcapotzalco: PATRIA S.A, 2014.
- [30] J. Guarella, J. P. Heredia, L. Rodríguez, and I. Bagatto, “Sensores y actuadores con motores,” Universidad Nacional de la plata, Buenos Aires, 2011.
- [31] A. Géron, *Hands-on Machine Learning whith Scikit-Learing, Keras and Tensorfow*, 2nd ed. Sebastopol: O’reilly, 2019.
- [32] S. Russell and P. Norving, *Artificial intelligence: A modern approach*, 4th ed. Upper Saddle River: Person, 2021.
- [33] B. Mahesh, “Machine Learning Algorithms-A Review ,” *Int. J. Sci. Res.*, vol. 9, no. 1, pp. 381–386, Jan. 2020, doi: 10.21275/ART20203995.
- [34] G. Hinton and T. Sejnowski, *Unsupervised learning: Foundations of neural computation*. Cambridg: MIT press, 1999.
- [35] I. El Naqa and M. J. Murphy, “Machine Learning in Radiation Oncology,” *Mach. Learn. Radiat. Oncol.*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3.
- [36] J. Colás Pasamontes, “2.4.1.4 Técnicas más utilizadas aplicadas al Reconocimiento de

- Habla,” *Estud. Lingüística del Español*, vol. 12, 2001, Accessed: Dec. 13, 2022. [Online]. Available: <http://elies.rediris.es/elies12/cap2414.htm>.
- [37] S. Clovis, “Sistema de reconocimiento automático del habla,” *Bit@Bit*, vol. 3, no. 5, pp. 69–75, Jun. 2021, [Online]. Available: <http://dicyt.uajms.edu.bo/revistas/index.php/bitabit/article/view/872/872>.
- [38] J. J. Morales, “Elaboración, evaluación y optimización de modelos ocultos de markov para la transliteración de palabras del inglés a katakana,” Escuela Politécnica Nacional, Quito, 2022.
- [39] G. Portillo Ramírez, “Reconocimiento de placas basado en modelos ocultos de Markov,” Benemérita Universidad Autónoma de Puebla, Puebla, México, 2021.
- [40] K. Gurney, *An Introduction to Neural Networks*, 1st ed. 2017.
- [41] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.
- [42] Microsoft, “¿Qué es el aprendizaje profundo?” <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-deep-learning/> (accessed Dec. 21, 2022).
- [43] Alpha Cephei, “VOSK Offline Speech Recognition API,” Jan. 02, 2020. <https://alphacephei.com/vosk/> (accessed Sep. 09, 2022).
- [44] E. Demirel, “Deep neural networks for automatic lyrics transcription,” Queen Mary University, Londres, 2022.
- [45] B. Tarján, G. Szaszák, T. Fegyó, and P. Mihajlik, “Investigation on N-Gram Approximated RNNLMs for Recognition of Morphologically Rich Speech,” *International Conference on Statistical Language and Speech Processing*, Springer Science and

- Business Media Deutschland GmbH, pp. 223–234, Sep. 2019.
- [46] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, “Improved Audio Scene Classification Based on Label-Tree Embeddings and Convolutional Neural Networks,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 6, pp. 1278–1290, 2017, doi: 10.1109/TASLP.2017.2690564.
- [47] V. T. Tran and W. H. Tsai, “Stethoscope-Sensed Speech and Breath-Sounds for Person Identification with Sparse Training Data,” *IEEE Sens. J.*, vol. 20, no. 2, pp. 848–859, 2020, doi: 10.1109/JSEN.2019.2945364.
- [48] N. Ziafat, H. F. Ahmad, I. Fatima, M. Zia, A. Alhumam, and K. Rajpoot, “Correct pronunciation detection of the arabic alphabet using deep learning,” *Appl. Sci.*, vol. 11, no. 6, pp. 1–19, 2021, doi: 10.3390/app11062508.
- [49] F. C. Ribeiro, R. T. S. Carvalho, P. C. Cortez, V. H. C. De Albuquerque, and P. P. R. Filho, “Binary Neural Networks for Classification of Voice Commands from Throat Microphone,” *IEEE Access*, vol. 6, pp. 70130–70144, 2018, doi: 10.1109/ACCESS.2018.2881199.
- [50] S. Fernández and K. Ramos, “Implementación de un sistema con inteligencia computacional para identificar dificultad respiratoria a partir del procesamiento digital de señales de voz,” Universidad Antonio Nariño, 2021.
- [51] M. Smit and A. N. Al-Assimi, “Cascade Deep Neural Networks Classifiers for Phonemes Recognition,” *ARN J. Eng. Appl. Sci.*, vol. 15, no. 7, pp. 1664–1670, 2020, doi: 10.36478/JEASCI.2020.1664.1670.
- [52] P. Arce, D. Salvo, G. Piñero, and A. Gonzalez, “FIWARE based low-cost wireless acoustic sensor network for monitoring and classification of urban soundscape,” *Comput.*

- Networks*, vol. 196, p. 108199, 2021, doi: 10.1016/j.comnet.2021.108199.
- [53] B. da Silva, A. W. Happi, A. Braeken, and A. Touhafi, “Evaluation of classical Machine Learning techniques towards urban sound recognition on embedded systems,” *Appl. Sci.*, vol. 9, no. 18, 2019, doi: 10.3390/app9183885.
- [54] E. Cerezuela-Escudero, A. Jimenez-Fernandez, R. Paz-Vicente, M. Dominguez-Morales, A. Linares-Barranco, and G. Jimenez-Moreno, “Musical notes classification with neuromorphic auditory system using FPGA and a convolutional spiking network,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2015-Septe, 2015, doi: 10.1109/IJCNN.2015.7280619.
- [55] P. P. Patange and J. S. R. Alex, “Implementation of ANN based speech recognition system on an embedded board,” *2017 Int. Conf. Nextgen Electron. Technol. Silicon to Software, ICNETS2 2017*, pp. 408–412, 2017, doi: 10.1109/ICNETS2.2017.8067968.
- [56] M. Bakouri *et al.*, “Steering a Robotic Wheelchair Based on Voice Recognition System Using Convolutional Neural Networks,” *Electron.*, vol. 11, no. 1, pp. 1–17, 2022, doi: 10.3390/electronics11010168.
- [57] M. S. I. Sharifuddin, S. Nordin, and A. M. Ali, “Voice control intelligent wheelchair movement using CNNs,” *Proc. - 2019 1st Int. Conf. Artif. Intell. Data Sci. AiDAS 2019*, pp. 40–43, 2019, doi: 10.1109/AiDAS47888.2019.8970865.
- [58] Spyder Website Contributors, “Spyder IDE.” <https://www.spyder-ide.org/> (accessed Dec. 09, 2022).
- [59] MINISTERIO DE AMBIENTE Y DESARROLLO SOSTENIBLE, “Resolución 2184 de 2019,” Dec. 26, 2019. <https://www.minambiente.gov.co/wp-content/uploads/2021/08/resolucion-2184-de-2019.pdf> (accessed Nov. 05, 2021).
- [60] P. López and S. Fachelli, “Producción. El diseño de la muestra,” in *Metodología de la*

- Investigación Social Cuantitativa*, Bellaterra, España, 2017.
- [61] M. Geier, “Python-sounddevice. Release 0.4.5-1-g841e0a1,” Aug. 2022. Accessed: Oct. 01, 2022. [Online]. Available: <https://docs.conda.io/miniconda.html>.
- [62] J. A. Cienfuegos, “Desarrollo de un asistente de voz para la generación de un cuaderno de bitácora digital,” Centro Universitario de la Defensa, Vigo, España, 2021.
- [63] M. Boada Oriols and J. A. Gómez Gutiérrez, *El gran libro angular*, 1st ed. Barcelona: Marcombo S.A, 2018.
- [64] Python, “JSON encoder and decoder,” Dec. 18, 2020. <https://docs.python.org/3.5/library/json.html> (accessed Nov. 02, 2022).
- [65] TowerPro, “MG946R.” <https://www.towerpro.com.tw/product/mg946r/> (accessed Dec. 10, 2022).
- [66] Samer I. Mohamed & Amr Abdelnabi, “Agent-Based Convolution and Reinforcement Learning,” *BEST Int. J. Manag. Inf. Technol. Eng. (BEST IJMITE)*, vol. 5, no. 12, pp. 17–28, 2017.
- [67] R. A. LLanos Castro and J. D. Celis Nuñez, “INTELIGENCIA COMPUTACIONAL CON PROCESAMIENTO DE VOZ PARA UNA APLICACIÓN DOMÓTICA EN EL DESARROLLO DEL INTERNET DE LAS COSAS,” Universidad Francisco de Paula Santander, 2017.
- [68] J. J. Peralta, C. A. Narvárez, M. P. Orellana, P. A. Patiño, and P. Cedillo, “Evaluación del reconocimiento de voz entre los servicios de Google y Amazon aplicado al Sistema Integrado de Seguridad ECU 911,” *Rev. Tecnológica-ESPOL*, vol. 33, no. 2, pp. 147–158, Oct. 2021, doi: <https://doi.org/10.37815/rte.v33n2.840>.
- [69] R. Errattahi, A. El Hannani, and H. Ouahmane, “Automatic speech recognition errors

detection and correction: A review,” *Procedia Comput. Sci.*, vol. 128, pp. 32–37, 2018, doi: 10.1016/j.procs.2018.03.005.

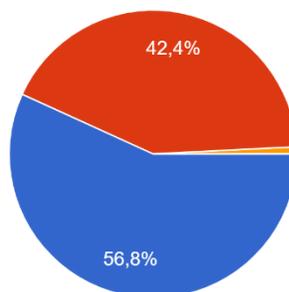
- [70] Azure-Microsoft, “Prueba de la precisión de un modelo de Habla personalizada,” Nov. 29, 2022. <https://learn.microsoft.com/es-es/azure/cognitive-services/speech-service/how-to-custom-speech-evaluate-data?pivots=speech-studio> (accessed Dec. 12, 2022).

ANEXOS

Anexo 1. Encuesta realizada a los estudiantes del Programa de Ingeniería Electrónica.

Contenedor blanco

5. Teniendo en cuenta la imagen, al momento de desechar estos objetos con que palabra los relaciona.

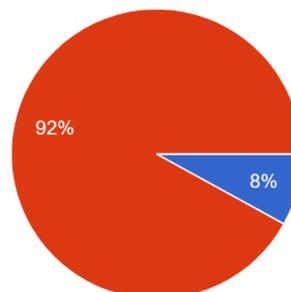


- Envases
- Botellas
- Fascos

Marca solo un óvalo.

- Envases
- Botellas
- Fascos

6. Teniendo en cuenta la imagen, al momento de desechar estos objetos con que palabra los relaciona.



- Hojas
- Papeles
- Láminas

Marca solo un óvalo.

- Hojas
- Papeles
- Láminas

7. Teniendo en cuenta la imagen, al momento de desechar estos objetos con que palabra los relaciona.



Marca solo un óvalo.

- Latas
- Potes

8. Teniendo en cuenta la imagen, al momento de desechar estos objetos con que palabra los relaciona.



Marca solo un óvalo.

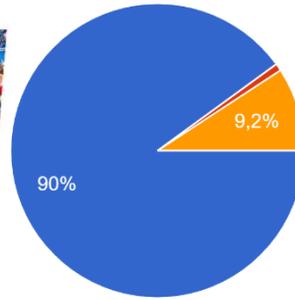
- Tapas
- Tapones
- Corcho latas

9. Teniendo en cuenta los materiales de las imágenes, al momento de desecharlos con que tipo de residuos lo asocia.

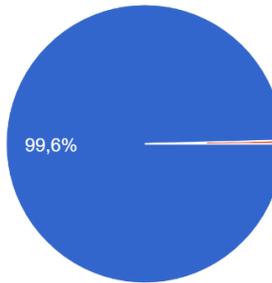


Marca solo un óvalo.

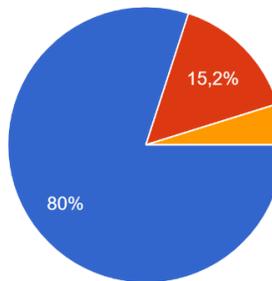
- Reciclables
- Reutilizables
- Renovables



- Latas
- Potes
- Metales



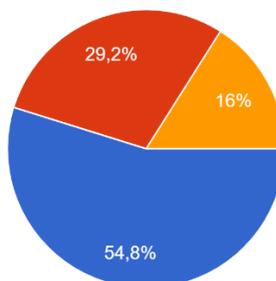
- Tapas
- Tapones
- Corcho latas



- Reciclables
- Reutilizables
- Renovables

Contenedor verde

10. Teniendo en cuenta la imagen, al momento de desechar estos objetos con que palabra los relaciona.

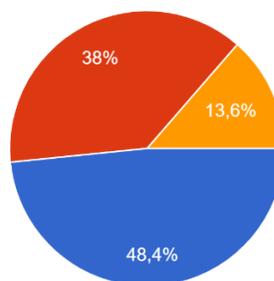


- Desperdicios
- Cáscaras
- Conchas

Marca solo un óvalo.

- Desperdicios
- Cáscaras
- Conchas

11. Teniendo en cuenta las imágenes, al momento de desechar estos objetos los relaciona con residuos de:

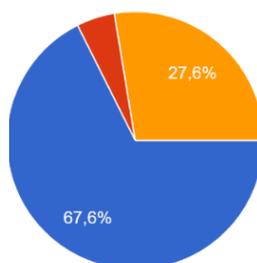


- Cultivo
- Jardinería
- Plantaciones

Marca solo un óvalo.

- Cultivo
- Jardinería
- Plantaciones

12. Teniendo en cuenta las cuatro imágenes anteriores. Para usted que palabra las generaliza, al momento de desecharlas.



- Residuos orgánicos
- Residuos aprovechables
- Residuos biodegradables

Marca solo un óvalo.

- Residuos orgánicos
- Residuos aprovechables
- Residuos biodegradables

Contenedor negro

13. Si se tiene en cuenta que estos 3 elementos han sido usados, con que palabra los relaciona al momento de desecharlos.



Marca solo un óvalo.

Papeles sucios

14. Teniendo en cuenta las imágenes, al momento de desechar estos objetos con que palabra los relaciona.



Marca solo un óvalo.

Envolturas

Recipientes sucios

Paquetes usados

15. Teniendo en cuenta las tres imágenes. Para usted que palabra las generaliza, al momento de desecharlas.

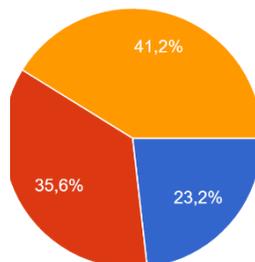


Marca solo un óvalo.

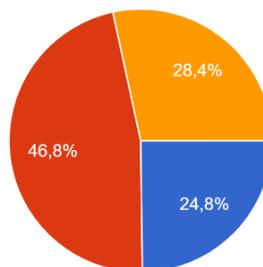
Residuos no aprovechables

Residuos no reutilizables

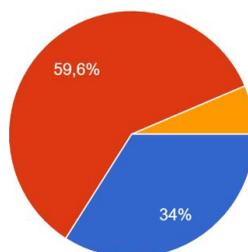
Residuos inertes



Papeles sucios
 Papeles contaminados
 Papeles usados



Envolturas
 Recipientes sucios
 Paquetes usados



Residuos no aprovechables
 Residuos no reutilizables
 Residuos inertes

Anexo 2. Carta de Veolia, con los desechos que se puede reciclar y los que no.

FORMATO
VHC-GR-F-02
Versión: 05



San José de Cúcuta, 21 de octubre de 2022.

Señor (a)
PAOLA ROCIO NOCUA ENCISO
paolarocione@ufps.edu.co

Ref: Respuesta PQR 1335187 - 30/09/2022

Apreciado Cliente:

Reciba un cordial saludo de Veolia, somos una empresa en la que trabajar para renovar el mundo, es nuestra razón de ser.

En atención al asunto en referencia, nos permitimos informarle de manera atenta que los residuos sólidos son todos los materiales que producimos en las actividades que realizamos diariamente y que pierden el valor para quienes lo poseen después de haber sido manipulados, razón por la cual son desechados.

Ahora bien, el reciclaje o aprovechamiento es un proceso cuyo objetivo es convertir desechos en nuevos productos o en materia para su posterior utilización. El tratamiento de reciclaje puede llevarse a cabo de manera total o parcial, según cada caso. Con algunos materiales, es posible obtener una materia prima, mientras que otros permiten generar un nuevo producto.

Los materiales no reciclables, por el contrario, son aquellos que, tras su uso principal y el agotamiento de su valor, no pueden ser aprovechados nuevamente.

BENEFICIOS DEL RECICLAJE

1. Reduce la necesidad de vertederos y/o rellenos sanitarios por tanto aumenta su vida útil
2. Evita la contaminación causada por la fabricación de productos de materiales vírgenes.
3. Ahorra energía.
4. Reduce las emisiones de gases de efecto invernadero que contribuyen al cambio climático.
5. Ahorra recursos naturales como son el uso de la madera, el agua y minerales.

QUÉ PODEMOS RECICLAR

Papel, cartón, vidrio, plásticos, madera, telas y componentes electrónicos.

Veolia Aseo Norte de Santander S.A.S E.S.P
Nit. 807.005.020-8
Avenida 4A # 8N-57 Zona Industrial
Cúcuta - Colombia
Teléfono : (607) 578 48 88
www.veolia.com.co/oriente

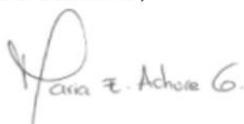


QUÉ NO PODEMOS RECICLAR

Papel plastificado, papel higiénico, servilletas de papel, envoltorios plásticos de alimentos, vajilla de cartón, material autoadhesivo, cerámica, espejos y cualquier material impregnado de combustibles fósiles.

Fue un gusto poder atenderle y recuerde que nuestro compromiso con usted es servirle a través de un equipo humano dispuesto siempre, si tiene alguna inquietud, con gusto le atenderemos a través de nuestra línea nacional de atención al cliente **018000950096** desde cualquier teléfono fijo o celular, de igual manera estamos a su disposición en nuestro correo electrónico co.servicioalcliente.aseo.orient@veolia.com o en nuestra página web <https://www.veolia.com.co/oriente>.

Cordialmente,



MARIA ESPERANZA ACHURE GRANDAS
Coordinadora de Servicio al Cliente

Transcriptor: Juanpvm.

Anexo 3. Codificación para el sistema de contenedores.

```
#!/usr/bin/env python3

import argparse
import queue
import sys
import sounddevice as sd
import RPi.GPIO as GPIO
import time
import json

from vosk import KaldiRecognizer, Model

GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(27, GPIO.OUT)
GPIO.setup(22, GPIO.OUT)

q = queue.Queue()

def int_or_str(text):
    """Helper function for argument parsing."""
    try:
        return int(text)
    except ValueError:
        return text

def callback(indata, frames, time, status):
    """This is called (from a separate thread) for each audio block."""
    if status:
        print(status, file=sys.stderr)
    q.put(bytes(indata))

parser = argparse.ArgumentParser(add_help=False)
parser.add_argument(
    "-l", "--list-devices", action="store_true",
    help="show list of audio devices and exit")
args, remaining = parser.parse_known_args()
if args.list_devices:
    print(sd.query_devices())
    parser.exit(0)
parser = argparse.ArgumentParser(
    description=__doc__,
    formatter_class=argparse.RawDescriptionHelpFormatter,
    parents=[parser])
parser.add_argument(
    "-f", "--filename", type=str, metavar="FILENAME",
    help="audio file to store recording to")
parser.add_argument(
    "-d", "--device", type=int_or_str,
    help="input device (numeric ID or substring)")
parser.add_argument(
    "-r", "--samplerate", type=int, help="sampling rate")
args = parser.parse_args(remaining)

Motor1_B = GPIO.PWM(17, 50) # GPIO 17 for PWM with 50Hz
Motor2_V = GPIO.PWM(27, 50) # GPIO 27 for PWM with 50Hz
Motor3_N = GPIO.PWM(22, 50) # GPIO 22 for PWM with 50Hz
```

```

Motor1_B.start(0)
time.sleep(2)
Motor2_V.start(0)
time.sleep(2)
Motor3_N.start(0)
time.sleep(2)

try:
    if args.samplerate is None:
        device_info = sd.query_devices(args.device, "input")
        # soundfile expects an int, sounddevice provides a float:
        args.samplerate = int(device_info["default_samplerate"])

    model = Model(lang="es")

    if args.filename:
        dump_fn = open(args.filename, "wb")
    else:
        dump_fn = None

    with sd.RawInputStream(samplerate=args.samplerate, blocksize = 8000,
                           device=args.device, dtype="int16", channels=1, callback=callback):
        print("#" * 90)
        print("Press Ctrl+C to stop the recording")
        print("#" * 90)

        rec = KaldiRecognizer(model, args.samplerate)
        while True:
            data = q.get()
            if rec.AcceptWaveform(data):
                #Definición de Listas para contenedores#
                blanco=["residuos reciclables", "tapas", "latas", "cartón","envases", "papeles",
                       "plástico","vidrio","madera"]
                verde=["desperdicios", "cultivo", "residuos orgánicos"]
                negro=["papeles usados", "recipientes sucios", "residuos no reutilizables",
                       "espejos", "cerámica"]

                jres = json.loads(rec.Result())
                valor_jres = jres.values() #Diccionario de valores de dicc
                lista_valor = list(valor_jres) #Lista de valores

                #Ciclos de Listas para el control de contenedores#
                for i in lista_valor:
                    for b in blanco:
                        if(b==i):
                            Motor1_B.ChangeDutyCycle(2)
                            time.sleep(5)
                            Motor1_B.ChangeDutyCycle(0)
                            time.sleep(0.5)
                            print("Contenedor 1 abierto")
                            Motor1_B.ChangeDutyCycle(8)
                            time.sleep(0.5)
                            print("Contendor 1 cerrado")
                            Motor1_B.ChangeDutyCycle(0)

                    for v in verde:
                        if(v==i):

```

```

        Motor2_V.ChangeDutyCycle(2)
        time.sleep(5)
        Motor2_V.ChangeDutyCycle(0)
        time.sleep(0.5)
        print("Contenedor 2 abierto")
        Motor2_V.ChangeDutyCycle(8)
        time.sleep(0.5)
        print("Contendor 2 cerrado")
        Motor2_V.ChangeDutyCycle(0)

    for n in negro:
        if(n==i):
            Motor3_N.ChangeDutyCycle(2)
            time.sleep(5)
            Motor3_N.ChangeDutyCycle(0)
            time.sleep(0.5)
            print("Contenedor 3 abierto")
            Motor3_N.ChangeDutyCycle(8)
            time.sleep(0.5)
            print("Contendor 3 cerrado")
            Motor3_N.ChangeDutyCycle(0)

    if i == ("detener"):
        print("\nSistema Apagado")
        parser.exit(0)
        Motor1_B.stop()
        Motor2_V.stop()
        Motor3_N.stop()
        GPIO.cleanup()

except Exception as e:
    parser.exit(type(e).__name__ + ": " + str(e))

```

Anexo 4. Consentimiento informado, prueba N°1.

CONSENTIMIENTO INFORMADO

TÍTULO: Sistema de contenedores controlado por voz para la correcta clasificación de los residuos sólidos.

GRUPO INVESTIGADOR

Paula Linned Suescun Vera, estudiante. Cel: 3107935991
Email: paulalinnedsv@ufps.edu.co

Paola Rocío Nocua Enciso, estudiante. Cel: 3134796846
Email: paolarocione@ufps.edu.co

Mario Joaquín Illera Bustos, IEO, MSc. Cel: 3185200403
Email: mariojoaquinib@ufps.edu.co

Geiner Giovanni Barbosa Casanova, IEO, MSc. Cel: 3112068964
Email: geinergiovannybc@ufps.edu.co

INTRODUCCIÓN

A través del presente documento lo invitamos a participar en una investigación que se desarrollará en los laboratorios del departamento de electricidad electrónica de la Universidad Francisco de Paula Santander. En él encontrará los elementos necesarios para tomar su decisión de participar en la misma o no hacerlo. Es importante que lea toda la información y si tiene alguna pregunta no dude en hacerla que los investigadores con gusto la responderán.

OBJETIVO DE LA INVESTIGACIÓN

Este estudio tiene como objetivo desarrollar un sistema de contenedores controlado por voz que ayude a la correcta clasificación de los residuos sólidos. Basado en un modelo entrenado con redes neuronales profundas.

PROCEDIMIENTOS

Para comprobar el funcionamiento del modelo en cuanto al reconocimiento de voz para cualquier usuario se hará uso de la voz de los participantes, los cuales leerán de manera fuerte y clara frente a un micrófono una lista de palabras que anteriormente han sido seleccionadas. Además, se tomarán algunas evidencias fotográficas con el fin de hacer el reporte de pruebas. Estas pruebas se realizarán en los laboratorios del departamento de electricidad electrónica de la Universidad Francisco de Paula Santander.

CARACTERÍSTICAS GENERALES DE LA INVESTIGACIÓN

Su participación es totalmente voluntaria. Esto quiere decir que, si usted desea negarse a participar en el estudio, lo puede hacer sin tener que dar explicaciones y sin que esto le genere algún tipo de problema jurídico.

Usted no se verá sometido a riesgo alguno, ya que no se utilizaran altos voltajes en el funcionamiento de los equipos.

Directamente usted no obtendrá ningún beneficio económico o en especie por participar en este estudio.

La participación en este estudio no tiene ningún costo para usted, ya que todo corre por cuenta del presupuesto del proyecto de investigación. Así mismo, no existirá ningún tipo de gasto en desplazamiento, ya que la totalidad de las pruebas se realizará dentro de la Universidad Francisco de Paula Santander.

Las fotografías que se tomen podrán ser utilizadas en el informe final del proyecto y en posibles publicaciones, pero la información personal que se obtendrá durante las pruebas permanecerá en secreto y no será proporcionada a ninguna persona diferente a los investigadores del proyecto. Para las evidencias se le asignará un código y de esta manera será nombrado si se requiere.

En caso de cualquier inquietud o duda, usted se puede contactar con alguna de las personas del grupo investigador (Paula Linned Suescum Vera, Paola Rocío Nocua Enciso, Mario Joaquín Illera Bustos y Geiner Giovanni Barbosa Casanova) en los teléfonos o correos electrónicos que aparecen al principio de este documento.

CONSENTIMIENTO

Después de haber leído comprensivamente toda la información contenida en este documento en relación con el estudio, y de haber recibido por parte de los investigadores explicaciones verbales y satisfactorias respuestas a mis inquietudes, yo _____ en pleno uso de mis facultades, libre y voluntariamente autorizo participar en esta investigación.

Además, expresamente autorizo a los investigadores tomar evidencias fotográficas para utilizar esta información en el informe final del proyecto y en posibles publicaciones con fines académicos y de investigación. En constancia, firmo este documento de consentimiento informado en la ciudad de Cúcuta el día ____ del mes de _____ del año _____.

Nombre _____ Firma _____

Cédula de ciudadanía No. _____ de _____

Código de participante: ____

Anexo 5. Registro de los resultados de la prueba N°4.

Tabla 1. Registro de resultados prueba N°4, parte 1.

Palabra	D	ND
Envases	x	
Latas	x	
Tapas	x	
Papeles	x	
Cartón	x	
Cartón		x
Cartón	x	
Cartón	x	
Vidrio	x	
Plástico	x	
Madera	x	
Residuos reciclables	x	

Tabla 2. Registro de resultados prueba N°4, parte 2.

Palabra	D	ND
Vidrio	x	
Vidrio	x	
Residuos no reutilizables	x	
Residuos no reutilizables		x
Residuos cultivo	x	
Residuos cultivo	x	
Latas	x	
Latas	x	
Papeles	x	
Papeles	x	
Cerámica	x	
Cerámica	x	
Cartón	x	
Cartón	x	
Residuos orgánicos	x	
Residuos orgánicos	x	
Madera	x	
Madera	x	
Residuos reciclables	x	
Residuos reciclables	x	
Espejos	x	
Espejos	x	
Envases	x	
Envases	x	
Recipientes sucios	x	
Recipientes sucios		x
Desperdicios	x	
Desperdicios	x	
Plástico	x	
Plástico	x	
Papeles usados	x	
Papeles usados	x	
Tapas		x
Tapas	x	
Cartón		x
Cartón	x	

Desperdicios	x	
Residuos cultivo		x
Residuos cultivo	x	
Residuos cultivo	x	
Residuos cultivo	x	
Residuos orgánicos	x	
Residuos orgánicos	x	
Residuos orgánicos		x
Residuos orgánicos	x	
Recipientes sucios	x	
Papeles usados	x	
Espejos	x	
Cerámica	x	
Residuos no reutilizables	x	
Residuos no reutilizables	x	
Residuos no reutilizables		x
Residuos no reutilizables	x	
Cerámica	x	
Cerámica	x	
Desperdicios	x	
Desperdicios	x	
Envases	x	
Envases	x	
Espejos	x	
Espejos	x	
Latas	x	
Latas	x	
Madera	x	
Madera	x	
Papeles	x	
Papeles	x	
Papeles usados	x	
Papeles usados	x	
Plástico		x
Plástico	x	
Recipientes sucios	x	
Recipientes sucios		x
Residuos cultivo	x	
Residuos cultivo	x	
Residuos no reutilizables	x	
Residuos no reutilizables	x	
Residuos orgánicos	x	
Residuos orgánicos	x	
Residuos reciclables	x	
Residuos reciclables	x	
Tapas	x	
Tapas	x	
Vidrio	x	
Vidrio	x	

INSTRUCCIONES Y MANUAL DE USUARIO PARA EL SISTEMA DE CONTENEDORES CONTROLADO POR VOZ

En este documento se presentan las instrucciones, requerimientos y procedimientos para el uso correcto del prototipo de sistema de contenedores controlado por voz.

1. Acceder al código del sistema.

Se accede al entorno de desarrollo *Spyder* desde la Raspberry pi 3B+. Una vez abierto se oprime el icono “open file” y se busca el archivo “Prueba_4.py”, como se observa en la Figura 1.

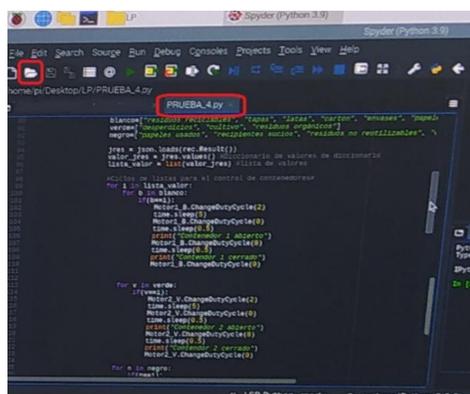


Figura 1. Acceso al código del sistema de contenedores.

2. Verificar las librerías para la ejecución de la codificación.

Cuando se trata de ejecutar el código por primera vez es probable que sea necesaria la instalación o actualización de algunas librerías. Para esto, puede dirigirse al enlace <https://pypi.org/> para buscar e instalar las librerías solicitadas.

3. Inicializar la codificación.

Con todas las librerías necesarias instaladas, al ejecutar el código de

programación se empieza una descarga automática del modelo en español. Una vez aparezca en la consola el mensaje: “#####Press Ctrl+C to stop the recording #####” significa que el sistema está listo para escuchar los comandos de voz.

4. Pronunciación de los comandos por parte del usuario.

El alcance del micrófono es de 30 cm, por lo tanto, el usuario al acercarse al micrófono no debe exceder dicha distancia. De esta manera, menciona la palabra que se relacione con el tipo de residuo que desea desechar. En cada uno de los contenedores se disponen unas etiquetas con las palabras permitidas por el sistema, como se ve en la Figura 2.



Figura 2. Palabras permitidas por el sistema de contenedores.

Cuando el usuario menciona, por ejemplo, ‘Espejos’ debe esperar 3 segundos para que el contenedor de ‘residuos no reutilizables’ se abra, y tiene un tiempo de 5 segundos para que se introduzca el desecho y proceda a cerrarse. En caso de que el sistema no capte la palabra mencionada debe repetirla con mayor claridad.

5. Detener el sistema de contenedores.

Si se desea detener el sistema, el usuario puede mencionar la palabra ‘detener’ o lo puede hacer con *Ctrl+C*. En la consola aparecerá el mensaje: “Sistema apagado”, lo que indica que el sistema ya no está escuchando.

MANUAL DE IMPLEMENTACIÓN TÉCNICA PARA EL SISTEMA DE CONTENEDORES CONTROLADO POR VOZ

Para la implementación del código de programación del sistema de contenedores controlado por voz se requiere la descarga e instalación del lenguaje de programación Python. Los paquetes y las librerías necesarias se encuentran en la Tabla 1 con su versión y descripción. Además, se necesita la configuración del sistema embebido Raspberry Pi 3B+ para la construcción del prototipo.

Para el reconocimiento de voz se usó una herramienta de red neuronal de código abierto y fuera de línea denominada Vosk-API la cual se configuró para usar el modelo en español de 39 MB. Para probar esta herramienta se tomó como base el script “*test_microphone.py*” proporcionado por la página oficial Alpha Cephei en el siguiente enlace https://github.com/alphacep/vosk-api/blob/master/python/example/test_microphone.py. Se configuró el modelo en idioma español y se manipuló la salida del sistema para el control de los servomotores.

Tabla 1. requerimientos para la ejecución del script del sistema de contenedores.

Herramienta/Librería/Paquete	Versión	Descripción
Spyder	5.1.5	Entorno de desarrollo para el lenguaje de programación
Python	3.9.2	Lenguaje de programación
Sounddevice	0.4.5	Módulo para reproducir y grabar sonidos
Vosk	0.3.44	Librería con el modelo de reconocimiento de voz
Pulseaudio	-	Servidor de sonido multiplataforma
Pip3	22.3.1	Administrador de paquetes

Al realizar la construcción del sistema se contaba con un micrófono de conexión Jack y por esto fue necesario realizar una configuración adicional a la Raspberry, usando una tarjeta de sonido externa (Jack - USB). Primero se deben actualizar los repositorios en la consola con el comando “*sudo apt update*”, luego de esto se actualiza el servidor de sonido con el comando “*sudo apt install pulseaudio*”, al finalizar se reinicia la tarjeta. Una vez encienda, primero se debe seleccionar el botón de sonido y segundo se seleccionan los perfiles de dispositivo, cuando se abra la ventana emergente, el puerto Jack se establece como puerto de salida de audio y el puerto de USB se establece como puerto de entrada como se observa en la Figura 1.

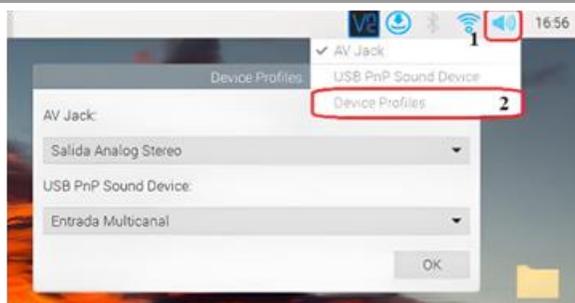


Figura 1. Configuración de las entradas y salidas de sonido de la Raspberry Pi 3B+.

Con la configuración del micrófono lista, se procede a hacer la conexión de los servomotores, como se observa en su diagrama circuital (ver Figura 2). Allí se presentan las conexiones a los pines de comunicación 17, 27 y 22, teniendo en cuenta la configuración GPIO.BCM. Además, se tiene en cuenta el punto común de los pines de alimentación positiva y negativa que deben tener los 3 servomotores y la Raspberry Pi 3B+ para su correcto funcionamiento.

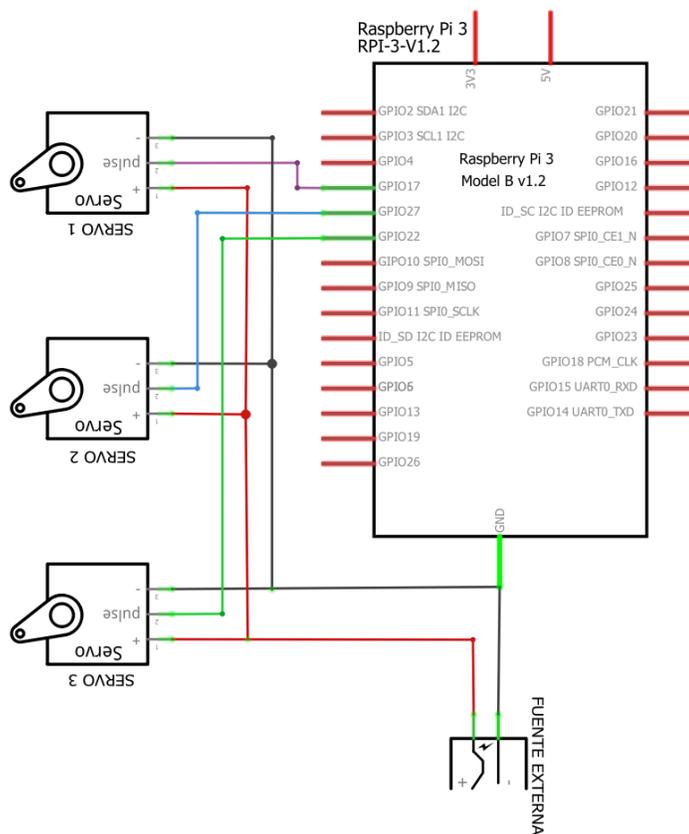


Figura 2. Diagrama circuital de los servomotores y la Raspberry Pi 3B+.