

System Call: plataforma distribuida para la recolección de desperdicios de ciudades superpobladas

System Call: smart trash collection in densely populated cities

Cristian Ronquillo Lamilla, Juan García Plúas, Michelle Varas Chiquito, Grace Loor Alcívar

Facultad de Ingeniería Industrial, Universidad de Guayaquil. Ecuador, 090112

Autor de correspondencia: grace.loora@ug.edu.ec

Vol. 01, Issue 01 (2022): July

DOI: 10.53591/easi.v1i1.1770

Submitted: May 4, 2022

Published: July 14, 2022

Engineering and Applied Sciences in Industry

University of Guayaquil. Ecuador

Frequency/Year: 2

Web:

<https://revistas.ug.edu.ec/index.php/easi>

Email:

easi-publication.industrial@ug.edu.ec

How to cite this article:

Ronquillo et al. (2022). System Call: plataforma distribuida para la recolección de desperdicios de ciudades superpobladas. *Journal of Engineering and Applied Sciences in Industry*, 1(1), 10-20. <https://doi.org/10.53591/easi.v1i1.1770>

Articles in journal repositories are freely open in digital form. Authors can reproduce and distribute the work in any non-commercial site and grant the journal right of first publication with the work simultaneously licensed under a CC BY-NC-ND 4.0

Resumen. En este trabajo investigativo se evidenció la ineficiente recolección de basura, que ocasiona un grave problema de contaminación. El objetivo primordial es demostrar que la implementación de una arquitectura inteligente permite que la recolección de basura sea más sencilla. La metodología fue descriptiva, cuantitativa, con uso de una simulación de sensores, prototipo de IoT. Los resultados previstos indicaron que, con la recolección optimizada de basura, la reducción de tiempo y costos se genera el desarrollo de ciudades inteligentes y estándares de sana convivencia. La implementación del SYSTEM CALL: SMART TRASH COLLECTION (STC), arquitectura avanzada, permite reducir desechos, detectando contenedores sobrecargados mediante arquitectura distribuida, con cola de mensaje RabbitMQ. En conclusión, la utilización de esta arquitectura escalable es útil para ciudades que quieren reducir el impacto ambiental.

Palabras claves: Gestión inteligente de residuos, Internet de las cosas, ciudades inteligentes, prototipo de IoT, transmisión de datos, recogida de datos, sensores de IoT, simulación de datos.

Abstract. In this research work, the inefficient garbage collection was evidenced, which causes a serious pollution problem. The main objective is to demonstrate that the implementation of a smart architecture makes garbage collection easier. The methodology was descriptive, quantitative, with the use of a sensor simulation, IoT prototype. The expected results indicated that, with the optimized collection of garbage, the reduction of time and costs is generated by the development of smart cities and standards of healthy coexistence. The implementation of SYSTEM CALL: SMART TRASH COLLECTION (STC), advanced architecture, allows waste reduction, detecting overloaded containers through distributed architecture, with RabbitMQ message queue. In conclusion, the use of this scalable architecture is useful for cities which require to reduce environmental impact.

Keywords: Smart waste management, Internet of Things, smart cities, IoT prototype, data transmission, data collection, IoT sensors, data simulation.

1. INTRODUCCIÓN

La recolección de basura no es un tema del que muchos se preocupen, a pesar de que, con el pasar de los años, la población va aumentando a un ritmo veloz, lo que quiere decir que el incremento de desechos, basura, residuos, etc., se hace mucho más notable cada vez, especialmente en las grandes ciudades y su recolección, de igual manera es un problema.

Debido al crecimiento de la población urbana, la recolección de desechos sólidos se ha vuelto más complicada o mejor dicho, es más desafiante para las ciudades, por ello, una investigación proveniente de la revisión literaria, expone que, según la ONU Medio Ambiente, solo un porcentaje ligeramente superior a la mita (55%) de la basura, tiene un manejo óptimo en el mundo entero (Boggiano, 2021).

Teniendo en cuenta que esta problemática no solo trae consigo el incremento de la basura, también hace referencia a problemas medioambientales y la sostenibilidad de la ciudad (Courcelle et al., 1998). De modo que, la solución planteada consiste en una arquitectura distribuida que, captura y procesa la información de recolección de basura, tales como: las de rutas de recolección, distancia de los sensores del camión, etc.

Este tipo de arquitectura aplicado o propuesta presentada proporciona una mejor forma o manera de implementar rutas de recolección de basura en las distintas ciudades inteligentes que se explicara en su respectiva metodología, desarrollando los distintos puntos y objetivos presentados (Bass et al., 2012).

Al respecto, se revisó un estudio realizado por Pacheco (2021), en donde se expuso la problemática que atravesó un proyecto urbanístico en Tunja, Colombia, debido a los procedimientos que exigen las autoridades públicas, para evitar que la basura u otro tipo de obstáculos llegue hasta los acueductos y alcantarillados ubicados en este proyecto, razón por la cual, fue necesario el uso de una plataforma tecnológica denominada Open Smartflex, así como una aplicación de ArcMap, para favorecer el cumplimiento de la legislación de esa nación, en materia de diseños hidráulicos y sanitarios eficientes, a lo que se añadió la utilización de un programa Collector para el control y auditoría del proyecto en mención.

Otra investigación realizada por Apacani (2020), en la Universidad Privada Domingo Savio de Bolivia, se centró en el desarrollo de una aplicación Web, basado en la metodología SCRUM, con el firme propósito de potenciar la gestión de transportación de los desperdicios sólidos en el sector urbano, para el mejoramiento de un servicio urbano, que no era eficiente, porque no atendía los reclamos de la ciudadanía, con cambios intempestivos de las rutas y horarios de los camiones recolectores de basura, por consiguiente, la aplicación tecnológica en mención, podía contribuir con la solución de esta problemática.

Mientras tanto, la revisión literaria prosiguió hasta Argentina, en donde se pudo conocer de una investigación efectuada por Difabio et al. (2020), quien se refirió al desarrollo de un software que pueda ser instalada en teléfonos inteligentes o smartphones, para el mejoramiento del sistema de recolección de desechos sólidos en Viedma, Argentina. La aplicación móvil en cuestión debía contar con sistema operativo Android, bajo metodología SCRUM, demostrando que, además de minimizar los impactos ambientales que genera la contaminación causada por los residuos sólidos, se pudo evidenciar la asociación entre las concepciones del Internet de las cosas y las ciudades inteligentes (Smart Cities).

A nivel de la ciudad de Guayaquil, se encontró una investigación efectuada por Crespo y Vivanco (2019), en donde se planteó el desarrollo de una plataforma tecnológica, con el propósito de solucionar el problema del embotellamiento del tráfico vehicular en la localidad, para lo cual se empleó algoritmos matemáticos que, permitan el aprovechamiento del análisis de los puntos georreferenciados y para el establecimiento de una aplicación Web que, además de brindar una opción para mejorar la vialidad en el tráfico urbano, también puede ser utilizado como una opción viable en la recolección de desechos sólidos.

El estado del arte demuestra que las plataformas tecnológicas pueden contribuir con el fortalecimiento de los sistemas de recolección de residuos sólidos en todo el mundo, debido a que, no solo han sido propuestos en investigaciones antecedentes, sino que, también se pudo observar que en las ciudades inteligentes ya se han puesto en marcha, con resultados positivos.

Por consiguiente, como objetivo general se cita el siguiente: demostrar que la implementación de una arquitectura inteligente permite que la recolección de basura sea más sencilla. Mientras que las preguntas de investigación que se van a despejar indican: ¿cuál idea de arquitectura sería más efectiva? ¿Qué diseño y modelo de arquitectura de software es escalable y completo?

2. MARCO TEÓRICO

En este marco teórico se pretende analizar, desde el punto de vista de la revisión bibliográfica, los fundamentos conceptuales del servidor de cola de mensaje RabbitMQ, la plataforma de desarrollo Python, la base de datos NoSQL y el *Internet of Things*, con el propósito de facilitar el análisis de los mismos, en la sección inherente a los resultados.

2.1. Servidor de cola de mensaje: RabbitMQ

Para procesar las operaciones y comunicar las diferentes partes de un sistema es necesario la existencia de una cola de mensaje, estas pueden simplificar de forma bastante significativa la escritura del código para las aplicaciones,

a su vez mejoran el rendimiento a la vez que estas escalan. Según Caiza et al. (2020), en la actualidad existe una variedad de estos servicios en línea, algunos de estos ejemplos son Kafka, Exchange Server, y RabbitMQ, siendo este último el designado a ser participe para la arquitectura de este proyecto.

RabbitMQ es un servidor de intercambio de mensajes, su sistema de cola almacena los mensajes de los remitentes y los almacena, esperando que los receptores los utilicen. Rabbit pertenece a la categoría de middleware de mensajería. De forma simplificada, se define una cola en RabbitMQ, que almacenará los mensajes enviados por el productor hasta que la aplicación consumidora reciba el mensaje y lo procese. El productor publica un mensaje, pero no lo envía directamente al consumidor, sino que lo pasa al intercambio (Vera, 2018).

En la transmisión de los mensajes, según Vivas (2020), participan cuatro estaciones:

- Productor: crea los mensajes.
- Intercambiador: entrega los mensajes.
- Cola: almacena los mensajes.
- Consumidor: procesa el mensaje.

Bajo lo aseverado en los párrafos, se sostiene que es beneficioso el uso de RabbitMQ en este estudio, el cual requiere un elevado nivel de confiabilidad que, este servicio en línea puede dotar, el cual, además, permite un enrutamiento flexible y mecanismos seguros para la autenticación de datos, generando un alto grado de precisión en el envío de mensajes a la ciudadanía, por parte del personal perteneciente a la empresa responsable de la recolección de basura.

2.2. Plataforma de desarrollo: Python

Como vimos anteriormente las colas de mensaje son los encargados de simplificar la escritura del código, pero ¿qué código precisamente? Al existir una variedad de lenguajes de programación, algunos más populares que otros, es difícil determinar en qué lenguaje será desarrollado el sistema. Para comodidad de muchos programadores, Python es una de las opciones más versátiles que cuenta con las funcionalidades necesarias para realizar este tipo de trabajos con bastante fluidez.

Python es un lenguaje de programación creado por Guido van Rossum a principios de los 90, y su nombre fue inspirado por el comediante inglés "Monty Python". Su sintaxis es simple y clara. La escritura dinámica, un administrador de memoria, una gran cantidad de bibliotecas disponibles y las potentes funciones del lenguaje hacen que desarrollar aplicaciones con Python sea fácil, rápido y, lo que es más importante, divertido (Mendoza & Trujillo, 2022).

La sintaxis de Python es tan simple que a veces parece un pseudocódigo. El gran mito sobre Python es su bajo rendimiento. Esto no es del todo correcto, porque si bien es un lenguaje interpretado y suele ser más lento que los lenguajes compilados, Python se diferencia de otros lenguajes interpretados en que ha implementado toda la biblioteca estándar en lenguaje C, lo que hace que sus funciones originales sean muy efectivas (Chun, 2001).

En síntesis, para facilitar la ejecución de aplicación Web, es indispensable contar con una plataforma tecnológica que viabilice el uso de herramientas fiables, bajo un lenguaje de programación adecuado. Queda claro que la elección de Python, se debe a su amplitud, rapidez y diseño, como características esenciales para garantizar la optimización de la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos.

2.3. Base de datos NoSQL: MongoDB Atlas

Los cuellos de botellas son un problema cotidiano en aplicaciones distribuidas de gran escala, en este caso al ser una aplicación dirigida no solo a operadores de recolección, sino al público en general, los cuales pueden tener acceso a la información, es importante determinar el servidor de la base de datos. Lo más recomendable son aquellas bases de tipo NoSQL, puesto que su escalabilidad se realiza de forma horizontal. Muchos datos generados por redes sociales, Internet de las cosas, sensores, transacciones, etc., están creciendo significativamente cada día. MongoDB es un tipo de software que permite a los usuarios crear y manipular bases de datos en forma de los llamados documentos, que se almacenan en formato BSON (Gómez & Carrión, 2021).

A diferencia de las bases de datos relacionales, la creación de una base de datos no necesita definir su arquitectura de antemano. En comparación con las bases de datos relacionales, esta función permite a los desarrolladores crear un sistema más flexible para futuras modificaciones de la estructura del documento. MongoDB es uno de los sistemas de base de datos NoSQL más populares, por lo que a menudo se contrasta con los sistemas relacionales y no relacionales (Marrero et al., 2021).

Esto significa que, la base de datos NoSQL, Mongo DB Atlas, puede trabajar con pequeños dispositivos, que mejoran la agilidad en las iteraciones y sobre todo, son flexibles, permitiendo que se puedan realizar cambios en la

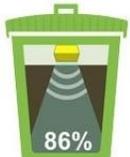
estructura del documento, según sea conveniente, a lo que se añade, el acceso a un número mayor de usuarios, beneficios por los cuales se ha elegido a esta base de datos, como uno de los fundamentos esenciales para la garantizar la optimización de la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos.

2.4. Internet of things: sensors

La conexión de múltiples medios dentro de un sistema automático, pertenecientes a la red es denominado Internet de las Cosas (Iot), esto permite el intercambio de información o la ejecución de acciones asíncronas; pero para ello es necesario la inclusión de ciertos dispositivos específicos en el sistema. Estos dispositivos son los encargados de proporcionar la comunicación junto con los demás elementos, sus límites van desde chips, antenas o sensores de cualquier tipo y escala (Barrio, 2020).

En este proyecto, según Novillo et al. (2018), los dispositivos pertenecientes al IoT son los sensores listados en la Tabla 1.

Tabla 1. Información de los sensores

Sensor	¿Para qué sirve?	Emite	Visualización
GPS, Ublox NEO 7M	Sirve para dar una posición más precisa de la ubicación del contenedor de basura que emite la alerta del nivel de contenido	Ubicación (Altitud y Longitud)	
Sensor de distancia, VL53L0X	Sirve para medir la distancia entre dos ubicaciones (Ubicación del contenedor y Ubicación del Camión de la basura)	Distancia (Distancia en cm)	
Sensor de ultrasonido	Sirve para medir el nivel del llenado de un contenedor de basura (Emite un sonido y mide el tiempo que tarda en recibir la señal, dependiendo del tiempo se determina el nivel de llenado).	Nivel de llenado (Nivel contenido - %)	

Fuente: Ramalho (2020)

En consecuencia, el internet de las cosas describe los sensores y dispositivos diversos que se conectan al Internet, para promover el intercambio de datos entre los mismos, entrelazándose con la automatización de los procesos, por consiguiente, representa una de las bases de este estudio, en donde se pretende garantizar la optimización de la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos.

3. METODOLOGÍA

La metodología aplicada en la presente investigación, es cuantitativa, descriptiva, de campo, basada en una simulación de la plataforma tecnológica *Smart Trash Collection* (STC), mediante la cual se pretende optimizar la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos, lo que además se puede demostrar que, la implementación de una arquitectura inteligente permite que la recolección de basura sea más sencilla, en cumplimiento del objetivo de la investigación. Para el efecto, se utilizó como instrumento investigativo, la matriz de datos que permitió su recopilación adecuada.

En cuanto al proceso investigativo, este inicia con el diseño de la arquitectura de la plataforma tecnológica *Smart Trash Collection* (STC), para luego proseguir con el desarrollo del tablero Kanban y la simulación del modelamiento de datos, para optimizar la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos y mejorar la comunicación con los usuarios.

El proceso investigativo culmina con la lista de datos de los contenedores, en una base electrónica que incluye las distancias recorridas, la longitud y altitud de estos vehículos, las horas de cada día, en que se realiza la recolección de desechos, así como el nivel y el estado actual de cada uno de ellos.

4. RESULTADOS Y DISCUSIÓN

4.1. Arquitectura

El diseño de la arquitectura permite expresar gráficamente, cómo la plataforma tecnológica *Smart Trash Collection* (STC), puede optimizar la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos y mejorar la comunicación con la ciudadanía, para simplificar el proceso de recolección de desechos y hacerlo más eficiente, para lo cual se utilizan los diagramas *Landscape* y de arquitectura (Fig. 1-2).

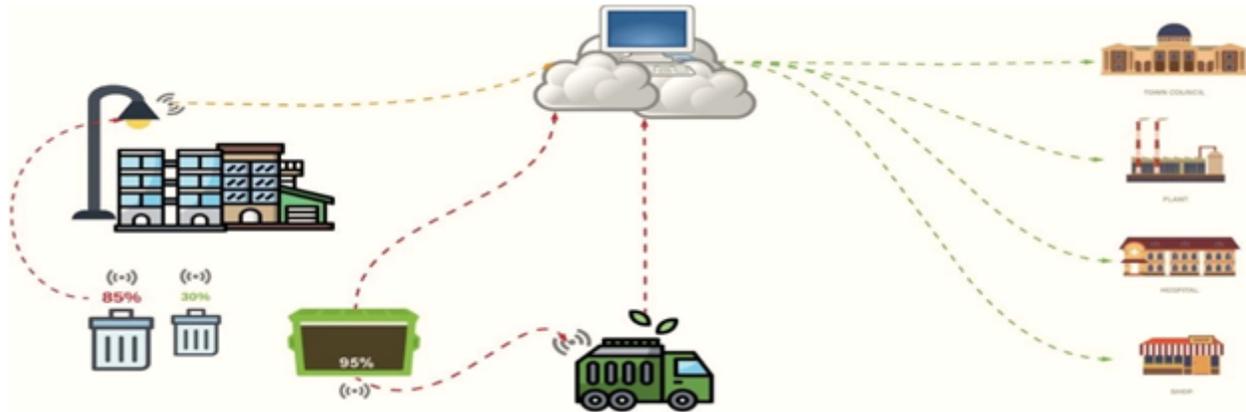


Figura 1. Diagrama Landscape

Como se puede apreciar en el diagrama *Landscape*, la comunicación entre los carros recolectores de desperdicios sólidos, se encuentran intercomunicados con la empresa responsable de este servicio que, a su vez, debe interactuar continuamente con el I. Municipio de Guayaquil, así como con la ciudadanía en general, beneficiario de este servicio público, donde la base de datos NoSQL MongoDB Atlas, interconecta a cada uno de los involucrados en el proyecto, a través de los dispositivos tecnológicos pertinentes.

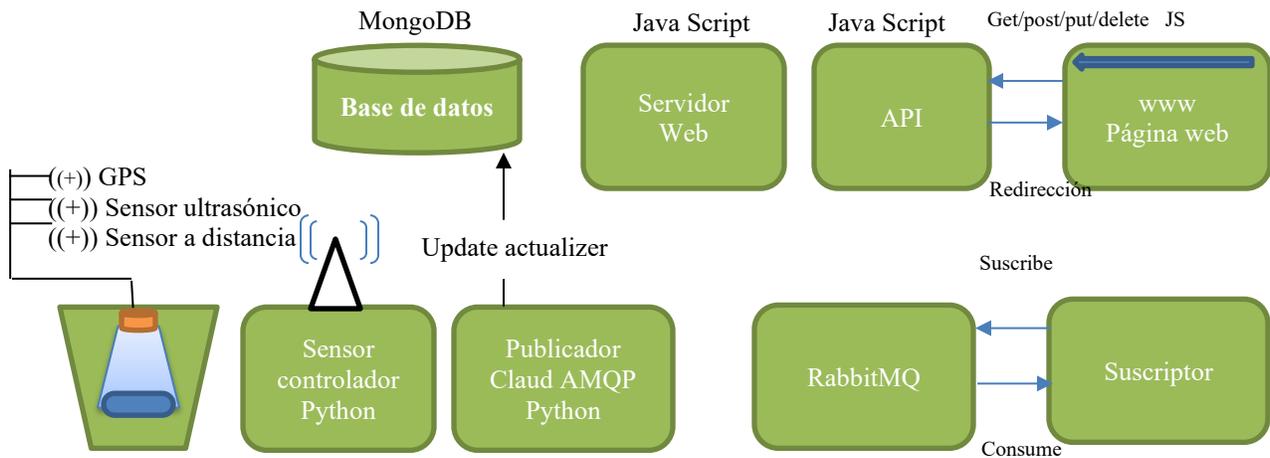


Figura 2. Diagrama de la arquitectura

Con relación al diagrama de arquitectura, se presenta la fluencia de datos, desde la base NoSQL MongoDB Atlas, cuyos datos fluyen hacia el servidor Web y la respectiva página web, mientras que, la plataforma de desarrollo: Python, actúa de publicador conectado a los sensores de control que, a su vez, permiten el trabajo eficiente del servidor de cola de mensaje: RabbitMQ, que facilita el envío y archivo de los mensajes a los suscriptores del servicio de recolección de residuos sólidos.

4.2. Metodología de desarrollo: Tablero Kanban

Llevar el control del flujo de trabajo, es una tarea importante para equipos pequeños de trabajo, sobre todo el evitar la sobrecarga de tareas a uno o más miembros del equipo; la mejor manera de gestionar el desarrollo general de las tareas es mediante la visualización de las tareas y sus estados. La metodología Kanban permite llevar el trabajo con fluidez hasta que las tareas sean completadas en su totalidad en un tiempo estimado (Yépez & Armijos, 2020).

En la Figura 3 se presenta el detalle del tablero Kanban asociado a los procesos, para optimizar la distribución de las rutas de transporte en el servicio de recolección de residuos sólidos y mejorar la comunicación con la ciudadanía, bajo el uso de la plataforma tecnológica *Smart Trash Collection* (STC).

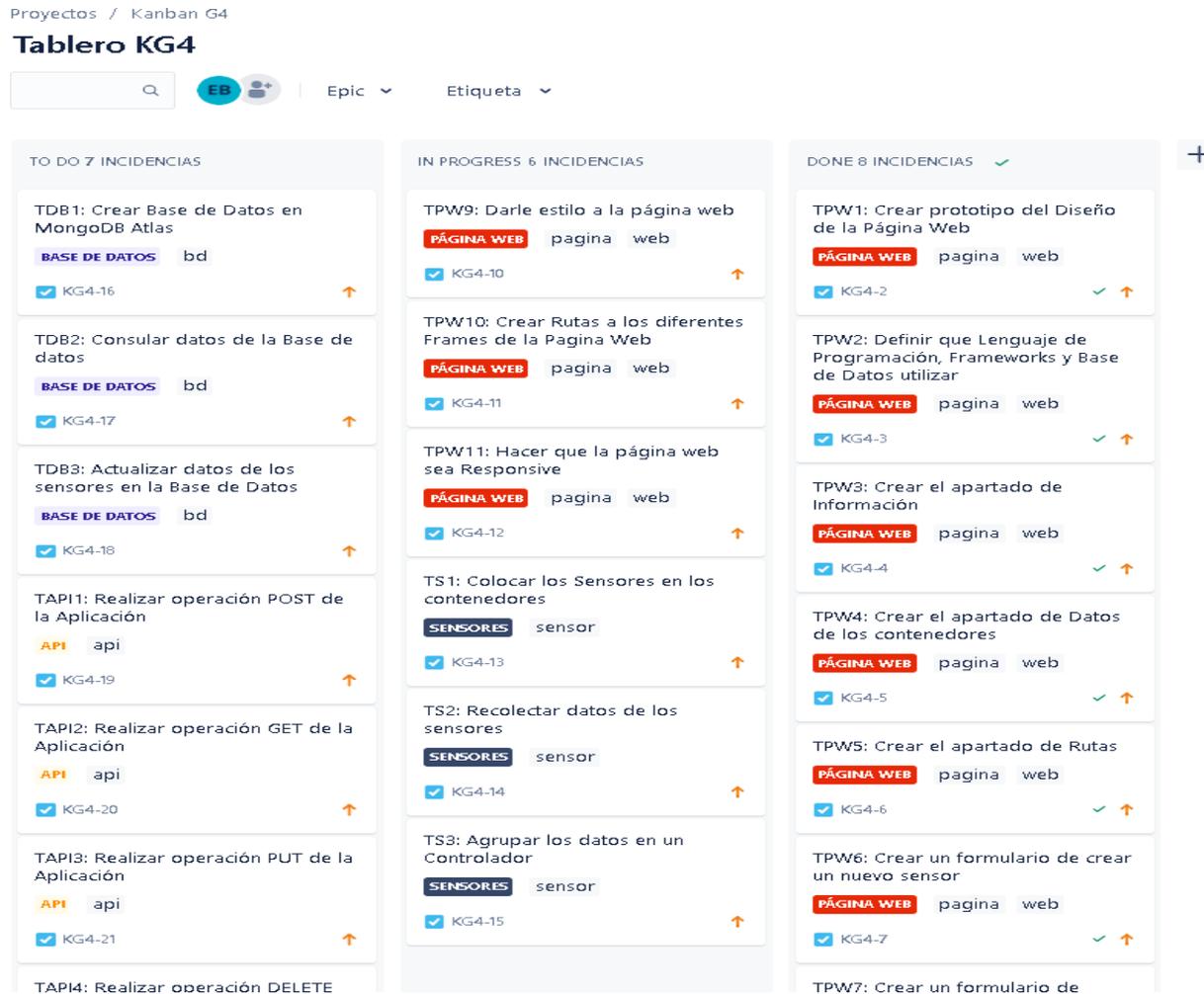


Figura 3. Tablero Kanban

El flujo del proceso indica que, el primer paso para la implementación de la tecnología en el servicio de recolección de residuos sólidos consiste en la creación de la base de datos NoSQL MongoDB Atlas, la cual debe ser consultada y actualizada convenientemente, para luego realizar las operaciones POST, GET y PUT de la aplicación. Por otra parte, es necesario que se realice la creación del prototipo del diseño de la página Web, definiendo el lenguaje de programación, frameworks, desarrollando los apartados inherentes a la información, de los contenedores, rutas, formulario de sensores nuevos, diseñando el estilo de la página Web, para luego, colocar los sensores en los contenedores de los carros recolectores, agrupando los datos en los controladores, para promover la optimización en la distribución y recorrido de las rutas de transporte en el servicio de recolección de residuos sólidos, a la vez que se puede mejorar la comunicación con las comunidades beneficiarias.

4.3. Simulación de modelamiento de datos

Diseñada la página Web y desarrollada la base NoSQL MongoDB Atlas, se debe operar la aplicación correspondiente en el proceso de recolección de basura, para garantizar una óptima distribución y recorrido de las rutas de los vehículos correspondientes, así como el mensaje entre el personal que opera estas unidades y los habitantes de la colectividad a quienes sirven. Los resultados de la simulación con el sensor GPS se muestran en la Tabla 2.

Tabla 2. Tabla de datos: simulación sensor GPS

	ID: GPS	Latitud	Longitud	Altitud	Rumbo	Velocidad	Satélite
GPS 1	01	15°	300	86	5.00	33	45
GPS 2	02	32°	75	56	90.00	10	62
GPS 3	03	121°	13	20	270.00	45	13

Fuente: los autores

La tabla de datos que muestra los resultados de la simulación del sensor GPS, describe la ubicación de los vehículos, considerando 3 GPS, indicando su latitud, longitud y altitud correspondiente, así como el rumbo y la velocidad que recorren en tiempo real, cada uno de los medios de transporte que formaron parte de la simulación respectiva.

Estos resultados tienen algunas similitudes con los indicados en el estudio de Difabio et al. (2020), el cual también demuestra los datos referentes a la ubicación de los medios de transportación de desechos sólidos, cuyo propósito final consistió en minimizar la contaminación ambiental causada por estos residuos, con base en la utilización de la tecnología de punta, bajo metodología SCRUM y las concepciones del Internet de las cosas, para lograr la optimización del proceso de distribución de vehículos y contenedores de basura. Al realizarse la simulación con un sensor de distancia, la operación arrojó los datos mostrados en la Tabla 3.

Tabla 3. Simulación sensor de distancia

	ID Dist	Latitud	Longitud	Altitud
Distancia 1	01	12	10	34
Distancia 2	02	55	53	50
Distancia 3	03	9	23	33

Fuente: los autores

La Tabla 3 describe la distancia recorrida por los vehículos, considerando 3 distancias, indicando su latitud, longitud y altitud correspondiente, en un tiempo determinado. Estos resultados guardan alguna concordancia con los expresados en el estudio de Apacani (2020) quien, en los hallazgos de su simulación, demostró la mejor distribución de los vehículos y contenedores, identificando las características de la distancia en que se encuentran en cada caso específico. Al realizarse la simulación con un sensor ultrasonido, la operación arrojó los datos mostrados en el Tabla 4.

Tabla 4. Simulación sensor ultrasonido

	ID_ULT	Nivel	Estados
Ult-1	01	3	-
Ult-2	02	1	+
Ult3	03	8	+

Fuente: los autores

Los resultados de la simulación anterior describen los datos del ultrasonido, considerando 3 unidades, indicando su nivel y estados correspondientes en un tiempo determinado. Al realizarse la simulación final, la operación arrojó los siguientes datos (Tabla 5).

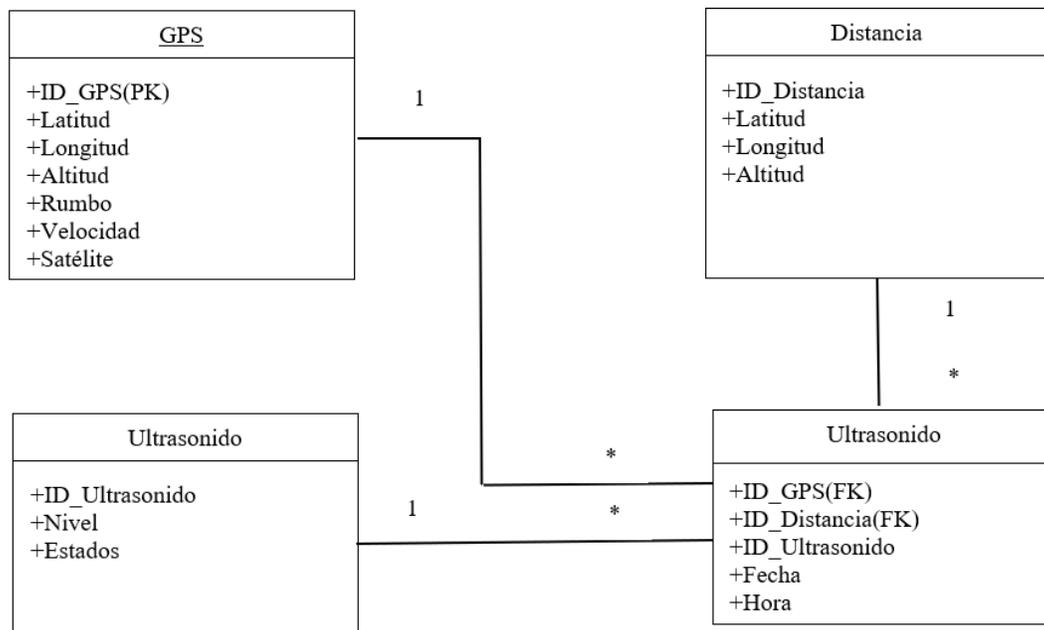
Tabla 5. Simulación final

	ID_GPS	ID_ULT	ID_Dist	Fecha	Hora
Sensor 1	01	02	03	12/01/20	13:30
Sensor 2	03	03	02	13/01/20	07:15
Sensor 3	02	01	01	12/01/20	22:10

Fuente: los autores

La tabla 5 describe los datos de los sensores GPS, distancia y ultrasonido, considerando 3 unidades, indicando además de los resultados obtenidos en las tablas anteriores, la fecha y hora en que se registraron los mismos datos en mención. Estos hallazgos tienen algunas similitudes con el estudio de (Bass et al., 2012), demostrando que es posible la optimización en la distribución de contenedores para evitar sobrecargas, y así minimizar la contaminación ambiental, bajo el uso de la herramienta tecnológica en cuestión.

A continuación, se presenta el esquema del modelo entidad/relación (Fig. 4), considerando el proceso de la simulación de los sensores señalados:

**Figura 4.** Modelo entidad/relación

4.4. Mapa y datos de contenedores

La Figura 5 muestra el mapa de los contenedores, información base para optimizar su distribución a lo largo de la ciudad de Guayaquil. Se observan los contenedores en diferentes colores, rojo, blanco, verde y amarillo, según su ubicación a lo largo de la ciudad de Guayaquil. Finalmente, se presenta la lista de los contenedores cuya base de datos se obtiene de las Tablas 2-4 (Fig. 6). La base de datos de los contenedores debe mantener una concordancia con aquellos obtenidos en la simulación final.

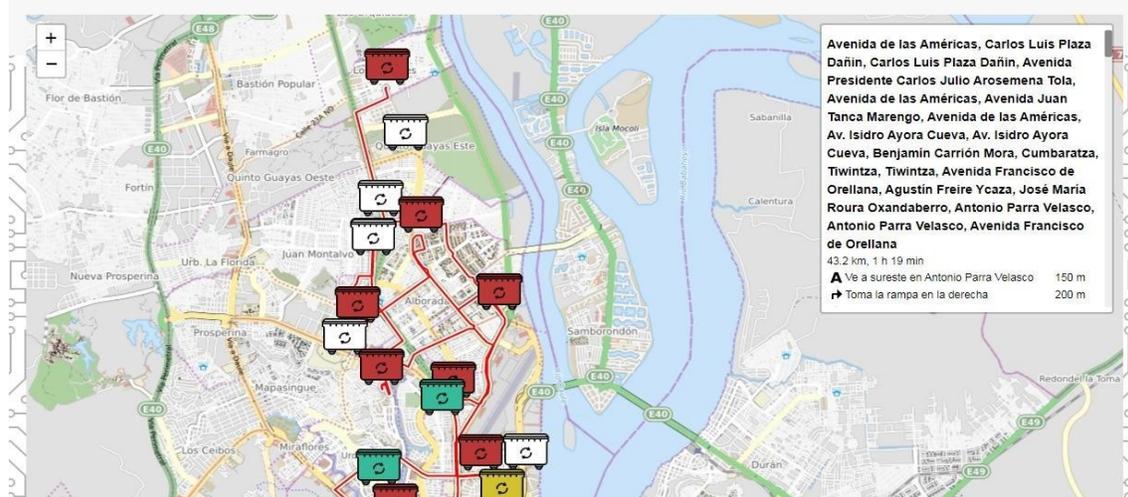


Figura 5. Mapa de contenedores distribuidos

Datos Rutas Problema Arquitectura Desarrollo Diseño Solución

Lista de los Datos de los Contenedores

New Sensors

ID	Distancia	Altitud	Longitud	Fecha	Hora	Nivel	Estado		
1	185.157508	-2.138895	-79.884006	15-02-2021	21:35:26	0 %	Inactivo		
2	205.15	-2.159881	-79.894718	15-02-2021	21:34:49	35 %	Activo		
3	10.25	-2.170901	-79.887401	02/10/21	22:50:00	100 %	Activo		
4	10.15	-2.170646	-79.879059	02/10/21	12:50:00	0 %	Inactivo		
5	1000.2	-2.177755	-79.883495	02/10/21	22:46:35	50 %	Activo		
6	500	-2.173715	-79.906529	02/10/21	22:46:35	25 %	Activo		
7	222.22	-2.180596	-79.903432	02/10/21	22:46:35	90 %	Activo		
8	1000.96	-2.156422	-79.892653	02/10/21	22:46:35	100 %	Activo		
9	1000.2	-2.107398	-79.901477	02/10/21	22:46:35	0 %	Inactivo		
10	233.03	-2.120365	-79.90613	02/10/21	22:46:35	0 %	Inactivo		
11	1000.2	-2.123576	-79.89863	02/10/21	22:46:35	75 %	Activo		
12	1000.2	-2.141425	-79.910485	02/10/21	22:46:35	78 %	Activo		
13	1000.2	-2.128002	-79.907524	02/10/21	22:46:35	80 %	Activo		
14	1000.2	-2.153873	-79.905893	02/10/21	22:46:35	97 %	Activo		
15	185.15750861478452	-2.138895	-79.884006	01/10/21	23:46:35	85 %	Activo		
16	185.15750861478452	-2.138895	-79.884006	15-02-2021	21:08:13	85 %	Activo		
17	12345	-2.12809	-79.9075	15-02-2021	21:08:56	0 %	Inactivo		
18	20	-2.147826	-79.912795	15-02-2021	21:10:53	100 %	Inactivo		
19	200	-2.094285	-79.904902	15-02-2021	20:22:13	85 %	Activo		

Figura 6. Datos de los contenedores

CONCLUSIÓN

La implementación de una arquitectura inteligente que permite la recolección de basura de un modo más eficiente y óptimo, reduciendo tiempos y costos de recolección, impacta positivamente en la gestión de residuos de ciudades inteligentes, y su manera o normas de convivencia. Esta implementación denominada *System call STC*, es una arquitectura de alto nivel que optimiza el complejo proceso de recolección de basura en ciudades densamente pobladas.

Se propone que la arquitectura más accesible para optimizar la distribución de los contenedores y los carros recolectores de residuos sólidos se encuentre conformada por la plataforma de desarrollo: Python, base de datos NoSQL MongoDB Atlas, servidor de cola de mensaje: RabbitMQ, *Internet of Things*, en base a sensores GPS de distancia y de ultrasonido.

Esta arquitectura también ayuda con los problemas ambientales ya que al reducir los desechos se reduce el impacto ambiental. Las ciudades inteligentes son capaces de entender el significado de esto, por lo que han optado por la aplicación de esta arquitectura. Se debe considerar las ventajas de esta implementación, sus ventajas son mayores que sus desventajas así que esta arquitectura es una herramienta útil y eficaz para las ciudades que desean reducir su impacto ambiental y mejorar su estilo de vida.

Declaración de conflicto de intereses

Los autores declaran que no existe ningún conflicto de interés potencial dentro de esta investigación, autoría y/o publicación de este artículo.

REFERENCIAS

- Apacani, C. (2020). Aplicación web para fortalecer la gestión del transporte en la recolección de residuos sólidos urbanos. *Ingeniería y sus alcances*, 4(10), 192-207. https://doi.org/http://repositorio.cidecuador.org/bitstream/123456789/1401/1/Articulo_No_2.pdf
- Barrio, M. (2020). *Internet de las Cosas*. Madrid: Reus. https://doi.org/https://books.google.es/books?hl=es&lr=&id=0BE5EAAAQBAJ&oi=fnd&pg=PA9&dq=Internet+de+las+cosas&ots=JK85K_BQEZ&sig=WXXRFWWIx14P58U9EhEMo1FHSsA#v=onepage&q=Internet%20de%20las%20cosas&f=false
- Bass, L., Clemente, P., & Kazman, R. (2012). *Software Architecture in Practice*. Boston: Addison-Wesley Professional. [https://doi.org/https://www.scirp.org/\(S\(oyulxb452alnt1aej1nfow45\)\)/reference/ReferencesPapers.aspx?ReferenceID=2055391](https://doi.org/https://www.scirp.org/(S(oyulxb452alnt1aej1nfow45))/reference/ReferencesPapers.aspx?ReferenceID=2055391)
- Boggiano, M. (2021). Diagnóstico y caracterización de los residuos sólidos domiciliarios de la ciudad de Trujillo – Perú, 2019-2020. *Ciencia y Tecnología*, 17(3), 61-72. <https://doi.org/10.17268/rev.cyt.2021.03.05>
- Caiza, G., Álvarez, E., Remache, E., Ortiz, A., & García, M. (2020). Comparación de AMQP y CoAP para la integración de las comunicaciones en el área de producción. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 1(E26), 652-666. <http://www.risti.xyz/issues/ristie26.pdf>
- Chun, W. (2001). *Core python programming*. Ohio: Prentice Hall Professional.
- Courcelle, C., Kestemont, M., Tyteca, D., & Installé, M. (1998). Assessing the economic and environmental performance of municipal solid waste collection and sorting programmes. In: Assessing the economic and environmental performance of municipal solid waste collection and sorting programmes. *Waste Management & Research*, 1(2), 253-262.
- Crespo, V., & Vivanco, J. (2019). *Plataforma tecnológica para contribuir la planeación urbana de la ciudad de Guayaquil dirigido a la transportación, enfocado a la integración de los algoritmos Kmeans, DbSCAN, Hierarchical Clustering estructurado y no estructurado al módulo de análisis*. Guayaquil: Universidad de Guayaquil. <http://repositorio.ug.edu.ec/handle/redug/39879>
- Difabio, L., Vivas, H., & Muñoz, H. (2020). Internet de las cosas aplicada a la trazabilidad de la recolección de residuos en ciudades inteligentes. *Revista Ibérica de Sistemas e Tecnologías de Informação (RISTI)*, 2(E29), 479-493. <http://www.risti.xyz/issues/ristie26.pdf>
- Gómez, E., & Carrión, J. (2021). Plataformas de visualización de datos tolerantes a fallos por medio de MongoDB. *Ecociencia*, 8(2), 15. <https://doi.org/10.21855/ecociencia.82.386>

- Marrero, L., Thomas, P., Pasini, A., et al. (2021). *Aspectos de ingeniería de software, bases de datos relacionales, y bases de datos no relacionales y bases de datos como servicios en la nube para el desarrollo de software híbrido*. UNLP. <http://sedici.unlp.edu.ar/handle/10915/120139>
- Mendoza, M., & Trujillo, B. (2022). *Evaluación, análisis y comparación del rendimiento de programas de procesamiento masivo implementados usando lenguajes de programación Java, Python y C++ sobre la plataforma Hadoop para clústeres de varios tamaños*. Ibarra: Universidad Técnica del Norte. <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/43527>
- Novillo, J., Hernández, D., Mazón, B., Molina, J., & Cárdenas, O. (2018). *Arduino y el internet de las cosas*. Buenos Aires: Área de Innovación y Desarrollo, S. L. <http://dx.doi.org/10.1/993/Ingytec.2018.45>
- Pacheco, M. (2021). *Aplicación de plataformas tecnológicas y normativas para la gestión técnica del desarrollo urbano en la empresa Veolia Aguas de Tunja S.A. E.S.P.* Universidad Santo Tomás. <https://doi.org/10.13140/RG.2.2.33942.47680>
- Ramalho, M. (2020). *SmartGC: una arquitectura de software para la recolección de basura en ciudades inteligentes*. Sao Paulo: UNLP.
- Vera, E. (2018). *Diseño de una arquitectura servidora para la gestión de soluciones basadas en Crowdsensing*. [Disertación de Maestría, Universidad Politécnica de Valencia]. <https://riunet.upv.es/handle/10251/94307>
- Vivas, O. (2020). *Migración de REDIS a RABBITMQ en aplicación Web de mantenimiento de enlace de rotos*. [Disertación de carrera, Universidad de Los Llanos]. <https://www.encisosystems.com/sites/default/files/2021-05/21vivas.pdf>
- Yépez, E., & Armijos, K. (2020). *Aplicación de la metodología kanban en el desarrollo del software para generación, validación y actualización de reactivos, integrado al sistema informático de control académico UNACH*. [Disertación de carrera, Universidad Nacional de Chimborazo]. <http://dspace.unach.edu.ec/handle/51000/6457>