REVISTA AMBIENTAL
AGUA, AIRE Y SUELO

ANÁLISIS DE HERRAMIENTAS PARA DESARROLLAR UN SISTEMA DE APOYO AMBIENTAL PARA IDENTIFICAR RESIDUOS SÓLIDOS

ANALYSIS OF TOOLS FOR DEVELOPING AN ENVIRONMENTAL SUPPORT SYSTEM TO IDENTIFY SOLID WASTE

NIÑO RONDÓN, C.V [1]; CASTRO-CASADIEGO S. A[2], ORTÍZ-FONSECA, D.M[3]

[1]Ing. Carlos Vicente Niño Rondón
Programa de Ingeniería Electrónica, Semillero de Investigación en Instrumentación Electrónica SIINE, Universidad Francisco de Paula Santander, carlosvicentenr@ufps.edu.co

[2]MSc. Sergio Alexander Castro Casadiego
Programa de Ingeniería Electrónica, Semillero de Investigación en Instrumentación Electrónica SIINE, Universidad Francisco de Paula Santander, sergio.castrocr@ufps.edu.co

[3]Ing (c). Danna Marcela Ortiz Fonseca
Programa de Ingeniería Electrónica, Semillero de Investigación en Instrumentación Electrónica SIINE, Universidad Francisco de Paula Santander, dannamarcelaof@ufps.edu.co

Resumen
Este artículo presenta una metodología detallada para el desarrollo de un sistema de apoyo ambiental destinado a la identificación de residuos sólidos. La metodología se divide en tres etapas fundamentales: selección de hardware, selección de software y diseño preliminar del algoritmo. En la primera etapa, se utiliza una matriz de selección para evaluar y comparar diversos aspectos clave del hardware, como procesadores, memoria RAM, lenguajes de programación, procesamiento gráfico, precio y disponibilidad en el mercado. Los resultados ponderados indican que el Raspberry Pi 4 y el Jetson Nano son las opciones más adecuadas, basadas en las necesidades específicas del proyecto. En la segunda etapa, se utiliza una matriz de selección similar para evaluar aspectos críticos del software, como el costo de la licencia, la disponibilidad de librerías, la integración con entornos de desarrollo, la conectividad con software de ofimática y la compatibilidad con inteligencia artificial. Python emerge como el lenguaje de programación ideal, con un peso del 52.35%, debido a su versatilidad y capacidad de procesamiento. Además, se presenta un algoritmo propuesto que aborda la preparación de datos, incluida la redimensión de imágenes, el etiquetado de datos de entrenamiento y validación, la conversión de imágenes a escala de grises y la normalización de datos. En conjunto, esta metodología ofrece una guía sólida para el desarrollo de un sistema que puede contribuir significativamente a la gestión ambiental y la sostenibilidad al identificar eficazmente los residuos

ISSN 1900-9178 Volumen 12 - 2021
UNIVERSIDAD DE PAMPLONA

sólidos. El enfoque propuesto puede tener un impacto positivo en entornos urbanos y rurales al mejorar la eficiencia de la gestión de residuos.

Palabras clave
residuos sólidos, aprendizaje profundo, inteligencia artificial, gestión ambiental.

Abstract
This paper presents a detailed methodology for the development of an environmental support system for solid waste identification. The methodology is divided into three fundamental stages: hardware selection, software selection, and preliminary algorithm design. In the first stage, a selection matrix is used to evaluate and compare several key hardware aspects, such as processors, RAM, programming languages, graphics processing, price and market availability. The weighted results indicate that the Raspberry Pi 4 and the Jetson Nano are the most suitable options, based on the specific needs of the project. In the second stage, a similar selection matrix is used to evaluate critical aspects of the software, such as licensing cost, library availability, integration with development environments, connectivity with office software and support for artificial intelligence. Python emerges as the ideal programming language, with a weight of 52.35%, due to its versatility and processing capacity. In addition, a proposed algorithm is presented that addresses data preparation, including image resizing, training and validation data labeling, grayscale image conversion, and data normalization. Taken together, this methodology provides a sound guide for the development of a system that can significantly contribute to environmental management and sustainability by effectively identifying solid waste. The proposed approach can have a positive impact on urban and rural environments by improving the efficiency of waste management.

Keywords
solid waste, deep learning, artificial intelligence, environmental management.

1. INTRODUCCIÓN

En un mundo que enfrenta crecientes desafíos ambientales, la gestión eficiente de los residuos sólidos se ha convertido en una prioridad imperativa. La acumulación descontrolada de desechos representa una amenaza significativa para los ecosistemas terrestres y acuáticos, así como para la salud humana en general. Con la creciente urbanización y la expansión de las poblaciones, la gestión adecuada de los residuos sólidos se ha vuelto aún más crítica. En este contexto, la tecnología desempeña un papel fundamental al ofrecer soluciones innovadoras y sostenibles (Perales Esteve et al., 2020)

El análisis de herramientas de hardware implica la evaluación de las tecnologías disponibles, como sensores y dispositivos de detección (Valencia et al., 2019), que permiten la recopilación de datos sobre los residuos sólidos. La elección adecuada de hardware es esencial para garantizar la adquisición precisa de información, ya que esta servirá como base para el análisis subsiguiente (Lenarduzzi et al., 2020).

Las herramientas de software desempeñan un papel crítico en la interpretación de los datos recopilados por el hardware (Graf et al., 2020). El procesamiento de imágenes, la inteligencia artificial y el aprendizaje automático son componentes clave en esta etapa. Se busca no solo identificar los tipos de residuos, sino también analizar su composición y características para una gestión óptima.

Finalmente, el diseño del algoritmo de programación ideal integra todas estas

herramientas en un sistema coherente y eficiente (Cavanzo-Nisso et al., 2017). El algoritmo debe ser capaz de procesar grandes volúmenes de datos en tiempo real y proporcionar resultados precisos y útiles para los encargados de la gestión de residuos.

A medida que se avanza en esta era de innovación tecnológica, es imperativo desarrollar sistemas de apoyo ambiental avanzados que puedan ayudar a abordar los problemas ambientales de manera efectiva (Perales Esteve et al., 2020). Este análisis de herramientas proporcionará una base sólida para la creación de un sistema de apoyo ambiental robusto y eficiente en la identificación de residuos sólidos, contribuyendo así a un futuro más sostenible y saludable para el planeta (Abdallah et al., 2020).

Este artículo se centra en el análisis exhaustivo de las herramientas necesarias para la creación de un sistema de apoyo ambiental que pueda identificar residuos sólidos de manera precisa y eficiente. Este análisis se divide en tres componentes esenciales: herramientas de hardware, herramientas de software y el diseño del algoritmo de programación ideal. Estos elementos son fundamentales para el desarrollo exitoso de un sistema que pueda abordar los desafíos actuales de la gestión de residuos sólidos.

2. METODOLOGIA

La metodología propuesta para el desarrollo de un sistema de apoyo ambiental destinado a la identificación de residuos sólidos se divide en tres etapas fundamentales: la selección de hardware mediante, la selección de y el diseño preliminar del algoritmo (Olabanji & Mpofu, 2020).

A) Selección de Hardware por Matriz de Selección:

En la primera etapa, se llevará a cabo un proceso meticuloso de selección de hardware, que se basará en una matriz de selección. Esta matriz se diseñará considerando criterios críticos como la precisión de detección, la escalabilidad, la durabilidad y la disponibilidad en el mercado. Se identificarán y evaluarán diferentes tipos de sensores, cámaras y dispositivos de recolección de datos que sean adecuados para la identificación de residuos sólidos. La matriz de selección proporcionará una puntuación ponderada para cada opción, lo que permitirá una elección informada del hardware óptimo para el sistema (Vidal et al., 2012).

Parámetros	Denotación
Procesador	A
Memoria RAM	B
Lenguajes	C
Procesamiento gráfico	D
Precio y disponibilidad	E
Tareas simultáneas	F

B) Selección de Software por Matriz de Selección:

En la segunda etapa, se abordará la selección de software utilizando una matriz de selección similar. Se evaluarán y compararán diversas plataformas de software de procesamiento de imágenes, análisis de datos y aprendizaje automático. Los criterios de selección incluirán la capacidad de procesamiento, la flexibilidad para adaptarse a diferentes tipos de residuos sólidos, la capacidad de actualización y expansión, y la capacidad de integración con el hardware seleccionado en la primera etapa. La matriz de selección permitirá identificar la solución de software más adecuada para el sistema.

C) Diseño Preliminar del Algoritmo:

ISSN 1900-9178 Volumen 12 - 2021
UNIVERSIDAD DE PAMPLONA

La tercera etapa se centrará en el diseño preliminar del algoritmo de programación. Utilizando las especificaciones del hardware y el software seleccionados, se elaborará un algoritmo que permita la identificación y clasificación eficiente de los residuos sólidos (Alqahtani et al., 2020). Este algoritmo se diseñará teniendo en cuenta la velocidad de procesamiento y la precisión de identificación como principales objetivos. Se considerarán técnicas de procesamiento de imágenes, aprendizaje automático y reconocimiento de patrones para desarrollar un algoritmo efectivo que pueda ser implementado en el sistema.

3. RESULTADOS

Para obtener los parámetros comparativos de tecnologías de hardware se establecieron los seis parámetros más importantes para la realización del proyecto en la tabla 1.

Tabla 1. Parámetros de hardware
En la tabla 2, se evidencia el peso que comprende cada uno de los parámetros.

Tabla 2. Peso de parámetros de hardware.

	A	B	C	D	E	F	S	F
A	X	5	2	8	1/5	10,00	25	0,33
B	1/5	X	5	5	1/5	5,00	15,40	0,20
C	1/2	1/5	X	5	1/3	5,00	11,03	0,14
D	1/8	1/5	1/5	X	2/5	2/5	1,33	0,02
E	5	5	3	5,00	X	5,00	23,00	0,30
f	1/10	1/5	1/5	3/5	1/5	X	1	0,02

Asimismo, en la tabla 3 se muestra la comparación de herramientas de hardware respecto a la velocidad de procesamiento.

Tabla 3. Comparación de herramientas de hardware respecto a la velocidad de procesamiento.

	Rpi 4	Jetson Nano	Arduino Uno	Nexys 3	S	F
Rpi 4	X	1	10	10	21,00	0,44
Jetson Nano	1	X	10	10	21,00	0,44
Arduino Uno	1/10	1/10	X	1/5	0,40	0,01
Nexys 3	1/10	1/10	5	X	5,20	0,11

A su vez, en la tabla 4 se realiza la comparación de herramientas de hardware respecto memoria RAM.

Tabla 4. Comparación de herramientas de hardware respecto memoria RAM.

	Rpi 4	Jetson Nano	Arduino Uno	Nexys 3	S	F
Rpi 4	X	5	10	6	21,00	0,58
Jetson Nano	1/5	X	6	5	11,20	0,31
Arduino Uno	1/10	1/6	X	1/3	0,60	0,02
Nexys 3	1/6	1/5	3	X	3,37	0,09

Por otro lado, en la tabla 5 se observará la comparación de herramientas de hardware respecto lenguaje de programación soportados

Tabla 5. Comparación de herramientas de hardware respecto lenguaje de programación soportados.

	Rpi 4	Jetson Nano	Arduino Uno	Nexys 3	S	F
Rpi 4	X	1	10	5	16,00	0,52
Jetso	1	X	5	5	11,	0,3

ISSN 1900-9178 Volumen 12 - 2021
UNIVERSIDAD DE PAMPLONA

n Nano					00	6
Arduino Uno	1/10	1	X	1	2,10	0,07
Nexys 3	1/5	1/5	1	X	1,40	0,05

En la tabla 6, se visualiza la comparación de herramientas de hardware respecto a la velocidad de procesamiento gráfico.

Tabla 6. Comparación de herramientas de hardware respecto a la velocidad de procesamiento gráfico.

	Rpi 4	**Jetson Nano**	**Arduino Uno**	**Nexys 3**	**S**	**F**
Rpi 4	X	1/5	1	2	3,20	0,16
Jetson Nano	5	X	5	1/2	10,50	0,51
Arduino Uno	1	1/5	X	1/2	1,70	0,08
Nexys 3	1/2	2	2	1/2	5,00	0,25

De igual forma, en la tabla 7 se observa la comparación de herramientas de hardware respecto precio y disponibilidad del mercado.

Tabla 7. Comparación de herramientas de hardware respecto precio y disponibilidad del mercado.

	Rpi 4	**Jetson Nano**	**Arduino Uno**	**Nexys 3**	**S**	**F**
Rpi 4	X	3/5	1/10	5	5,70	0,11
Jetson Nano	5	X	1/8	1/10	5,23	0,10
Arduino Uno	9	8	X	10	27,00	0,53
Nexys 3	5	8	1/10	X	13,10	0,26

En la tabla 8, se puede visualizar la comparación de herramientas de hardware respecto a las tareas simultaneas.

Tabla 8. Comparación de herramientas de hardware respecto a las tareas simultaneas.

	Rpi 4	**Jetson Nano**	**Arduino Uno**	**Nexys 3**	**S**	**F**
Rpi 4	X	3	2	5	10,00	0,54
Jetson Nano	1/3	X	2/3	1,67	2,67	0,15
Arduino Uno	1/2	1,5	X	2,5	4,50	0,25
Nexys 3	1/5	3/5	2/5	X	1,20	0,07

Con esto, se elabora la matriz de decisión para la selección de la herramienta de hardware.

Tabla 9. Peso de parámetros de hardware.

	A	**B**	**C**	**D**	**E**	**F**	**S**
Rpi	0,1439 0	0,1157	0,0749	0,0027	0,0333	0,0092	0,37977
JN	0,1439	0,0617	0,0515	0,0088	0,0305	0,0024	0,29899
AUNO	0,0027 4	0,0033	0,0098	0,0014	0,15755	0,0041	0,17789
N3	0,0356	0,0186	0,0066	0,0042	0,0764	0,0011	0,14255

De lo anterior, se puede observar que con un porcentaje del 37,97% el hardware seleccionado sería el Raspberry pi 4, en segundo lugar, el Jetson Nano con un 29,89%, y como los dos últimos el Arduino UNO con un 17,89% y Nexys 3 con un 14,25%, estas selecciones fueron realizadas a base de las necesidades del proyecto.

Para obtener los parámetros comparativos de tecnologías de software se

establecieron los cinco parámetros más importantes para la ejecución del sistema.

Tabla 10. Parámetros de software

Parámetros	Denotación
A	Costo de licencia
B	Disponibilidad de librerías
C	Entornos de desarrollo
D	Conectividad con ofimática
E	Comparación de herramientas de software respecto a la especialidad para inteligencia artificial

En la tabla 11, se evidencia el peso que comprende cada uno de los parámetros.

Tabla 11. Peso de parámetros de software.

	A	B	C	D	E	S	F
A	X	1	1	5	1	8	0,17
B	1	X	5	5	1	12	0,26
C	1	1/5	X	10	1	12,2	0,27
D	1/5	1/5	1/10	X	1/10	0,6	0,01
E	1	1	1	10	X	13	0,28

Inicialmente, en la tabla 12 se muestra la comparación respecto al tipo de licencias.

Tabla 12. Comparación de herramientas de software respecto al tipo de licencia.

	R	Python	Java	SQL	S	F
R	X	1/10	1/5	1/5	0,50	0,014
Python	10	X	5	5	20,00	0,554
Java	5	1/5	X	5	10,20	0,283
SQL	5	1/5	1/5	X	5,40	0,150

Asimismo, en la tabla 13 se muestra la comparación respecto a la capacidad de integración con entornos de desarrollo.

Tabla 13. Comparación de herramientas de software respecto a la capacidad de integración con entornos de desarrollo.

	R	Python	Java	SQL	S	F
R	X	1/10	1/10	1	1,20	0,03
Python	10	X	1	10	21,00	0,47
Java	10	1	X	10	21,00	0,47
SQL	1	1/10	1/10	X	1,20	0,03

En la tabla 14, se encuentra la comparación de herramientas de software respecto a la disponibilidad de paquetes y librerías.

Tabla 14. Comparación de herramientas de software respecto a la disponibilidad de paquetes y librerías.

	R	Python	Java	SQL	S	F
R	X	1/10	1/5	5	5,30	0,13
Python	10	X	5	10	25,00	0,61
Java	5	1/5	X	5	10,20	0,25
SQL	1/5	1/10	1/5	X	0,50	0,01

A su vez, en la tabla 15 se muestra la comparación de herramientas de software respecto conectividad ofimática.

Tabla 15. Comparación de herramientas de software respecto conectividad ofimática.

	R	Python	Java	SQL	S	F
R	X	1/10	1/10	1/10	0,30	0,01
Python	10	X	1	10	21,00	0,48
Java	1	1	X	10	21,0	0,4

	0				0	8
SQL	1	1/10	1/10	X	1,20	0,03

Del mismo modo, en la tabla 16 se realiza la comparación de herramientas de software respecto a la compatibilidad con inteligencia artificial.

Tabla 16. Comparación de herramientas de software respecto a compatibilidad con inteligencia artificial.

	R	**Python**	**Java**	**SQL**	**S**	**F**
R	X	1/5	1/5	5	5,40	0,19
Python	5	X	1	5	11,00	0,39
Java	5	1	X	5	11,00	0,39
SQL	1/5	1/5	1/5	X	0,60	0,02

Luego de realizar la comparación de las herramientas de software respecto a cada parámetro, se procede a obtener una matriz de decisión en la cual se exponen los resultados adquiridos de cada una de las tablas mencionadas anteriormente, por lo tanto, al realizar una comparación con el peso de los parámetros, se obtiene la tabla 17.

Tabla 17. Matriz de decisión para la herramienta de software.

	A	**B**	**C**	**D**	**E**	**S**
R	0,004239923	0,0395719	0,0044126	0,000014	0,0393586	0,087724
Python	0,169596925	0,18666	0,0772201	0,009085	0,0801749	0,523504
Java	0,086494432	0,0761573	0,0772201	0,009085	0,0801749	0,329899
SQ	0,00	0,00	0,00	0,0	0,00	0,0
L	3052745	37332	44126	0056	43732	16135

Para la realización del sistema de apoyo ambiental para la identificación de residuos sólidos, el lenguaje de programación ideal es Python, el cual tiene un porcentaje del 52,35%.

De igual forma, se presenta el algoritmo propuesto para las etapas de procesamiento de imagen, así como entrenamiento del sistema de detección y clasificación de residuos sólidos. Se realiza con el fin de obtener imágenes con la misma cantidad de pixeles, realizar el etiquetado de cada una en las carpetas de entrenamiento y la validación, y a su vez, las imágenes se cambiaron a un solo canal de escalas grises para obtener una mejor detección de borde.

Tabla 18. Algoritmo propuesto para redimensión, etiquetado y cambio de canal.

1	**Importar** librerías
2	**Cargar** el entrenamiento de residuos solidos
3	**Cargar** la validación de residuos solidos
4	**Redimensionar** la imagen tomara un tamaño de 200x200 pixeles
5	**Etiquetar** las imágenes de entrenamiento
6	**Etiquetar** las imágenes de validación
7	**Leer** las fotos de entrenamiento
8	**Asignar** etiqueta a cada imagen de entrenamiento
9	**Leer** imagen con etiqueta en entrenamiento
10	**Realizar** interpolación de las imágenes
11	**Representar** la imagen en un solo canal en la escala de grises
12	**Añadir** imagen en EDG
13	**Leer** las fotos de validación
14	**Asignar** etiqueta a cada imagen de validación
15	**Leer** imagen con etiqueta de validación

16	**Realizar** interpolación de imágenes
17	**Representar** la imagen en un solo canal en la escala de grises
18	**Añadir** imagen en EDG
19	**Normalizar** imágenes de entrenamiento y validación
20	**Realizar** capa densa
21	**Aplicar** Kernel
22	**Ejecutar** matriz max pooling
23	**Realizar** convolución
24	**Aplicar** Kernel
25	**Ejecutar** matriz max pooling
26	**Realizar** convolución con Drop Out
27	**Aplicar** Kernel
28	**Ejecutar** matriz max pooling

4. CONCLUSIONES

El desarrollo de un sistema de apoyo ambiental para la identificación de residuos sólidos es un proyecto que requiere una metodología estructurada y cuidadosa, como se ha delineado en este artículo. A partir de la selección de hardware y software mediante matrices de selección ponderadas, se pueden tomar decisiones informadas que optimicen la eficiencia y la efectividad del sistema.

En la etapa de selección de hardware, se evaluaron varios parámetros críticos, como el procesador, la memoria RAM, los lenguajes de programación soportados, el procesamiento gráfico, el precio y la disponibilidad en el mercado, así como la capacidad para realizar tareas simultáneas. Los resultados indicaron que el Raspberry Pi 4 y el Jetson Nano se destacan como las mejores opciones, con el Raspberry Pi 4 liderando en términos de ponderación. Estas elecciones se basaron en las necesidades específicas del proyecto, como la capacidad de procesamiento y la disponibilidad en el mercado.

En la selección de software, se evaluaron parámetros como el costo de la licencia, la disponibilidad de librerías, la capacidad de integración con entornos de desarrollo, la conectividad con software de ofimática y la compatibilidad con inteligencia artificial. Python se destacó como el lenguaje de programación ideal, con un peso del 52.35%, debido a su versatilidad, amplia comunidad de desarrolladores y robusta capacidad de procesamiento de datos.

Además, se presentó un algoritmo propuesto para las etapas de procesamiento de imagen y entrenamiento del sistema de detección y clasificación de residuos sólidos. Este algoritmo aborda la redimensión de imágenes, el etiquetado de datos de entrenamiento y validación, la conversión de imágenes a escala de grises y la normalización de datos, preparando los datos de entrada para el aprendizaje automático.

5. REFERENCIAS BIBLIOGRAFÍCAS

Abdallah, M., Abu Talib, M., Feroz, S., Nasir, Q., Abdalla, H., & Mahfood, B. (2020). Artificial intelligence applications in solid waste management: A systematic research review. In *Waste Management* (Vol. 109, pp. 231–246). Elsevier Ltd. https://doi.org/10.1016/j.wasman.2020.04.057

Alqahtani, F., Al-Makhadmeh, Z., Tolba, A., & Said, W. (2020). Internet of things-based urban waste management system for smart cities using a Cuckoo Search Algorithm. *Cluster Computing*. https://doi.org/10.1007/s10586-020-03126-x

Cavanzo-Nisso, G. A., Pérez-Pereira, M. R., & Villavisan-Buitrago, F. (2017). Medición de eficiencia de algoritmos de visión artificial implementados en raspberry pi y ordenador personal

mediante Python. *Ingenium*, *18*(35), 105–119.

Graf, J., Batchelor, W., Harper, S., Marlow, R., Carlisle, E., & Athanas, P. (2020). A practical application of game theory to optimize selection of hardware Trojan detection strategies. *Journal of Hardware and Systems Security*, *4*(2), 98–119. https://doi.org/10.1007/s41635-019-00089-3

Lenarduzzi, V., Taibi, D., Tosi, D., Lavazza, L., & Morasca, S. (2020). Open Source Software Evaluation, Selection, and Adoption: A Systematic Literature Review. *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*, 437–444. https://doi.org/10.1109/SEAA51224.2020.00076

Olabanji, O., & Mpofu, K. (2020). Pugh matrix and aggregated by extent analysis using trapezoidal fuzzy number for assessing conceptual designs. *Decision Science Letters*, *9*(1), 21–36. https://doi.org/10.5267/j.dsl.2019.9.001

Perales Esteve, M., Toral Marín, S., & Gutierrez Reina, D. (2020). Formación en Ingeniería y Cooperación Internacional: Diseño de drones acuáticos para monitorización de variables ambientales. *XIV Congreso de Tecnologías Aplicadas a La Enseñanza de La Electrónica*, 11–16.

Valencia, V. R., Vera, R. B., Cabrera, C. M., Guerra, J., & Saltos, E. (2019). Propuesta de un sistema de estacionamiento Inteligente con sensores IoT. *Revista Iberica de Sistemas y Tecnologias de Informacion*, *19*(4), 553–567. search.proquest.com/openview/5b2c56a6dda949cb7b5f8875c88b1437/1?pq-origsite=gscholar&cbl=1006393

Vidal, C. J. C. J., Bravo, J. J., Cajiao, E., Meza, P. P., Arango, S., Franco, D., Calderón, J. H., Jos, J., Cajiao, E., Pablo, G. P., Franco, A. S. D., & Hern, J. (2012). *Guía metodológica para la priorización de proyectos: un enfoque aplicado a la infraestructura, la logística y la conectividad.* Sello editorial Javeriano.