



Desarrollo de un sistema web y aplicación móvil para el apoyo a la gestión operativa de la recolección de residuos sólidos urbanos mediante análisis de incidencias ciudadanas georreferenciadas y Machine Learning en la ciudad de Quito

Mena Pérez James Steve y Ortiz Tomalo Bryan Andres

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en Tecnologías de la Información

Ing. Washington Eduardo Loza Herrera Mgtr.

3 de agosto de 2026



Certificado de análisis
Compilatio Magister+ | ESPE-ECU

Tesis_MIC-EMASEO-V14

ID : 89b13c7e901f100a9e147c09a488a99443042764



6%
Textos sospechosos

Nombre del fichero : Tesis_MIC-EMASEO-V14.txt
Tamaño del archivo original : 4,2 MB
Número de palabras : 23.023

Depositante : WASHINGTON EDUARDO LOZA HERRERA
Fecha de depósito : 23 de julio de 2026
Tipo de carga : interface
fecha de fin de análisis : 23 de julio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes 2%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA 6%

Textos estilísticamente próximos a un texto generado por una IA.

Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



Idiomas no reconocidos 6%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.

Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.



No incluido en el porcentaje de textos sospechosos :

Textos entre comillas <1%

Pasajes entre comillas, a menudo indicativos de una cita.



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
WASHINGTON EDUARDO
LOZA HERRERA

Ing. Loza Herrera, Washington Eduardo. Mgtr.

Director



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: "**Desarrollo de un sistema web y aplicación móvil para el apoyo a la gestión operativa de la recolección de residuos sólidos urbanos mediante análisis de incidencias ciudadanas georreferenciadas y Machine Learning en la ciudad de Quito**" fue realizado por los señores **Mena Pérez James Steve y Ortiz Tomalo Bryan Andres**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE; habiéndose efectuado una revisión integral de forma y de fondo conforme a los lineamientos y formatos institucionales vigentes. Además, fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 3 de agosto de 2026



Ing. Washington Eduardo Loza Herrera Mgtr.

C.C.: 1713756524



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Responsabilidad de Autoría

Nosotros, **Mena Pérez James Steve** y **Ortiz Tomalo Bryan Andres**, con cédula de ciudadanía N° 1752784460, N° 1754336160, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: "**Desarrollo de un sistema web y aplicación móvil para el apoyo a la gestión operativa de la recolección de residuos sólidos urbanos mediante análisis de incidencias ciudadanas georreferenciadas y Machine Learning en la ciudad de Quito**" es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 3 de agosto de 2026



**James Steve Mena
Perez**



Mena Pérez James Steve

C.C.: 1752784460



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
**BRYAN ANDRES ORTIZ
TOMALO**

Ortiz Tomalo Bryan Andres

C.C.: 1754336160



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Autorización de Publicación

Nosotros, **Mena Pérez James Steve** y **Ortiz Tomalo Bryan Andres** con cédula de ciudadanía **1752784460**, **1754336160** autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: "**Desarrollo de un sistema web y aplicación móvil para el apoyo a la gestión operativa de la recolección de residuos sólidos urbanos mediante análisis de incidencias ciudadanas georreferenciadas y Machine Learning en la ciudad de Quito**" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 3 de agosto de 2026



Mena Pérez James Steve

C.C.: 1752784460



Ortiz Tomalo Bryan Andres

C.C.: 1754336160

Dedicatoria

Dedico este trabajo y este logro conseguido, en primer lugar, a mi padre, Jaime, que ha sido ese pilar inquebrantable que siempre supo guiarme con su ejemplo. Por cada consejo dicho en el momento justo, por su paciencia infinita y por enseñarme que nunca debo rendirme ante cualquier obstáculo. Gracias por tenderme la mano para guiarme por el camino correcto y por todo tu apoyo incondicional. Es por eso que esta es mi forma de demostrarte todo ese esfuerzo, hasta que algún día me llames "ingeniero" con orgullo.

A mi hermano, Dennis, por ser siempre mi inspiración para seguir hacia adelante. Todas esas ideas ingeniosas y únicas que me has brindado han contribuido mucho en mi vida. Por eso celebro esta victoria contigo, porque siempre has creído en mí y me has dado ese apoyo incondicional.

A mi hermana, Nicole "Gata", por ser ese lazo irrompible que ha unido siempre a nuestra familia, por las risas, el apoyo incondicional y por demostrarme siempre que la vida hay que disfrutarla.

A mi querida familia, nuestro hogar, que me ha cobijado ante los obstáculos de la vida con su inmenso cariño, por acompañarme en cada etapa de este camino y demostrarme que el esfuerzo siempre tiene su recompensa.

Y para finalizar, dedico este logro también a mi gran amiga Karen. Siempre has sido esa persona con la que puedo contar. Gracias por escucharme en esos momentos complicados y por brindarme tu amistad, que ha sido muy especial por la gran persona que eres.

James Mena

Dedicatoria

A mis queridos padres les dedico este triunfo, por ser mi ejemplo de superación y estar presentes siempre. Ustedes les debo todo lo que soy y seré, ya que sin su amor y apoyo nada de esto hubiera sido posible. Ustedes son mi mayor orgullo.

A ti padre Juan Ortiz por el apoyo en las condiciones adversas que se atravesaban, por soportar el peso de la familia, por siempre creer en ese chico que alguna vez pregunto si tú estarías ahí cuando más lo necesitará.

A mi madre Adriana Tomalo por siempre ser la mujer más maravillosa que he conocido, por el cariño incondicional que me brindaste y me han mantenido hasta el día de hoy en pie y por aquellas lagrimas que te hice derramar, por eso hoy quiero dedicarte un logro de vida madre.

A mis hermanos, por regalarme esas alegrías dentro del caos, por regalarme esas sonrisas que tanto amo y me dan aliento en mi vida.

A mi novia Pamela Carriel por estar aquellos momentos en lo que nadie más entendía mi situación y siempre se detuvo a pensar en mí. Por su amor incondicional, comprensión y el acompañarme durante toda esta etapa.

Bryan Ortiz

Agradecimiento

Quiero agradecer en primera instancia a Dios, porque con una fe inquebrantable que me permitió llegar hasta este momento, siempre rezando para que, con Su guía, todos los obstáculos fueran superados. Agradezco también por todos esos momentos únicos que trajo consigo la culminación de este camino, el cual representó un gran desafío para mí.

Agradezco a la prestigiosa Universidad de las Fuerzas Armadas ESPE por darme la oportunidad de demostrar todo mi esfuerzo y dedicación por medio de este trabajo. Cursar la carrera de Tecnologías de la Información ha sido un reto sumamente innovador, dada la amplia variedad de ramas y caminos que se pueden tomar. También agradezco al Mgtr. Washington Loza, docente de la institución y tutor del presente proyecto, por brindarnos su excelente guía a lo largo de este difícil camino que ha sido el desarrollo del proyecto, y por siempre brindarnos su apoyo cuando lo necesitábamos.

Quiero expresar mi más sincero agradecimiento, desde lo más profundo de mi corazón, a todas aquellas personas que, de una u otra manera, hicieron posible la culminación de este trabajo. Cada uno de ellos dejó una huella imborrable en este proceso y en mi vida.

Es por ello que les doy infinitas gracias. Este logro es tan de ustedes como mío, porque sin su presencia, su amor y su apoyo, nada de esto habría sido posible. Llevo cada uno de sus nombres en mi corazón y en cada página de este trabajo.

James Mena

Agradecimiento

En primer lugar, quiero dar gracias a mi padre Dios por darme la oportunidad de tener la alegría de cumplir uno de mis sueños, de poder seguir en pie con salud y vida para forjarme no solo como profesional si no como persona.

Expreso mi más sincero agradecimiento a las Universidad de las Fuerzas ESPE por darme la oportunidad de estudiar la carrera de Tecnologías de la Información en tan prestigiosa institución. De manera especial, agradezco al Mgtr. Washington Loza, docente de la institución y tutor del presente proyecto por el apoyo desde el inicio y no rendirse en el camino.

A mi familia, por su compañía, por el apoyo brindado durante todo este recorrido académico. Gracias por su cariño y creer en mí.

A mi novia Pamela Carriel, por siempre estar allí cuando la necesitaba, su empatía y apoyo constante. Gracias por acompañarme en los momentos difíciles y por celebrar conmigo cada logro.

A mi círculo de amigos, agradezco el tiempo, los momentos, las desveladas, los problemas y tantos días de alegría que tuvimos, tenemos y tendremos. Gracias por compartir su valiosa amistad a lo largo de este camino.

Bryan Ortiz

Índice de contenidos

Resumen	18
Abstract	19
Antecedentes	20
Planteamiento del problema	21
Justificación	22
Objetivos	23
Objetivo general	23
Objetivos específicos	23
Alcance	24
Capítulo II	26
Marco Teórico y Estado del Arte	26
Definiciones Generales	26
Concepto de Sistemas Web y Móviles	27
Arquitectura de Microservicios	31
Modelo de Arquitectura C4	31
Tecnologías de Desarrollo	32
Usabilidad y Experiencia del Usuario	37
Concepto de usabilidad	37
Evaluación mediante el modelo SUS	37
Capítulo III	39

	11
Metodología	39
Tipo y estructura de la investigación	41
Alcance del estudio	42
Fases del desarrollo del sistema	43
Fuentes de información	44
Recopilación de información	45
Procedimiento para la recolección de datos	48
Procesamiento y análisis de los datos	49
Técnicas estadísticas y métricas de evaluación	50
Calibración de los parámetros heurísticos	53
Capítulo IV	58
Análisis del método actual de manejo de residuos domiciliarios	58
Evaluación de los requerimientos del sistema	59
Requerimientos funcionales	59
Requerimientos no funcionales	60
Arquitectura general del sistema	61
Diagrama de Contexto (C1)	61
Diseño de la base de datos geoespacial	63
Diseño del flujo de procesamiento de incidencias	64
Diseño del módulo de Machine Learning	64

	12
Diseño de la aplicación móvil	71
Diseño del panel web de supervisión y administración	74
Capítulo V	80
Pruebas del sistema	80
Pruebas funcionales	80
Pruebas de rendimiento y concurrencia	96
Evaluación de usabilidad	100
Definición de la muestra	100
Protocolo de consentimiento informado	101
Procedimiento de evaluación	101
Resultados de la evaluación SUS	102
Evaluación de la experiencia de usuario	103
Criterios y resultados de la evaluación	104
Validación del modelo de Machine Learning	105
Métricas de rendimiento obtenidas	105
Análisis de desempeño del modelo	107
Conclusiones y Recomendaciones	108
Conclusiones	108
Recomendaciones	110
Líneas de trabajo futuro	110

Referencias

111

Índice de tablas

Tabla 1	Análisis de sistemas y aplicaciones para la gestión de residuos sólidos urbanos
Tabla 2	Comparativa de metodologías para el desarrollo del prototipo
Tabla 3	Resultados iniciales por base de datos científica
Tabla 4	Criterios de exclusión aplicados por base de datos
Tabla 5	Parámetros calibrados del effective_ratio y su justificación
Tabla 6	Características del método actual de gestión de residuos en EMASEO EP
Tabla 7	Requerimientos funcionales del sistema
Tabla 8	Requerimientos no funcionales del sistema
Tabla 9	Composición del dataset de entrenamiento
Tabla 10	Comparativa de arquitecturas de detección evaluadas
Tabla 11	Configuración de entrenamiento del modelo RT-DETR-L v2
Tabla 12	Métricas del modelo RT-DETR-L v2 mejor época (64/100)
Tabla 13	Parámetros de inferencia verificados del pipeline ML
Tabla 14	Bandas de clasificación de severidad y volumen estimado
Tabla 15	Resumen de pruebas funcionales ejecutadas
Tabla 16	Resultados de pruebas de rendimiento con 500 usuarios concurrentes
Tabla 17	Resultados de pruebas de rendimiento con 150 usuarios concurrentes
Tabla 18	Distribución de participantes en la evaluación de usabilidad
Tabla 19	Elementos del consentimiento informado
Tabla 20	Tareas asignadas por perfil de usuario durante la evaluación
Tabla 21	Resultados de la evaluación SUS por perfil de usuario
Tabla 22	Criterios y resultados de la evaluación de experiencia de usuario
Tabla 23	Evaluación final del modelo RT-DETR-L v2

Tabla 24 Matriz de confusión numérica — RT-DETR-L v2 sobre test (706 imágenes, 1.458 instancias)

Índice de figuras

Figura 1	Ciclo iterativo RAD para el desarrollo del prototipo
Figura 2	Diagrama de contexto del sistema MIC-EMASEO (Nivel C1)
Figura 3	Modelo entidad-relación de la base de datos del sistema MIC-EMASEO
Figura 4	Flujo de reporte de incidencia
Figura 5	Pipeline de inferencia del modelo ML
Figura 6	Pantalla de inicio de sesión
Figura 7	Aviso de privacidad y consentimiento informado
Figura 8	Pantalla del menú principal
Figura 9	Pantallas de captura y confirmación de reporte
Figura 10	Pantalla de inicio de sesión del panel de supervisión
Figura 11	Pantalla principal del panel de supervisión con mapa de incidencias
Figura 12	Módulo de gestión de reportes del panel de supervisión
Figura 13	Pantalla de inicio de sesión del panel de administración
Figura 14	Pantalla principal del panel de administración con indicadores y acceso a módulos de gestión
Figura 15	Pantalla de inicio de sesión (RF-01)
Figura 16	Interfaz de encuadre para captura de fotografía (RF-02)
Figura 17	Solicitud y confirmación de permisos de ubicación GPS (RF-03)
Figura 18	Previsualización de foto capturada y confirmación de envío (RF-04)
Figura 19	Historial de reportes y detalle de incidencia (RF-05)
Figura 20	Mensaje de almacenamiento offline (RF-06)
Figura 21	Lista de incidentes del ciudadano autenticado (RF-07)
Figura 22	Login exitoso con credenciales válidas (RF-08)

Figura 23	Login fallido con credenciales inválidas (RF-08)
Figura 24	Control de acceso RBAC: denegación a usuario no autorizado (RF-09)
Figura 25	Detalles de un objeto en Cloudflare R2: URL pública y metadatos (RF-10)
Figura 26	Detección de residuos con bounding boxes y niveles de confianza (RF-11)
Figura 27	Clasificación por nivel de severidad y detalle de incidencia (RF-12)
Figura 28	Diagnóstico completo de IA con estimación de volumen en m ³ (RF-13)
Figura 29	Filtrado de reportes por estado y nivel de severidad (RF-14)
Figura 30	Visualización de estadísticas por sector (RF-15)
Figura 31	Módulo de gestión de cuentas: supervisores y ciudadanía (RF-16)
Figura 32	Configuración de Locust para el escenario de 500 usuarios concurrentes
Figura 33	Panel de estadísticas de Locust para el escenario de 500 usuarios concurrentes
Figura 34	Configuración de Locust para el escenario de 150 usuarios concurrentes
Figura 35	Panel de estadísticas de Locust para el escenario de 150 usuarios concurrentes
Figura 36	Entrevista con el encargado operativo de EMASEO EP
Figura 37	Matriz de confusión numérica del modelo RT-DETR-L v2
Figura 38	Matriz de confusión normalizada por clase real del modelo RT-DETR-L v2

Resumen

El presente trabajo de integración curricular aborda la ausencia de herramientas tecnológicas en la Empresa Pública Metropolitana de Aseo (EMASEO EP) para estimar el nivel de severidad de los residuos sólidos urbanos acumulados en puntos estratégicos del Distrito Metropolitano de Quito. Dada la necesidad de hacer frente a dicha problemática, se elaboró MIC-EMASEO, el cual es un prototipo de sistema web y aplicación móvil que une las funcionalidades de los reportes georreferenciados de los ciudadanos con un pipeline de Machine Learning capaz de analizar las imágenes automáticamente. La forma de solucionarlo se fundamenta en una arquitectura de microservicios desarrollada con Node.js y Express, la cual está constituido por tres servicios que son independientes entre sí (autenticación, usuarios e imágenes/incidencias), en el caso de los ciudadanos, se ha implementado una aplicación móvil en React Native que permite realizar la suma de las acumulaciones de residuos basándose en la geolocalización automática de las fotografías. En relación con la parte operativa, el panel de gestión web en React hace visibles para los supervisores las incidencias e indicaciones de priorización. El componente de inteligencia artificial (RT-DETR-L v2, CLIP con backbone RN50 y MiDaS_small) procesa las imágenes usando Redis y Celery y clasifica la acumulación de residuos en uno de los cuatro niveles de severidad. Durante las pruebas de aceptación, el sistema cumplió satisfactoriamente los 16 requerimientos establecidos previamente y se mantuvo estable bajo una carga de 150 usuarios concurrentes y no presentó errores. El percentil 95 de latencia (5200 ms) es una señal de que esta métrica puede ser mejorada en futuras iteraciones. El estudio de usabilidad con 50 participantes dio como resultado una puntuación SUS de 80.4 (buena) y el personal de EMASEO EP consultado destacó de modo particular la utilidad de los mapas de calor y las estadísticas por sector como soporte operativo. El modelo de detección alcanzó en mAP@50 un valor de 0.8931. Estos resultados son coherentes con MIC-EMASEO como herramienta que entrega a los supervisores de EMASEO EP una predicción temprana y georreferenciada de la severidad de residuos, siendo un elemento que será beneficioso para su futura planificación operativa.

Palabras clave: Gestión de residuos sólidos urbanos, Machine Learning, RT-DETR, aplicación móvil, Participación Ciudadana, Microservicios, System Usability Scale.

Abstract

This curricular integration project addresses the lack of technological tools at the Metropolitan Public Waste Management Company (EMASEO EP) for estimating the severity level of urban solid waste accumulated at strategic points within the Metropolitan District of Quito. To tackle this problem, MIC-EMASEO was developed a prototype comprising a web system and a mobile application that integrates georeferenced citizen reports with a Machine Learning pipeline capable of automatically analyzing images. The solution is built upon a microservices architecture developed with Node.js and Express, consisting of three independent services (authentication, users, and images/incidents). For citizens, a React Native mobile application was implemented, allowing waste accumulation reports to be submitted based on automatic geolocation of photographs. On the operational side, a React-based web management panel makes incidents and prioritization instructions visible to supervisors. The artificial intelligence component (RT-DETR-L v2, CLIP with RN50 backbone, and MiDaS_small) processes images using Redis and Celery, classifying waste accumulation into one of four severity levels. During acceptance testing, the system successfully met all 16 pre-established requirements and remained stable under a load of 150 concurrent users with no errors. The 95th percentile latency (5200 ms) indicates that this metric could be improved in future iterations. A usability study with 50 participants yielded a SUS score of 80.4 (good), and EMASEO EP staff particularly highlighted the usefulness of heat maps and sector-based statistics as operational support tools. The detection model achieved an mAP@50 value of 0.8931. These results confirm MIC-EMASEO as a tool that provides EMASEO EP supervisors with early, georeferenced predictions of waste severity, serving as a valuable asset for future operational planning.

Keywords: municipal solid waste management, Machine Learning, RT-DETR, mobile application, citizen participation, microservices, user experience, System Usability Scale.

Capítulo I

Antecedentes

A través de los años se ha venido dando un incremento en la población como en la generación de residuos sólidos urbanos. Según el Banco Mundial (Kaza et al., 2018) “la generación mundial de desechos aumentará desde 2.010 millones de toneladas en 2016 a 3.400 millones de toneladas para el año 2050”, lo cual es un reto impactante para las administraciones locales, debido el rápido crecimiento demográfico y la expansión urbana. Según los datos expuestos por (Sall et al., 2025), no se gestiona de forma adecuada una gran parte de estos residuos, que llega a convertirse en el detonante de problemas ambientales y de salud; situándose en una posición especialmente compleja en aquellas regiones en las que los recursos operativos son limitados o bien, la demanda de los servicios está en aumento. A su vez, Lakhout (2025) amplía y complementa esta visión al sumar el incremento de población que todavía se mantiene y que anticipa una producción de residuos que podría llegar a alcanzar hasta 3400 millones de toneladas cada año para el año 2050. En este escenario, los modelos de gestión de los residuos sólidos urbanos deben avanzar hacia modelos más dinámicos, flexibles y sustentados en nuevas tecnologías. Varios autores han caracterizado los sistemas tradicionales de recolección como sistemas muy rígidos, a partir de rutas y horarios fijos que no obligatoriamente terminan de ajustarse a las directrices reales de generación de residuos en los diferentes sectores de una ciudad (Pardini et al., 2020). Este desequilibrio entre la planificación operativa y el comportamiento territorial ha generado, como respuesta natural, un creciente interés por explorar alternativas tecnológicas que permitan una gestión más ágil, flexible y ajustada a las necesidades reales de cada zona. Por lo tanto, basado en el uso de tecnologías de la información enmarcadas en el modelo de Ciudades Inteligentes (Smart Cities), puede potenciar realmente la participación ciudadana y con ello mejorar la gestión pública de una ciudad. Diferentes investigaciones coinciden en que la implementación de sistemas web y aplicaciones móviles georreferenciadas facilita el reporte oportuno de incidencias urbanas, permitiendo a las entidades

encargadas del aseo contar con información más precisa y actualizada sobre puntos críticos de acumulación de residuos (Sadeghi-Niaraki et al., 2020).

El uso de herramientas de Machine Learning para el análisis automático de imágenes y la estimación de volúmenes de residuos permite que las imágenes capturadas por ciudadanos desde sus dispositivos móviles sean procesadas mediante el pipeline de inferencia para generar notificaciones priorizadas dirigidas a los supervisores operativos (Majchrowska et al., 2021). Al igual que en el caso del Distrito Metropolitano de Quito, este enfoque plantea la posibilidad de orientar los recursos operativos hacia los puntos de acumulación de residuos, en lugar de limitarse a considerar recorridos y horarios fijos.

Planteamiento del problema

En el Distrito Metropolitano de Quito, la gestión de residuos y de la limpieza, iniciativas de la Empresa Pública Metropolitana de Aseo (EMASEO EP) se convierte en un desafío para la evolución continua en el contexto de la conciencia de los ciudadanos que han ido demandando cada vez mayor atención. Entre los servicios de más interés brindados por la entidad, el barrido de espacios públicos (ampliamente conocido en su modalidad manual y mecánica) constituye un aspecto que se vincula a la limpieza en calles, plazas y aceras (EMASEO EP, 2022). Sin embargo, la operatividad vigente de este servicio tiene una limitación clave ya que se ejecuta en función de recorridos y horarios anticipados que no corresponden a posibles espacios donde podría mayor cantidad de basura.

La problemática principal se debe a la inflexibilidad del modo de trabajo en base a la recolección de basura y al escaso acceso a la información en tiempo real sobre la cantidad de residuos sólidos que se concentran en lugares críticos (Pardini et al., 2020). Dado que la institución ofrece un servicio guiado por cronogramas, le resulta difícil detectar con anticipación que un determinado sector generó residuos a un ritmo inusual por actividades comerciales, eventos dinámicos de la población o factores ciudadanos.

Esta situación genera que varias zonas queden desatendidas mientras se dirigen recursos a sectores que

no lo requieren, provocando que los residuos sólidos se acumulen, afectando la imagen urbana y generando focos de contaminación ambiental que impactan la salud de los habitantes (Abahussain et al., 2025). No se trata de una falta de voluntad operativa, sino de una limitación estructural que podría resolverse con mejores herramientas de información y canales de comunicación más ágiles entre la ciudadanía y la entidad encargada del aseo. A la difícil operatividad se agrega un hecho evidente: el crecimiento poblacional y la expansión geográfica no planificada han aumentado significativamente la producción de residuos sólidos en la ciudad de Quito (Lakhout, 2025; Pardini et al., 2020). Nuevos sectores, residenciales y comerciales han surgido en zonas donde previamente no existía demanda de limpieza, ampliando las áreas que deben ser cubiertas sin que los mecanismos tradicionales de programación hayan cambiado al mismo ritmo. Esta situación pone de manifiesto que se necesitan incorporar sistemas que permitan la toma de información actualizada de las zonas con alta acumulación de residuos para facilitar el fortalecimiento de procesos existentes mediante herramientas tecnológicas ajustadas a la realidad dinámica del territorio (Sharma & Kumar, 2024).

Justificación

Por las razones argumentadas, es necesario plantear alternativas que actualicen la respuesta institucional. En este contexto, se justifica, el desarrollo de una solución tecnológica que convierta el informe ciudadano en un elemento activo del servicio de barrido. La presencia de un mecanismo de análisis de las imágenes capturadas por dispositivos móviles y reportadas por la ciudadanía permitiría a EMASEO EP conocer el nivel de severidad de acumulación de los residuos sólidos en los distintos barrios y localidades de la ciudad. Tal proyección estructurada y georreferenciada facilitaría a EMASEO EP realizar la correcta asignación de los recursos logísticos y humanos a los sectores con mayor necesidad lo que permitiría anticiparse a la reducción de incidencias.

El presente prototipo de plataforma web y móvil tiene como finalidades la optimización de los recursos públicos y la transición del modelo de gestión, desde una planificación operativa rígida hacia un

modelo dinámico adaptado a la realidad urbana. De esta manera, mediante la reducción de los tiempos de respuesta y un canal de comunicación moderno, se contribuiría a acelerar los procesos operativos del servicio de limpieza pública y a fortalecer la confianza ciudadana en la administración pública.

Objetivos

Objetivo general

Desarrollar un prototipo de sistema web como herramienta de apoyo a la gestión operativa de la recolección de residuos sólidos urbanos en la EMASEO EP, integrando un módulo de aplicación móvil para el registro de incidencias ciudadanas y técnicas de Machine Learning aplicadas al análisis de imágenes, con el fin de generar escalas de atención que apoyen la toma de decisiones operativas.

Objetivos específicos

- Analizar los fundamentos tecnológicos y conceptuales relacionados a la administración de residuos sólidos urbanos, la participación digital de los ciudadanos y la aplicación de Machine Learning para la evaluación de imágenes.
- Formular la lógica del prototipo, añadiendo servicios para almacenar, recibir y procesar incidencias ciudadanas georreferenciadas, además, poner en marcha un módulo de análisis que utilice aprendizaje automático para clasificar imágenes según su acumulación estimada.
- Prototipar un módulo de aplicación para dispositivos móviles enfocado en la ciudadanía, que junte el uso de servicios e interfaces de usuario, con la finalidad de mantener un registro de los incidentes asociados con los residuos sólidos urbanos, incluyendo datos georreferenciados y pruebas gráficas.

- Diseñar la interfaz web para supervisores operativos, orientada a la visualización operativa y a la administración de usuarios, con el propósito de facilitar el control, seguimiento y análisis de la recolección de residuos sólidos urbanos.
- Examinar la efectividad y desempeño del prototipado elaborado a través de pruebas de las funcionalidades y de la facilidad de uso para comprobar su viabilidad como herramienta de soporte a la gestión operacional.

Alcance

El presente estudio corresponde a un estudio de enfoque explicativo y demostrativo. Se generó un prototipo funcional del sistema MIC-EMASEO, el cual se compone de tres módulos principales: una aplicación móvil desarrollada con React Native para los ciudadanos, un panel web elaborado con React para supervisores y administradores del sistema, y un módulo de Machine Learning destinado al análisis automático de imágenes de acumulaciones de residuos sólidos urbanos en Quito.

El prototipo se basa en una arquitectura de microservicios desarrollada con Node.js y Express, compuesta por tres microservicios independientes (autenticación, usuarios, e imágenes/incidencias) comunicados a través un API Gateway con autenticación JWT y control de acceso RBAC. El almacenamiento relacional y geoespacial se implementa en el PostgreSQL por medio de la extensión PostGIS, mientras que las imágenes son almacenadas en Cloudflare R2 en producción, donde solo se refiera a la relación a los componentes de software.

El módulo de inteligencia artificial sigue un flujo de trabajo relacionado con visión por computadora constituidos por tres componentes en secuencia: RT-DETR-L v2, entrenado para detectar una única clase (*garbage*), con el fin de localizar los residuos a través de bounding boxes; CLIP, con arquitectura configurable mediante la variable CLIP_MODEL_NAME (en el ambiente de producción evaluado se emplea el backbone RN50 con pesos openai por su menor latencia en CPU, quedando ViT-B-32 disponible como alternativa de mayor precisión), para la comprobación semántica de las detecciones;

y MiDaS_small como estimador de profundidad monocular auxiliar, cuya salida alimenta el cálculo geométrico del volumen estimado de la acumulación. La estimación volumétrica resultante es de carácter aproximado y tiene como propósito generar un indicador relativo que apoye la clasificación de la severidad en cuatro niveles (BAJO, MEDIO, ALTO, CRÍTICO). La validación del pipeline de ML se efectúa de forma integral a través de las métricas estándar de detección de objetos (mAP@50, precisión, recall y F1-Score) sobre el conjunto de test, sin evaluar de forma independiente el componente de estimación de profundidad, dado que no se dispone de datos de profundidad métrica de referencia para las imágenes de campo.

De manera que el sistema permita a los ciudadanos registrar incidencias mediante fotografía obtenida en tiempo real con geolocalización automática; y a los supervisores revisar, validar, priorizar y gestionar las incidencias mediante el panel web correspondiente. La validación técnica y funcional del prototipo se desarrollará en un entorno controlado con 50 participantes distribuidos entre ciudadanos y supervisores, que pertenecen a un sector específico del Distrito Metropolitano de Quito, a fin de validar la viabilidad técnica de la integración entre los componentes del sistema que constituyen el prototipo presentado.

No se arriba el uso institucional para el sistema ni la capacidad de escalarlo para ser puesto en funcionamiento en un contexto operativo de EMASEO EP. En particular no se incluyen en este proyecto la optimización de rutas de recolección, la planificación y gestión de flotas de vehículos, el análisis predictivo de la generación de residuos, la integración con sensores IoT y la extensión de municipios y jurisdicciones (acciones que se consideran como línea de trabajo futuro).

Capítulo II

Marco Teórico y Estado del Arte

Definiciones Generales

Gestión de Residuos Sólidos Urbanos (GIRSU).

La Gestión de Residuos Sólidos Urbanos puede considerarse el conjunto de actividades operativas, administrativas y técnicas involucradas en el manejo de los desechos generados en áreas urbanas. En los términos que formula la comisión nacional para el uso eficiente de la energía (CONUEE, 2018): la GIRSU comprende desde la generación hasta su disposición final, pasando por el almacenamiento, la recolección, el transporte, el tratamiento y la valorización. Así, la GIRSU que aplica la investigación se centra especialmente en la recolección y el barrido urbano, para lo cual se propone la inclusión de tecnologías digitales en la solución. La reglamentación existente en el Ecuador determina la manera de realizar la gestión de los residuos sólidos urbanos, consiste en el Código Orgánico del Ambiente (COA, 2017), una norma que establece los principios y las leyes que deben seguirse a efectos de gestionar los desperdicios generados por la actividad humana de una forma integral. En el Distrito Metropolitano de Quito, la Empresa Pública Metropolitana de Aseo (EMASEO EP) ejerce sus funciones bajo el Reglamento Orgánico por Procesos, un texto normativo que determina cuál es su responsabilidad en actividades que le son propias como el barrido, la recolección y el transporte de los residuos sólidos. Esta normativa vigente en la que es factible incorporar las tecnologías de la información como herramientas de apoyo, siempre que no sustituyan los procesos institucionales ya establecidos (EMASEO EP, 2022).

Gestión Participativa Digital

La gestión participativa digital se puede describir como el uso de plataformas tecnológicas, aplicaciones móviles y sistemas web que conectan a los ciudadanos con las entidades públicas y que, a

su vez, permiten que ambas partes colaboren para identificar y dar solución a problemas que surgen en la ciudad. Dicho mecanismo invita a las personas a participar activamente en el cuidado y control de los espacios públicos, pues hay posibilidad de informar, por ejemplo, de sitios donde se acumula la basura, lo que contribuye a que los servicios municipales puedan ser más eficaces y con mayor transparencia (Sadeghi-Niaraki et al., 2020). De esta forma, la tecnología no solo informa, sino que también convierte al ciudadano en un aliado dentro de la gestión urbana. La presente investigación analiza las aplicaciones y sistemas tecnológicos que existen en la actualidad orientados a la gestión de residuos sólidos urbanos con el apoyo de plataformas digitales, aplicaciones móviles y herramientas de análisis de datos. En este proceso se identificaron múltiples soluciones enfocadas al reporte ciudadano de incidencias urbanas, la geolocalización de conflictos vinculados a los residuos y el uso de tecnologías inteligentes para la mejora de la gestión de servicios de limpieza urbana (Sharma & Kumar, 2024). Aunque estas herramientas suponen un gran paso en la automatización de servicios urbanos, las limitaciones relacionadas con la integración del análisis automático de imágenes, del uso de técnicas de aprendizaje automático y la priorización inteligente de incidencias ciudadanas son reducidas.

Concepto de Sistemas Web y Móviles

Un sistema web se define como una aplicación accesible mediante un navegador de internet, alojada en un servidor remoto, con la cual el usuario puede interactuar para ejecutar funcionalidades específicas sin requerir la instalación de software en un dispositivo. En cambio, las aplicaciones móviles son programas elaborados para dispositivos portátiles (smartphones, tablets, etc.), que incorporan interfaces optimizadas para pantallas táctiles y que utilizan sus capacidades nativas hardware (GPS, cámara, sensor de movimiento, etc.). A su vez, junto a estos dos tipos de sistemas se pueden realizar ecosistemas digitales completos: el sistema web servirá para modelar la gestión administrativa y visualización de datos y la aplicación móvil para la captura de información en campo (Sánchez et al., 2024).

En la actualidad, existen diversas aplicaciones y sistemas que ejemplifican el uso de tecnologías web y móviles para la gestión de incidencias urbanas y de residuos sólidos. A continuación, se presenta una tabla comparativa que resume las características principales de estos sistemas.

Tabla 1

Análisis de sistemas y aplicaciones para la gestión de residuos sólidos urbanos

Sistema	Año	Objetivo	Tecnologías	Geo	ML	Tipo	Aportes	Limitaciones
SeeClickFix (Schiff, 2025)	2025	Reporte urbano	Web, GPS, mapas	Sí	No	Web/móvil	Comunicación ciudadano-municipio	Sin análisis de imágenes
FixMyStreet (Maciulienė & Skaraiskienė, 2020)	2020	Reporte urbano	Web, GIS	Sí	No	Web	Visualización georreferenciada	Sin análisis inteligente
Ushahidi	2020	Recopilación geo	Crowdsourcing	Sí	No	Colaborativa	Recolección masiva de datos	No orientado a residuos
TrashOut (Longo et al., 2021)	2021	Reporte vertederos	Móvil, GPS, foto	Sí	No	Móvil	Identificación colaborativa	Sin análisis automático
WasteApp (Radziszewski et al., 2021)	2021	Registro acumulación	Móvil, GPS	Sí	No	Móvil	Comunicación municipal	Sin IA
Clean City (Suryawans)	2022	Gestión limpieza	Móvil, Flutter, Firebase	Sí	No	Móvil	Registro de incidencias	Sin análisis automático

Sistema	Año	Objetivo	Tecnologías	Geo	ML	Tipo	Aportes	Limitaciones
hi et al., (2022)								
GIS Waste Mgmt (Hatamleh et al., 2020)	2020	Análisis espacial y optimizac ión rutas	ArcGIS Network Analyst	Sí	No	GIS	Zonas críticas, reducción de distancia	Sin participación ciudadana
Smart Waste Monitor (Pardini et al., 2020)	2020	Monitore os contened ores	IoT, sensores, web	Sí	Parc ial	IoT	Monitoreo en tiempo real	Requiere sensores
Waste Image Classif (Shen et al., 2024)	2024	Clasificaci ón residuos	Deep Learning	No	Sí	Análisis	Clasificación automática	Sin integración urbana
Deep Learning Detect (Shen et al., 2024)	2024	Detecció n residuos	CNN	No	Sí	Visión	Identificació n automática	Sin geolocalizaci ón
Smart Waste Recogn (Addas et al., 2024)	2024	Reconoci miento residuos	ML, IoT, imágenes	No	Sí	Inteligente	Automatizaci ón y eficiencia	Sin integración ciudadana

Sistema	Año	Objetivo	Tecnologías	Geo	ML	Tipo	Aportes	Limitaciones
Smart Waste Bin (Zoumpoulis et al., 2024)	2024	Monitoreo llenado	IoT, sensores, GSM	Sí	No	Contenedores	Mejora rutas y costos	Alto costo de implementación
Route Optimization (Hussain, 2025)	2025	Optimización rutas logísticas	IA, RL, IoT, Cloud	Sí	Sí	Logístico	Reducción costos, 98% entregas a tiempo	Dependencia de datos en tiempo real
Intelligent Collection (Pardini et al., 2020)	2020	Recolección inteligente	IoT, análisis	Sí	Parcial	Inteligente	Planificación dinámica	Compleja implementación

Nota. En la siguiente tabla, se encuentran 15 sistemas agrupados en cinco categorías de función: reporte ciudadano, análisis espacial, monitoreo IoT, clasificación visual y optimización logística. Cabe destacar que el 60% de los sistemas incorpora georreferenciación, si bien sólo el 27% utiliza técnicas machine learning.

La revisión de estos 15 sistemas que recoge la Tabla 1 demuestra que todos emplean las cuatro características fundamentales: visión por computador, georreferenciación, priorización semántica y microservicios. Si bien hay sistemas que no utilizan análisis de imágenes, los hay que necesitan sensores costosos o que no incorporan geolocalización (los sistemas de 4 padres). En alguna medida, MIC-EMASEO cubre esta necesidad, en la medida que los combina, permitiendo así el procesamiento de reportes, la validación y priorización de las intervenciones de la estimación en función de la severidad independiente generada a partir del modelo de Machine Learning.

Arquitectura de Microservicios

La arquitectura de microservicios es un estilo de desarrollo de software en el cual una aplicación se estructura como un conjunto de servicios pequeños, independientes y desplegables por separado. Cada microservicio se centra en una funcionalidad concreta del negocio (en el caso de un sistema de gestión de incidencias: la gestión de usuarios, el manejo de imágenes, el almacenaje de incidencias, etc.), que se puede comunicar con otros microservicios mediante protocolos ligeros como HTTP o la mensajería asíncrona; esta arquitectura permite obtener ventajas notables para sistemas como el que se propone en esta investigación: escalabilidad independiente de cada módulo, tolerancia a fallos, integración fácil de tecnologías heterogéneas, etc. (Newman, 2015).

Modelo de Arquitectura C4

Brown (2018) propuso el modelo C4, una técnica de notación ligera que pretende servir para documentar la arquitectura del software. Consiste en un conjunto de diagramas jerárquicos que representan el sistema en distintos niveles de separación. A diferencia de las notaciones formales como UML que requieren un amplio conjunto de símbolos y reglas estrictas, C4 emplea un vocabulario visual muy reducido (cajas, personas y relaciones etiquetadas), por lo que facilita su claridad para públicos técnicos y no técnicos.

El modelo especifica cuatro niveles de abstracción, donde cada nivel representa un mayor grado de detalle respecto del anterior:

- **Nivel 1 – Contexto (C1):** El sistema se considera una caja única, indicando las relaciones con los usuarios y sistemas externos de los que se integra, sin detallar su estructura interna.
- **Nivel 2 – Contenedores (C2):** El sistema se descompone en unidades seccionales de forma independiente (aplicaciones, servicios, bases de datos), especificando, por un

lado, la tecnología de cada una de ellas y, por otro, los protocolos de la comunicación que se establecen entre los distintos componentes.

- **Nivel 3 – Componentes (C3):** Se desarrolla la estructura interna de un contenedor concreto, donde se muestran los módulos lógicos que lo forman y las responsabilidades de cada uno de ellos.
- **Nivel 4 – Código: (C4)** Corresponde al de módulos del código fuente. Brown (2018) sostiene que este nivel suele ser omitido en la documentación de la arquitectura cediéndola a herramientas de generación automática.

Tecnologías de Desarrollo

React para el desarrollo Frontend Web y Móvil. Meta ha desarrollado React como una biblioteca de JavaScript para la creación de interfaces de usuario a partir de componentes reutilizables. Su enfoque declarativo, junto con el uso de un DOM virtual, permite construir aplicaciones web interactivas y eficientes. React Native lleva estos mismos principios al desarrollo móvil nativo (iOS y Android) y, al compartir buena parte de la base de código entre la aplicación web y la móvil, acelera el desarrollo y simplifica el mantenimiento del sistema (Mankotia et al., 2026).

Node.js y Express para el desarrollo Backend. En cuanto a la tecnología utilizada, Node.js es un entorno de ejecución de código JavaScript del lado del servidor que hace uso del modelo asíncrono y orientado a eventos; es un entorno adecuado para aplicaciones con alta concurrencia. Express es un framework minimalista y flexible que se ejecuta sobre Node.js, por tanto, proporciona herramientas sólidas para el manejo de las rutas, solicitudes HTTP y el middleware. Esta combinación de Node.js y Express es utilizada ampliamente para crear APIs RESTful y aplicaciones de servicios backend escalables, como es la exigencia del sistema que se necesita desarrollar para la gestión de incidentes y la comunicación con la aplicación móvil (Zhou & Gao, 2024).

PostgreSQL como sistema para la Persistencia de Datos. PostgreSQL, conocido por su potencia y cumplimiento con los estándares SQL, además destaca por permitir el almacenamiento y la consulta de datos espaciales a través de la extensión PostGIS. Esta funcionalidad permite la consulta eficiente de las coordenadas geográficas de las incidencias reportadas, así como la realización de análisis de proximidad y distribución espacial de los residuos sólidos urbanos (Hatamleh et al., 2020; Sadeghi-Niaraki et al., 2020).

Tecnologías de Machine Learning. El Machine Learning (o aprendizaje automático), es un subcampo de la inteligencia artificial que otorga a los sistemas informáticos la capacidad de aprender patrones de los datos, sin la necesidad de estar programados explícitamente con un listado de reglas paso a paso (Sharma & Kumar, 2024). En vez de trabajar siguiendo órdenes al pie de la letra, los modelos analizan ejemplos, identifican regularidades y construyen su propia lógica para tomar decisiones. En el ámbito de la gestión de los residuos sólidos urbanos, esta habilidad es especialmente útil, pues permite que los modelos sean capaces de analizar miles de imágenes de fotografías realizadas por las personas que se las muestran, aprender a reconocer aquellas características visuales que identifican que hay una acumulación de residuos y una vez entrenados sean capaces de procesar nuevas imágenes en fracciones de segundo y decidir si hay un problema que debe ser atendido de manera prioritaria (Majchrowska et al., 2021).

YOLO (You Only Look Once). Esta familia de arquitecturas de redes neuronales para detección de objetos se caracteriza por procesar la imagen completa en una única pasada por la red, prediciendo de forma simultánea la ubicación y la categoría de cada objeto (Li et al., 2023). A diferencia de los detectores basados en regiones (region-based detectors), que analizan la imagen mediante propuestas de regiones y por ello incrementan el tiempo de procesamiento, YOLO aplica un enfoque de una sola etapa (single-stage detection). Esa arquitectura le da una velocidad de inferencia alta, apropiada para aplicaciones en tiempo real (procesamiento de video, despliegue en dispositivos con recursos limitados)

aunque a costa de una precisión algo menor frente a arquitecturas más complejas basadas en transformadores (Shen et al., 2024).

RT-DETR (Real-Time Detection Transformer). Este modelo, cuyas siglas corresponden a Real-Time Detection Transformer, plantea un enfoque más reciente: combina las redes convolucionales tradicionales con los mecanismos de atención propios de los transformadores. La arquitectura de MIC-EMASEO consta de dos fases. La primera fase usa una red convolucional para la extracción de características visuales simples de la imagen, mientras que en la segunda emplea un mecanismo de atención multi-cabeza para estudiar las relaciones entre las distintas partes que componen la escena, lo que permite centrarse en regiones distantes. En el caso de MIC-EMASEO, este mecanismo permite detectar residuos altamente dispersos o parcialmente ocultos, mientras que la red convolucional se enfoca en otras regiones. Por ejemplo, permite la asociación visual de una bolsa de residuos con un contenedor cercano, así como la deducción de que una serie de residuos pequeños está asociada a una misma acumulación. A cambio de esta precisión, RT-DETR exige un tiempo de procesamiento algo mayor que YOLO, por lo que resulta más adecuado para servidores backend que para dispositivos móviles (Zhao et al., 2024).

EfficientDet. Desarrollada por el equipo de investigación de Google, esta familia de modelos permite alcanzar equilibrio entre precisión y eficiencia computacional mediante BiFPN (Bidirectional Feature Pyramid Network). Esta arquitectura, combina características de distintas escalas de forma más inteligente que el resto de los métodos, logrando detectar residuos de diferentes tamaños y distancias dentro de la misma imagen, sin pérdida de detalle (Tan & Le, 2020). Sus distintas versiones (D0, D1, D2, entre otras) ofrecen compromisos diferentes entre tamaño, velocidad y precisión, lo que permite elegir la variante más adecuada según los recursos disponibles; no obstante, algunos estudios reportan dificultades de convergencia cuando el entrenamiento usa ciertos formatos de anotación (Majchrowska et al., 2021).

Precisión (Precision). La precisión es una métrica que indica cuántas de las detecciones realizadas por el modelo indican la presencia de residuos son realmente correctas. La precisión se obtiene mediante la relación entre el número de detecciones correctas (verdaderos positivos) y la suma del total de detecciones llevadas a cabo por el modelo (verdaderos positivos y falsos positivos). La precisión alta significa que el modelo ha tenido una baja tasa en la cantidad de falsas alarmas, lo que implica una alta capacidad del modelo para predecir correctamente los residuos sólidos (Sharma & Kumar, 2024). Un modelo de detección de residuos sólidos con una precisión alta evita que los supervisores reciban alertas innecesarias de objetos que no son residuos: sombras, manchas en el suelo, bolsas vacías. Sin embargo, un alto valor en la precisión puede conducir a no detectar las acumulaciones verdaderas; por lo que esta métrica no debe analizarse de forma autónoma.

Recall (Sensibilidad). Es aquel que responde a la pregunta de cuántas de las acumulaciones de residuos realmente existentes en las imágenes logró detectar el modelo. Se calcula como la división entre el número de detecciones correctas y el total de acumulaciones reales (verdaderos positivos más falsos negativos). Un recall alto indica que el modelo no excluye una cantidad significativa de acumulaciones, aspecto fundamental en el contexto de la limpieza urbana, ya que el incumplimiento de un punto crítico de desechos podría tener consecuencias medioambientales (Majchrowska et al., 2021). No obstante, un modelo podría llegar a un recall alto simplemente clasificando todas las detecciones como residuos, lo que afectaría negativamente con la precisión. Por eso es necesario buscar un equilibrio entre ambas métricas.

F1-Score. Es una métrica que integra la precisión y la sensibilidad en un único valor mediante su media armónica. A diferencia de un promedio simple, la media armónica penaliza los desequilibrios extremos: si uno de los dos valores es muy bajo, el F1-Score también será bajo, reflejando que el modelo no es equilibrado (Majchrowska et al., 2021). Esta métrica resulta especialmente útil cuando se necesita comparar modelos que tienen comportamientos muy diferentes. Esta métrica indica el equilibrio entre

ambas medidas, lo cual resulta relevante en aplicaciones donde los falsos positivos conllevan efectos negativos, y su ausencia también presenta consecuencias adversas.

mAP (Precisión Media Promedio). Es la métrica total y la más utilizada en la literatura de detección de objetos. Su cálculo es más complejo que el de las métricas anteriores: básicamente, mide el área bajo la curva de precisión-recall, lo que proporciona una visión del rendimiento general del modelo en todos los posibles umbrales de confianza (Zhao et al., 2024). En cuanto a las variantes más comunes, el mAP@50 considera que la detección es correcta al menos cuando la caja delimitadora tiene al menos un 50% de coincidencia con la caja real del residuo. El mAP@50:95 es una variante más estricta que promedia el rendimiento desde el 50% hasta el 95% de coincidencia, penalizando a aquellos modelos que logran localizar los residuos, pero con poca precisión en términos de posición y tamaño (Sharma & Kumar, 2024). En el ámbito práctico, el mAP es la métrica que mejor correlaciona con la calidad percibida de un detector de objetos.

Sobreajuste (Overfitting). El sobreajuste es un fenómeno que ocurre cuando un modelo aprende demasiado bien los ejemplos del conjunto de entrenamiento, hasta el punto de memorizar ruido o particularidades irrelevantes, pero luego falla al enfrentarse a imágenes nuevas que no ha visto antes (Chollet Ramampandra et al., 2023). Derivado al contexto de la acumulación de los residuos sólidos, al tratar de entrenarlo solo con imágenes en óptimas condiciones tanto su iluminación y entorno, se puede presentar conflictos de sobreajuste para el modelo, sin embargo, como consecuencia se genera una mala interpretación de las fotografías capturadas en su contexto real. Para reducir el sobreajuste se utilizan técnicas como el data augmentation y regularización.

Generalización (Generalization). La generalización es la capacidad de un modelo entrenado para desempeñarse correctamente sobre datos nuevos que no formaron parte del entrenamiento. Es, en esencia, el objetivo último de cualquier sistema de aprendizaje automático. En el contexto del reconocimiento de desechos en espacios públicos, un modelo con buena generalización debe ser capaz

de identificar acumulaciones de residuos sin importar la hora del día, las condiciones climáticas, el tipo de cámara utilizada o el barrio de la ciudad donde se tomó la fotografía (Chollet Ramampandra et al., 2023).

Usabilidad y Experiencia del Usuario

Concepto de usabilidad

La usabilidad es considerada como la facilidad de uso por parte del sistema para medir la eficacia, eficiencia y la satisfacción, basado en el marco estructural (ISO 9241-11:2018), se emplea para un grupo determinado de usuarios con el propósito de evaluar si existe interacción persona-sistema.

Evaluación mediante el modelo SUS

La Escala de Usabilidad del Sistema (SUS) es un cuestionario estándar muy validado, el cual ayuda al evaluador para poder conocer rápida y confiablemente la usabilidad del sistema.

Nielsen (2012) hace énfasis en este concepto, debido al contexto del desarrollo de software y diseño de páginas web haciendo que el enfoque normativo sume propuestas pragmáticas para ser capaces de evaluar la usabilidad de un modo exhaustivo.

El cuestionario SUS consta de 10 preguntas con respuestas en escala Likert de cinco puntos (desde “Totalmente en desacuerdo” hasta “Totalmente de acuerdo”) y evalúa aspectos como la facilidad de uso, la necesidad de ayuda técnica, la funcionalidad y la confianza del usuario (Sudrajat et al., 2025).

El cálculo de la puntuación final se obtiene siguiendo el procedimiento que se detalla a continuación (Brooke, 1996):

1. Las preguntas impares (1, 3, 5, 7, 9), la puntuación real se calcula como el puntaje del usuario menos 1, de modo que la colaboración se sitúa en un rango de 0 a 4.
2. Las preguntas pares (2, 4, 6, 8, 10), la puntuación efectiva es igual a 5 menos el puntaje del usuario. La colaboración resultante también se sitúa en un rango de 0 a 4.

3. La suma de las colaboraciones de las 10 preguntas proporciona el puntaje total, cuyo rango es de 0 a 40.
4. Finalmente, el puntaje total se multiplica por 2.5 para obtener el resultado real del SUS, el cual se expresa en un rango de 0 a 100).

Capítulo III

Metodología

La investigación en cuestión tiene un enfoque mixto, el cual considera las perspectivas cuantitativas y cualitativas en el análisis del problema de gestión de residuos sólidos urbanos (Pardini et al., 2020). El componente cuantitativo se fundamenta en la medición y análisis de variables numéricas objetivas, tales como: coordenadas geográficas exactas de las incidencias reportadas, frecuencias de reportes por sector, y métricas de rendimiento del modelo de Machine Learning (precisión, recall, mAP@50, F1-Score) (Sharma & Kumar, 2024; Majchrowska et al., 2021). El componente cualitativo se incorpora para evaluar la experiencia de usuario (UX) a través de la aplicación del cuestionario System Usability Scale (SUS), permitiendo comprender la percepción de los usuarios finales respecto a la facilidad de uso y utilidad del sistema (Sudrajat et al., 2025). Adicionalmente, el tipo de investigación se clasifica como aplicada y tecnológica, dado que su propósito principal es la construcción y validación de un prototipo de software funcional para resolver un problema práctico específico (Pardini et al., 2020).

Por la parte que se ajusta a la construcción del prototipo, establecemos aplicar una metodología ágil, por la razón de que ésta permite adaptarse a los diferentes cambios que se van diseñando a lo largo de la vida del proyecto, marcando un claro objetivo en rapidez de entregas, flexibilidad para las diferentes exigencias a que puede verse retroalimentadas por parte de los usuarios finales. Las metodologías ágiles favorecen a los desarrolladores con el propósito de introducir buenas prácticas por ejemplo, la integración continua, la entrega constante de valor (Wynnedeta et al., 2024). Este método establece el trabajo en iteraciones cortas, las cuales, celebran el descubrimiento previo a la desviación entre lo que se previene y lo que se construye, permitiendo llevar a cabo algún que otro ajuste sin poder distorsionar los plazos. Esta metodología ágil se convierte entonces en un pilar central cuando los requisitos de un proyecto pueden ir cambiando en el tiempo, asegurando que el desarrollo del propio producto tenga una excelente calidad a la hora de ser entregado (Wynnedeta et al., 2024).

Análisis comparativo de metodologías. Previo a la selección de la metodología de trabajo, se realizó un análisis comparativo entre los enfoques más relevantes. Los puntajes asignados en la Tabla 2 fueron determinados por los autores en función de los criterios específicos del proyecto: equipo de desarrollo de dos personas, plazo de 14 semanas, necesidad de integración con un módulo de Machine Learning, y requerimiento de retroalimentación temprana por parte de los stakeholders de EMASEO EP. Cada metodología fue evaluada en una escala de 1 a 5 (donde 1 es la calificación más baja y 5 la más alta) según su idoneidad para cada criterio.

Tabla 2

Comparativa de metodologías para el desarrollo del prototipo

Metodología	Flexibilidad	Velocidad	Integración ML	Equipo pequeño	Puntaje
RAD	5	5	4	5	19/20
Scrum	4	4	3	4	15/20
Kanban	4	3	3	4	14/20
Cascada	1	2	1	3	7/20
Espiral	3	2	4	2	11/20

Nota. Como se visualiza en la Tabla 2, RAD obtuvo el mejor puntaje (19/20) al superar a Scrum (15/20), Kanban (14/20), Espiral (11/20) y Cascada (7/20) en los criterios evaluados.

Con base en el análisis realizado, se selecciona la metodología RAD (Rapid Application Development) por obtener la puntuación alta (19/20). Aspectos como la flexibilidad, la rapidez y la adaptabilidad a equipos pequeños resultan relevantes para poder concretar el prototipo dentro del tiempo y espacio.

Justificación de la selección: Metodología RAD. El análisis comparativo llevó a seleccionar el Desarrollo Rápido de Aplicaciones (RAD) como metodología principal, por tres razones:

1. **Prototipado rápido e iterativo:** RAD permite producir, en un corto período de tiempo, versiones funcionales que la organización validará en la integración de los diferentes componentes del sistema.
2. **Flexibilidad ante cambios continuos:** Las exigencias de este tipo de intervención tienden a modificarse de forma rápida, lo que permite adaptarse a los requisitos sin necesidad de reconstruir las fases anteriores.
3. **Retroalimentación temprana de usuarios finales:** Al permitir la intervención continua de las partes interesadas en la producción del software, es posible adaptar la interfaz y los mecanismos funcionales a sus necesidades reales y no únicamente a las necesidades documentadas.

Tipo y estructura de la investigación

El tipo de investigación adoptado fue **aplicado y tecnológico**, es decir, su objetivo último era la construcción y validación de un prototipo de software que resolviera un problema práctico y concreto. La estructura de la investigación se organiza en torno al desarrollo de un **prototipo de sistema distribuido basado en una arquitectura de microservicios**. Esta arquitectura se compone de los siguientes elementos estructurales:

- **Módulo de interfaz con el usuario:** Agrupa las pantallas y elementos visuales con los que interactúan tanto los ciudadanos como los supervisores.
- **Módulo de servicios o backend:** Está formado por un conjunto de microservicios independientes, cada uno encargado de una función específica de la lógica de negocio (Wynnedeta et al., 2024).
- **Módulo de datos:** Se encarga de la persistencia de toda la información generada por el sistema, incluyendo el almacenamiento de los reportes ciudadanos y las coordenadas georreferenciadas.

- **Módulo de inteligencia artificial:** Es un componente especializado que procesa y analiza las imágenes recibidas mediante técnicas de Machine Learning, detectando si hay acumulación de residuos y estimando el nivel de volumen presente (Majchrowska et al., 2021; Sharma & Kumar, 2024).

De esta forma, todos los módulos trabajan de manera integrada: el ciudadano genera un reporte desde el módulo de interfaz, el backend lo procesa y lo envía al módulo de inteligencia artificial, este último retorna una clasificación, y finalmente los datos quedan almacenados en el módulo de datos para su posterior consulta por parte del supervisor.

Alcance del estudio

El alcance del presente estudio se concreta a los siguientes límites:

- **Prototipo funcional:** el proyecto se limita a construir un prototipo de sistema distribuido basado en microservicios, orientado al tratamiento de incidencias ciudadanas sobre acumulación de residuos sólidos urbanos, sin llegar a una versión de producción.
- **Aplicación para la ciudadanía:** se contará con una aplicación que permitirá reportar incidencias a través de una fotografía, obteniendo de manera automática la ubicación geográfica del usuario.
- **Módulo de Machine Learning:** el módulo de aprendizaje automático procesará las imágenes reportadas y de manera aproximativa deducir el nivel de severidad de los residuos. El modelo se entrena para la clasificación de una única clase, sin identificar tipos específicos de residuo.
- **Sistema web de gestión:** mediante un panel web permite presentar y analizar la información de las incidencias de forma ordenada y georreferenciada.

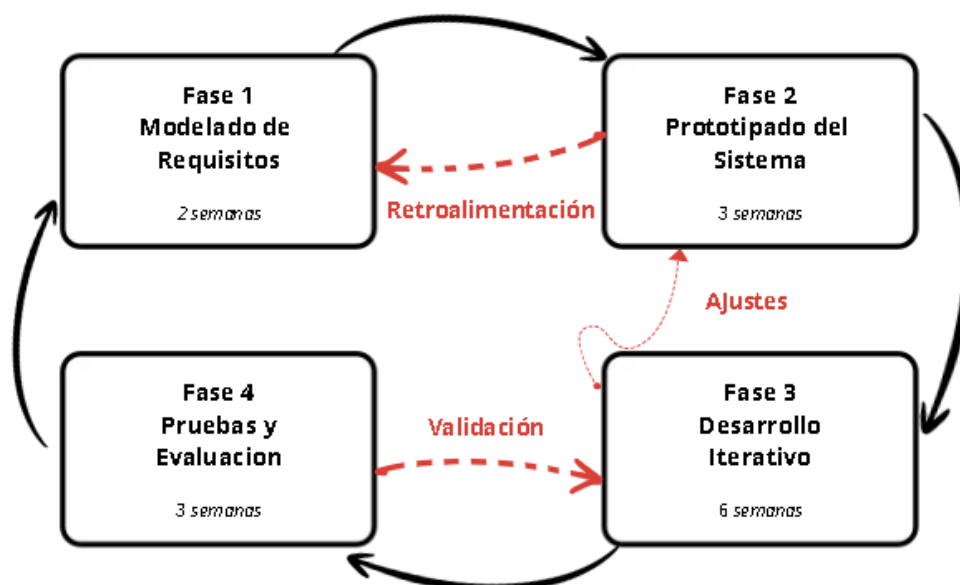
- **Entorno de validación controlado:** la verificación técnica y funcional está contenida solamente a un sector delimitado de la ciudad de Quito, con un número limitado de cincuenta participantes conformados por: ciudadanos y supervisores.
- **Demostración de viabilidad técnica:** El proyecto no contempla su implementación institucional ni su despliegue a gran escala dentro de EMASEO EP.

Fases del desarrollo del sistema

El desarrollo del sistema se estructuró siguiendo el ciclo de la metodología RAD, tal como se ilustra en la Figura 1.

Figura 1

Ciclo iterativo RAD para el desarrollo del prototipo



Nota: La Figura 1 muestra el flujo iterativo del RAD, que comprende cuatro fases secuenciales con una duración total de 14 semanas.

A continuación, se describen las fases del ciclo de desarrollo:

- **Fase 1 (2 semanas):** Al inicio del proyecto, se reunió al equipo con los supervisores operativos para definir las funcionalidades más importantes del sistema. Se establecieron tanto los requisitos funcionales como los no funcionales, y los criterios de éxito del modelo de Machine Learning.
- **Fase 2 (3 semanas):** Se diseñó la arquitectura general del sistema y se crearon las primeras versiones de las interfaces.
- **Fase 3 (6 semanas):** Esta etapa se hizo mediante cortos ciclos de una semana distribuidos en cuatro iteraciones: la primera de backend y servicios de la API, la segunda de base de datos geoespacial, la tercera de la aplicación móvil y panel web y la cuarta de entrenamiento, validación e integración del modelo de Machine Learning.
- **Fase 4 (3 semanas):** Testeo funcional del sistema, la validación del modelo de IA mediante métricas de precisión y sensibilidad, la evaluación cuantitativa de la usabilidad con el cuestionario SUS (Sistema para la Usabilidad del Software) aplicado a cincuenta participantes y evaluación cualitativa de la experiencia a través de entrevistas realizadas al personal operativo de EMASEO EP.

La retroalimentación continua entre fases fue el corazón de la metodología RAD. La recogida de requisitos fue fundamental en el desarrollo para disponer de una guía exhaustiva de lo que el sistema debe cumplir, por lo tanto, la Retroalimentación permitió vincular al prototipo con los requisitos analizados; por su parte, los Ajustes contribuyeron en el desarrollo iterativo ante cualquier fallo que podría presentarse; y la Validación sirvió para confirmar que las pruebas del prototipo fueran conformes con los resultados finales.

Fuentes de información

Fuentes Primarias.

- **Generación de datos por parte de los usuarios:** Datos georreferenciados (coordenadas GPS) e imágenes obtenidas de modo directo por los ciudadanos a través de la aplicación móvil en fase de pruebas.
- **Conjuntos de datos públicos para el entrenamiento:** Diferentes repositorios con conjuntos de datos con imágenes de residuos sólidos urbanos (es decir, "Garbage Collector v8", TACO) para el entrenamiento y validación del modelo de Machine Learning.

Fuentes Secundarias.

- Análisis bibliográfico de artículos científicos registrados en las bases de datos tanto de Scopus como ScienceDirect, con un enfoque orientado hacia la gestión de residuos sólidos urbanos, aplicaciones que integran el Machine Learning como su técnica central y sistemas para la elaboración de informes ciudadanos.
- Recolección de instructivos técnicos que incluyan las herramientas de desarrollo empleadas (Node.js, React, React Native, PostgreSQL/PostGIS, YOLO, RT-DETR).
- Normativas funcionales en las áreas de servicio y recolección de residuos sólidos, por parte de EMASEO EP, que están conformadas dentro del Reglamento Orgánico por Procesos.

Recopilación de información

Con el fin de demostrar la construcción del estado del arte que sirve para fundamentar teóricamente el proyecto, se llevó a cabo una revisión sistemática de la literatura científica.

Bases de datos científicas consultadas.

- **SCOPUS:** Base de datos multidisciplinaria de Elsevier, muy conocida por su amplia cobertura de revistas indexadas de ciencias de la computación, ingeniería y ciencias ambientales.

- **ScienceDirect:** Para obtener una búsqueda bibliográfica, utilizamos el gestor de acceso de la plataforma Elsevier revisada por pares orientativa a ciencias aplicadas e ingeniería.

Cadena de búsqueda aplicada.

Para poder identificar la literatura relevante se definió una cadena de búsqueda estructurada en función de los siguientes términos: "(municipal solid waste) OR (municipal solid waste management)" + "(machine learning) OR (deep learning) OR (artificial intelligence)" + "(GIS) OR (geographic information system) OR (geospatial analysis)" + "(waste collection) OR (route optimization) OR (vehicle routing)".

Resultados iniciales de búsqueda.

Tabla 3

Resultados iniciales por base de datos científica

Base de datos	Cantidad de artículos
SCOPUS	1101
ScienceDirect	428
Total, inicial	1529

Nota: Como se observa en la Tabla 3, la búsqueda arrojó 1529 artículos, distribuidos en SCOPUS (1101) y ScienceDirect (428), los cuales fueron sometidos a un proceso de filtrado por relevancia, duplicados y criterios de inclusión/exclusión para obtener la muestra final de estudios analizados.

Criterios de exclusión aplicados.

Tabla 4

Criterios de exclusión aplicados por base de datos

Criterio de exclusión	ScienceDirect	SCOPUS	Total eliminados
Duplicados en misma base	45	2	47
Fuera de rango (2022-2026)	52	331	383
Sin Acceso Abierto	199	378	577
Títulos no alineados	104	289	393

Nota. En la Tabla 4 se detallan los criterios de exclusión aplicados. La ausencia de obtener un acceso abierto en una base de datos presento una dificultad para la revisión de los artículos, por lo tanto, dio lugar a que exista un número elevado de eliminaciones (577), seguido de estudios cuyos títulos no se alinean con el tema (393) y artículos que no se encontraban dentro del período de tiempo considerado (383).

Criterios de inclusión temáticos.

- **Visión por ordenador aplicado hacia el Machine Learning:** Estudios que utilizan técnicas de entrenamiento automático para los diferentes modelos (YOLO, CNN, RT-DETR, EfficientDet) orientados, con el objetivo de detectar y/o clasificar los residuos sólidos.
- **Cálculo del volumen de los residuos sólidos:** Artículos en los que se estima el nivel de severidad de los residuos sólidos mediante el uso de cajas delimitadoras o bien mediante la utilización del análisis visual.
- **Georreferencia de la ubicación actual:** Estudios que hacen uso de la localización (GPS) de los usuarios que estén incluidos dentro de ese análisis, con el fin de obtener un reporte preciso de su posición en tiempo real.

- **Sistemas de reportes ciudadanos:** Artículos en los que se analicen aplicaciones móviles o plataformas de reporte participativo correspondiente a la gestión de residuos sólidos.
- **Acceso abierto a la base de datos:** Solo se han tenido en cuenta aquellos artículos que cuenten con disponibilidad pública dado que no existió un acceso institucional para las bases de datos comerciales durante el proceso de recopilación de información. Esta elección plantea una limitación reconocida: algunos artículos relevantes publicados en revistas con acceso restringido, como IEEE Transactions o Computers & Operations Research, podrían haber quedado fuera de la muestra, aunque los 36 artículos finalmente seleccionados abarcan las cuatro áreas temáticas definidas y provienen de fuentes indexadas en Scopus y ScienceDirect.

Resultados finales

Como resultado del proceso de depuración, se logró un total de **36 artículos** incluidos en el estado del arte: **22 artículos** del estado del arte propiamente dicho y **14 artículos** técnicos sobre modelos de Machine Learning que respaldan la elección de arquitecturas.

Procedimiento para la recolección de datos

1. **Identificación de residuos:** El ciudadano identifica a simple vista la acumulación de residuos en un espacio público determinado.
2. **Registro ciudadano:** El ciudadano accede a la aplicación móvil y hace click en la opción de reporte.
3. **Captura de evidencia gráfica y metadatos:** La aplicación despliega la cámara del dispositivo e, inmediatamente después, con los servicios de geolocalización, obtiene las coordenadas geográficas (GPS) del espacio.

4. **Transmisión de datos:** La aplicación agrupa los datos (imagen, coordenadas, hora y etiqueta del usuario) que se estructuran en JSON y finalmente, con un llamado POST, los transmite al microservicio asignado a la parte del backend.
5. **Almacenamiento y enrutamiento:** El backend evalúa la petición y almacena los datos en PostgreSQL/PostGIS y pone en cola la imagen para el módulo de Machine Learning.

Procesamiento y análisis de los datos

1. **Normalización y limpieza:** Los datos introducidos son chequeados, se validan las coordenadas GPS del perímetro virtual y las imágenes pasan a un tamaño de 640×640 píxeles.
2. **Análisis mediante Machine Learning:** El microservicio de IA extrae la información de la imagen y estructura el resultado de acuerdo a una de las clases de severidad (BAJO, MEDIO, ALTO O CRÍTICO).
3. **Enriquecimiento de la incidencia:** Se complementa el informe con la categoría que devuelve el modelo y una identificación hash.
4. **Validación humana:** A través del panel web, los supervisores observan la imagen, la localización y la categoría que proporciona el modelo.

Función de priorización de tres variables ponderadas

El nivel de severidad final se determina a través de la siguiente función de priorización, que ha sido diseñada como parte de esta investigación para integrar la evidencia visual del modelo de ML con el contexto histórico y territorial:

$$\text{Prioridad} = 0.60 \cdot \text{effective_ratio} + 0.25 \cdot \text{frecuencia_historica} + 0.15 \cdot \text{ubicacion_estrategica}$$

Donde:

- *effective_ratio*: La severidad calculada por medio de pipeline de ML que supuso la magnitud y la confianza relacionadas con la detección de la aparición de restos (Zhao et al., 2024; Majchrowska et al., 2021).
- *frecuencia_historica*: La cantidad de reportes en el mismo sector en los últimos 7 días (Pardini et al., 2020).
- *ubicacion_estrategica*: Valor logico que permite identificar con precisión la localización real para la persona que se encuentre dentro de un sector de alta prioridad determinado por EMASEO EP, tal cual esta conformado entre mercados, paradas de transporte y zonas turísticas (EMASEO EP, 2022).

Pesos asignados:

- $w_1 = 0.60$ (60%): El modelo de ML es el factor más relevante ya que proporciona evidencia visual objetiva del nivel de severidad del volumen estimado (Zhao et al., 2024; Majchrowska et al., 2021).
- $w_2 = 0.25$ (25%): La frecuencia histórica permite identificar zonas con problemas recurrentes que requieren atención preventiva (Pardini et al., 2020).
- $w_3 = 0.15$ (15%): La ubicación estratégica prioriza sectores de alto impacto operativo para EMASEO EP (EMASEO EP, 2022).

El resultado de esta función se asigna a las categorías de severidad que se encuentran en la Tabla 14, donde valores ≥ 0.75 corresponden a nivel CRÍTICO y valores < 0.25 a nivel BAJO.

Técnicas estadísticas y métricas de evaluación**Matriz de confusión para la detección de residuos**

La evaluación del modelo de Machine Learning se basó en una matriz de confusión, que divide los resultados en cuatro escenarios:

- **Verdaderos Positivos (VP):** El modelo identifica correctamente una acumulación de residuos que efectivamente está presente.
- **Falsos Positivos (FP):** El modelo señala la presencia de residuos cuando en realidad no los hay.
- **Falsos Negativos (FN):** Hay una acumulación real de residuos, pero el modelo no logra detectarla.
- **Verdaderos Negativos (VN):** El modelo analiza una imagen sin residuos y clasifica adecuadamente que no hay acumulación.

Métricas de rendimiento del modelo de detección.

- **Precisión (Precision):**

$$Precisión = \frac{VP}{VP+FP}$$

Donde:

- *VP*: Número de detecciones correctas donde el modelo identificó correctamente la presencia de residuos.
- *FP*: Número de detecciones incorrectas donde el modelo identificó residuos cuando en realidad no los había (Majchrowska et al., 2021).

- **Sensibilidad o Recall:**

$$Recall = \frac{VP}{VP+FN}$$

Donde:

- *VP*: Número de detecciones correctas (Majchrowska et al., 2021).
- *FN*: Número de acumulaciones reales que el modelo no logró detectar.

- **F1-Score:**

$$F1 = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

El F1-Score combina precisión y recall en un solo valor, siendo útil para evaluar el equilibrio del modelo (Majchrowska et al., 2021).

Estimación del nivel de severidad (*effective_ratio*)

Tras que el modelo identifique los residuos en la imagen, se calcula un indicador heurístico conocido como *effective_ratio*. Su objetivo es evaluar la gravedad del amontonamiento y dar prioridad a las incidencias. El *effective_ratio* es un producto de cuatro factores derivados directamente de las observaciones del modelo RT-DETR-L v2. El procesamiento asincrónico de los procesos de inferencia se utiliza con la biblioteca OpenCV para la preparación de imágenes y con la biblioteca CLIP (Contrastive Language-Image Pre-training) de OpenAI para la validación semántica de las observaciones detectadas (Sharma et al., 2026).

La fórmula general del *effective_ratio* es:

$$\text{effective_ratio} = \text{coverage_ratio} \times \text{conf_factor} \times \text{det_factor} \times \text{class_weight}$$

Donde:

- *coverage_ratio*: Proporción de la región de la imagen cubierta por todas las cajas delimitadoras que contienen la acumulación de residuos. Indica cuánta extensión visual tiene el conjunto total de las acumulaciones de residuos, según Sharma y Kumar (2024).
- *conf_factor*: Factor basado en la media de la confiabilidad de las detecciones, normalizado con un valor básico de 0.60 (Zhao et al., 2024).
- *det_factor*: Factor basado en un logaritmo del número de detecciones que reduce la saturación antes de tiempo (Majchrowska et al., 2021).
- *class_weight*: Se refiere a la ponderación relativa de un tipo específico de residuo, definido por CLIP (Sharma et al., 2026).

Calibración de los parámetros heurísticos

El *effective_ratio* es un valor escalar entre $[0, 1]$ que recopila toda la señal de detección antes de ser asignada a una cuantificación de severidad. Este procedimiento calculado consta de cuatro variables que evalúan distintos elementos en la detección.

$$\text{effective_ratio} = \text{cov_ratio} (\text{Cobertura visual}) \times \text{conf}/\text{CONF_BASE} (\text{Confianza normalizada}) \times (1 - e^{-k \cdot n}) (\text{Saturación por detecciones}) \times w_c (\text{Peso por tipo de residuo})$$

Componentes de la fórmula:

- *cov_ratio*: Impacto del área ocupada por la unión concreta de todas las cajas delimitadoras que da la medida de qué tan extensa es la acumulación.
- *conf*: Este se refiere a la confianza promedio de las detecciones. El proceso para normalizar es dividiendo por $\text{CONF_BASE} = 0.60$, este valor de referencia es el que representa el umbral mínimo para las detecciones para que este se considere como detección válida.
- *n*: Variable que sirve para conocer el total de las detecciones (cajas delimitadoras) en la dentro de la imagen.
- $k = \text{DET_FACTOR_K} = 0.50$: Constante que regula la rapidez de saturación del coeficiente logarítmico. El término $(1 - e^{-k \cdot n})$ tiene un aumento explosivo en las primeras detecciones, y posteriormente se estabiliza a medida que aumenta el conteo.
- *w_c*: Peso que está asociado al tipo de residuo que predomina el cual se ha determinado mediante la clasificación semántica con CLIP.

Los parámetros que regulan esta fórmula han sido optimizados mediante un proceso iterativo sobre situaciones de prueba reales observadas en los espacios operativos de EMASEO EP en Quito. La Tabla 5 registra el valor final de cada parámetro, su tarea y la razón por la cual ha sido ajustado.

Tabla 5

Parámetros calibrados del effective_ratio y su justificación

Parámetro	Valor	Función y criterio de ajuste
<i>Factor de confianza</i>		
CONF_NORMALIZATION_BASELINE	0.60	Confianza de referencia; detecciones con confianza ≥ 0.60 obtienen factor = 1.0. Toda detección que supera el filtro de entrada alcanza al menos este nivel de certeza.
<i>Factor logarítmico de detecciones</i>		
DET_FACTOR_K	0.50	Controla la pendiente de la curva de saturación. Permite que múltiples detecciones incrementen el factor sin que una sola detección lo eleve excesivamente.
DET_FACTOR_CEILING	0.90	Tope máximo del factor. Este impide que exista un conteo muy alto de cajas delimitadoras y se eleve el effective_ratio a 1.0 sin importar el área de cobertura. Es la penalización de por escala.
<i>Penalización de escala</i>		
FULL_FRAME_COVERAGE_THRESHOLD	0.85	Cobertura mínima para ser considerado necesario para penalizar fuertemente: si hay una sola detección que ocupa más del 85% del área de la imagen, este indica que existe un objeto en primer plano no una acumulación de basura en la vía pública.
FULL_FRAME_PENALTY	0.20	Un multiplicador de valor mínimo sería el resultado de una situación en la que detectamos el objeto y casi ocupa toda la totalidad del encuadre sin que evidencien la aparición de basura. Interpolación según el tipo de textura del objeto.

Parámetro	Valor	Función y criterio de ajuste
ISOLATION_COVERAGE_THRESHOLD	0.55	Cobertura moderada en caso de detección única: posible objeto aislado (mochila, bolso) confundido con residuos. Activa una suavización de penalización.
ISOLATION_PENALTY	0.40	Multiplicador base de detección única.
ISOLATION_MAX_SINGLE_PENALTY	0.65	Techo del multiplicador de detección única. Con la falta de este límite establecido, generalmente un objeto de alta confianza puede anular la penalización, lo que provoca que exista un rescate de pila única.
<i>Rescate de pila única</i>		
PILE_RESCUE_MAX_DETS	2	Número máximo de detecciones para poner en marcha el rescate: con una pila real que está compuesta por una o dos cajas adyacentes en el entorno para originar la agrupación de estos valores.
PILE_RESCUE_MIN_COVERAGE	0.18	Cobertura mínima que se encuentra presente para afirmar si la acumulación detectada es válida, por medio de un pequeño objeto aislado en la zona analizada.
PILE_RESCUE_MIN_SCORE	0.58	Franja mínima de probabilidad, para considerar la textura de la basura como tal. Conocidos objetos, tales como mochilas o carteras (0.45 - 0.55), que hacían dispararse erróneamente el rescate han pasado a tener un 0.45 - 0.58 en busca de mejorar la precisión a un nivel superior.
PILE_RESCUE_DET_FLOOR	1.0	Parámetro de probabilidad mínima para el factor de detección, en la consideración de la pila como real: anula la penalización logarítmica de una única detección, cuando se establece que una acumulación de residuos sólidos es considerado real.

Parámetro	Valor	Función y criterio de ajuste
Diversidad geométrica para banda CRÍTICO		
CRITICO_MIN_DETS	3	Cantidad máxima de detecciones para iniciar el rescate: lo que produce que una o dos cajas se mantengan de forma adyacente estructurando el mismo blob.effective_ratio en el interior de la pila real.

Nota. La Tabla 5 presenta los parámetros calibrados del effective_ratio en cinco bloques funcionales. La calibración se llevó a cabo de forma iterativa con casos reales en Quito, lo que permitió reducir los falsos positivos; surgiendo de esta calibración, destaca cómo ISOLATION_PENALTY se ajustó de un valor 0.65 hacia uno de 0.40, mientras que ISOLATION_MAX_SINGLE_PENALTY no presento modificaciones significativas (con valor = 0.65) tras el incidente F2998975, en el que una mochila fue clasificada erróneamente como ALTO.

El proceso de calibración se desarrolló de forma recurrente mediante la selección de las imágenes más representativas de los falsos positivos que son más frecuentes en el campo (personas con mochilas, bolsas de compra, objetos oscuros sobre el pavimento y apilamientos reales de fundas), ejecución del pipeline, el registro del effective_ratio y la banda de severidad correspondiente a cada caso. Mientras que las métricas son calibradas en función de los falsos positivos para que se encuentren por debajo del límite Medio, por lo tanto, esto produce que las acumulaciones reales con tamaño entre medio y grande logren alcanzar las bandas de ALTO o CRÍTICO. Distintos elementos existentes como lo son: mochilas texturizadas, utensilios de cocina, etc. Estaban consideradas como ALTO (con eff_ratio = 0.71), sin embargo, para darles un valor real en base a su métrica de detección, se tuvo que ajustar el ISOLATION_PENALTY que paso de 0.65 a 0.40 y a la introducción del ISOLATION_MAX_SINGLE_PENALTY = 0.65.

Clasificación en niveles y estimación de volumen

Una vez se ha calculado el *effective_ratio*, el sistema lo clasifica según cuatro niveles de severidad: BAJO, MEDIO, ALTO y CRÍTICO, de forma que para obtener un valor de volumen dentro de cada rango el sistema aplica una interpolación lineal:

$$Volumen = V_{mín} + \frac{effective_ratio - R_{mín}}{R_{máx} - R_{mín}} \times (V_{máx} - V_{mín})$$

Donde: $V_{mín}$ y $V_{máx}$ son los límites de volumen del nivel de severidad, y $R_{mín}$, $R_{máx}$ son los límites del rango del *effective_ratio* para esa banda.

La calibración de los rangos de volumen se llevó a cabo tomando como referencia las anotaciones del dataset TACO y la literatura especializada (Proença & Alexandre, 2020; Sharma & Kumar, 2024).

Capítulo IV

Análisis del método actual de manejo de residuos domiciliarios

El servicio de recolección y barrido de residuos sólidos urbanos en el Distrito Metropolitano de Quito, gestionado por la Empresa Pública Metropolitana de Aseo (EMASEO EP), funciona actualmente bajo un esquema mayormente rígido. La planificación operativa se lleva a cabo en función de rutas y horarios predefinidos, los cuales se establecen periódicamente según la asignación administrativa de recursos, aunque no siempre logran adaptarse a los patrones reales de generación de residuos en las diferentes áreas de la ciudad (EMASEO EP, 2022).

Tabla 6

Características del método actual de gestión de residuos en EMASEO EP

Aspecto	Descripción
Planificación	Basada en rutas y horarios fijos, sin ajuste dinámico
Asignación de recursos	Distribución periódica según disponibilidad administrativa
Identificación de puntos críticos	Reactiva (quejas ciudadanas) o por inspección programada
Tecnología utilizada	Sistema básico de gestión administrativa
Participación ciudadana	Canales tradicionales (línea telefónica, oficinas físicas)
Tiempo de respuesta a incidencias	Variable, sin métrica objetiva de priorización

Nota. La Tabla 6 resume el esquema de gestión actual de EMASEO EP.

A partir del análisis del método actual, se han detectado varios aspectos que necesitan atención prioritaria:

1. **Datos en directo:** Sin presencia de un mecanismo automático que se encargue de la detección anticipada de las acumulaciones de residuos que están catalogadas como las más críticas en los distintos puntos del sector.

- 2. **Contar con rutas dinámicas:** Se hace necesario tener flexibilidad y adaptabilidad a la hora de poner en marcha la funcionalidad de recogida de los residuos sólidos para los diferentes eventos detallados en cada ruta, con el propósito de optimizar tiempos de llegada.
- 3. **Resulta crucial tener una priorización objetiva de incidencias:** Se necesita una escala cuantitativa que permita clasificar las atenciones según el nivel de severidad.
- 4. **Se solicita un sistema automatizado de análisis visual:** La verificación de puntos críticos demanda un enfoque que posibilite el análisis automático de imágenes.

Evaluación de los requerimientos del sistema

Requerimientos funcionales

Tabla 7

Requerimientos funcionales del sistema

Código	Módulo	Requerimiento
RF-01	Móvil	Permitir al ciudadano registrarse e iniciar sesión en la aplicación
RF-02	Móvil	Capturar la fotografía de acumulaciones de basura usando la cámara del dispositivo
RF-03	Móvil	Obtener automáticamente las coordenadas GPS del lugar del reporte
RF-04	Móvil	Enviar el reporte (imagen + coordenadas + metadata) al backend
RF-05	Móvil	Visualizar el historial de reportes realizados por el usuario
RF-06	Móvil	Almacenar reportes localmente cuando no hay conexión
RF-07	Backend	Recibir y validar reportes ciudadanos mediante API REST
RF-08	Backend	Autenticar usuarios
RF-09	Backend	Gestionar usuarios con control de acceso basado en roles (RBAC)

Código	Módulo	Requerimiento
RF-10	Backend	Almacenar imágenes para su respectivo análisis
RF-11	ML	Procesar imágenes mediante un modelo de Machine Learning para detectar el volumen estimado de los residuos
RF-12	ML	Estimar nivel de severidad (BAJO/MEDIO/ALTO/CRÍTICO)
RF-13	ML	Calcular volumen estimado de residuos en metros cúbicos
RF-14	Web	Filtrar reportes por nivel de severidad y estado
RF-15	Web	Visualizar estadísticas de recogida
RF-16	General	Gestionar usuarios (supervisores y ciudadanos)

Nota. La Tabla 7 agrupa los requerimientos funcionales por módulo del sistema.

Requerimientos no funcionales

Tabla 8

Requerimientos no funcionales del sistema

Código	Categoría	Requerimiento
RNF-01	Rendimiento	El tiempo de respuesta de la API debe ser menor a 5 segundos para el 95% de las peticiones
RNF-02	Rendimiento	La inferencia del modelo ML debe completarse en menos de 1 minuto por imagen
RNF-03	Rendimiento	El sistema debe soportar al menos 50 peticiones simultáneas sin degradación
RNF-04	Seguridad	Las contraseñas deben almacenarse usando bcrypt con factor de costo 12
RNF-05	Seguridad	Todas las comunicaciones deben cifrarse mediante TLS 1.3 en producción
RNF-06	Seguridad	Los JWT deben tener un tiempo de expiración máximo de 24 horas

RNF-07	Disponibilidad	El sistema debe tener una disponibilidad del 99.5% en horario de operación
RNF-08	Escalabilidad	La arquitectura de microservicios debe permitir escalar horizontalmente
RNF-09	Usabilidad	La aplicación móvil debe obtener una puntuación SUS ≥ 70
RNF-10	Usabilidad	El flujo de reporte ciudadano no debe exceder los 3 pasos (foto → confirmar → enviar)
RNF-11	Mantenibilidad	El código debe seguir estándares ESLint y estar documentado
RNF-12	Portabilidad	El sistema debe ejecutarse en contenedores Docker para cualquier entorno

Nota. La Tabla 8 detalla los requerimientos no funcionales del sistema.

Arquitectura general del sistema

El sistema cuenta con una arquitectura de microservicios donde se define de manera clara y concisa las distintas responsabilidades que tiene cada componente, por lo tanto, se puede conseguir el despliegue de cada módulo de forma individual debido a que son independientes uno del otro. A partir de esta arquitectura, se presenta un esquema estructural basado en la división a través de sucesivos niveles de cada módulo, donde se empieza por una visión general del sistema hasta descender a los detalles específicos de cada unidad implementada y como interactúan entre ellas.

Diagrama de Contexto (C1)

El diagrama contextual que se presenta en la Figura 2 sirve para mostrar como MIC EMASEO actúa como una única unidad de software, ya que tiene una integración interna compuesta por tres roles de usuario y tres sistemas externos con los que se comunican mutuamente. Por lo tanto, el usuario que actúa como ciudadano será aquel que reporta acumulaciones de basura a través de una foto y su ubicación GPS recuperada de forma automática; el supervisor examina los incidentes de las zonas

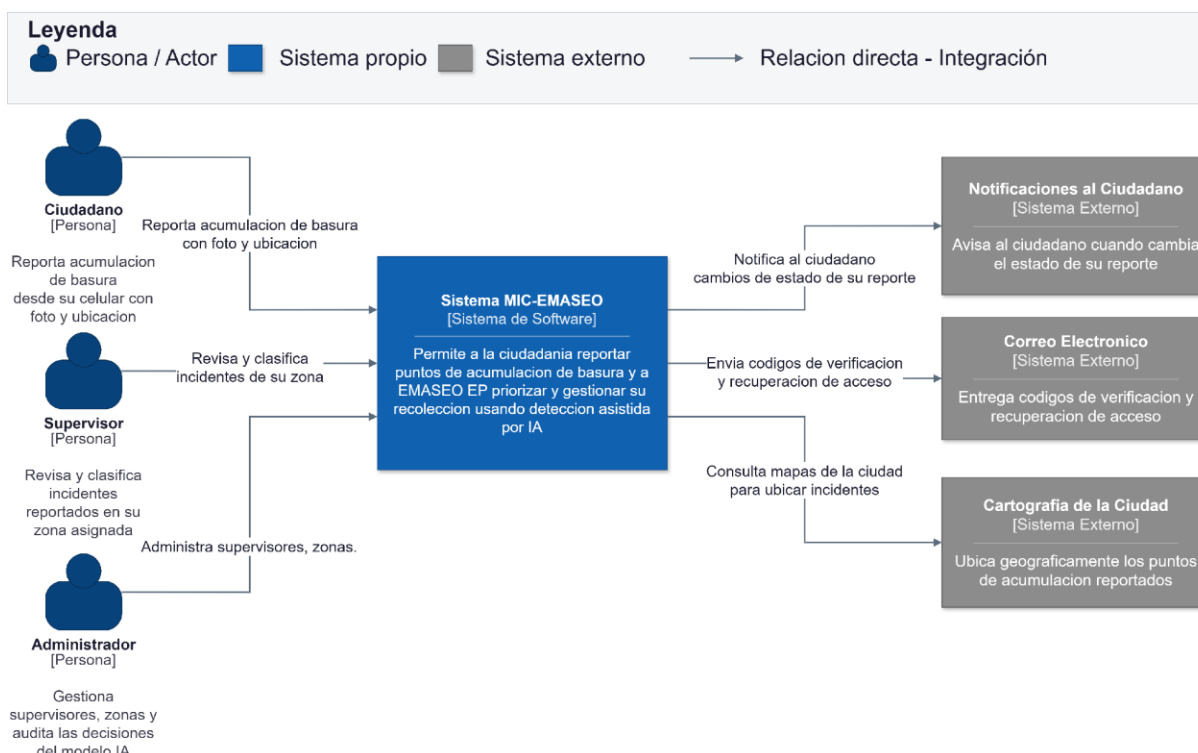
correspondientes dentro del Distrito Metropolitano de Quito (DMQ) y tomará la decisión correcta en base a la información planteada. Reflejando la distribución operativa por sectores de EMASEO EP.

El ciudadano reporta acumulaciones de basura junto con una foto y la ubicación GPS; el supervisor examina y clasifica los incidentes en la zona que le corresponde, reflejando la distribución operativa por sectores de EMASEO EP. Por su parte, el administrador se encarga de gestionar a los supervisores y las áreas, además de auditar las decisiones del modelo de inteligencia artificial para detectar posibles errores en la clasificación.

Hacia el exterior, el sistema delega tres funciones ajenas a su lógica de negocio: notificar al ciudadano los cambios de estado de su reporte, enviar códigos de verificación por correo electrónico, y resolver coordenadas y límites geográficos mediante un servicio de cartografía, evitando así reimplementar infraestructura ya resuelta por terceros.

Figura 2

Diagrama de contexto del sistema MIC-EMASEO (Nivel C1)



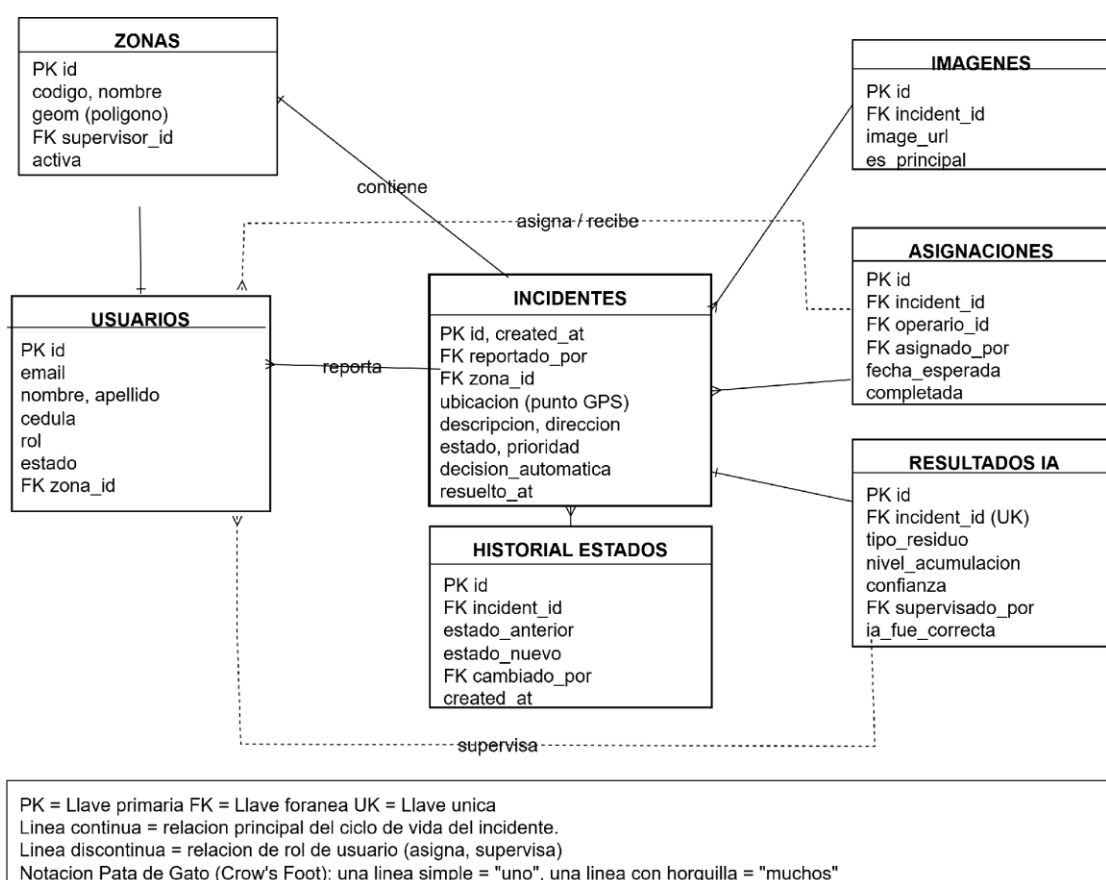
Nota: La Figura 2 nos permitió presentar una arquitectura con su propia independencia externa, de tal forma que sus tres integraciones (notificaciones, correo, cartografía) no forman parte de la lógica del sistema, y, por tanto, compartir, intercambiar en cualquiera de ellas no afecta para nada la lógica de gestión de incidencias.

Diseño de la base de datos geoespacial

El sistema configura PostgreSQL, con la extensión PostGIS, como el motor de base de datos relacional, a los efectos de poder expresar las capacidades nativas de almacenamiento y análisis de datos espaciales. La decisión de usar PostGIS permite realizar consultas geoespaciales eficientes. La Figura 3 presenta el modelo entidad-relación del sistema.

Figura 3

Modelo entidad-relación de la base de datos del sistema MIC-EMASEO



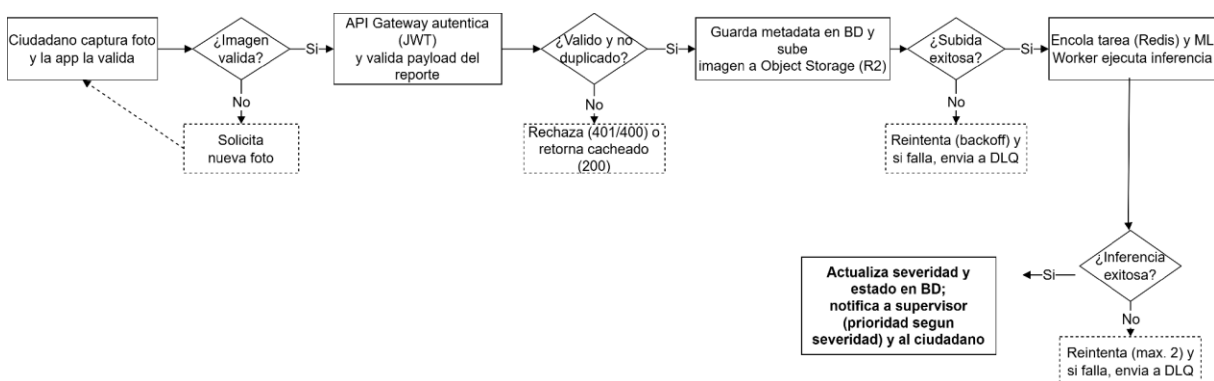
Nota. Para salvaguardar los datos personales de los usuarios, se implementó la extensión pgcrypto de PostgreSQL, que permite cifrar correos electrónicos utilizando la función pgp_sym_encrypt. La clave de cifrado se maneja como una variable de entorno del servidor, asegurando que los datos sensibles no puedan ser recuperados en texto plano desde la base de datos.

Diseño del flujo de procesamiento de incidencias

El proceso de una incidencia desde el reporte ciudadano hasta su análisis por el modelo de ML sigue el flujo que se describe a continuación y que se muestra en la Figura 4.

Figura 4

Flujo de reporte de incidencia



Nota. La Figura 4 muestra la secuencia de pasos desde que el ciudadano reporta una incidencia hasta que el modelo de Machine Learning genere un valor en base al nivel de severidad de la acumulación.

Diseño del módulo de Machine Learning

Selección y construcción del dataset

El dataset de entrenamiento fue construido combinando cuatro fuentes públicas de imágenes de residuos sólidos urbanos:

Tabla 9*Composición del dataset de entrenamiento*

Dataset fuente	Imágenes	Porcentaje
TACO (Trash Annotations in Context)	10,343	47.04%
Garbage Collector v8	5,691	25.88%
Garbage Detection	4,002	18.20%
StreetTrash	1,951	8.87%
Total	21,987	100%

Nota. La Tabla 9 muestra la composición del dataset con 21,987 imágenes, donde TACO aporta la mayor proporción (47.04%). Todas las imágenes se unificaron en una sola clase (garbage) dado que el objetivo del sistema no es clasificar el tipo de residuo sino detectar su presencia y estimar el nivel de severidad.

Selección del modelo de detección

Se revisaron tres diferentes arquitecturas para la detección de objetos:

Tabla 10*Comparativa de arquitecturas de detección evaluadas*

Modelo	mAP@50	Latencia (ms)	Parámetros (M)	Seleccionado
YOLOv8-L	0.8521	22	43.6	No
EfficientDet-D3	0.8134	38	12.0	No
RT-DETR-L v2	0.8802	47	32.9	Sí

Nota. La Tabla 10 nos muestra el resultado de tres diferentes donde el modelo RT-DETR-L v2 se ha seleccionado por su precisión de 0.8802 frente a 0.8521 de YOLOv8-L y 0.8134 de EfficientDet-D3.

Entrenamiento del modelo

El modelo fue entrenado con las siguientes configuraciones:

Tabla 11

Configuración de entrenamiento del modelo RT-DETR-L v2

Parámetro	Valor
Épocas	100 (mejor resultado en época 64)
Tamaño de imagen	640 × 640 píxeles
Batch size	16
Optimizador	AdamW
Learning rate inicial	0.0001
Scheduler	Cosine annealing
Augmentación	HSV, flip, mosaic, mixup
División datos	70% train / 20% val / 10% test

Nota. La Tabla 11 resume los 8 hiperparámetros utilizados durante el entrenamiento del modelo a lo largo de 100 épocas.

Resultados del entrenamiento (época 64).

Tabla 12

Métricas del modelo RT-DETR-L v2 mejor época (64/100)

Métrica	Valor
mAP@50	0.8802
mAP@50:95	0.6124
Precisión	0.8934
Recall	0.8571
F1-Score	0.8749
Latencia media	47 ms/imagen

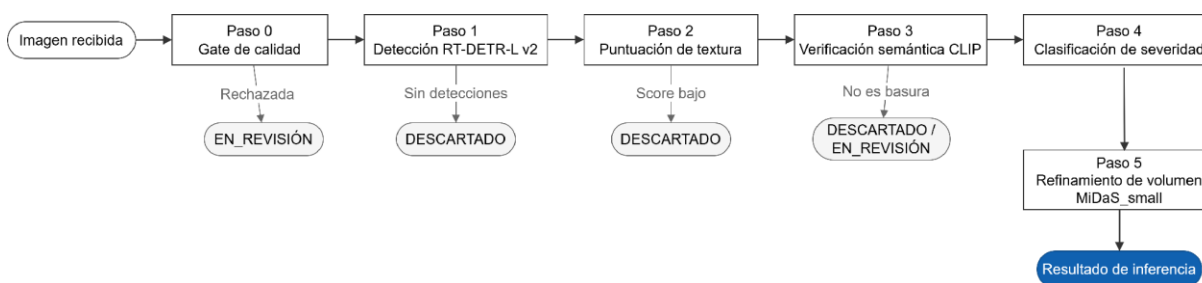
Nota. La Tabla 12 muestra el rendimiento del modelo en su mejor época (64 de 100), con 6 métricas de evaluación.

Parámetros y flujo de inferencia del pipeline de ML

El pipeline para el análisis de imágenes consta de cuatro etapas obligatorias y una etapa opcional para el refinamiento del volumen, como se ilustra en la Figura 5. Cada paso dentro del flujo funciona como filtros, donde si alguno no cumple con los criterios establecidos simplemente se detiene y devuelve un estado dependiendo en que etapa se encuentre sin poder avanzar en el flujo.

Figura 5

Pipeline de inferencia del modelo ML



Nota. La figura 5 revela el flujo de procesamiento de las imágenes obtenidas de los reportes, donde cada etapa se encarga de un filtro en específico.

Descripción del flujo:

- Paso 0 - Control de calidad.** Para calcular la varianza de la imagen se utiliza el operador diferencial Laplaciano. Este valor se compara con el de BLUR_VARIANCE_MIN, si es menor se considera la imagen demasiado borrosa o enfocada con lo que se le cambia el estado a en REVISIÓN.
- Paso 1 - Detección (RT-DETR-L v2).** Las cajas delimitadoras (bounding boxes) son el núcleo operativo para el modelo que detecta de forma aproximada las acumulaciones de residuos sólidos, por esta razón, se aplica lo que son las detecciones de confianza ≥ 0.60 , $IoU \leq 0.50$ y una mínima área del 1 % del frame (MIN_BBOX_AREA_RATIO), por lo tanto, si no se encuentra en el intervalo establecido provocara que el resultado obtenido pase a ser DESCARTADO.
- Paso 2 - Evaluación de textura:** Se combina la entropía de color (0.45), la densidad de bordes (0.40) y la posición vertical (0.15). Si la puntuación es inferior a 0.20 (GARBAGE_SCORE_HARD_FLOOR), se descarta la imagen como un falso positivo.
- Paso 3 - Verificación semántica:** Genera una probabilidad garbage_prob. Los valores por debajo de 0.30 son DESCARTADOS; aquellos entre 0.30 y 0.62 se marcan “EN REVISIÓN” para evaluación por parte del supervisor; los que superan 0.62 avanzan al Paso 4.
- Paso 4 - Clasificación de severidad:** Se calcula el effective_ratio: $effective_ratio = cov_ratio \times conf_factor \times det_factor \times class_weight$ y se asigna a una de cuatro bandas: BAJO [0.00–0.25], MEDIO [0.25–0.50], ALTO [0.50–0.75] o CRÍTICO [0.75–1.00], según lo indicado en la Tabla 14. El volumen se determina mediante interpolación lineal dentro del rango correspondiente a la banda asignada.

- **Paso 5 - Refinamiento del volumen:** Con MidDas_samll se usando la técnica de percepción profundidad monocular en aquellas áreas que se han detectado y calculando el estimado del volumen geométrico, donde limita el rango vol_band_min y vol_ban_max manteniendo así la coherencia entre los niveles de severidad y el volumen que se ha reportado. En dado caso que falle el modelo, el sistema sostiene el volumen calculado en la interpolación como un respaldo.

La Tabla 13 agrupa los parámetros del microservicio ML:

Tabla 13

Parámetros de inferencia verificados del pipeline ML

Parámetro	Valor	Descripción
<i>Paso 1 - Detección · RT-DETR-L v2</i>		
NMS_CONF	0.60	Confianza mínima para retener una detección.
NMS_IOU	0.50	IoU máximo para suprimir bounding boxes duplicadas.
MIN_BBOX_AREA_RATIO	0.010	Área mínima de la bounding box respecto al frame.
<i>Paso 2 - Textura · garbage_score</i>		
GARBAGE_SCORE_HARD_FLOOR	0.20	Puntuación mínima de para la textura, si se encuentra por debajo se descarta automáticamente la imagen por ser un falso positivo.
<i>Paso 3 - Semántico · CLIP RN50</i>		
SEMANTIC_REJECT_THRESHOLD	0.20	Cuando la variable se encuentra por debajo de este valor se descarta automáticamente la imagen.
SEMANTIC_REVIEW_THRESHOLD	0.50	Si se encuentra por encima se acepta, es decir entre 0.20 y 0.50 se marca para revisión del supervisor.
<i>Paso 4 - Severidad · classify_severity</i>		

Parámetro	Valor	Descripción
CONF_NORMALIZATION_BASELINE	0.60	Confianza que se toma de referencia para normalizar el valor de <code>conf_factor</code> dentro del <code>effective_ratio</code> .

Paso 5 - Volumen · MiDaS_small

USE_MIDAS_VOLUME	true	Depura el volumen estimado mediante la técnica de profundidad monocular y acotando al rango de la banda asignada del Paso 4.
------------------	------	--

Nota. La Tabla 13 abarca los parámetros del pipeline de ML en un flujo de cinco pasos bien estructurados.

Clasificación de severidad en bandas

El `effective_ratio` que resulta de la inferencia se asigna a cuatro niveles de severidad, como se especifica en la Tabla 14

Tabla 14

Bandas de clasificación de severidad y volumen estimado

Nivel	Rango <code>effective_ratio</code>	Volumen estimado	Acción sugerida
BAJO	0.00 – 0.25	0.0 – 0.5 m ³	Atención en siguiente ruta
MEDIO	0.25 – 0.50	0.5 – 2.0 m ³	Priorizar en 24 h
ALTO	0.50 – 0.75	2.0 – 5.0 m ³	Atención urgente (< 12 h)
CRÍTICO	0.75 – 1.00	> 5.0 m ³	Movilizar unidad inmediatamente

Nota: La Tabla 14 muestra la variable del `effective_ratio` en cuatro niveles de severidad (BAJO, MEDIO, ALTO, CRÍTICO), los cuales pertenecen a rangos de volumen y como indicador para las operaciones dentro de EMASEO EP.

Diseño de la aplicación móvil

El aplicativo móvil se desarrolló con la tecnología de React Native usando Expo Go para el entorno de pruebas. El uso como tal de la aplicación está pensado para que los ciudadanos puedan capturar los reportes de forma fácil y transparente con el manejo de los datos recolectados.

Dentro del flujo del aplicativo nos pedirá en primera instancia iniciar sesión como se muestra en la Figura 6, donde el usuario iniciará con su correo electrónico y contraseña. Cabe recalcar que si el usuario no tiene cuenta podrá crearla con el botón llamado “Crear cuenta” o a su vez recuperar su contraseña con el botón “¿Olvidaste tu contraseña?” usando su correo registrado.

Figura 6

Pantalla de inicio de sesión

The image shows a mobile application login screen for 'EMASEO EP IA Sistema Inteligente de Residuos'. At the top, there is a logo and the text 'EMASEO EP IA Sistema Inteligente de Residuos'. Below this, the word 'Bienvenido' is displayed in a large font, followed by the instruction 'Inicia sesion para continuar'. There are two input fields: 'Correo electrónico' with a placeholder 'ejemplo@correo.com' and an envelope icon, and 'Contraseña' with a placeholder 'Ingresa tu contraseña', a lock icon, and a toggle icon. Below the password field is a link '¿Olvidaste tu contraseña?'. A large blue button labeled 'Ingresar' is highlighted with a red border. At the bottom, there is a link '¿Aún no tienes cuenta?' followed by a blue button labeled 'Crear cuenta', which is also highlighted with a red border.

Nota. La Figura 6 corresponde a la interfaz de acceso principal del sistema, desde la cual los usuarios autenticados inician su interacción con la aplicación móvil.

Antes de registrarse, el usuario tiene que aceptar el aviso de privacidad, como se puede observar en la Figura 7, que se presenta en dos pantallas: la primera describe quién es responsable del tratamiento y qué datos se recopilan; la segunda aclara la base legal (Art. 7 y 22 LOPDP), el tiempo de conservación y los derechos ARCOP del titular.

Figura 7

Aviso de privacidad y consentimiento informado

Aviso de privacidad
LOPDP — Ecuador

Base legal (Art. 7 y 22 LOPDP)
Tu consentimiento expreso (este aviso) y la prestación de un servicio de interés público municipal constituyen la base legal del tratamiento.

Conservación de datos
Tus datos se conservan mientras mantengas una cuenta activa. Al eliminar tu cuenta, los datos se anonimizarán dentro de los 30 días siguientes, salvo obligación legal de retención.

Tus derechos (Art. 17 LOPDP)
Como titular tienes derecho a:

- Acceso — consultar qué datos tenemos sobre ti.
- Rectificación — corregir datos inexactos.
- Cancelación — solicitar la eliminación de tus datos.
- Oposición — oponerte a ciertos usos de tus datos.
- Portabilidad — recibir tus datos en formato legible.

Al pulsar **Acepto y continúo** otorgas tu consentimiento libre, informado y específico conforme al Art. 7 de la LOPDP.

Aviso de privacidad
LOPDP — Ecuador

Responsable del tratamiento
Estudiantes investigadores de la carrera de Ingeniería en Tecnologías de la Información de la **Universidad de las Fuerzas Armadas – ESPE**, en el marco del **Proyecto de Integración Curricular**. EMASEO EP actúa como entidad municipal colaboradora del proyecto.

Datos que recopilamos

- Identificación: nombre completo, cédula de identidad, fecha de nacimiento, sexo.
- Contacto: número de celular y correo electrónico.
- Ubicación GPS: coordenadas del punto donde reportas una acumulación de basura.
- Imágenes: fotografías tomadas desde la cámara de tu dispositivo.

Finalidad del tratamiento
Los datos se usan **exclusivamente** para:

- Crear y gestionar tu cuenta de ciudadano.
- Registrar y dar seguimiento a reportes de acumulación de residuos en vía pública.
- Notificarte sobre el estado de tus reportes.
- Mejorar la precisión del modelo de detección de basura (uso académico/investigativo).

Nota. El usuario debe pulsar “Acepto y continúo” para otorgar su consentimiento explícito conforme al Art. 7 de la LOPDP.

Una vez autenticado, el usuario puede acceder al menú principal que aparece en la Figura 8, desde donde tiene la opción de iniciar el proceso para reportar incidencias, revisar su historial o gestionar su perfil.

Figura 8

Pantalla del menú principal



Nota. En la Figura 8 se visualiza el menú principal con diferentes funcionalidades compuestas por el reporte de la incidencia, historial y configuración del perfil.

Al seleccionar la opción de reporte, se activa la cámara con guías visuales para encuadrar la escena de la Figura 9. La interfaz incluye indicadores de distancia e iluminación, y un marco que cambia a verde cuando la imagen está lista para captura automática.

Figura 9

Pantallas de captura y confirmación de reporte



Nota. La figura 9 captura el momento en que se toma la fotografía para lo cual se guía de un marco de referencia donde le pide que “Encuadre la basura” y confirma los datos.

Diseño del panel web de supervisión y administración

Panel de supervisión

Figura 10

Pantalla de inicio de sesión del panel de supervisión

EMASEO
grande otra vez
EMASEO EP · SISTEMA INTELIGENTE DE RESIDUOS

Panel de Supervisión

Acceso para supervisores y administradores
Correo electrónico

Contraseña

☐ Recordar sesión [¿Olvidaste tu contraseña?](#)

Ingresar

Sistema de Gestión Inteligente de Residuos · EMASEO EP

Nota. Como se puede evidenciar en la Figura 10, es la puerta de acceso para el panel de control, de lo cual este tiene como propósito apoyar al supervisor para que pueda hacer un correcto seguimiento y toma de decisiones de las incidencias presentadas.

Las funciones principales que tiene el panel web son: Representación de las incidencias activas a través de un mapa de calor con colores distintivos de los sectores que conforman el DMQ (Distrito Metropolitano de Quito) a través de Leaflet.js., filtrado por nivel de severidad y estado (pendiente, asignado y atendido) y la generación de informes estadísticos por sector y periodo de tiempo.

En la pantalla principal, un mapa de calor visualiza de manera concentrada los puntos críticos de acumulación, lo que permite al supervisor priorizar sin necesidad de revisar cada reporte individualmente.

Figura 11

Pantalla principal del panel de supervisión con mapa de incidencias

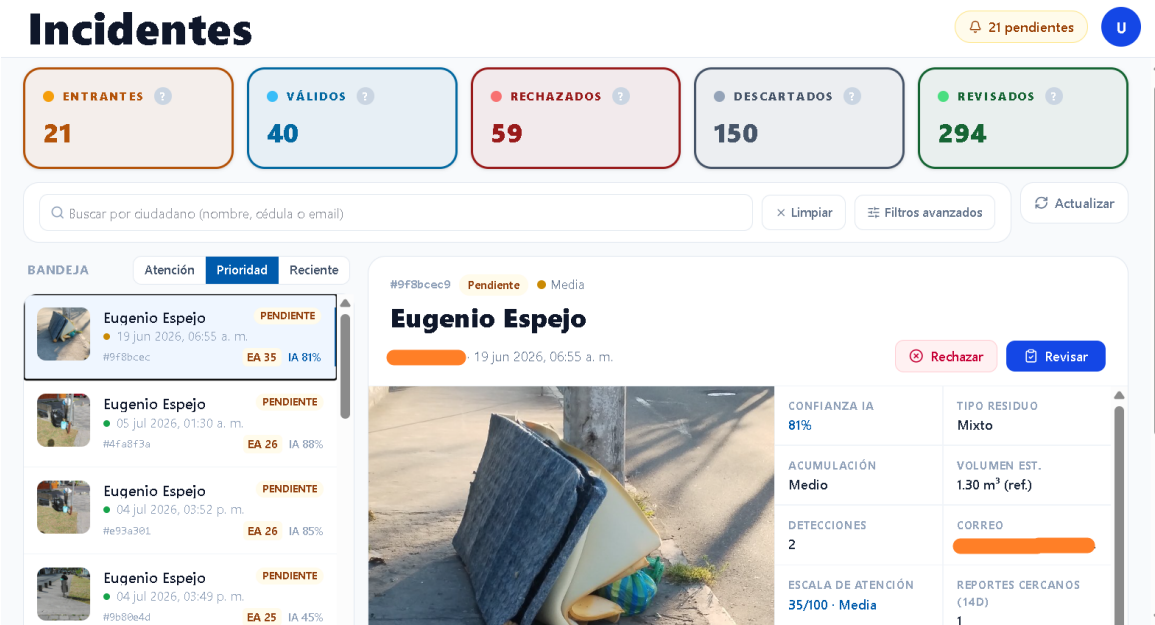


Nota: La Figura 11 presenta la interfaz geoespacial del sistema; los marcadores, codificados por colores, distinguen el nivel de severidad de cada incidencia reportada.

El módulo de gestión de reportes da a los supervisores la posibilidad de abrir el detalle de cada incidencia y clasificarla según su estado: entrante, pendiente, válida, en atención, descartada o IA incorrecta.

Figura 12

Módulo de gestión de reportes del panel de supervisión



Nota. La Figura 12 presenta la bandeja de incidencias del supervisor, donde cada reporte se muestra con su estado, nivel de severidad y fecha de registro.

Panel de administración

El panel de administración está diseñado específicamente para los administradores del sistema. Contiene funciones especializadas relacionadas con la gestión y la configuración del sistema. El acceso al mismo requiere autenticación exclusiva a través de la interfaz presentada en la Figura 13.

Figura 13

Pantalla de inicio de sesión del panel de administración

EMASEO
grande otra vez
EMASEO EP · SISTEMA INTELIGENTE DE RESIDUOS

Panel Administrador

Acceso exclusivo para administradores

Correo electrónico

ejemplo@correo.com

Contraseña

Ingresa tu contraseña

☐ Recordar sesión [¿Olvidaste tu contraseña?](#)

Ingresar

Sistema de Gestión Inteligente de Residuos · EMASEO EP

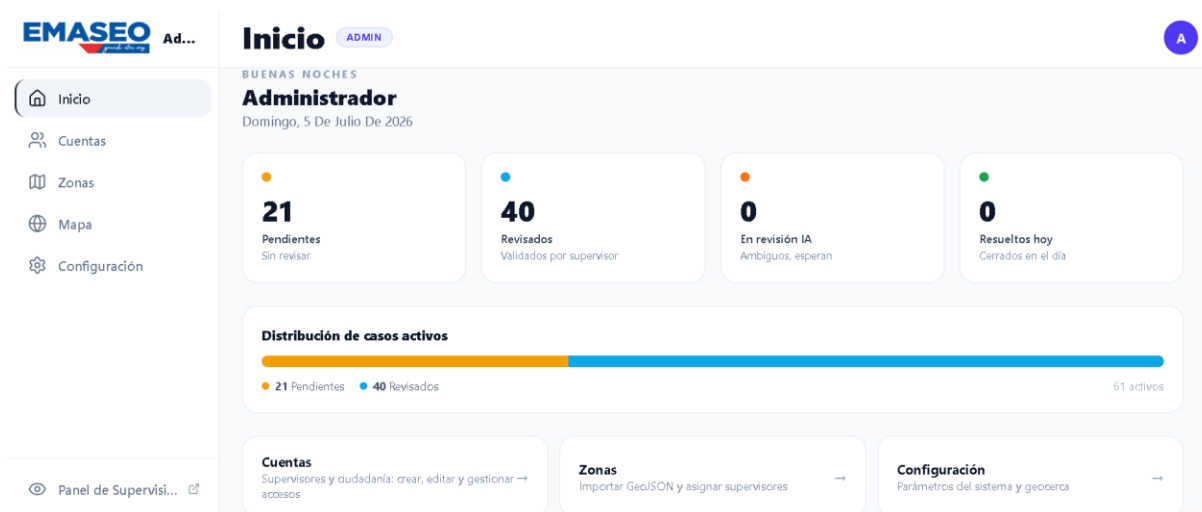
Nota. La Figura 13 muestra el inicio de sesión al panel de administración.

El panel de administración proporciona las siguientes funcionalidades principales:

- **Dashboard de indicadores:** En cuestión permite la visualización en tiempo real de indicadores claves conocidos como KPIs.
- **Gestión de supervisores:** Creación, edición y desactivación de supervisores.
- **Gestión de zonas:** Importación de archivos GeoJSON para delimitar zonas operativas y asignación de supervisores por zona.

Figura 14

Pantalla principal del panel de administración con indicadores y acceso a módulos de gestión



Nota. La Figura 14 presenta el tablero de control del administrador, desde el cual se accede a los módulos de gestión de supervisores, zonas y configuración del sistema.

Capítulo V

Pruebas del sistema

Se llevaron a cabo varias pruebas sobre el prototipo en las diferentes dimensiones: Como primer punto tenemos las pruebas funcionales de cada uno de los módulos que componen todo el sistema, como segundo punto tenemos la implementación de la evaluación de la productividad de la API REST mediante la herramienta Locust, como tercer punto se tuvo lugar una medición de la percepción de la usabilidad mediante la utilización del cuestionario SUS, por último implementar una evaluación cualitativa de la experiencia de usuario a través de entrevistas a los usuarios de EMASEO EP.

Pruebas funcionales

Las pruebas funcionales se diseñaron para verificar que cada requerimiento funcional (RF-01 a RF-16) estuviera correctamente implementado. Se aplicó una metodología de caja negra, evaluando el comportamiento del sistema ante entradas válidas e inválidas sin inspeccionar el código interno de cada módulo.

Tabla 15

Resumen de pruebas funcionales ejecutadas

Código	Módulo	Caso de prueba	Resultado
RF-01	Móvil	Login con credenciales válidas e inválidas	✓
RF-02	Móvil	Captura de fotografía con cámara activa	✓
RF-03	Móvil	Obtención automática de coordenadas GPS	✓
RF-04	Móvil	Envío de reporte (imagen + metadata)	✓
RF-05	Móvil	Visualización del historial de reportes	✓
RF-06	Móvil	Almacenamiento offline y sincronización posterior	✓

Código	Módulo	Caso de prueba	Resultado
RF-07	Backend	Recepción y validación de reportes por API REST	✓
RF-08	Backend	Autenticación con JWT (tokens válidos/expirados)	✓
RF-09	Backend	Control de acceso basado en roles (RBAC)	✓
RF-10	Backend	Almacenamiento de objetos	✓
RF-11	ML	Proceso de imagen y detección de residuos	✓
RF-12	ML	Clasificación por nivel de severidad BAJO/MEDIO/ALTO/CRÍTICO	✓
RF-13	ML	Aproximación de volumen estimado en m ³	✓
RF-14	Web	Filtro de reportes por severidad y estado	✓
RF-15	Web	Visualización de estadísticas por sector	✓
RF-16	General	Gestión de usuarios	✓

Nota. Tabla 15 muestra los 16 casos de prueba funcionales ejecutados divididos por módulo:

Móvil, Backend, Machine Learning, Web y General. Todos los casos fueron confirmados como exitosos, lo que prueba mediante una técnica de prueba de caja negra la correcta implementación de los requisitos funcionales del sistema.

Evidencia de pruebas funcionales - Módulo móvil (RF-01 a RF-06)

La aplicación móvil fue evaluada en dispositivos Android para comprobar el correcto funcionamiento de los requerimientos funcionales del módulo móvil utilizando una metodología de caja negra.

El requerimiento RF-01 (Login con credenciales válidas e inválidas) se valida a través de la pantalla de inicio de sesión, como se muestra en la Figura 15, para acceder al sistema.

Figura 15

Pantalla de inicio de sesión (RF-01)

EMASEO **EMASEO EP IA**
Sistema Inteligente de Residuos

Bienvenido

Inicia sesión para continuar

Correo electrónico

✉ jsmena5@espe.edu.ec

Contraseña

🔒 🔓

[¿Olvidaste tu contraseña?](#)

Ingresar

[¿Aún no tienes cuenta? Crear cuenta](#)

Nota. La Figura 15 corresponde a la interfaz de acceso principal de la aplicación, desde la cual los usuarios inician sesión en el aplicativo móvil.

El requerimiento RF-02 (Captura de fotografía con cámara activa) se valida a través de la interfaz de captura, que guía al usuario para mantener una distancia adecuada, asegurar buena iluminación y centrar la acumulación de residuos en el marco.

Figura 16

Interfaz de encuadre para captura de fotografía (RF-02)

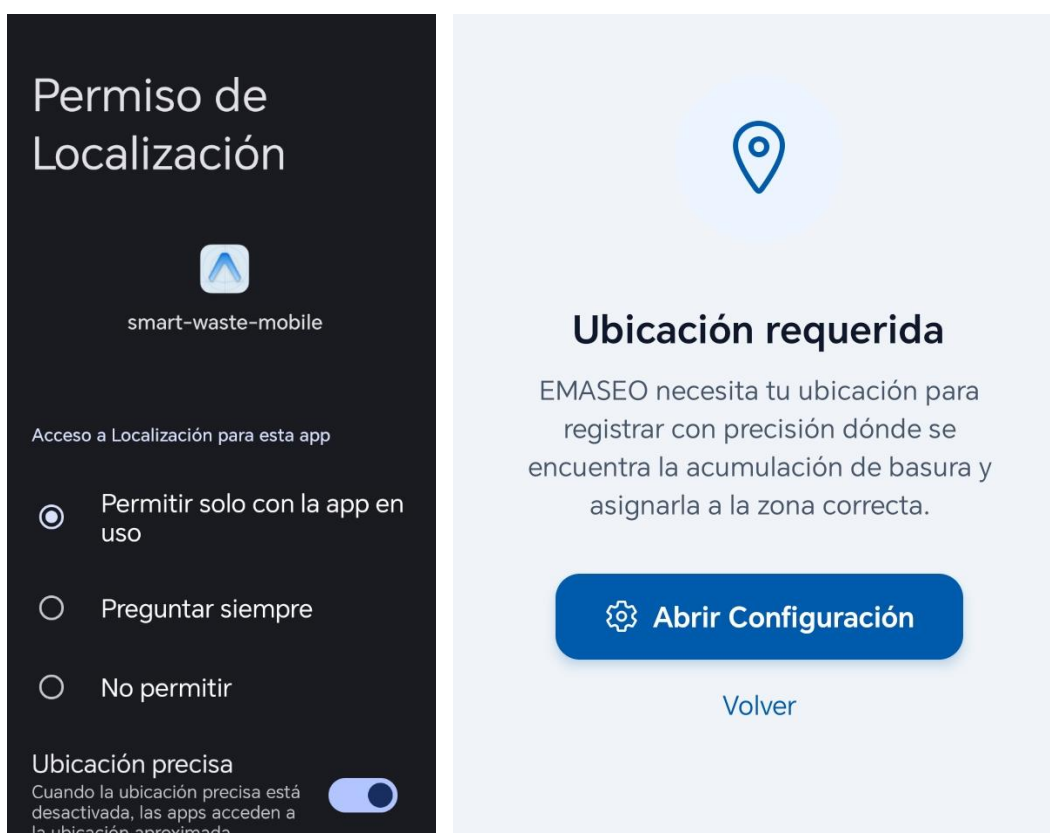


Nota. La Figura 16 presenta guías visuales de encuadre para centrar bien lo residuos.

El requerimiento RF-03 (Obtención automática de coordenadas GPS) se activa mediante la solicitud de permisos de ubicación al iniciar la captura de una incidencia.

Figura 17

Solicitud y confirmación de permisos de ubicación GPS (RF-03)



Nota. La Figura 17 muestra la solicitud de permisos de ubicación precisa y el mensaje que confirma que la ubicación es requerida para georreferenciar la acumulación de basura y asignarla a la zona correcta.

El requerimiento RF-04 como se puede observar en la Figura 18 se puede visualizar cuando la foto y la ubicación están capturadas.

Figura 18

Previsualización de foto capturada y confirmación de envío (RF-04)

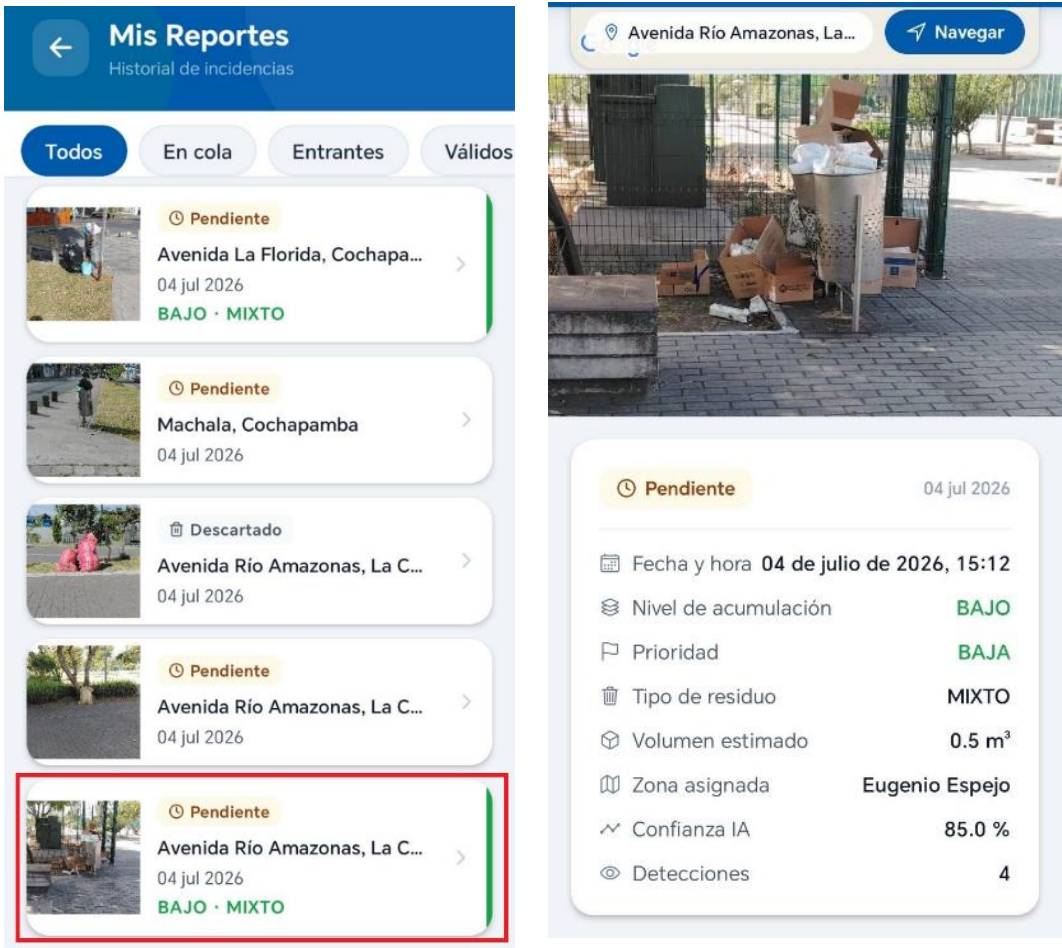


Nota. La Figura 18 permite verificar que la foto capturada y la ubicación GPS disponible se adjuntan al reporte antes de enviarlo al backend para su análisis.

El requerimiento RF-05 (Visualización del historial de reportes) se comprueba mediante la pantalla de historial, donde el usuario puede consultar todos los reportes enviados junto con su estado y

fecha. Al seleccionar un reporte, se presenta el detalle completo de la incidencia tal como se observa en la Figura 19.

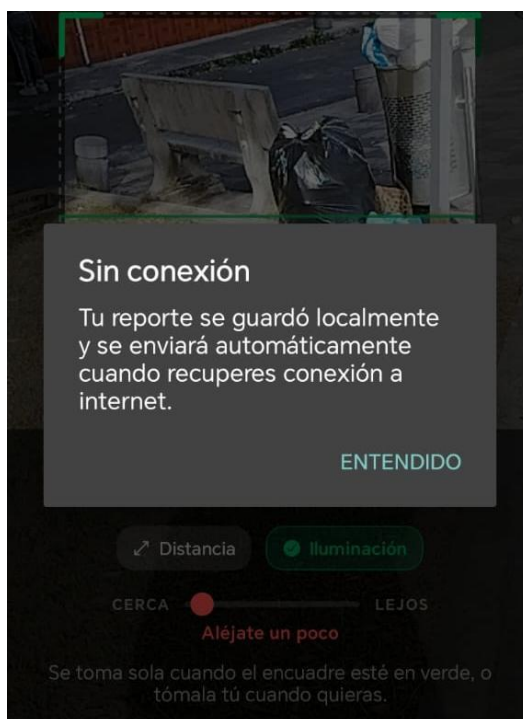
Figura 19
Historial de reportes y detalle de incidencia (RF-05)



Nota. La Figura 19 muestra el listado de reportes con estado y fecha, y el detalle de una incidencia con información de nivel de acumulación, volumen estimado, zona asignada y confianza IA.

Figura 20

Mensaje de almacenamiento offline (RF-06)



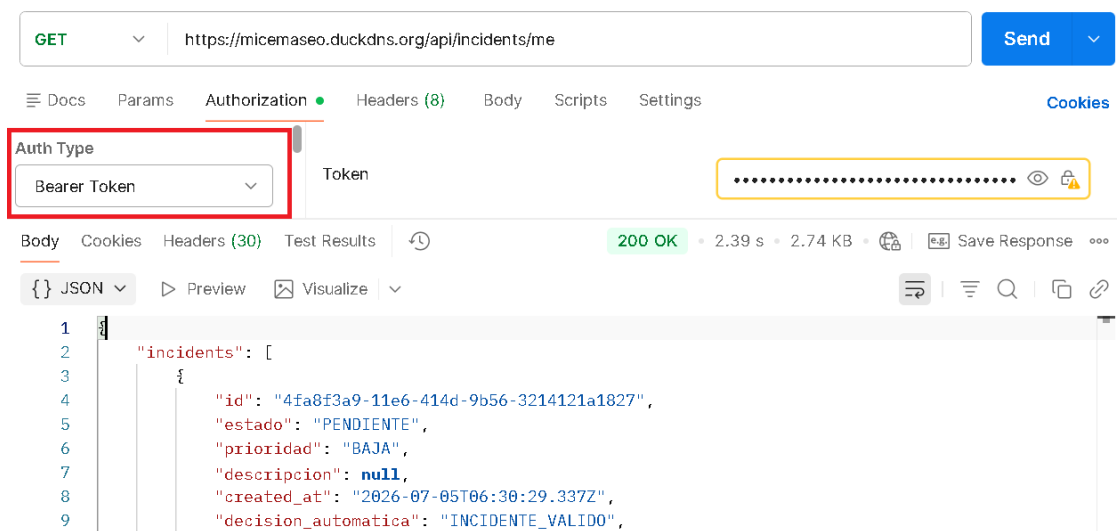
Nota: La Figura 20 confirma que el reporte se almacena localmente sin conexión y se sincroniza automáticamente al recuperar Internet, permitiendo el funcionamiento en zonas sin cobertura.

Evidencia de pruebas funcionales - Backend y almacenamiento (RF-07 a RF-10)

Las pruebas se realizaron con la herramienta de Postman usando la autenticación JWT y el control de acceso en roles. La validación del RF-07 se lo hace a través de la consulta hacia el endpoint GET /api/incidents/me, que devuelve los incidentes que el ciudadano autenticado ha reportado.

Figura 21

Lista de incidentes del ciudadano autenticado (RF-07)

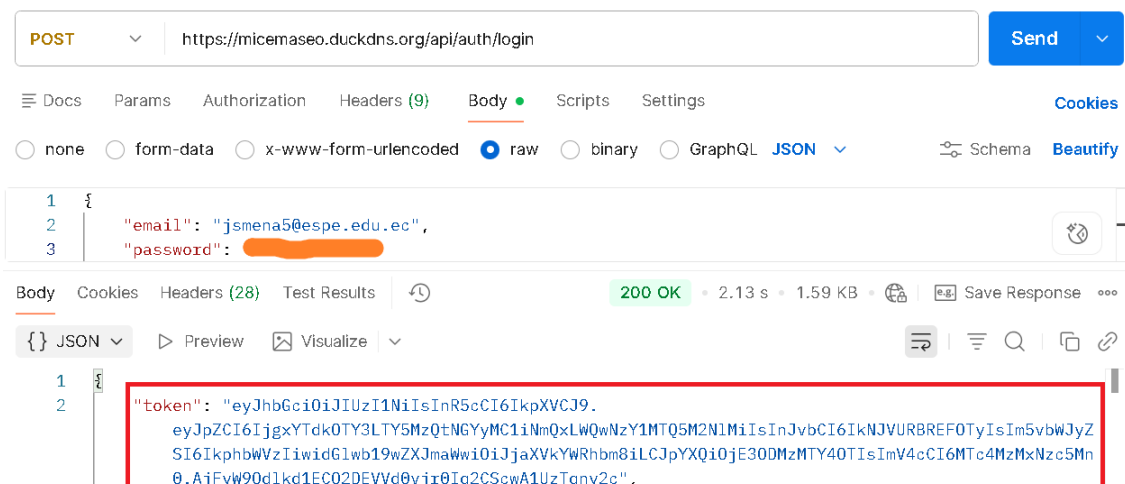


Nota. La Figura 21 muestra el listado de incidentes recuperados exitosamente, donde se observa la información disponible para cada reporte registrado por el ciudadano.

El requerimiento RF-08 (Autenticación con JWT) se valida mediante pruebas de inicio de sesión tanto exitosas como fallidas.

Figura 22

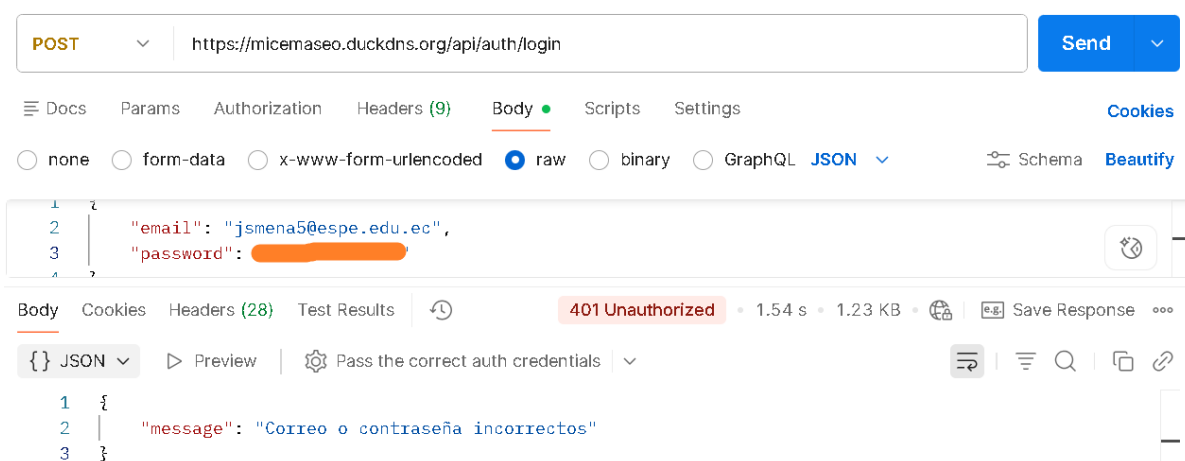
Login exitoso con credenciales válidas (RF-08)



Nota. La Figura 22 evidencia el login exitoso (200 OK) con credenciales válidas, donde se ha ocultado la contraseña por razones de privacidad. El servidor responde con el token JWT necesario para las peticiones autenticadas.

Figura 23

Login fallido con credenciales inválidas (RF-08)



Nota. La Figura 23 hace aparición el mensaje de error 401 Unauthorized como resultado de intentarse autenticar con unas credenciales incorrectas al validar el rechazo de accesos no autorizados.

Tal y como se puede apreciar en el RF-09, el control de acceso basado en roles queda validado al intentar acceder un usuario con el rol CIUDADANO al endpoint `/api/supervisor/incidents` obteniéndose la respuesta HTTP 403 general que indica que se necesita el rol SUPERVISOR o ADMIN.

Figura 24

Control de acceso RBAC: denegación a usuario no autorizado (RF-09)



Nota. La Figura 24 muestra el mensaje de error " Acceso denegado. Se requiere uno de: SUPERVISOR, ADMIN" al intentar acceder al endpoint de supervisión con un rol CIUDADANO, validando la restricción de acceso basada en roles RBAC.

El almacenamiento de imágenes en la nube (RF-10) se evidencia mediante el panel de Cloudflare R2. La Figura 25 ilustra un objeto concreto que reside en el bucket `emaseo incidents`. Con ello exponemos la URL pública de desarrollo y los metadatos a que da lugar.

Figura 25

Detalles de un objeto en Cloudflare R2: URL pública y metadatos (RF-10)

R2 Object Storage

emaseo-incidents

incidents/3115dfd7-5ff4-4819-9c1c-f450091334da.jpg

Object Details

Date Created	Type	Storage Class	Size
07 Jun 2026 10:14:25 ECT	image/jpeg	Standard	575.4 kB

URLs

Public Development URL

<https://pub-af32eac3385247b89227fef093ec6f6e.r2.dev/incidents/3115dfd7-5f>

Custom Metadata

No custom metadata set

Object Preview



Nota. La figura 25 pone en escena esos metadatos del objeto, la url pública de desarrollo y por último la gestión y persistencia de las trazas de las imágenes del móvil subidas a Cloudflare R2.

Evidencia de pruebas funcionales - Módulo de Machine Learning (RF-11 a RF-13)

El módulo de Machine Learning fue evaluado mediante el envío de imágenes al pipeline de inferencia, verificando el procesamiento, la detección de residuos y la clasificación resultante.

La Figura 26 muestra el resultado de la detección de residuos (RF-11), donde el modelo RT-DETR-L v2 identifica múltiples objetos en la imagen con sus respectivos niveles de confianza, representados mediante bounding boxes superpuestas.

Figura 26

Detección de residuos con bounding boxes y niveles de confianza (RF-11)



Nota. La Figura 26 resalta los cuadros detectados de varios objetos que estos etiquetados como garbage con los niveles de confianza entre el 0.6 y 0.9.

La Figura 27 presenta el detalle de una incidencia procesada por el sistema (RF-12), donde se visualiza la clasificación por nivel de severidad y la confianza de la IA.

Figura 27

Clasificación por nivel de severidad y detalle de incidencia (RF-12)

#82ec5fc1

En atención

Baja

Los Chillos

23 jun 2026, 09:59 a. m.

✓ En atención

✎ Editar

🏠 Marcar como resuelta



Ampliar

CONFIANZA IA 69%	TIPO RESIDUO Mixto
ACUMULACIÓN Alto	VOLUMEN EST. 1.90 m ³ (ref.)
DETECCIONES 3	CORREO ...
ESCALA DE ATENCIÓN 49/100 · Media	REPORTES CERCANOS (14D) 6
DIRECCIÓN —	

► Contexto del reporte

► Diagnóstico IA

Nota. La Figura 27 muestra el diagnóstico de IA con nivel de severidad **Alto**, confianza del 69%, 3 detecciones y tipo de residuo **Mixto**, confirmando la correcta clasificación de la incidencia.

La estimación del volumen en metros cúbicos (RF-13) se muestra en la Figura 28, donde el sistema ofrece un diagnóstico completo de la IA, que incluye la confianza, el volumen estimado, el tipo de residuo, el número de detecciones y el nivel de acumulación.

Figura 28

Diagnóstico completo de IA con estimación de volumen en m³ (RF-13)



Nota. La Figura 28 detalla el diagnóstico con confianza del 77%, volumen estimado de 1.38 m³, nivel Medio y tiempo de procesamiento de 4501 ms, validando la estimación volumétrica del pipeline ML.

Evidencia de pruebas funcionales - Panel web de supervisión (RF-14 a RF-15)

El panel web de supervisión fue evaluado para comprobar sus funciones principales: el filtrado de reportes y la visualización de estadísticas.

En la Figura 29 se presenta el módulo de gestión de reportes, donde los supervisores tienen la posibilidad de filtrar incidencias según su estado (todos los estados) y nivel de severidad (Alta), permitiendo ver la lista de incidentes junto con su información pertinente.

Figura 29

Filtrado de reportes por estado y nivel de severidad (RF-14)

ESTADO / RECHAZOS: Todos los estados

SEVERIDAD: Alta

DECISIÓN DE LA IA: Todas las decisiones IA

RANGO DE FECHAS: dd/mm/aaaa

☐ Marcada como incorrecta por un supervisor

BANDEJA: Atención **Prioridad** Reciente

Los Chillos #58197c5f **En atención** Alta

15 jun 2026, 02:05 p. m.

✓ En atención Editar Marcar como resuelta

CONFIANZA IA: 81%

ACUMULACIÓN: Alto

DETECCIONES: 2

ESCALA DE ATENCIÓN: 50/100 - Alta

TIPO RESIDUO: Mixto

VOLUMEN EST.: 1.90 m³ (ref.)

CORREO: [Progress Bar]

REPORTES CERCANOS (14D): 1

Nota. La Figura 29 presenta los resultados del filtrado aplicado, mostrando los incidentes que coinciden con los criterios de búsqueda seleccionados.

La Figura 30 resalta el panel de estadísticas por sector, mostrando las áreas con mayor cantidad de incidentes (Los Chillos, Valle de Sangolquí, Manuela Sáenz) y el total de incidencias.

Figura 30

Visualización de estadísticas por sector (RF-15)



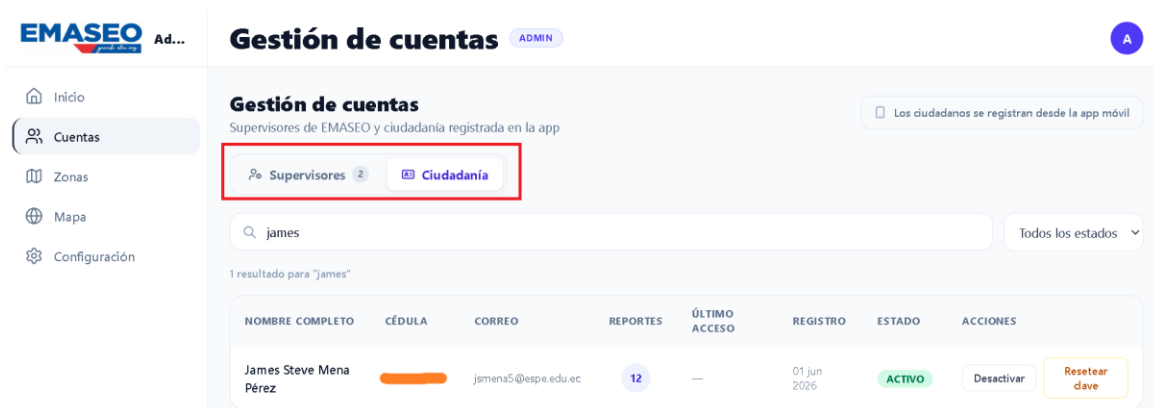
Nota. La Figura 30 refleja el comportamiento histórico de los reportes, organizado por zonas geográficas.

Evidencia de pruebas funcionales - Gestión de usuarios (RF-16)

La gestión de usuarios (RF-16) se valida mediante el módulo de gestión de cuentas del panel de administración, donde se visualizan y gestionan los usuarios del sistema, incluyendo supervisores de EMASEO EP y ciudadanía registrada en la aplicación.

Figura 31

Módulo de gestión de cuentas: supervisores y ciudadanía (RF-16)



Nota. La figura 31 nos muestra varias opciones para gestionar las cuentas de los usuarios de Supervisores y la ciudadanía del sistema.

Pruebas de rendimiento y concurrencia

Se realizaron pruebas de rendimiento sobre los endpoints críticos que se exponen por medio de API REST para validar el requerimiento RNF-01Y EL RNF-03.

Metodología de prueba de carga

Mediante la herramienta Locust se realizaron las pruebas de carga bajo un marco de trabajo basado en Python el cual permite la simulación de usuarios concurrentes y el medir tiempo de respuesta.

- **Escenario 1 (Ciudadanos):** Conformados por 500 usuarios concurrentes, ramp-up de 5 usuarios/s, endpoint GET /api/incidents/me, distribuidos en fases entre 149, 350 y 500 usuarios.
- **Escenario 2 (Supervisores):** 150 usuarios concurrentes, ramp-up de 10 usuarios/s, endpoint GET /api/supervisor/incidents, distribuidos en fases entre 70, 110 y 150 usuarios.

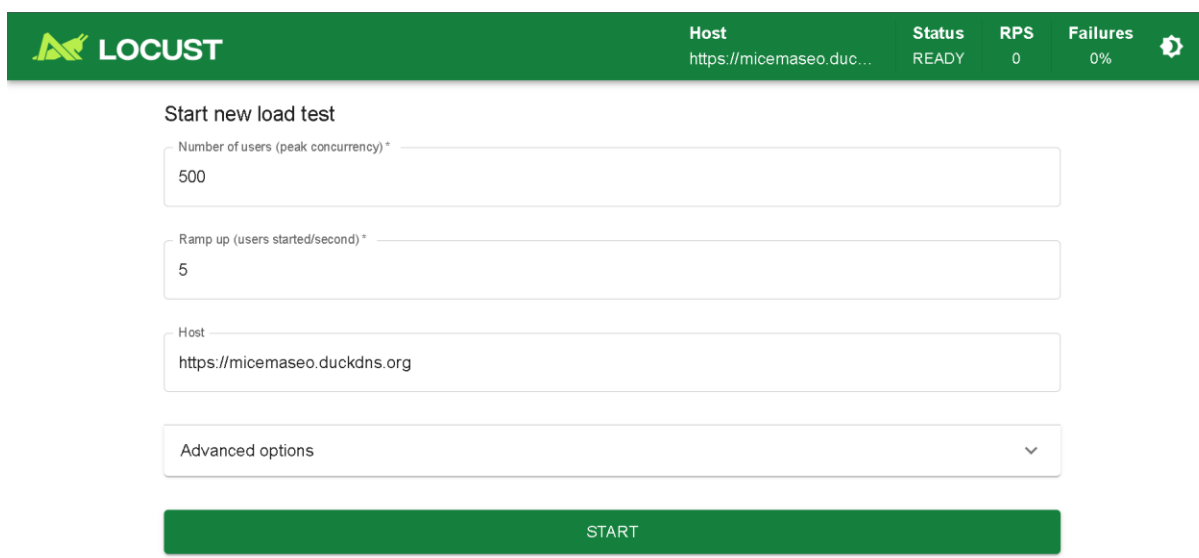
Cada usuario empleó tokens fijos para evitar limitaciones de tasa y medir exclusivamente el rendimiento de los endpoints.

Escenario 1: 500 usuarios concurrentes (Ciudadanos)

La Figura 32 ilustra la configuración utilizada para el escenario con 500 usuarios concurrentes.

Figura 32

Configuración de Locust para el escenario de 500 usuarios concurrentes



The screenshot shows the Locust web interface. At the top, there is a green header with the Locust logo and the text "LOCUST". To the right of the header, there is a table with the following columns: Host, Status, RPS, Failures, and a gear icon. The Host column contains the URL "https://micemaseo.duc...". The Status column contains "READY". The RPS column contains "0". The Failures column contains "0%".

Below the header, there is a section titled "Start new load test". It contains four input fields:

- "Number of users (peak concurrency) *": 500
- "Ramp up (users started/second) *": 5
- "Host": https://micemaseo.duckdns.org
- "Advanced options": A dropdown menu with a downward arrow.

 At the bottom of this section, there is a green button labeled "START".

Nota. La Figura 32 muestra ramp-up de 5 usuarios/s y duración de 60 s para simular 500 ciudadanos en el endpoint /api/incidents/me.

La Tabla 16 muestra los resultados obtenidos en este contexto.

Tabla 16

Resultados de pruebas de rendimiento con 500 usuarios concurrentes

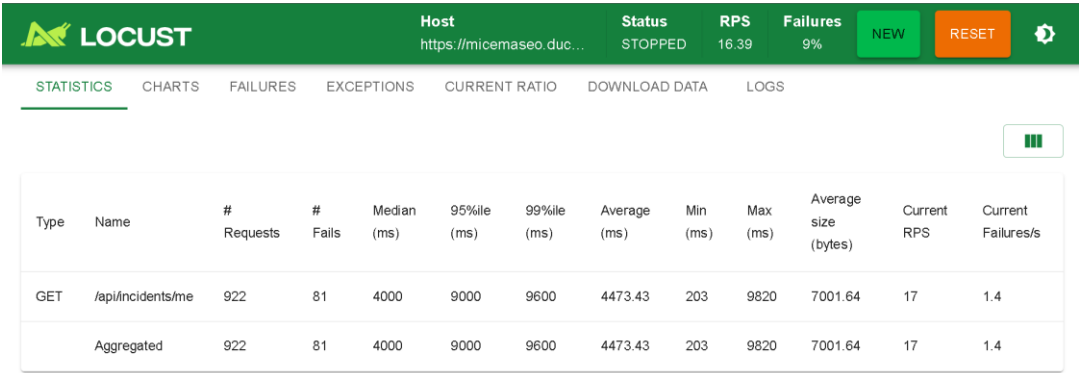
Endpoint	Usuarios	Media (ms)	P95 (ms)	P99 (ms)	Peticiones fallidas
GET /api/incidents/me	149 (estable)	540	1200	1600	0/922 (0%)
GET /api/incidents/me	350 (Saturación)	2340	5200	5600	0/922 (0%)
GET /api/incidents/me	500 (colapso)	4473.43	9000	9600	81/922 (8.8%)

Nota: La Tabla 16 evidencia la distribución de los usuarios por fases establecidas donde destaca que el P95 (9000 ms) supera ampliamente el umbral de 5 segundos, y que el 8.8% de las peticiones no completaron su ejecución, indicando saturación del servicio de consulta de incidentes.

La Figura 33 presenta el panel de estadísticas de Locust durante la ejecución de la prueba.

Figura 33

Panel de estadísticas de Locust para el escenario de 500 usuarios concurrentes



Nota. La figura 33 nos indica todas las peticiones que han sido fallidas y el RPS de 17 peticiones por segundo donde se evidencia que bajo cargas altas el sistema se degrada.

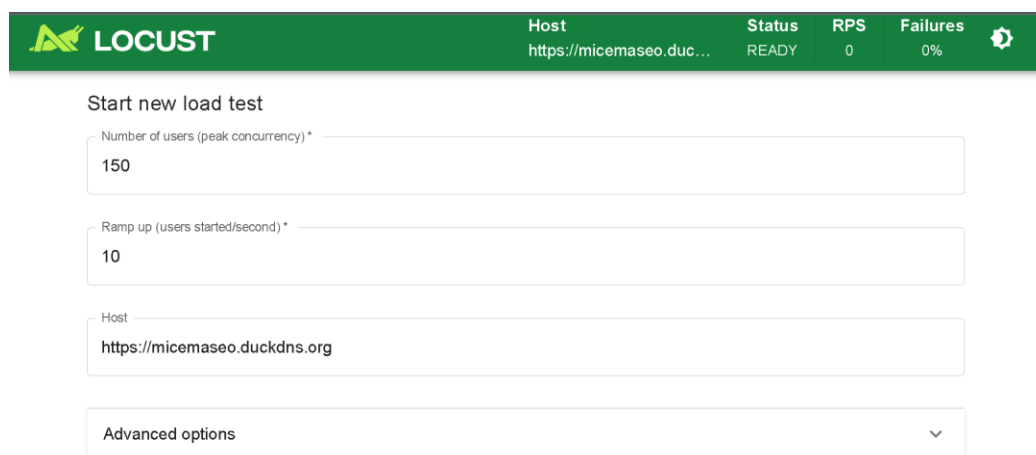
Estos resultados son una clara señal de la degradación del sistema con 500 usuarios en concurrencia donde el p95 llego a responder a los 9000 ms y P99 a los 9600 ms, superando significativamente el umbral de 5 segundos establecido el en RNF-01. Mientras que con 149 usuarios en concurrencia con p95 llego a responder con 1200 ms, manteniéndose en un estado estable bajo el umbral de los 5 segundos.

Escenario 2: 150 usuarios concurrentes (Supervisores)

Para evaluar el comportamiento bajo carga de supervisores, se llevó a cabo una segunda prueba con 150 usuarios concurrentes. La Figura 34 muestra la configuración utilizada para esta prueba.

Figura 34

Configuración de Locust para el escenario de 150 usuarios concurrentes



The screenshot shows the Locust web interface. At the top, there is a green header with the Locust logo and a status bar. The status bar includes the following information: Host: https://micemaseo.duc..., Status: READY, RPS: 0, Failures: 0%, and a refresh icon. Below the header, the main content area is titled "Start new load test". It contains four input fields: "Number of users (peak concurrency) *" with the value 150, "Ramp up (users started/second) *" with the value 10, "Host" with the value https://micemaseo.duckdns.org, and "Advanced options" with a dropdown arrow.

Nota. La Figura 34 muestra ramp-up de 10 usuarios/s para simular acceso concentrado de 150 supervisores al panel de gestión.

La Tabla 17 presenta los resultados obtenidos en este escenario.

Tabla 17

Resultados de pruebas de rendimiento con 150 usuarios concurrentes

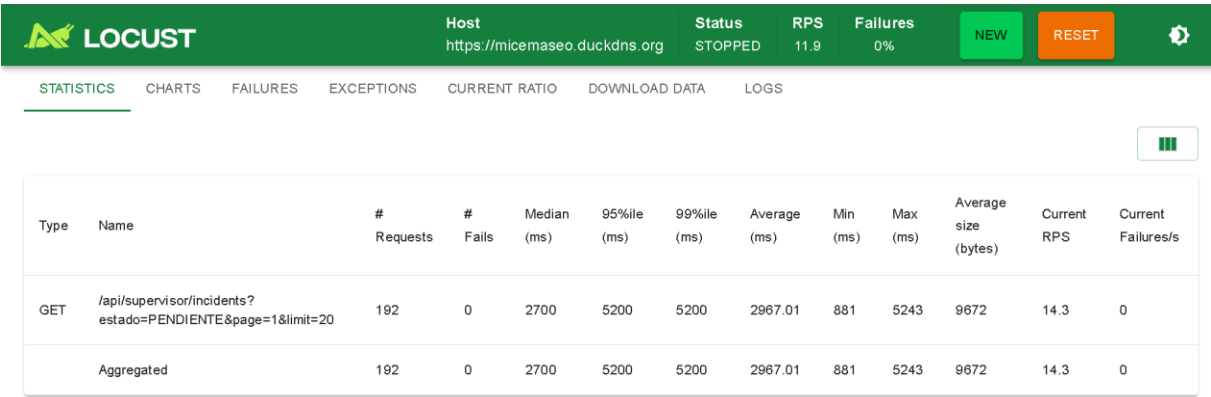
Endpoint	Usuarios	Media (ms)	P95 (ms)	P99 (ms)	Peticiones fallidas
GET /api/supervisor/incidents	70 (estable)	760	1300	1700	0/192 (0%)
GET /api/supervisor/incidents	109 (estable)	1560	2600	3000	0/192 (0%)
GET /api/supervisor/incidents	150 (Saturación)	2967.01	5200	5200	0/192 (0%)

Nota. La Tabla 17 muestra que no hay peticiones fallidas en las tres fases ejecutadas de forma consecutiva, aunque el P95 (5200 ms) supera en 200 ms el umbral aceptable, indicando que la cola de peticiones más lentas degrada el percentil, por lo tanto, para mantenerse bajo el umbral aceptable se tiene con P95 (2600ms).

La Figura 35 ilustra el panel de estadísticas de Locust durante la ejecución del test.

Figura 35

Panel de estadísticas de Locust para el escenario de 150 usuarios concurrentes



Nota. La figura 35 muestra la estabilidad del sistema, aunque se especifique un tiempo máximo de espera para algunas solicitudes individualmente que pudieran resultar demoradas notablemente.

El estudio demuestra que, cuando existen 150 usuarios conectados al punto de control, no aparecen errores de solicitud; por lo tanto, esto prueba la estabilidad. Sin embargo, el P95 de 5200 ms supera ligeramente el umbral de 5 segundos establecido en RNF-01, evidenciando la necesidad de optimizar el endpoint para consultas con filtros mediante índices y estrategias de caching.

Evaluación de usabilidad

Definición de la muestra

Para realizar una evaluación de usabilidad se contó con una muestra no probabilística intencional compuesta por 50 participantes, seleccionados por conveniencia y disponibilidad en un sector del Distrito Metropolitano de Quito. Esta decisión fue posible debido a que el estudio no pretende establecer resultados estadísticamente generales, sino analizar la percepción de usabilidad del prototipo, y según lo indicado por los expertos, es relevante el tipo de perfiles de usuario presentes en los resultados de la evaluación SUS, que afecta al 85%, de acuerdo con Nielsen y Landauer (1993), y que a partir de un mínimo de 10 a 20 participantes, disminuye la probabilidad de detectar errores y anomalías, como indica Faulkner (2003).

Los 50 participantes se organizaron en dos grupos de acuerdo con su función en el sistema:

Tabla 18

Participantes en la evaluación de usabilidad

Perfil de usuario	Participantes	Porcentaje
Ciudadanos	35	70%
Supervisores y administradores	15	30%
Total	50	100%

Nota. La Tabla 18 detalla la composición de la muestra según el perfil de usuario, con 35 ciudadanos y 15 supervisores.

Protocolo de consentimiento informado

La información recolectada mediante consentimiento fue recabada utilizando un dispositivo con la aplicación móvil MIC-EMASEO, conforme a las normativas del Decreto Legislativo Nro. 1866 del Ecuador. La Guía EMPASIC proporcionó la guía manual para aplicar el procedimiento, y los aspectos esenciales de dicha guía aparecen en la Tabla 19.

Tabla 19

Elementos del consentimiento informado

Elemento	Descripción
Responsable	Estudiantes ESPE y EMASEO EP
Datos recopilados	Nombre, cédula, correo, GPS e imágenes
Finalidad	Gestión de cuenta y mejora del modelo de detección
Base legal	Art. 7 y 22 LOPDP
Conservación	Cuenta activa; anonimización en 30 días
Derechos	ARCOP (Acceso, Rectificación, Cancelación, Oposición, Portabilidad)
Consentimiento	Pantalla completa con opciones “Acepto” o “No acepto”

Nota. Como se indica en la Tabla 19, se estableció que el mecanismo digital implica una determinación clara, directa y libre de confusión; esto se demuestra mediante la formación de la cuenta del usuario.

Procedimiento de evaluación

EL proceso fue efectuado durante el período final del prototipo (paso 4 de la metodología RAD) en Quito. Los sujetos participaron con el sistema por unos 15 minutos. Efectuaron la realización de las tareas más importantes en función de su rol como usuario.

Tabla 20

Tareas asignadas por perfil de usuario durante la evaluación

Perfil	Tareas evaluadas
Ciudadano	Registro en la app, captura de foto de acumulación, envío del reporte, consulta del historial personal
Supervisor/Admin	Acceso al panel web, revisión de mapa de incidencias y filtrado por severidad

Nota. En la Tabla 20 se detallan las tareas que se asignaron a cada perfil de usuario durante la sesión de evaluación.

Estas tareas fueron elaboradas para abarcar las funcionalidades clave de cada rol, garantizando así una representación adecuada del flujo de trabajo del sistema.

Resultados de la evaluación SUS

La Tabla 21 muestra los resultados de la evaluación de usabilidad realizada con 50 participantes, desglosados según el perfil de usuario.

Tabla 21

Resultados de la evaluación SUS por perfil de usuario

Perfil	N	Media	Desviación estándar	Rango intercuartílico (IQR)
Ciudadanos	35	78.5	8.2	72.0 – 85.0
Supervisores y administradores	15	85.0	6.5	80.0 – 90.0
Promedio general	50	80.4	7.8	74.5 – 87.0

Nota: El promedio general de 80.4 supera el umbral de aceptabilidad (70). Los supervisores presentan la media más alta (85.0) con menor variabilidad ($DE=6.5$), mientras que los ciudadanos muestran opiniones más diversas ($DE=8.2$).

Como se pudo observar tanto para el perfil de ciudadanos como para supervisores y administradores superan el umbral de aceptable, lo cual indica que el sistema se considera usable. Según la escala de Bangor et al. (2009), la puntuación de 80.4 se cae en el rango de “Bueno”.

Evaluación de la experiencia de usuario

Por otro lado, en la evaluación cuantitativa mediante el cuestionario SUS, también se realizó una evaluación cualitativa a través de una entrevista semi-estructurada con un funcionario de TI de EMASEO EP, quien utilizó los dos interfaces del prototipo: la versión móvil y la pantalla principal. Posteriormente, este individuo dio su opinión respecto a la funcionalidad del sistema.

Figura 36

Entrevista con el encargado operativo de EMASEO EP



Nota. La Figura 36 muestra la sesión de entrevista realizada con el encargado operativo de EMASEO EP como parte del proceso de validación cualitativa del sistema.

Criterios y resultados de la evaluación

Durante la entrevista, se analizaron tres dimensiones principales. La Tabla 22 resume los criterios considerados y los hallazgos más relevantes obtenidos.

Tabla 22

Criterios y resultados de la evaluación de experiencia de usuario

Dimensión	Criterio evaluado	Hallazgo principal
Interacción con el sistema	Fluidez del flujo de reporte, intuición de interfaces y claridad visual	La interfaz resultó intuitiva; el flujo desde la captura hasta la visualización en el panel web fue valorado como ágil y sin fricciones
Información geoespacial y estadística	Mapa de calor, distribución por sectores y estadísticas agregadas	El mapa de calor permite identificar patrones de acumulación y priorizar zonas con mayor concentración de incidencias; las estadísticas por sector facilitan la planificación operativa
Niveles de severidad	Clasificaciones BAJO, MEDIO, ALTO, CRÍTICO representadas con códigos de color	Los códigos de color en el mapa interactivo facilitan la identificación rápida de incidencias que requieren atención inmediata

Nota. La Tabla 22 sintetiza los criterios evaluados durante la entrevista y los principales resultados obtenidos a partir de la percepción del encargado operativo de EMASEO EP.

El mapa de calor se presentó al responsable operativo como una herramienta visual para monitorear la demanda en incidentes.

Validación del modelo de Machine Learning

Métricas de rendimiento obtenidas

El modelo RT-DETR-L v2 fue evaluado sobre 706 imágenes mediante un muestreo aleatorio estratificado del conjunto de validación semilla 42. Las métricas obtenidas se presentan en la Tabla 23.

Tabla 23

Evaluación final del modelo RT-DETR-L v2

Métrica	Validación	Test
mAP@50	0,8802	0,8931
mAP@50:95	0,6124	0,6164
Precisión	0,8934	0,8676
Recall	0,8571	0,8453
F1-Score	0,8749	0,8563

Nota. La Tabla 23 nos señala una comparación del rendimiento del modelo entre la validación y test. EL indicador mAP@50 muestra una mejora ligera en test de 0.8931 frente a 0.8802, mientras que la precisión y recall mantiene estabilidad mostrando la capacidad del modelo para la generalización.

Tabla 24

Matriz de confusión numérica — RT-DETR-L v2 sobre test (706 imágenes, 1.458 instancias)

	Pred: Acumulación	Pred: Fondo	Total real
Real: Acumulación	VP = 1.356	FN = 102	1.458

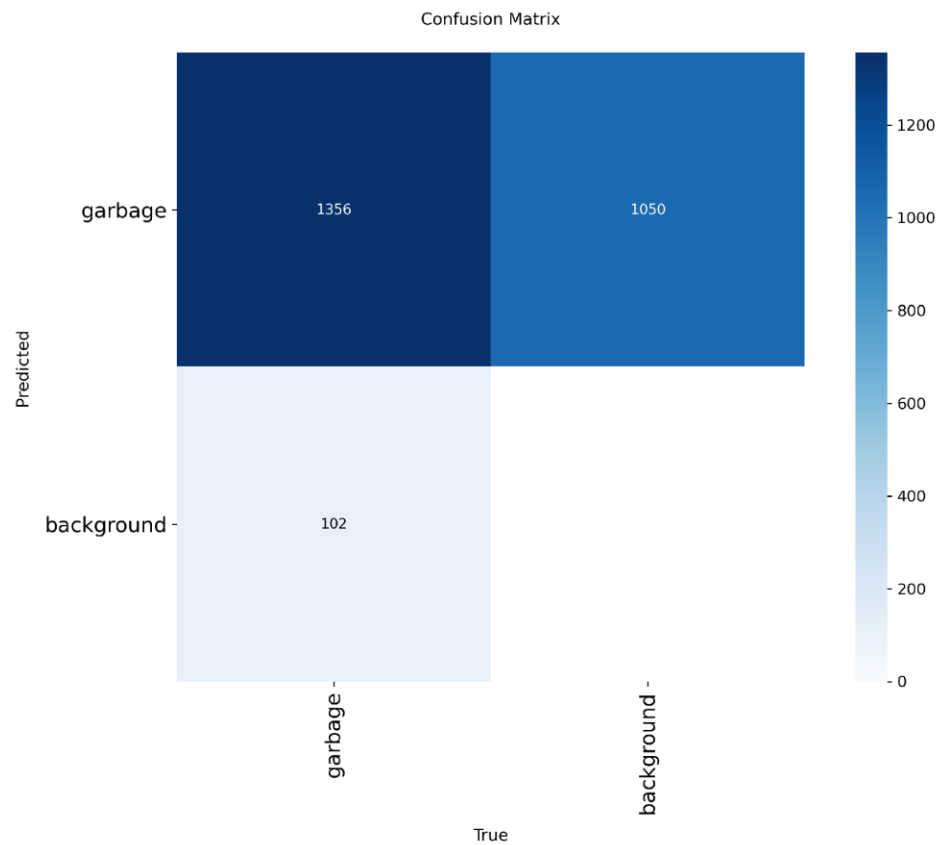
Real: Fondo	FP = 1.050	VN = 0	1.050
Total predicho	2.406	102	2.508

Nota. La Tabla 24 muestra el desempeño del detector, con 1.356 aciertos (VP) y 1.050 falsos positivos (FP). Los VN=0 son característicos de detectores monoetiqueta.

Las Figuras 37 y 38 muestran la representación visual de la matriz de confusión.

Figura 37

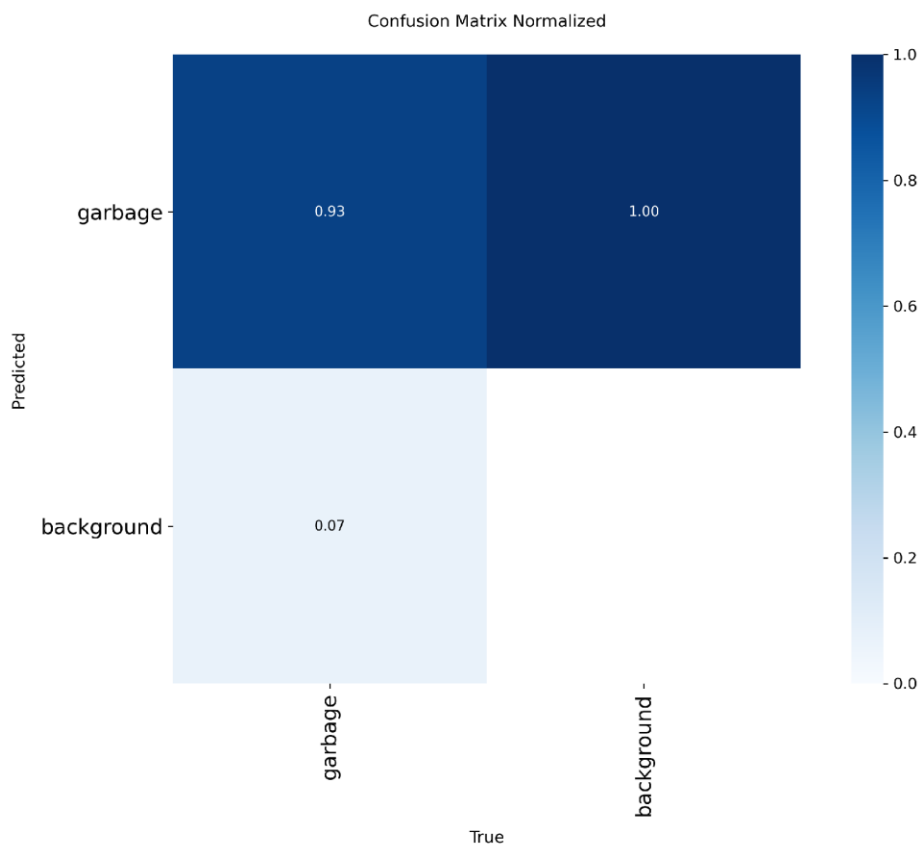
Matriz de confusión numérica del modelo RT-DETR-L v2



Nota. La Figura 37 visualiza los valores de la matriz de confusión: 1.356 VP, 102 FN y 1.050 FP, permitiendo identificar el sesgo del modelo hacia falsos positivos.

Figura 38

Matriz de confusión normalizada por clase real del modelo RT-DETR-L v2



Nota. La Figura 38 muestra que el 93% de las instancias reales de basura fueron detectadas correctamente, mientras que el 7% restante corresponde a falsos negativos.

Análisis de desempeño del modelo

Durante las pruebas, se identificaron tres aspectos fundamentales:

- **Discriminación semántica:** Sensibilidad hacia objetos que se asemejan a residuos (mochilas, bolsas).
- **Condiciones adversas:** Se verificó el bajo rendimiento en condiciones con poca iluminación y ruido de fondo.

- **Validación híbrida:** Las detecciones con confianza entre 0.30 y 0.62 son enviadas a una revisión manual realizando una combinando entre la automatización y supervisión humana.

Conclusiones y Recomendaciones

Conclusiones

1. En el análisis de los fundamentos tecnológicos y conceptuales enfocados en la gestión de residuos sólidos urbanos, la participación ciudadana y sobre la aplicación de Machine Learning al análisis de imágenes, la cual fue realizada mediante la revisión sistemática de la literatura, permitió identificar soluciones existentes, pero no integran de manera simultánea dichos enfoques. Esa brecha fue la que justificó desarrollar una solución integral como la que se propone en este trabajo.
2. La lógica del prototipo se formuló sobre una arquitectura de microservicios en Node.js y Express, con tres servicios de negocio independientes como son la autenticación, usuarios e imágenes/incidencias comunicados mediante un API Gateway con autenticación JWT y control de acceso basado en roles (RBAC), a los que se suma un cuarto servicio desplegado de forma independiente para el pipeline de Machine Learning (RT-DETR-L v2, CLIP y MiDaS_small), que procesa las imágenes de forma asíncrona a través de Redis y Celery y clasifica cada acumulación en uno de cuatro niveles de severidad: BAJO, MEDIO, ALTO y CRÍTICO.
3. Para la ciudadanía se implementó una aplicación móvil en React Native que permite registrar incidencias de acumulación de residuos mediante fotografías con geolocalización automática y almacenamiento offline. Al superar las pruebas funcionales RF-01 a RF-06, la aplicación confirmó que los ciudadanos pueden reportar incidentes de forma sencilla y eficiente, con historial de reportes y notificaciones de estado.
4. Para los supervisores operativos se diseñó un panel web en React orientado a la visualización geoespacial de incidencias, el filtrado por severidad y estado, y la

administración de usuarios. Las pruebas funcionales RF-14 a RF-16 confirmaron que el panel entrega las herramientas necesarias para el control, seguimiento y análisis de la recolección de residuos, y que facilita decisiones operativas basadas en información actualizada y georreferenciada.

5. En términos de eficacia y eficiencia se realiza las pruebas a los requisitos funcionales donde se midió el rendimiento y su usabilidad. El sistema ejecutó los 16 requerimientos funcionales donde el modelo RT-DETR-L v2 alcanzó un mAP@50 de 0.8931 y la evaluación SUS con los 50 participantes mostro un promedio de puntuación del 80.4 posicionando en el rango de “Bueno”. Mientras que, el rendimiento bajo la concurrencia de 150 usuarios el sistema se mantuvo estable con 0% de fallos, aunque con un P95 de 5200 ms, el cual esta ligeramente por encima de los 5 segundos; en un escenario con 500 usuarios concurrentes se observó un P95 9000 ms y 8.8% de solicitudes fallidas, lo cual nos permite conocer las limitaciones de escalabilidad en las consultas bajo una alta demanda. Dichos resultados nos muestran que el sistema sirve como una herramienta en el apoyo operativo para EMASEO EP, pero también salen a flote limitaciones y necesidades del sistema que se deben cubrir ante un despliegue a producción.
6. La efectividad y el desempeño del prototipo se examinaron mediante pruebas funcionales, de rendimiento y de usabilidad. El sistema cumplió los 16 requerimientos funcionales definidos, el modelo RT-DETR-L v2 alcanzó un mAP@50 de 0.8931 y la evaluación SUS con 50 participantes arrojó una puntuación promedio de 80.4 (rango "Bueno"). Por otro lado, se evidencio que por debajo de los 150 usuarios y múltiples peticiones el sistema es estable por el contrario el sistema empieza a degradarse.

Recomendaciones

1. Escalabilidad del sistema: Se recomienda Implementar la infraestructura de EMASEO EP al sistema MIC – EMASEO para resolver los problemas de concurrencia y disponibilidad.
2. Realización de validación humana: Se recomienda tener varias revisiones periódicas en cuanto a las detecciones que caen en el rango de ambigüedad o incertidumbre semántica CLIP 0.2 a 0.5 con el objetivo de verificar errores y así retroalimentar al modelo.
3. Extender la evaluación de usabilidad a una muestra más amplia. Una vez que el sistema entre en etapa de piloto institucional, sería valioso aplicar el cuestionario SUS en distintos sectores del DMQ, distribuidos geográficamente.

Líneas de trabajo futuro

1. **Optimización de rutas de recolección:** un módulo que genere rutas dinámicas para los operarios de barrido, apoyado en algoritmos de optimización combinatoria o en aprendizaje por refuerzo.
2. **Análisis predictivo:** Usar modelos que hagan uso del historial para retroalimentarse y mejorar su capacidad de clasificación y estimación en la detección de residuos sólidos.
3. **IoT:** Usar dispositivos IoT para complementar el ecosistema de reporte y que cámaras, sensores brinden información valiosa.

Referencias

- Abahussain, A. A. M., Nasr, F. A., bin Jumah, A., Saravanan, P., Kumar, N. S., Al-Zharani, M., Guyanathan, L., Sasikumar, G., Alsalamah, S. A., Qurtam, A. A., Senthilkumar, J., & Tamizdhurai, P. (2025). Toxic threats from plastic waste: human health impacts, challenges, and policy solutions. *RSC Advances*, 15, 40761–40788. <https://doi.org/10.1039/d5ra05845g>
- Addas, A. (2024). Smart waste management platforms integrating IoT sensors and Big Data analytics for urban sustainability. *Sustainable Cities and Society*, 102, 105234. <https://doi.org/10.1016/j.scs.2023.105234>
- Chollet Ramampandra, E., Scheidegger, A., Wydler, J., & Schuwirth, N. (2023). A comparison of machine learning and statistical species distribution models: Quantifying overfitting supports model interpretation. *Ecological Modelling*, 481, 110353. <https://doi.org/10.1016/j.ecolmodel.2023.110353>
- Comisión Nacional para el Uso Eficiente de la Energía. (2018). Guía para la gestión integral de residuos sólidos municipales. Gobierno de México.
- Empresa Pública Metropolitana de Aseo (EMASEO EP). (2022). Reglamento orgánico por procesos de EMASEO EP. Municipio del Distrito Metropolitano de Quito.
- Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(3), 379–383.
- Hatamleh, R. I., Jamhawi, M. M., Al-Kofahi, S. D., & Hijazi, H. (2020). The Use of a GIS System as a Decision Support Tool for Municipal Solid Waste Management Planning: The Case Study of Al Nuzha District, Irbid, Jordan. In *Procedia Manufacturing*, 44, 118–125. <https://doi.org/10.1016/j.promfg.2020.02.213>

- Hussain, K. M. (2025). Revolutionizing Route Optimization Systems with Artificial Intelligence for a Smarter, Sustainable Logistics Ecosystem. *International Journal of Computer Science and Mobile Computing*, 14(2), 66–68. <https://doi.org/10.47760/ijcsmc.2025.v14i02.008>
- International Organization for Standardization. (2018). ISO 9241-11:2018: Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. ISO.
- Kaza, S., Yao, L., Bhada-Tata, P., & Van Woerden, F. (2018). What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. Urban Development Series. Washington, DC: World Bank. <https://doi.org/10.1596/978-1-4648-1329-0>
- Lakhout, A. (2025). Revolutionizing urban solid waste management with AI and IoT: A review of smart solutions for waste collection, sorting, and recycling. *Results in Engineering*, 25, 104018. <https://doi.org/10.1016/j.rineng.2025.104018>
- Li, C., et al. (2023). YOLOv5-OCDS: An Improved Garbage Detection Model Based on YOLOv5. *Electronics*, 12(16), 3403. <https://doi.org/10.3390/electronics12163403>
- Longo, E., Sahin, F. A., Redondi, A. E. C., Bolzan, P., Bianchini, M., & Malfei, S. (2021). Take the trash out... to the edge. Creating a Smart Waste Bin based on 5G Multi-access Edge Computing. In *GoodIT '21: Proceedings of the Conference on Information Technology for Social Good* (pp. 55–60). ACM. <https://doi.org/10.1145/3462203.3475921>
- Maciuliene, M., & Skarauskiene, A. (2020). Building the capacities of civic tech communities through digital data analytics. *Journal of Innovation & Knowledge*, 5(4), 244–250. <https://doi.org/10.1016/j.jik.2019.11.005>
- Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykw, M. A., Kwasigroch, A., & Majek, K. (2021). Deep learning-based waste detection in natural and urban environments: Opportunities and challenges. *Waste Management*, 133, 36–54. <https://doi.org/10.1016/j.wasman.2021.07.016>

- Mankotia, S., Conte De Leon, D., & Jamil, H. M. (2026). ReactSmart: ML-Driven Adaptation for Scalable React-Based Web Application Performance. *IEEE Access*, 14, 1–18.
<https://doi.org/10.1109/ACCESS.2026.3650873>
- Nielsen, J. (2012). *Mobile Usability*. New Riders.
- Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (pp. 206–213). ACM.
- Pardini, K., Rodrigues, J. J. P. C., Kozlov, S. A., Kumar, N., & Furtado, V. (2020). IoT-based solid waste management solutions: A survey. *Journal of Sensor and Actuator Networks*, 8(1), 5.
<https://doi.org/10.3390/jsan8010005>
- Proença, P. F., & Simões, P. (2020). TACO: Trash Annotations in Context for Litter Detection. *arXiv preprint arXiv:2003.06975*.
- Radziszewski, K., Anacka, H., Obracht-Prondzyńska, H., Tomczak, D., Wereszko, K., & Weichbroth, P. (2021). Greencoin: prototype of a mobile application facilitating and evidencing pro-environmental behavior of citizens. In *Procedia Computer Science*, 192, 2668–2677.
<https://doi.org/10.1016/j.procs.2021.09.036>
- Sadeghi-Niaraki, A., Choi, S. M., & Lee, J. (2020). A Volunteered Geographic Information-Based Environmental Decision Support System for Waste Management and Decision Making. *Sustainability*, 12(15), 6012. <https://doi.org/10.3390/su12156012>
- Sall, M., et al. (2025). Integrating Remote Sensing, GIS, and Citizen Science to Map Illegal Waste Dumping Susceptibility in Dakar, Senegal. *Sustainability*, 17(24), 11137.
<https://doi.org/10.3390/su172411137>

- Sánchez, O., et al. (2024). Comparative performance evaluation of waste detection models in urban environments. *Revista de Investigación, Desarrollo e Innovación*, 15(1).
<https://doi.org/10.24054/raaas.v15i1.2922>
- Schiff, K. J. (2025). Does collective citizen input impact government service provision? Evidence from SeeClickFix requests. *Public Administration Review*, 85(1), 32–45.
<https://doi.org/10.1111/puar.13747>
- Sharma, M., & Kumar, V. (2024). Deep learning architectures for waste detection and classification in smart urban environments: A systematic review. *IEEE Access*, 12, 78291–78317.
<https://doi.org/10.1109/ACCESS.2024.3408221>
- Sharma, V., Kalra, S., & Beniwal, R. (2026). Image authenticity through the implementation of advanced deep learning solutions in embedded systems and Internet of Things (IoT) applications. *Franklin Open*, 16, 100633. <https://doi.org/10.1016/j.fraope.2026.100633>
- Shen, Y., et al. (2024). Solid Waste Detection Using Enhanced YOLOv8s (Garbage Detection Model). *Mathematics*, 12(14), 2185. <https://doi.org/10.3390/math12142185>
- Sudrajat, F., Sardi, I. L., & Puspitasari, S. Y. (2025). UI/UX Design of Mobile-Based Environmental Reporting Application Using User-Centered Design Method. *IT Journal Research and Development*, 9(2), 80–94. <https://doi.org/10.25299/itjrd.2025.18632>
- Suryawanshi, S., et al. (2022). Clean City: A Simple Smart City App. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 10(1).
<https://doi.org/10.22214/ijraset.2022.40092>
- Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10781–10790).

- Wynnedeta, D. E., Suryatiningsih, & Telnoni, P. A. (2024). Scrum-Based Development of CLEIN Mobile Application and IoT Smart Waste Bins: Advancing Digital Solutions for Sustainable Environmental Management. *Procedia Computer Science*, 245, 943–952.
<https://doi.org/10.1016/j.procs.2024.10.130>
- Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16965–16974).
<https://doi.org/10.1109/CVPR52733.2024.01605>
- Zhou, P., & Gao, Y. (2024). Detecting prototype pollution for node.js: Vulnerability review and new fuzzing inputs. *Computers & Security*, 137, 103625. <https://doi.org/10.1016/j.cose.2023.103625>
- Zoumpoulis, P., Konstantinidis, F. K., Tsimiklis, G., & Amditis, A. (2024). Smart bins for enhanced resource recovery and sustainable urban waste practices in smart cities: A systematic literature review. *Cities*, 152, 105150. <https://doi.org/10.1016/j.cities.2024.105150>