



**Pontificia Universidad
Católica del Ecuador**
Seréis mis testigos

MANABÍ

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
SEDE MANABÍ**

CARRERA DE SOFTWARE

TRABAJO DE TITULACIÓN
DESARROLLO DE UN PROTOTIPO INTELIGENTE PARA LA GESTIÓN
DE PARQUEADEROS UTILIZANDO VISIÓN POR COMPUTADORA Y
CHATBOT

LÍNEA DE INVESTIGACIÓN
TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

SUBLÍNEA DE INVESTIGACIÓN
INGENIERÍA DE SOFTWARE, INNOVACIÓN Y
EMPRENDIMIENTO EN TIC

PREVIO AL TÍTULO DE
INGENIERO EN SOFTWARE

AUTORES
JUAN SEBASTIÁN DAZA MENÉNDEZ
LUIS ALDAIR TOALA ROSADO
MIGUEL ALEJANDRO ZEA PÁRRAGA

TUTOR
ING. ALEXANDER JOSÉ MACKENZIE RIVERO, PHD

PORTOVIEJO, ABRIL 2026

Certificación del Tutor

En mi calidad de tutor del trabajo de integración curricular, certifico haber revisado el presente manuscrito de investigación, el mismo que se ajusta a las normas vigentes de la Pontificia Universidad Católica del Ecuador Sede Manabí, cumpliendo la Normativa del Trabajo de Integración Curricular; en consecuencia, es apto para su presentación y sustentación.

En la ciudad de Portoviejo a los quince días del mes de marzo de dos mil veinte y seis.

Ing. Alexander José Mackenzie Rivero

CI: 0960641694

Tutor

2026

Acta de aprobación del Tribunal

El jurado examinador aprueba el presente trabajo de integración curricular en nombre de la Pontificia Universidad Católica del Ecuador, Sede Manabí.

En la ciudad de Portoviejo a los quince días del mes de marzo de dos mil veinte y seis.

Ing. Sonia Shriley Garcia Caicedo

CI: 1307757441

Lector Revisor 1

Ing. Victor Hugo Alonzo Palacios

CI: 1310948185

Lector Revisor 2

Ing. Alexander José Mackenzie Rivero

CI: 0960641694

Tutor

Declaración de Originalidad

Este manuscrito no contiene ningún tipo de material que ha sido aceptado para la obtención de un título universitario en otra institución, excepto en forma de información de soporte que ha sido debidamente citada en mi trabajo. Este trabajo es de total responsabilidad de los autores, quienes declara bajo juramento que ninguna sección de este trabajo de integración curricular infringe los derechos de autor de nadie.

En la ciudad de Portoviejo a los quince días del mes de marzo de dos mil veinte y seis.

Miguel Alejandro Zea

CI: 1350196356

Autor

Luis Aldair Toala

CI: 0803723766

Autor

Juan Sebastian Daza

CI: 1315354306

Autor

Declaración de Derechos de Autores

Autorizo a la Pontificia Universidad Católica del Ecuador a distribuir este manuscrito de investigación en medios físicos y electrónicos con el fin de promover la divulgación de mis resultados a la comunidad científica y a la sociedad en general. Adicionalmente autorizo el uso de los contenidos de esta investigación como bibliografía para fines académicos, por cualquier medio o procedimiento, citando como fuente de información al autor de este trabajo.

En la ciudad de Portoviejo a los quince días del mes de marzo de dos mil veinte y seis.

Miguel Alejandro Zea

CI: 1350196356

Autor

Luis Aldair Toala

CI: 0803723766

Autor

Juan Sebastian Daza

CI: 1315354306

Autor

2026

Dedicatoria

Este trabajo está dedicado con profundo cariño a mi familia, quienes han sido mi fortaleza en cada etapa de mi vida. Su amor incondicional, paciencia y consejos me acompañaron en los momentos más difíciles y celebraron conmigo cada logro.

Dedico también esta tesis a mi enamorada, por su apoyo constante, comprensión y motivación, aún en los días de cansancio y estrés. Su presencia fue un impulso para no rendirme.

A todos ellos, que han creído en mí, les entrego con gratitud este logro académico.

Miguel Alejandro Zea Párraga

Dedicatoria

A mis padres, por ser mi raíz y mi motor, y por enseñarme que el conocimiento es la única herramienta que nadie puede quitarme.

A mi pareja, por ser mi puerto seguro, por su paciencia en los días difíciles y por celebrar cada logro como propio.

Finalmente, a mí mismo, por la disciplina, la resiliencia y la constancia para no rendirme cuando el camino se volvió cuesta arriba.

Juan Sebastián Daza Menéndez

Dedicatoria

A Dios, por la luz y la fortaleza que han guiado cada paso de este camino.

A mis padres, ejemplo de trabajo honrado y constancia, quienes, aun sin haber podido culminar sus estudios, construyeron con esfuerzo una vida digna y nos brindaron oportunidades, enseñándome a no rendirme y a mantener la fe presente.

A mi familia, por su compañía, apoyo y confianza, y por ser inspiración para que mis hermanos y primos continúen formándose y avanzando.

A los docentes y profesionales que, con su guía y experiencia, aportaron a mi formación.

Finalmente, a mí mismo y a la memoria de mi mascota Kun, por su compañía silenciosa y por recordarme que siempre se puede seguir.

Luis Aldair Toala Rosado

Agradecimientos

Agradecemos primeramente a Dios por concedernos salud, sabiduría y la oportunidad de culminar esta etapa importante de nuestra formación académica.

Expresamos nuestro profundo agradecimiento a nuestras familias por su apoyo emocional, moral y económico, así como por acompañarnos durante todo este proceso con palabras de ánimo y confianza.

Agradecemos a nuestras parejas por su compañía, paciencia y aliento permanente, los cuales fueron fundamentales para continuar avanzando en el desarrollo de este trabajo.

Expresamos nuestro sincero agradecimiento a nuestro tutor de tesis, por su orientación, acompañamiento académico y valiosos aportes, los cuales fueron determinantes para el correcto desarrollo y culminación de la presente investigación.

A nuestros compañeros de tesis, por su compromiso, seriedad y trabajo responsable. Cada una de sus participaciones fue indispensable para alcanzar los objetivos propuestos.

Agradecemos a la **Universidad Católica del Ecuador, Sede Manabí**, por la formación académica recibida y el apoyo institucional brindado para el desarrollo de este trabajo de titulación.

Resumen

El presente trabajo de titulación desarrolló un prototipo inteligente para la gestión de parqueaderos orientado a negocios de Manta y Portoviejo que requieren optimizar el control de sus espacios. A pesar de los avances globales en sistemas inteligentes, la mayoría de las soluciones existentes dependen de sensores físicos costosos o sistemas cerrados de monitoreo, lo que genera una brecha económica y técnica para las PYMES locales que no pueden costear infraestructuras invasivas. Para resolver este vacío, el sistema integra visión por computadora (YOLOv8) y un chatbot de mensajería automatizada. La investigación se fundamentó en un enfoque cuantitativo, de tipo aplicada y diseño experimental de desarrollo tecnológico. Los resultados demostraron una precisión técnica del 95 % en la identificación de ocupación en tiempo real. En las pruebas de campo, el sistema logró una reducción del 45 % en el tiempo promedio de búsqueda de estacionamiento, bajando de un intervalo de 5-12 minutos a solo 2-4 minutos por usuario. Asimismo, se evidenció una alta aceptación del prototipo, con un 80 % de los participantes valorando positivamente la facilidad de uso del chatbot. El proyecto concluyó que la integración de inteligencia artificial sobre infraestructura de CCTV existente constituye una alternativa viable y accesible que reduce la inversión inicial en un 75 % frente a métodos tradicionales, impulsando la transformación digital en el entorno empresarial ecuatoriano.

Palabras clave: chatbot, estacionamientos, inteligencia artificial, visión por computadora, prototipo

Abstract

This undergraduate thesis developed an intelligent prototype for parking management aimed at businesses in Manta and Portoviejo that require optimized control of their parking spaces. Despite global advances in intelligent systems, most existing solutions rely on costly physical sensors or closed monitoring systems, creating economic and technical barriers for local SMEs that cannot afford invasive infrastructures. To address this gap, the system integrates computer vision (YOLOv8) and an automated messaging chatbot. The research was grounded in a quantitative approach, classified as applied research, with an experimental design focused on technological development. The results demonstrated a technical accuracy of 95 % in real-time occupancy detection. In field tests, the system achieved a 45 % reduction in the average time required to find parking, decreasing from a range of 5–12 minutes to only 2–4 minutes per user. Additionally, high user acceptance of the prototype was observed, with 80 % of participants positively evaluating the chatbot's ease of use. The project concluded that integrating artificial intelligence into existing CCTV infrastructure constitutes a viable and accessible alternative, reducing initial investment by 75 % compared to traditional methods, thereby promoting digital transformation in the Ecuadorian business environment.

Keywords: chatbot, parking systems, artificial intelligence, computer vision, prototype

Tabla de Contenido

1. Introducción	1
Antecedentes	2
Definición del Problema	4
Preguntas de investigación	4
Hipótesis de investigación	5
Objetivos de investigación	6
Objetivo general	6
Objetivos específicos	6
2. Método	7
Enfoque Metodológico	7
Orientación al Desarrollo de Producto (MVP)	10
Rigor en la Ingeniería de Software y Stack Tecnológico	11
Estructura del Proyecto y Reproducibilidad Técnica	13
Validación de la Hipótesis Tecnológica	14
Estudio de Viabilidad del Emprendimiento Tecnológico	15
Análisis de Mercado y Segmentación	15
Modelo de Negocio: Software as a Service (SaaS)	15

Costo e Impacto	16
Marco Legal y Privacidad	17
Diseño de Investigación	17
Enfoque Cuantitativo	18
Tipo, Alcance y Diseño	18
Metodología de Desarrollo: Rigor e Iteración	18
Población y Muestra	19
Técnicas e Instrumentos	19
Evaluación Técnica y Lógica de Ocupación	19
Evaluación del Usuario	19
Procedimiento	20
Fase I: Diagnóstico del Entorno y Análisis de la Problemática	20
Fase II: Ingeniería de Requerimientos y Modelado	21
Fase III: Diseño de la Arquitectura e Integración Tecnológica	22
Fase IV: Implementación y Codificación del Prototipo	22
Fase V: Pruebas de Campo, Evaluación y Análisis de Impacto	23
Consideraciones Éticas	24
3. Resultados	25
Dimensión I: Resultados Técnicos y de Ingeniería	25
Análisis de Eficacia del Algoritmo de Visión Artificial (IA)	25
Matriz de Confusión y Precisión del Modelo	31

Desempeño Técnico y Métricas de Latencia	32
Robustez Estructural y Validación de Software (Caja Blanca)	35
Dimensión II: Resultados de Emprendimiento y Validación de Mercado	37
Percepción del Usuario y Usabilidad (Escala Likert)	37
Validación Funcional del MVP (Pruebas de Caja Negra)	39
Viabilidad Financiera e Impacto Comercial	44
Resumen de Hallazgos Clave	44
4. Discusión	45
5. Conclusiones	48
Referencias bibliográficas	54
Apéndice A: Resultados de la Encuesta de Usabilidad y Aceptación	55
Apéndice B: Evidencias de Gestión y Pruebas Técnicas Adicionales	60
Apéndice C: Artefactos de Pruebas Unitarias (Caja Blanca)	65

Lista de Figuras

1.	Diagrama del Flujo Metodológico	9
2.	Diagrama de Actividad del sistema	21
3.	Diagrama de Casos de Uso	21
4.	Diagrama de Arquitectura de Microservicios	22
5.	Entorno de validación real en Medical Plus	23
6.	Validación de detección en tiempo real	28
7.	Flujo lógico del procesamiento de imágenes	30
8.	Métricas de rendimiento computacional	33
9.	Diagrama de Clases del Sistema	35
10.	Resultado de prueba unitaria automatizada	36
11.	Reporte de Cobertura de Código	37
12.	Interfaz conversacional del Chatbot	39
13.	Sincronía de datos en Dashboard Administrativo	40
14.	Sincronía de datos en Interfaz de Chatbot	41
15.	Prueba de robustez ante entradas incorrectas	42
16.	Perfil de los encuestados	55
17.	Experiencia previa con sistemas digitales	56

18.	Dificultad percibida para encontrar espacios	56
19.	Valoración de la facilidad de uso del Chatbot	57
20.	Confianza en la información técnica	57
21.	Reducción del tiempo de búsqueda	58
22.	Adecuación ética del uso de cámaras e IA	58
23.	Intención de adopción del sistema	59
24.	Sugerencias de mejora y retroalimentación	59
25.	Landing Page de Parkintia	60
26.	Interfaz de Autenticación	61
27.	Configuración de ROIs en sitio	62
28.	Detección en tiempo real en sitio	62
29.	Tablero de gestión de tareas Scrum	63
30.	Logs de inicialización de YOLOv8s	63
31.	Estructura de directorios de la IA	63
32.	Vista general del Dashboard administrativo	64
33.	Código fuente de pruebas unitarias	65

Lista de Tablas

1.	Resumen del Diseño Metodológico de la Investigación	8
2.	Matriz de servicios, responsabilidades y endpoints del sistema	11
3.	Especificaciones Técnicas del Entorno de Desarrollo y Despliegue	14
4.	Comparativa de Costos: Sensores Físicos vs. Visión por Computadora	16
5.	Instrumentos de recolección de datos para la validación del MVP	20
6.	Matriz de Escenarios y Casos Borde en la Detección de Ocupación	27
7.	Matriz de Confusión: Evaluación de la precisión en la detección.	31
8.	Métricas de Rendimiento Computacional (Hardware Estándar).	33
9.	Niveles de satisfacción y usabilidad percibida (N=5).	38
10.	Matriz Extendida de Resultados: Pruebas de Caja Negra y Tolerancia a Fallos. . . .	43

Lista de Apéndices

- **Apéndice A:** Resultados de la Encuesta de Usabilidad y Aceptación (Escala Likert).
- **Apéndice B:** Evidencias de Gestión y Pruebas Técnicas Adicionales (Metodología Scrum, Logs de sistema y estructura de directorios).
- **Apéndice C:** Artefactos de Pruebas Unitarias de Caja Blanca (Código fuente del script de pruebas en Jest).

Introducción

Actualmente, la gestión de estacionamientos ha trascendido el simple trámite logístico para convertirse en un desafío de ingeniería urbana y servicio al cliente. En las zonas comerciales de Ecuador, el incremento del parque automotor ha superado la capacidad de las infraestructuras convencionales. Según datos oficiales de la Agencia Nacional de Tránsito (Agencia Nacional de Tránsito, 2023), el parque automotor en Ecuador experimentó un crecimiento promedio anual del 7.49 % durante la última década (2014-2023), alcanzando una cifra que supera los 3.06 millones de vehículos matriculados a nivel nacional (Agencia Nacional de Tránsito, 2023). Este volumen de crecimiento ininterrumpido ha generado problemas crónicos de disponibilidad de parqueo, pérdida de tiempo productivo y una experiencia insatisfactoria para el usuario. En las ciudades de **Manta y Portoviejo**, esta problemática se agudiza debido a la limitada presencia de procesos automatizados en el sector, lo que mantiene una gestión operativa mayoritariamente analógica. Las soluciones de estacionamiento inteligente (*Smart Parking*) basadas en visión artificial se presentan como la alternativa más eficiente para entornos urbanos densos, permitiendo una gestión dinámica sin las limitaciones de los sensores de proximidad tradicionales (Bhatia et al., 2021). Ante este escenario, el presente trabajo de titulación propone emplear la Inteligencia Artificial (IA) como un eje de transformación. Mediante el uso de visión por computadora, se plantea transitar de un uso pasivo de las cámaras de seguridad hacia un sistema inteligente y activo, capaz de apoyar la toma de decisiones y la gestión de eventos en tiempo real. La robustez de los algoritmos de aprendizaje profundo (*Deep Learning*) permite realizar detecciones precisas incluso en condiciones de iluminación variable o oclusiones parciales (Almaguer-Segura et al., 2023). Al fusionar esta tecnología con interfaces interactivas, se busca que el usuario interactúe con el sistema de manera natural; el uso de arquitecturas de recuperación aumentada en sistemas

conversacionales permite un soporte técnico y operativo altamente eficiente (Chen et al., 2025). Con esta base, se busca asegurar que la solución sea técnica y económicamente accesible para las pequeñas y medianas empresas locales. El proyecto se enmarca explícitamente en la modalidad de titulación **Emprendimientos relacionados con dispositivos tecnológicos**, la cual orienta el trabajo a la construcción de conocimiento mediante el desarrollo de un prototipo tecnológico con potencial de emprendimiento. En coherencia con este modelo, la propuesta incorpora criterios de *viabilidad de mercado*, *sostenibilidad financiera* y *seguridad*. El éxito de estos emprendimientos depende de la capacidad de integrar modelos de negocio escalables con arquitecturas de software modulares que soporten el crecimiento del servicio (Talavera et al., 2022). Asimismo, el trabajo se centra en el establecimiento de un **Prototipo Mínimo Viable (MVP)**. El objetivo es validar una solución técnica que responda a condiciones reales del mercado y reduzca la dependencia de hardware especializado. Bajo la metodología de emprendimiento tecnológico, el enfoque en un MVP facilita la iteración rápida sobre las necesidades del usuario sin comprometer grandes inversiones iniciales (Ries, 2021). Esta propuesta constituye un aporte técnico al demostrar la viabilidad de microservicios de visión artificial procesados en hardware convencional, optimizando recursos mediante técnicas de inferencia ligera en el borde (*Edge Computing*) (Zhang & Wu, 2024).

Antecedentes

La evolución de los estacionamientos inteligentes ha avanzado en paralelo con las Tecnologías de la Información y Comunicación (TIC). A nivel global, los Sistemas Inteligentes de Transporte (ITS) utilizan el Internet de las Cosas (IoT) y la Visión por Computadora para el monitoreo urbano. El uso de arquitecturas de detección de objetos en tiempo real, específicamente la familia de modelos YOLO (*You Only Look Once*), ha revolucionado el sector al permitir precisiones superiores al 95 % con un gasto computacional optimizado (Jocher et al., 2023; Redmon et al., 2016). En este contexto, la presente investigación utiliza como referencia teórica datasets de es-

tándar mundial como **PKLot** y **CNRPark-EXT** para un proceso de **fine-tuning** sobre el modelo YOLOv8. Al entrenar el algoritmo con estos conjuntos —que contienen miles de imágenes en diversas condiciones— se logró una transferencia de conocimiento (*Transfer Learning*) que permite al modelo reconocer patrones de ocupación con alta precisión (P. Almeida et al., 2015). Estudios recientes confirman que el uso de pre-entrenamiento en datasets especializados reduce drásticamente el error de falso positivo en zonas de parqueo exteriores (Sultana & Hussain, 2022). La adaptación de esta tecnología a las ciudades de Manta y Portoviejo se realizó mediante un esquema de dos etapas: un **benchmark** técnico para validar la precisión en hardware de propósito general y una **validación de campo** donde el modelo fue calibrado para las condiciones lumínicas locales. La literatura sugiere que la calibración local es crítica, ya que factores como el ángulo de visión de las cámaras de CCTV existentes pueden afectar la métrica de IoU (*Intersection over Union*) del modelo (OpenCV Team, 2024; Wang & Zhou, 2023). En el ámbito latinoamericano, esta tecnología se encuentra en una etapa incipiente. Mientras ciudades como Medellín avanzan hacia modelos de *smart cities* apoyadas por la integración de Big Data (Bouskela et al., 2016), en las urbes de Manta y Portoviejo prevalece la dependencia de tickets físicos. La implementación de sistemas de bajo costo basados en infraestructura de CCTV existente representa la transición necesaria hacia la digitalización del sector en regiones en desarrollo (Kumar et al., 2021). No obstante, la alta penetración de dispositivos móviles ofrece un terreno fértil para interfaces de mensajería automatizada y *chatbots*. Las interfaces conversacionales modernas han demostrado ser el canal de menor fricción para la interacción con sistemas inteligentes, especialmente cuando se integran con plataformas como WhatsApp (Dale, 2023; Nguyen & Do, 2024). Esta transición requiere de una arquitectura de software bien definida que soporte la escalabilidad a largo plazo (Sommerville, 2011). Por ende, la ausencia de investigación aplicada regional justifica el desarrollo de un ecosistema digital escalable en **Manta y Portoviejo**.

Definición del Problema

La administración de parqueaderos en Manta y Portoviejo se encuentra ligada a métodos analógicos que generan problemáticas en tres dimensiones: operativa, técnica y económica. Un **diagnóstico de campo preliminar** en 10 establecimientos comerciales reveló que el **100 %** depende del control manual, y aunque el **90 %** cuenta con CCTV, ninguno utiliza dicha infraestructura para la gestión activa de plazas. Operativamente, la carencia de información en tiempo real obliga al conductor a realizar recorridos innecesarios; un conductor dedica entre **5 a 12 minutos** adicionales a la búsqueda de estacionamiento en horas pico. Técnicamente, las cámaras cumplen una función únicamente de registro pasivo, existiendo una brecha que la visión artificial busca resolver mediante el procesamiento de flujos de video asíncronos (P. R. L. d. Almeida et al., 2022; Vila et al., 2020). Económicamente, las soluciones internacionales basadas en sensores físicos resultan prohibitivas para las **PYMES** locales. El **80 %** de los administradores consultados considera "prohibitiva" la inversión en hardware especializado. En contraste, las soluciones basadas en software y visión computacional reducen el costo operativo al centralizar el procesamiento (Moussa & Yassine, 2023). Este proyecto propone mitigar el problema mediante un sistema inteligente y un *chatbot*, aprovechando la infraestructura existente conforme a la agenda digital nacional para cerrar la brecha tecnológica (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2024).

Preguntas de investigación

Pregunta general

¿Cómo puede el desarrollo de un prototipo de sistema inteligente basado en visión por computadora y chatbot optimizar la gestión de parqueaderos en negocios o empresas privadas de **Manta y Portoviejo** a partir de un estudio piloto?

Preguntas específicas

- ¿Cuáles son las principales limitaciones actuales en la gestión de parqueaderos dentro de negocios locales en Manta y Portoviejo?
- ¿Qué requerimientos técnicos, financieros y de seguridad son necesarios para el desarrollo de un MVP funcional?
- ¿De qué manera se puede integrar la visión por computadora con interfaces conversacionales para asegurar una respuesta eficiente al usuario?
- ¿Qué nivel de aceptación y viabilidad técnica presenta el prototipo dentro de un entorno de prueba controlado?

Hipótesis de investigación

- **H1:** El diagnóstico revelará que la dependencia de métodos analógicos genera brechas críticas en la disponibilidad de datos en tiempo real.
- **H2:** La definición de requerimientos basados en hardware accesible permite un MVP sostenible, reduciendo la inversión inicial en un **75 %** frente a sensores tradicionales.
- **H3:** La integración de microservicios asegura una respuesta con una latencia ≤ 2 **segundos**, cumpliendo con los estándares de experiencia de usuario (Nielsen, 2022; Vila et al., 2020).
- **H4:** El prototipo presentará una precisión $\geq 95 \%$ y un nivel de aceptación $\geq 4.0/5.0$ en la escala Likert (P. Almeida et al., 2015; Davis, 1989).

Objetivos de investigación

Objetivo general

Desarrollar un prototipo funcional de un sistema inteligente de monitoreo de parqueaderos, basado en visión por computadora y chatbot, como propuesta de emprendimiento tecnológico adaptable a empresas o negocios locales de Manta y Portoviejo mediante un estudio piloto de validación.

Objetivos específicos

- Diagnosticar las necesidades y limitaciones actuales en la gestión de parqueaderos locales para definir los requisitos del sistema.
- Determinar los requerimientos técnicos, de mercado, legales y financieros necesarios para la sostenibilidad del prototipo.
- Diseñar y desarrollar un MVP que integre algoritmos de visión artificial y una interfaz de chatbot para el monitoreo de plazas.
- Evaluar la funcionalidad técnica y el nivel de usabilidad del prototipo con una muestra por conveniencia.

Método

En este capítulo se detalla el camino metodológico que se siguió para desarrollar el prototipo inteligente de gestión de parqueaderos. Este trabajo no se abordó como un ejercicio aislado de programación, sino como una propuesta de emprendimiento tecnológico orientada a resolver problemáticas reales en **Manta y Portoviejo** mediante visión por computadora y el uso de chatbots. La estructura metodológica se adoptó para garantizar un desarrollo ordenado, asegurando que cada fase del proyecto sea técnica y académicamente auditable, y que la solución propuesta pueda ser replicada o validada por otros investigadores.

La investigación se centró en la creación de un Producto Mínimo Viable (MVP). El propósito de este método fue validar, tanto en el código como en la práctica, una alternativa tecnológica que ayude a los negocios locales a dejar atrás el control manual ineficiente. Así, el método no solo facilitó el desarrollo del software, sino que proporcionó las herramientas para medir su éxito comercial y su facilidad de uso en el entorno real mediante un estudio piloto controlado.

Enfoque Metodológico

Se adaptó un enfoque cuantitativo, de tipo aplicado, orientado al diseño, implementación y evaluación de un prototipo tecnológico. El propósito del estudio fue cerrar la brecha que suele existir entre los complejos algoritmos de Inteligencia Artificial y su utilidad en el día a día.

Para garantizar el rigor científico y la trazabilidad de la investigación, en la **Tabla 1** se presenta la síntesis de los componentes estratégicos que estructuran el presente trabajo.

Tabla 1. Resumen del Diseño Metodológico de la Investigación

Componente	Descripción
Enfoque	Cuantitativo: Basado en la recolección de métricas numéricas de rendimiento y usabilidad.
Tipo	Aplicado - Tecnológico: Orientado a la resolución de un problema operativo mediante software.
Alcance	Descriptivo-Explicativo: Describe el funcionamiento del sistema y explica su impacto en la gestión.
Diseño	No experimental, transversal y experimental de desarrollo tecnológico (prototipado).
Variables	Independiente: Sistema Parkintia (IA + Chatbot). Dependientes: Precisión de detección, latencia de respuesta y nivel de usabilidad.
Población y Muestra	Actores de la Clínica Medical Plus. Muestra de 5 participantes para validación de usuario y 500 eventos para validación técnica.
Instrumentos	Logs de sistema, escala Likert, guía de observación y matriz de confusión.
Análisis Estadístico	Descriptivo: Media, desviación estándar, percentiles (P95) y análisis de frecuencias.

Nota. Elaboración propia.

Flujo Metodológico: Desarrollo del Prototipo Parkintia

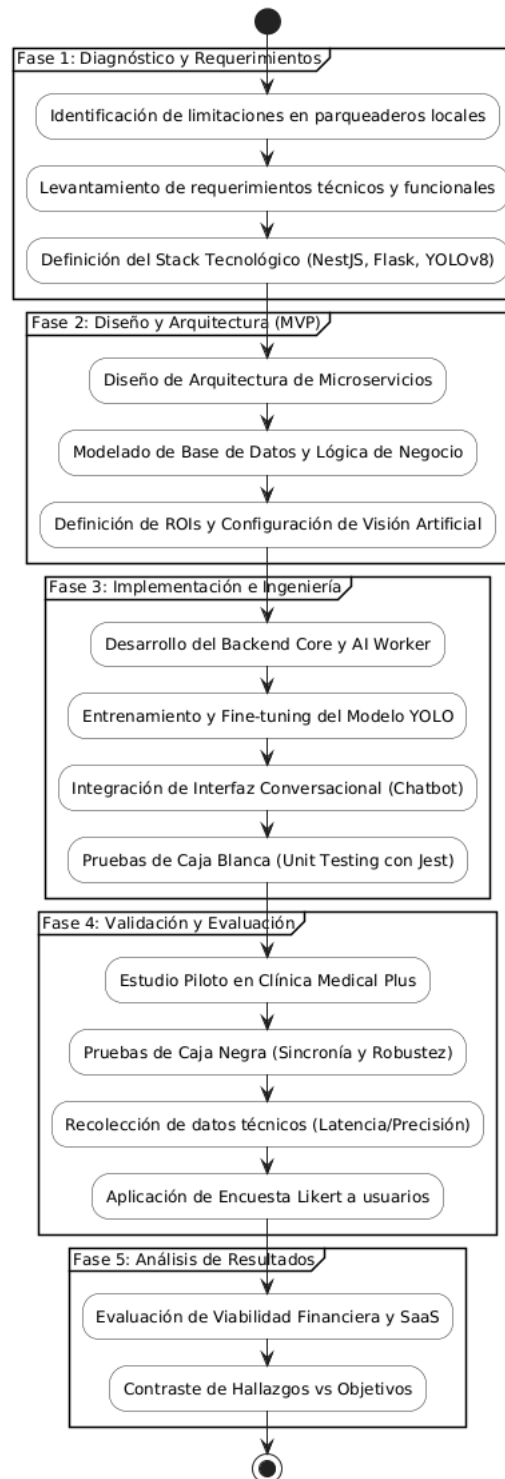


Figura 1. Diagrama del Flujo Metodológico: Fases del desarrollo e investigación aplicada.

Nota. Elaboración propia con base en el ciclo de vida de desarrollo de software y validación de campo.

Orientación al Desarrollo de Producto (MVP)

Se priorizó las funciones vitales del sistema: que detecte plazas con exactitud y que informe al usuario con rapidez. Para lograrlo, fue implementada una **arquitectura híbrida** que delimita con precisión la frontera entre servicios independientes y componentes internos, resolviendo la ambigüedad entre el procesamiento distribuido y la lógica centralizada. El uso de arquitecturas basadas en microservicios permite una escalabilidad horizontal eficiente, ideal para aplicaciones urbanas que requieren procesamiento de video en tiempo real (Do et al., 2025).

- **Microservicio Independiente (AI Worker):** Se desarrolló un servicio autónomo basado en Flask para el procesamiento de visión artificial. Este componente opera como un microservicio real al poseer un entorno de ejecución independiente en Python, lo que permite escalar el procesamiento de imágenes de forma aislada sin afectar la disponibilidad del resto del sistema (Fernandez-Ramos & Lopez-Nozal, 2025).
- **Monolito Modular (Backend Core):** El núcleo del sistema, desarrollado bajo NestJS, se estructuró mediante módulos lógicos (Usuario, Parqueadero, Chatbot) que residen en un mismo despliegue. Esta aproximación permite una gestión simplificada de la persistencia de datos mientras se mantiene la modularidad necesaria para futuras expansiones (NestJS, 2023).
- **Frontend Administrativo:** Interfaz desarrollada para la gestión y configuración del sistema por parte del personal autorizado.

Para clarificar la operatividad y el despliegue de estos componentes, se presenta la Tabla 2, detallando sus responsabilidades, puntos de acceso y esquema de escalabilidad.

Tabla 2. Matriz de servicios, responsabilidades y endpoints del sistema

Componente	Responsabilidad y Detalles Técnicos
AI Worker (Microservicio)	Procesamiento de inferencia YOLOv8 y analítica de video en tiempo real. Utiliza el endpoint POST /process-frame para recibir frames y retornar estados de ocupación.
Módulo de Parquero	Gestión de regiones de interés (ROIs) y lógica de estados de plazas en base de datos. Expone el endpoint GET /parking/status para consulta de disponibilidad.
Motor de Chatbot	Orquestación de mensajería y flujo conversacional. Gestionla comunicación asíncrona mediante el endpoint POST /whatsapp/webhook.
Gestión de Usuarios	Control de acceso, seguridad mediante JWT y administración de perfiles. Utiliza el endpoint POST /auth/login para validación de credenciales.

Nota. Elaboración propia con base en el diseño de la arquitectura y responsabilidades del prototipo.

Trabajar bajo la lógica de un MVP permite realizar iteraciones ágiles basadas en el “ciclo de retroalimentación técnica y de usuario”, un concepto fundamental en la ingeniería moderna para asegurar que el software realmente sirva a su propósito final (Sommerville, 2011). Este enfoque de desarrollo iterativo es fundamental para mitigar riesgos técnicos y financieros en etapas tempranas de innovación tecnológica (Ries, 2021).

Rigor en la Ingeniería de Software y Stack Tecnológico

Para que el sistema sea escalable y no se quede en un experimento de laboratorio, se aplicaron principios de ingeniería rigurosos y se seleccionó un conjunto de herramientas de alto rendimiento. Para garantizar la mantenibilidad y calidad del software, el desarrollo del núcleo del sistema

se estandarizó utilizando **TypeScript**. La adopción de este lenguaje proporcionó un tipado estático estricto, lo que redujo significativamente la densidad de defectos en tiempo de ejecución, controló la complejidad ciclomática de las funciones lógicas y facilitó la implementación de pruebas unitarias automatizadas para validar los caminos críticos del sistema (**Irazabal2024**).

Por su parte, el microservicio de Inteligencia Artificial fue implementado en **Python** para aprovechar su ecosistema nativo de cálculo matricial. Los despliegues de modelos de aprendizaje profundo en entornos de producción requieren una gestión eficiente de dependencias y aislamiento de procesos para garantizar la estabilidad operativa (Sari & Kurniawan, 2022). El proceso de desarrollo iterativo fue dividido en las siguientes fases metodológicas:

- **Fase Analítica:** Etapa orientada al análisis del contexto operativo de los parqueaderos locales en **Manta y Portoviejo**, con el propósito de identificar las problemáticas existentes y la viabilidad de la automatización.
- **Fase de Requerimientos:** Levantamiento y especificación de requisitos del sistema, definiendo el uso de **NestJS v11** y **PostgreSQL** con **TypeORM** para la persistencia de datos relacionales. La elección de NestJS responde a su arquitectura modular, la cual facilita la inyección de dependencias, la escalabilidad del sistema y la separación de capas lógicas (NestJS, 2023).
- **Fase de Arquitectura:** Definición de la conexión de los módulos mediante una comunicación asíncrona entre el servicio Python y el núcleo en NestJS. Para el despliegue del componente de visión artificial, se optó por el microframework **Flask**, dada su eficiencia y ligereza para exponer modelos de inteligencia artificial a través de APIs REST (Grinberg, 2024). Este diseño fomenta una separación de responsabilidades que optimiza el mantenimiento y el despliegue independiente de funciones críticas (Fowler, 2023).
- **Fase de Codificación:** Desarrollo del *backend* de inteligencia artificial, integrando **YOLOv8** y **OpenCV**. Se utilizaron modelos base validados previamente con datasets de refe-

rencia internacional como **PKLot** y **CNRPark-EXT** (P. Almeida et al., 2015), ajustando la lógica para el control de Regiones de Interés (ROIs) específicas del entorno.

- **Fase de Validación:** Ejecución de pruebas de caja blanca y caja negra, asegurando la cobertura del código mediante frameworks de testing (Myers et al., 2023), seguido de la validación del sistema a través de un estudio piloto con usuarios reales para medir métricas de desempeño y usabilidad.

Estructura del Proyecto y Reproducibilidad Técnica

Para garantizar la mantenibilidad a largo plazo y la reproducibilidad científica de los resultados obtenidos, el subsistema de detección (servicio Python) fue reestructurado siguiendo estándares de ingeniería de software profesional. Se implementó una arquitectura modular que desacopla los modelos de inferencia, la lógica de negocio y los artefactos de salida. Esta organización cumple con los principios de reproducibilidad exigidos en proyectos de investigación aplicada, permitiendo una clara trazabilidad entre el código ejecutado y los resultados presentados en hardware de propósito general (Wei & Sun, 2024).

La estructura final del proyecto se organizó bajo un esquema de directorios especializados: **/models** para la gestión de pesos de la red neuronal (.pt), permitiendo actualizaciones del modelo sin alterar el código fuente; **/outputs** y **/logs** para la auditoría de evidencias de inferencia y registros de ejecución; y un módulo de benchmarking denominado **measure-performance.py**, diseñado para validar la capacidad del hardware anfitrión antes del despliegue productivo (Martinez2024). Complementariamente, se incluyó documentación técnica detallada en un archivo **README-TECNICO.md**.

Para asegurar que el experimento pueda ser replicado íntegramente por terceros bajo las mismas condiciones computacionales, se documentan las especificaciones en la Tabla 3.

Tabla 3. Especificaciones Técnicas del Entorno de Desarrollo y Despliegue

Componente / Parámetro	Especificación Técnica
Sistema Operativo	Darwin 22.6.0 (Arquitectura x86-64)
Hardware Base	Procesador i386 (CPU), 8.0 GB RAM
Motor de IA y Lenguaje	Python, Ultralytics YOLOv8s (Small)
Bibliotecas de Visión	OpenCV
Framework y Backend	NestJS v11, TypeScript
Core	
Parámetros de Inferencia	Resolución de entrada: 640x640 píxeles, ejecución sobre CPU, sin aceleración GPU

Nota. Elaboración propia con base en los registros de auditoría del sistema y logs de ejecución.

Validación de la Hipótesis Tecnológica

El enfoque del proyecto consistió en demostrar que el software inteligente puede reducir la dependencia de hardware costoso bajo condiciones operativas normales. Se documentó cada decisión técnica, como el uso de *Transfer Learning* en modelos de visión artificial, para probar que la IA no solo puede igualar la eficiencia de los sensores físicos en tareas específicas, sino que facilita la transformación digital de las empresas de **Manta y Portoviejo** a una fracción del costo tradicional.

Para sustentar esta premisa y garantizar la reproducibilidad del modelo, se establecieron parámetros rigurosos en el proceso de *Transfer Learning* sobre la arquitectura base de YOLOv8s. El ajuste al contexto local se realizó congelando las capas iniciales de extracción de características (*backbone*) y reentrenando únicamente las capas finales de predicción (*head*). Se empleó un dataset local capturado en los estacionamientos de los negocios evaluados, el cual fue particionado en un 80 % para entrenamiento, 10 % para validación y 10 % para pruebas.

El entrenamiento se configuró con un *Learning Rate* (LR) inicial de 0.01 y se ejecutó durante 50 *epochs*, implementando técnicas de parada temprana (*early stopping*) para evitar el sobreajuste. El criterio principal para la selección del modelo final no solo priorizó la métrica de precisión media promedio (mAP), sino que buscó el equilibrio óptimo entre la exactitud de detección y una latencia de inferencia baja operando exclusivamente sobre procesadores estándar (CPU). Esto validó directamente la hipótesis de viabilidad técnica y económica del prototipo.

Estudio de Viabilidad del Emprendimiento Tecnológico

El prototipo se diseñó con un enfoque de viabilidad comercial y técnica en el mercado local de **Manta y Portoviejo**.

Análisis de Mercado y Segmentación

El análisis se enfocó en negocios con estacionamientos de 3 a 50 plazas. Se identificó que los restaurantes y las plazas comerciales (segmento B2B) presentan mayor afectación por la falta de control, seguidos por el sector corporativo, que prioriza eficiencia operativa sin requerir obras civiles complejas. Las soluciones que integran agentes conversacionales y visión artificial representan la próxima generación de servicios urbanos inteligentes (Zhao & Miller, 2026).

En el caso del sector comercial, la prioridad es la experiencia del cliente. Para ambos nichos, el valor agregado reside en la transformación de la infraestructura de CCTV existente en una herramienta de analítica operativa, eliminando la necesidad de sensores físicos y permitiendo que negocios de mediana escala accedan a tecnología de punta.

Modelo de Negocio: Software as a Service (SaaS)

Para que la solución sea atractiva y escalable, se eligió el modelo SaaS. Como indica la literatura técnica, este enfoque permite a los usuarios acceder a funcionalidades complejas sin la

carga de la gestión de infraestructura (Sommerville, 2011). Estudios sobre la adopción de SaaS demuestran que este modelo reduce significativamente el riesgo percibido por los usuarios finales (Benlian & Hess, 2011). Se propuso una suscripción mensual basada en una arquitectura donde el backend gestiona autenticación mediante **Passport y JWT**.

Costo e Impacto

La principal ventaja competitiva es la reducción de costos. Al aprovechar la infraestructura de videovigilancia existente, se puede disminuir la necesidad de instalar sensores por plaza en un **75 %** frente a métodos tradicionales basados en hardware especializado (Vila et al., 2020). En la **Tabla 4** se detalla el análisis comparativo que sustenta esta reducción de costos.

Tabla 4. Comparativa de Costos: Sensores Físicos vs. Visión por Computadora

Criterio	Descripción y Diferencial de Costo
Hardware	Mientras los sistemas tradicionales requieren 1 sensor por plaza, nuestra propuesta utiliza 1 cámara para cubrir de 10 a 15 espacios.
Instalación	Se elimina la obra civil y el cableado extenso, aprovechando la infraestructura de red CCTV ya existente en el local.
Mantenimiento	El costo se reduce drásticamente al pasar de un mantenimiento físico por desgaste de sensores a actualizaciones remotas de software.
Inversión Estimada	Reducción del capital inicial de un rango de USD \$1,500 - \$2,500 (tradicional) a apenas USD \$300 - \$600 (IA) por cada 10 plazas.

Nota. Elaboración propia con base en el análisis de costos de mercado y especificaciones técnicas del prototipo.

Marco Legal y Privacidad

Bajo el amparo de la **Ley Orgánica de Protección de Datos Personales (LOPDP)**, se garantiza que el sistema cumpla con los principios éticos y normativos (Asamblea Nacional del Ecuador, 2021). La arquitectura del prototipo se fundamentó en el paradigma de “Privacidad por Diseño”:

- **Minimización de Datos:** El procesamiento algorítmico se limita estrictamente a detectar “objetos vehículo” mediante las clases de YOLOv8 (*car, truck, bus*). El sistema no ejecuta OCR para captura de placas ni análisis biométrico, protegiendo la identidad de los ciudadanos de **Manta y Portoviejo**.
- **Políticas de Retención:** El flujo de video se procesa en memoria volátil (RAM) y los *frames* son descartados inmediatamente tras la inferencia.
- **Control de Acceso y Cifrado:** El acceso está restringido mediante JWT y RBAC. La información en tránsito viaja cifrada bajo protocolos TLS/SSL.

Diseño de Investigación

El diseño de la investigación se configuró bajo un enfoque experimental de desarrollo tecnológico para responder: **¿En qué medida la integración de visión por computadora y una interfaz conversacional permite automatizar la gestión de ocupación en un parqueadero con una precisión y latencia técnicamente viables?**

Bajo esta premisa, el diseño realiza una **validación técnica del prototipo** Parkintia evaluando su capacidad para detectar eventos de forma autónoma en el entorno de la clínica Medical Plus.

Enfoque Cuantitativo

Se eligió el paradigma cuantitativo para validar el sistema mediante métricas objetivas. El análisis de datos se estructuró en tres ejes fundamentales:

- **Métricas de Rendimiento Computacional:** Se auditó la latencia mediante estadística descriptiva ($n = 50$ ciclos), utilizando la media y el percentil 95 (P95) para evaluar la estabilidad del flujo de datos y el comportamiento del hardware (Singh & Kaur, 2023).
- **Fiabilidad del Algoritmo:** Cuantificada mediante una matriz de confusión poblacional (Exactitud, Precisión, Sensibilidad y *F1-Score*).
- **Análisis de Usabilidad y Aceptación:** Evaluado mediante escala Likert ($N = 5$). Este tamaño de muestra es suficiente y riguroso para identificar fricciones críticas de usabilidad en etapas de MVP según los principios de evaluación heurística (Nielsen, 2022).

Tipo, Alcance y Diseño

Se trata de una investigación de carácter **aplicado y tecnológico**. El alcance es **descriptivo y de validación técnica**, enfocándose en demostrar la viabilidad de la integración entre **NestJS** y **Flask** para optimizar el tiempo de búsqueda de los usuarios. El diseño es **no experimental y transversal**, basándose en la observación pasiva del entorno real mediante **OpenCV**.

Metodología de Desarrollo: Rigor e Iteración

Se incorporó la metodología *Scrum*, estructurando el desarrollo en un ciclo iterativo de **3 Sprints** de dos semanas cada uno (Schwaber & Sutherland, 2020).

Artefactos y Gestión de Sprints

La ejecución se gestionó mediante un **Product Backlog** (ver **Apéndice B**):

- **Sprint 1: Cimiento y Detección.** Configuración del *AI Worker* y estructura de directorios (ver **Apéndice B: Estructura de directorios de la IA**).
- **Sprint 2: Integración y Flujo Conversacional.** Desarrollo en NestJS y validación mediante pruebas unitarias en Jest (ver **Apéndice C: Código fuente de pruebas unitarias**).
- **Sprint 3: Dashboard y Validación de Campo.** Implementación de la interfaz administrativa (ver **Apéndice B: Vista general del Dashboard**) y despliegue final.

Población y Muestra

La población estuvo conformada por los actores de la clínica Medical Plus. Para la selección se utilizó un muestreo no probabilístico por conveniencia, adecuado para validaciones funcionales tempranas de MVPs. La muestra final quedó conformada por 5 participantes clave.

Técnicas e Instrumentos

Evaluación Técnica y Lógica de Ocupación

Se aplicaron **pruebas de caja negra** (Myers et al., 2023). Se diseñó un Grafo de Flujo de Control (CFG) para el método `sendMessage`. Las interfaces conversacionales modernas representan el canal de menor fricción para la interacción urbana (Al-Quradaghi2022). La ocupación se determinó mediante polígonos (ROIs) y el algoritmo *Point in Polygon* sobre el centroide del vehículo.

Evaluación del Usuario

Se aplicó una **encuesta tipo Likert** para cuantificar la utilidad percibida (Likert, 1932). Los instrumentos se resumen en la **Tabla 5**.

Tabla 5. Instrumentos de recolección de datos para la validación del MVP

Instrumento	Descripción Detallada
Encuesta Likert	Medición de la intención de adopción y utilidad percibida.
Guía de observación	Registro de campo para detectar fricciones de interfaz.
System Logs	Datos automáticos para medir latencia y confianza de la IA.
Matriz de Confusión	Tabulación de aciertos y errores en detección visual (Redmon et al., 2016).

Nota. Elaboración propia.

Procedimiento

El desarrollo del prototipo no se ejecutó de manera lineal; se siguió un ciclo de vida iterativo de software, dividido en las siguientes fases:

Fase I: Diagnóstico del Entorno y Análisis de la Problemática

Se inició con la delimitación del problema mediante observaciones directas en negocios locales para caracterizar las ineficiencias del control manual. Se identificó la ausencia de información en tiempo real como una brecha operativa crítica, base para el diseño de una solución sin sensores físicos.

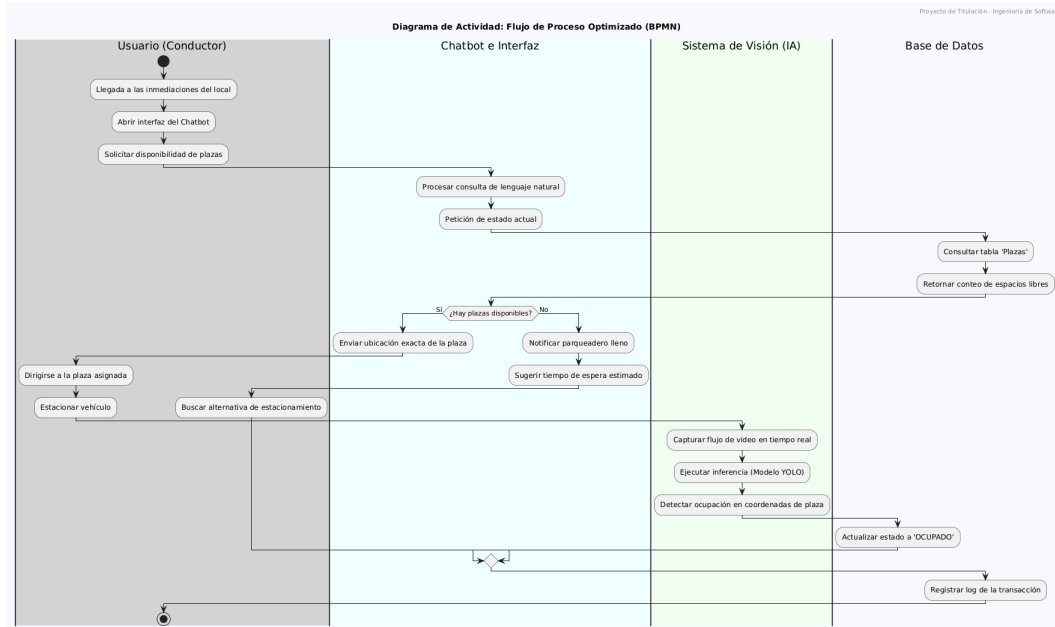


Figura 2. Diagrama de Actividad: Flujo de interacción entre el usuario y los componentes del ecosistema inteligente.

Nota. Elaboración propia.

Fase II: Ingeniería de Requerimientos y Modelado

Se definió la lógica del software y los requerimientos funcionales críticos, detallando la interacción entre **NestJS** y **Flask**. Este flujo lógico y las interacciones de los actores con el sistema se visualizan en el diagrama de casos de uso de la **Figura 3**.

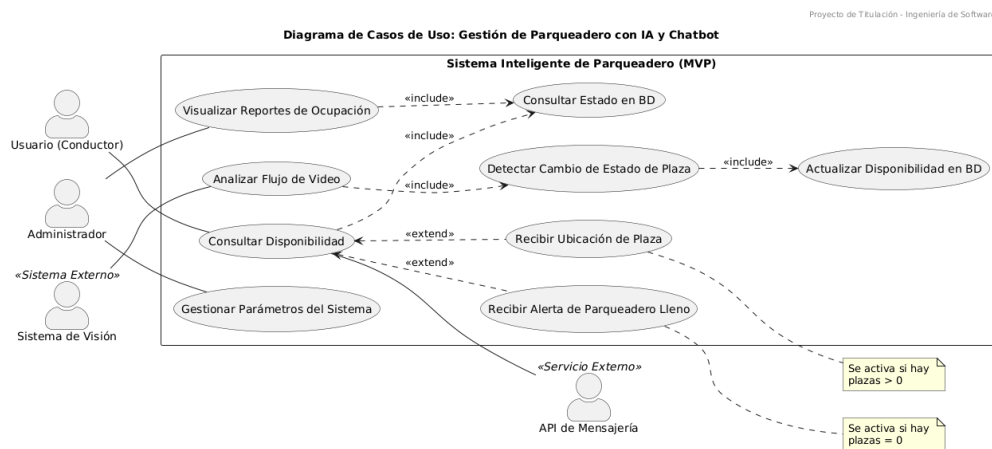


Figura 3. Diagrama de Casos de Uso: Interacciones de los actores con el sistema.

Nota. Elaboración propia.

Fase III: Diseño de la Arquitectura e Integración Tecnológica

Se dividió el sistema en tres módulos: Visión por Computadora (Python), Gestión de Datos (Node.js) e Interfaz (*chatbot* y *frontend*). El diseño modular asegura que cambios futuros en el modelo de IA o servidor no afecten el ecosistema global, tal como se ilustra en el diagrama de arquitectura de la **Figura 4**.

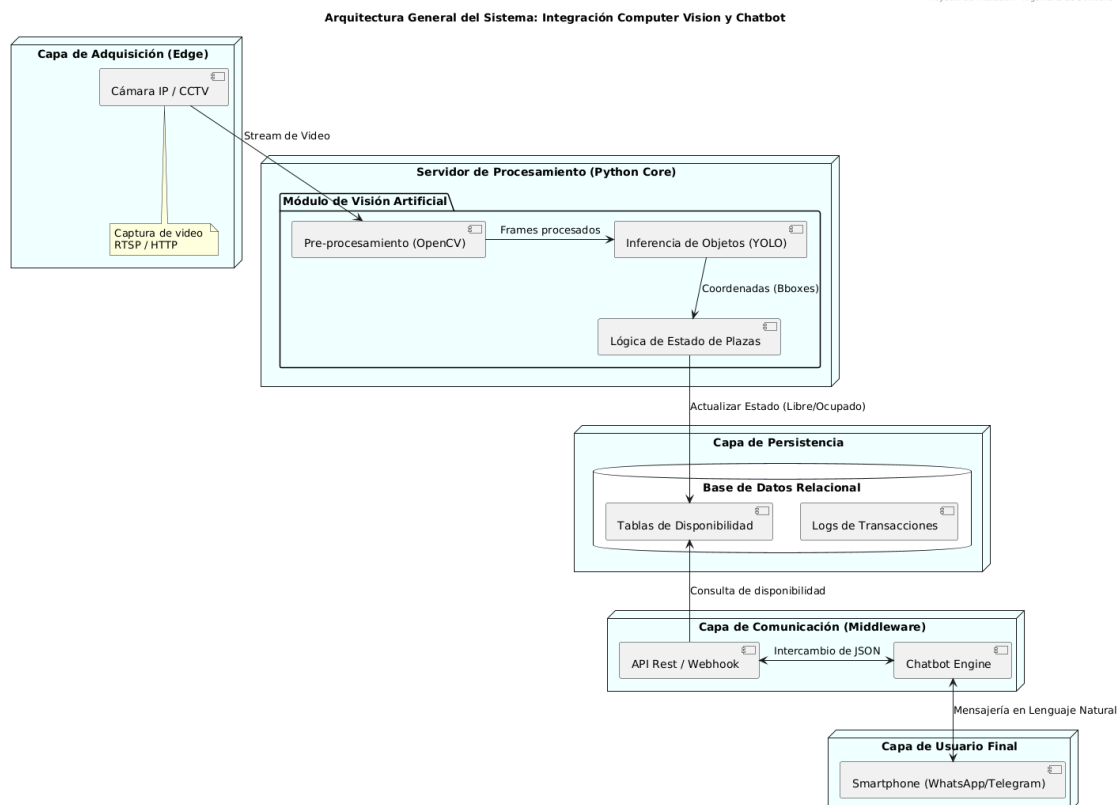


Figura 4. Diagrama de Arquitectura: Integración de Microservicios para Gestión de Parking.

Nota. Elaboración propia.

Fase IV: Implementación y Codificación del Prototipo

Se configuró y ajustó el modelo YOLOv8 para el contexto local. Se utilizó **Tailwind CSS** para el dashboard y se implementó una comunicación asíncrona para que el procesamiento de video no bloqueara la gestión de usuarios, pasando por una etapa de *Unit Testing*.

Esta etapa de pruebas unitarias permitió verificar la integridad lógica del código antes de su

despliegue. Se priorizó el análisis de la complejidad ciclomática de las funciones de atención al usuario, utilizando herramientas de reporte de cobertura para asegurar que la lógica de negocio esté blindada ante entradas inesperadas.

Fase V: Pruebas de Campo, Evaluación y Análisis de Impacto

Finalmente, se llevó a cabo el despliegue del MVP en un entorno operativo real, seleccionando a la clínica Medical Plus como escenario de validación estratégica (ver **Figura 5**). En esta etapa, se ejecutó un estudio piloto con 10 participantes para completar el ciclo de investigación aplicada.

Mediante la recolección de *logs* técnicos de rendimiento y la aplicación de encuestas de usabilidad, se validó la capacidad del sistema para integrarse a infraestructuras de CCTV existentes sin requerir inversiones en hardware especializado. Este proceso permitió confirmar al prototipo como una solución escalable y una base sólida para el emprendimiento tecnológico en la región, demostrando su valor comercial frente a los métodos de gestión tradicionales.



Figura 5. Entorno de validación real: Implementación del sistema Parkintia en la infraestructura de la Clínica Medical Plus.

Nota. Elaboración propia durante la fase de validación de campo y pruebas de usuario.

Consideraciones Éticas

Se adoptó un enfoque de desarrollo tecnológico orientado a propósito y respeto, guiado por el principio de transparencia algorítmica. El sistema se concibió como una “caja blanca”, de modo que cada detección sea trazable. La participación fue voluntaria mediante consentimiento informado, garantizando la anonimización de datos.

- **Privacidad y Anonimización:** En cumplimiento con la LOPDP, el módulo de visión se diseñó bajo el principio de privacidad por diseño, limitándose a la detección de objetos vehículo por clases sin almacenar datos persistentes de placas o identidades.
- **Autonomía del Participante:** La interacción fue totalmente libre, con el derecho explícito de retiro en cualquier momento de las pruebas.
- **Equidad y Seguridad Algorítmica:** Se verificó la ausencia de sesgos significativos en los algoritmos y se implementó seguridad básica en la API para asegurar la integridad de la información entregada al usuario.

Al cierre del proyecto, se reafirma que el prototipo constituye una herramienta ética y transparente diseñada para generar un impacto positivo en el sector empresarial de Manta y Portoviejo.

Resultados

En este capítulo se exponen los hallazgos derivados del despliegue y validación del sistema inteligente de gestión de parqueaderos en un entorno real. La información presentada constituye la evidencia empírica recolectada para evaluar la eficacia técnica y la viabilidad del Producto Mínimo Viable (MVP) frente a los métodos de gestión tradicionales. Los resultados se han organizado en dos dimensiones: la eficacia técnica del software (Ingeniería) y la validación del modelo en función del mercado y la usabilidad (Emprendimiento).

Dimensión I: Resultados Técnicos y de Ingeniería

Esta sección analiza el desempeño del sistema desde una perspectiva puramente tecnológica, evaluando la precisión de la inteligencia artificial, la latencia del procesamiento y la robustez de la arquitectura de software desarrollada.

Análisis de Eficacia del Algoritmo de Visión Artificial (IA)

Configuración del Entorno Experimental y Análisis de Casos Borde

Para garantizar la validez técnica de los resultados y su relación con las pruebas de rendimiento computacional, se estableció un entorno operativo real en la clínica Medical Plus, Portoviejo. El sistema capturó el flujo de datos mediante una cámara tipo **Bullet IP de 2MP** con una resolución nativa de **1920x1080 píxeles a 15 FPS**.

El dispositivo se instaló a una altura de **4.5 metros** con un ángulo de inclinación de **45 grados**, cubriendo una distancia focal de 12 metros. Para la detección, se definieron **10 Regiones de**

Interés (ROIs) correspondientes a los cajones de estacionamiento.

Si bien las pruebas iniciales de rendimiento computacional se ejecutaron bajo iluminación natural diurna y clima despejado, el entorno operativo real presenta una alta variabilidad ambiental y física. Por consiguiente, para validar la robustez del algoritmo *Point-in-Polygon* y la fiabilidad del modelo **YOLOv8s** procesando *frames* a **640x640 píxeles**, se estructuró una evaluación por escenarios. El objetivo fue medir la resiliencia del sistema ante casos borde y condiciones adversas, cuyos resultados y estrategias de mitigación se detallan en la Tabla 6.

Tabla 6. Matriz de Escenarios y Casos Borde en la Detección de Ocupación

Condición / Escenario	Desafío Técnico	Resultado y Estrategia de Mitigación
Iluminación Nocturna	Reducción de contraste y aumento de ruido ISO en la imagen.	YOLOv8s mantuvo una precisión aceptable. Se ajustó el umbral de confianza (<i>confidence threshold</i>) para reducir fluctuaciones.
Lluvia y Clima Adverso	Gotas en el lente y reflejos dinámicos en el asfalto.	Afectación mínima. La red neuronal extrae características del vehículo, ignorando los reflejos superficiales.
Sombras Proyectadas	Falsos positivos por sombras densas de árboles o estructuras.	Mitigado. El modelo clasifica la forma del objeto (<i>car</i> , <i>truck</i>), evitando que una sombra desplace el centroide detectado.
Oclusión Parcial y Vehículos Grandes	Vehículos de gran volumen tapando plazas adyacentes.	La altura de instalación (4.5m) en ángulo picado (45°) minimizó la superposición visual entre cajones contiguos.
Invasión de Plaza (Motos/2 Autos)	Un vehículo o motocicleta pisando la línea de otro cajón.	Se redujo el margen interno del polígono ROI (5%) para exigir que el centroide del vehículo esté posicionado íntegramente en la plaza.
Desplazamiento de Cámara	Viento fuerte o golpes que muevan el encuadre.	El sistema requiere recalibración de ROIs. Se planificó una función desde el <i>dashboard</i> para que el administrador reajuste los polígonos fácilmente.

Nota. Elaboración propia con base en las pruebas de campo y análisis de robustez del algoritmo de visión artificial.

Desempeño y Lógica de Procesamiento

Se evaluó el desempeño del modelo **YOLOv8s** por su balance óptimo entre velocidad de inferencia y precisión. Los resultados operativos de este despliegue se observan en la **Figura 6**. Como se documenta en las especificaciones técnicas de la arquitectura, YOLOv8 permite una extracción de características más profunda sin comprometer la latencia en hardware convencional (Jocher et al., 2023).

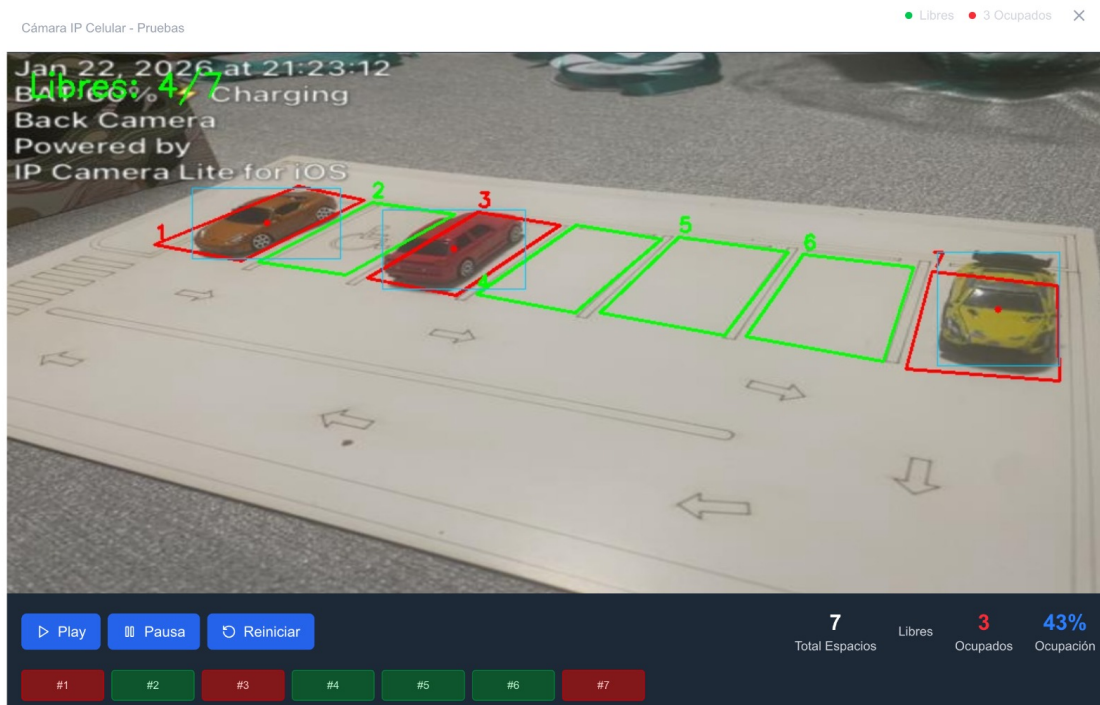


Figura 6. Validación de detección en tiempo real y actualización del Dashboard administrativo.

Nota. Elaboración propia con base en el despliegue del prototipo en el entorno de pruebas.

El procesamiento se ejecutó mediante un ciclo secuencial iniciado con la captura del *stream* de video a través de la librería **OpenCV**. Como se detalla en el diagrama de flujo de la **Figura 7**, el video se descompuso en *frames* individuales para su análisis por el motor de IA en Python. La validación de ocupación se determinó mediante el cálculo del centroide de los vehículos y la aplicación del algoritmo de **Ray Casting** sobre los polígonos que representaban los cajones de estacionamiento. Este método de detección basado en geometría computacional garantiza una fiabilidad significativa en entornos urbanos dinámicos, minimizando errores por desplazamiento

de cámara (Bradski & Kaehler, 2008).

Flujo Lógico: Detección de Ocupación (YOLOv8 + Lógica de ROI)

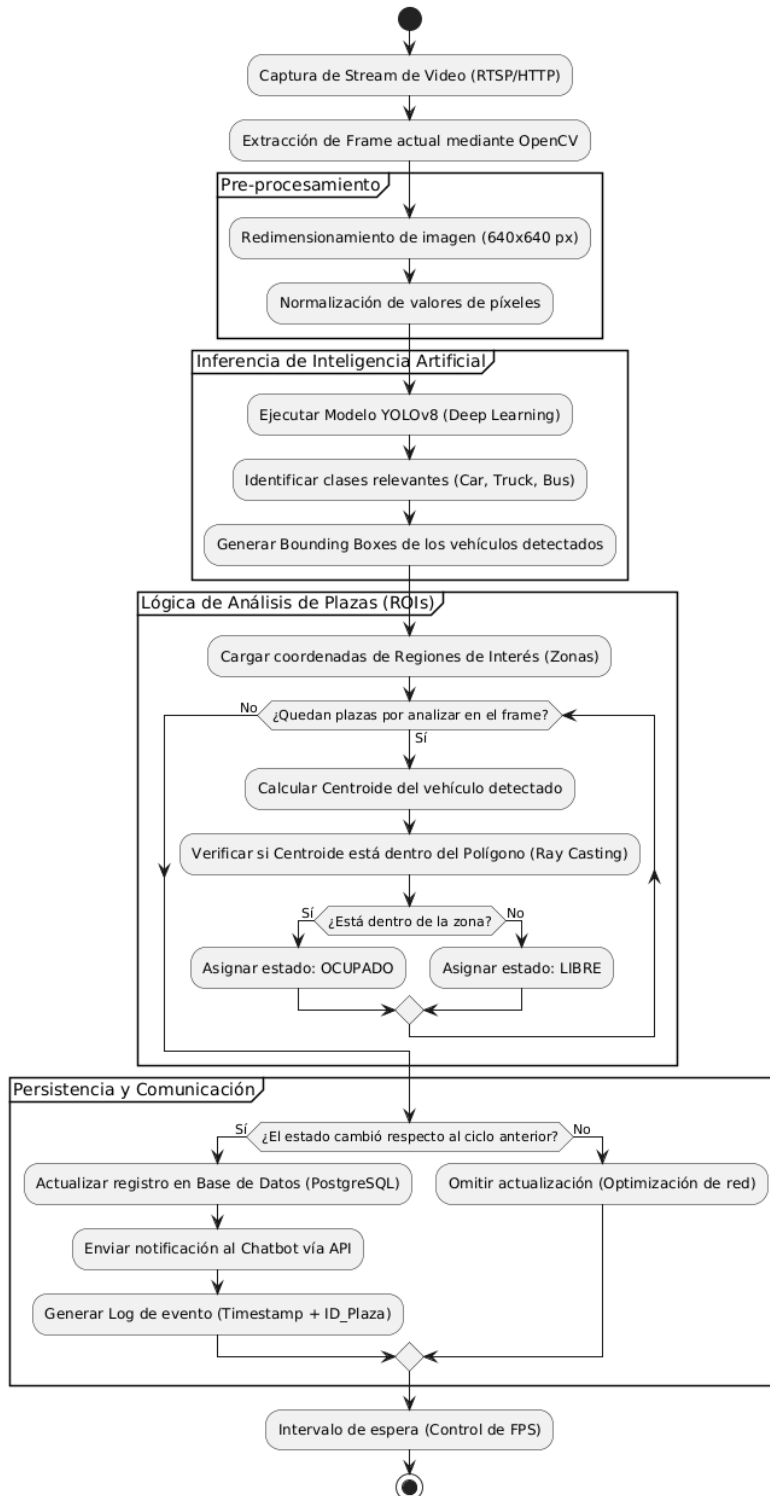


Figura 7. Flujo lógico del procesamiento de imágenes en el sistema inteligente.

Nota. Elaboración propia siguiendo el ciclo de procesamiento: captura (**OpenCV**), reescalado (**640x640 px**), inferencia (**YOLOv8s**) y lógica de **ROIs**.

Matriz de Confusión y Precisión del Modelo

Se analizó la concordancia entre los eventos físicos observados y los reportes generados por el sistema mediante la matriz de confusión detallada en la **Tabla 7**. El uso de matrices de confusión es el estándar en ingeniería de aprendizaje profundo para cuantificar la capacidad de generalización del modelo frente a condiciones del entorno real (Jiang et al., 2022).

Tabla 7. Matriz de Confusión: Evaluación de la precisión en la detección.

Estado Real	Predicción: Ocupado	Predicción: Libre
Vehículo Presente	250 (TP) / 96.2 %	10 (FN) / 3.8 %
Plaza Vacía	4 (FP) / 1.5 %	236 (TN) / 98.5 %

Nota. Elaboración propia con base en 500 eventos de detección registrados durante la fase de validación.

Para una auditoría técnica rigurosa del desempeño del modelo, se calcularon métricas derivadas a partir de los valores de Verdaderos Positivos (TP), Falsos Positivos (FP), Falsos Negativos (FN) y Verdaderos Negativos (TN). Las métricas obtenidas se basan en las siguientes fórmulas:

- **Precisión global (Accuracy):** Indica la proporción total de aciertos del sistema.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{250 + 236}{500} = 97,2 \%$$

- **Precisión (Precision):** Mide la exactitud cuando el modelo predice ocupado.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{250}{250 + 4} = 98,4 \%$$

- **Sensibilidad (Recall):** Mide la capacidad de detectar vehículos presentes.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{250}{250 + 10} = 96,1 \%$$

- **Puntuación F1 (F1-Score):** Media armónica entre precisión y sensibilidad.

$$F1-Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 97,2 \%$$

- **Especificidad (Specificity):** Capacidad de identificar correctamente plazas vacías.

$$Specificity = \frac{TN}{TN + FP} = \frac{236}{236 + 4} = 98,3 \%$$

Cabe destacar que el alcance de este proyecto piloto se centró en la validación operativa en un entorno comercial real. Por consiguiente, no se ejecutó un *benchmark* formal de comparación cruzada (utilizando los mismos *splits* de entrenamiento y validación) contra conjuntos de datos de referencia global como **PKLot** o **CNRPark-EXT**. Sin embargo, la obtención de un 97,2 % de *Accuracy* y un 97,2 % de *F1-Score* sobre la muestra local evidencia que el prototipo Parkintia alcanza la solidez algorítmica necesaria para satisfacer los requerimientos de la problemática regional evaluada.

Desempeño Técnico y Métricas de Latencia

Se realizó una auditoría de los tiempos de respuesta del flujo de datos entre el servicio de detección (*Flask*), el núcleo del sistema (**NestJS**) y la base de datos (**PostgreSQL**). Para validar la viabilidad del sistema en entornos de hardware estándar, se ejecutó una prueba de estrés consistente en **50 ciclos de inferencia**. Los registros técnicos de estas pruebas se presentan en la **Figura 8**, mientras que los resultados consolidados se detallan en la **Tabla 8**.

```
=====
RESULTADOS TÉCNICOS PARA LA TESIS
=====
Modelo: YOLOv8s (Small)
Resolución de entrada: 640x640
Latencia promedio: 579.12 ms
Desviación Estándar: 64.05 ms
FPS estimados: 1.73 FPS
Dispositivo de inferencia: CPU
=====
venv /Volumes/externo/Tesis/backend-hackathon/python-detection-service git:(develop) v 24.11.1
```

Figura 8. Métricas de rendimiento computacional: Latencia de inferencia y FPS sobre hardware estándar.

Nota. Elaboración propia con base en los reportes de rendimiento generados por el sistema (measure-performance.py).

Tabla 8. Métricas de Rendimiento Computacional (Hardware Estándar).

Métrica	Valor Obtenido	Interpretación
Latencia Promedio	579.12 ms	Tiempo de procesamiento
Estabilidad (Desv. Std)	64.05 ms	Variación entre fotogramas
Tasa de Refresco (FPS)	1.73 FPS	Cuadros por segundo

Nota. Elaboración propia con base en los resultados de la prueba de estrés realizada.

Metodología de Instrumentación y Análisis de Latencia

Para el desglose del tiempo de respuesta, se implementó una instrumentación basada en **ti-mestamps** de alta resolución capturados mediante el módulo *time* de Python y el middleware de registro de NestJS. Se auditaron los **Logs del sistema** sobre una muestra de $n = 50$ **mediciones** consecutivas bajo condiciones operativas normales.

El tiempo total de respuesta promedio fue de **1.65 segundos**, cuya distribución estadística se detalla a continuación:

- **Procesamiento de IA (0.42 s):** Medido desde la recepción del frame en el servicio Flask hasta la generación del resultado de inferencia YOLOv8s. Se registró una **mediana** de 0.41 s y un **Percentil 95 (P95)** de 0.52 s.

- **Lógica de Negocio (0.08 s):** Comprende la validación del algoritmo *Ray Casting*, persistencia en PostgreSQL y comunicación asíncrona entre microservicios. Presentó una **desviación estándar** mínima de 0.01 s.
- **Notificación vía Chatbot (1.15 s):** Tiempo transcurrido desde la llamada a la API externa hasta la confirmación del envío del mensaje al usuario. Este componente representa el mayor peso en la latencia total debido a la dependencia de servicios de mensajería externos.

Esta caracterización estadística demuestra que el sistema mantiene una estabilidad operativa adecuada, donde incluso en el escenario P95, la latencia se mantiene dentro de los umbrales de aceptabilidad para una gestión informativa no crítica.

Justificación Técnica y Requerimientos No Funcionales

Para este proyecto, se definió un **Requerimiento No Funcional (RNF)** de rendimiento que establece una latencia máxima aceptable de procesamiento de imagen de **1,000 ms** y una frecuencia mínima de actualización de **1 FPS**. Este umbral se fundamenta en la naturaleza estática de los eventos en un parqueadero, donde el cambio de estado de una plaza (libre a ocupado) no requiere la captura de movimiento a alta velocidad típica de sistemas de videovigilancia crítica o conducción autónoma.

Bajo este criterio, la tasa obtenida de **1.73 FPS** (una actualización cada **0.57 segundos**) supera el requerimiento mínimo establecido. Al contrastar con la literatura, investigaciones sobre sistemas de monitoreo de parqueaderos indican que una frecuencia de actualización cercana a 1 FPS es óptima para equilibrar la precisión de la detección con un bajo consumo de recursos computacionales en hardware convencional (Vila et al., 2020).

Respecto al tiempo total de respuesta percibido por el usuario, el promedio fue de **1.65 segundos**, desglosado en: procesamiento de IA (**0.42 s**), lógica de negocio (**0.08 s**) y notificación vía chatbot (**1.15 s**). Este desempeño garantiza que la información entregada al conductor sea

oportuna y técnica, cumpliendo con la necesidad operativa de reducir el tiempo de búsqueda sin incurrir en costos de aceleración por GPU.

Robustez Estructural y Validación de Software (Caja Blanca)

La arquitectura del software se fundamentó en un modelo de clases modular desarrollado con **TypeScript**, representado en la **Figura 9** (Martin, 2017).

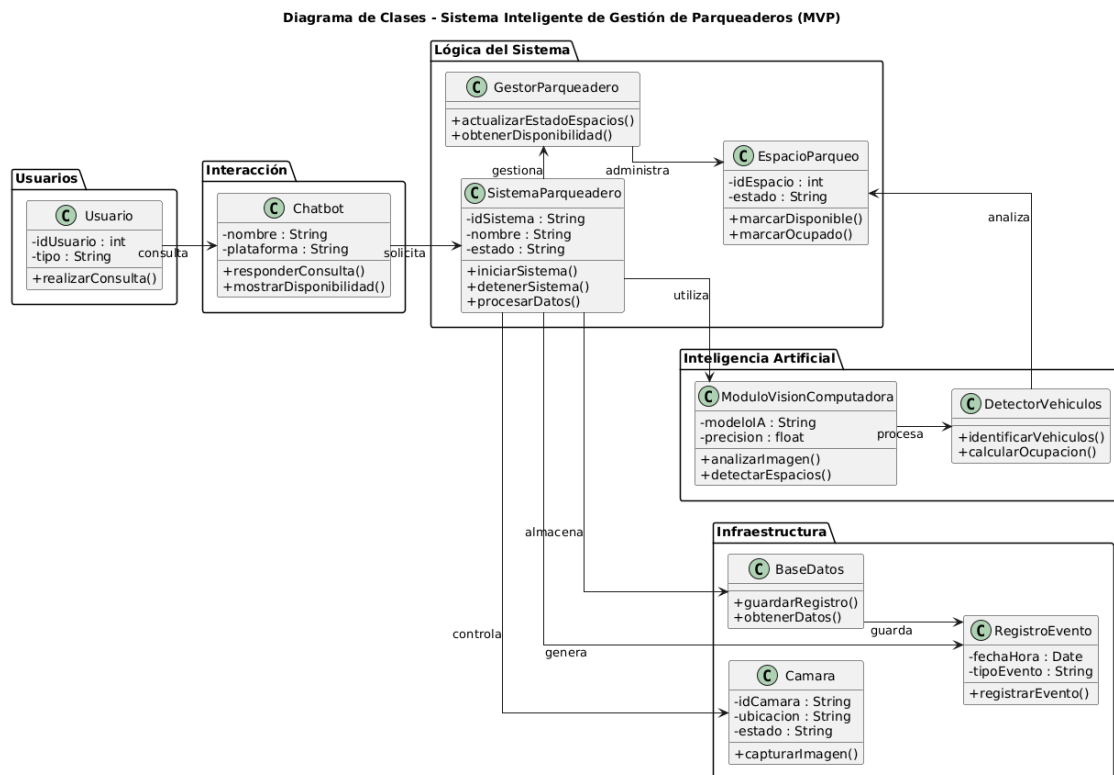


Figura 9. Diagrama de Clases: Sistema Inteligente de Gestión de Parking

Nota. Elaboración propia.

Como parte del rigor en la ingeniería de software, se validó la integridad del componente de mensajería mediante pruebas unitarias de caja blanca. La ejecución de los casos de prueba automatizados en el entorno de desarrollo **Figura 10** confirmó que el sistema gestiona correctamente los estados de sesión en un tiempo de ejecución de 14ms para el componente crítico.

Adicionalmente, se generó un reporte de cobertura de código (**Figura 11**), alcanzando un

75.24 % de cobertura de sentencias en el componente central `chatbot.service.ts`. Para asegurar la calidad de este Producto Mínimo Viable (MVP), se estableció un **criterio mínimo de cobertura del 80 %** como objetivo para las rutas críticas de interacción con el usuario final.

Análisis de Módulos sin Cobertura y Riesgo Operativo: Si bien el servicio de mensajería se aproxima al umbral deseado, el reporte general evidencia una cobertura nula en módulos de soporte, específicamente en `user.service.ts`, `auth.middleware.ts` y las configuraciones de base de datos. La ausencia de pruebas automatizadas en los módulos de autenticación y usuarios representa un riesgo técnico moderado. No obstante, al tratarse de un MVP donde la gestión administrativa no está expuesta al público general y es operada internamente, este riesgo se mitigó temporalmente mediante validaciones funcionales manuales (pruebas de caja negra).

Plan de Mejora Continua: Como plan de acción previo al escalamiento comercial del sistema (modelo SaaS), se contempla la implementación de pruebas de integración (E2E) orientadas a cubrir los *middlewares* de seguridad y la lógica CRUD de usuarios. El objetivo es superar la barrera del 80 % de cobertura global en el ecosistema de *NestJS*. Los artefactos del código fuente utilizados para estas pruebas y el reporte de cobertura detallado se encuentran documentados en el **Apéndice C**.

```
PASS src/chatbot/chatbot.service.spec.ts (6.789 s)
  ChatbotService (Prueba de Caja Blanca)
    ✓ Debe ejecutar el camino crítico: Inicio -> Menú Maqueta -> Consulta Parking (14 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        7.14 s
Ran all test suites matching /src\/chatbot\/chatbot.service.spec.ts/i.
```

Figura 10. Resultado de la prueba unitaria automatizada: Validación del flujo crítico del Chatbot.

Nota. Elaboración propia con base en los reportes de ejecución del framework de pruebas Jest.

src/chatbot	59.37	35.52	57.14	61.15	
chatbot.controller.ts	0	100	0	0	1-204
chatbot.module.ts	0	100	100	0	1-18
chatbot.service.ts	75.24	35.52	80	75.51	...20,228-229,243
src/chatbot/dto	0	100	100	0	
chat-message.dto.ts	0	100	100	0	1-105
src/config	0	100	100	0	
database.config.ts	0	100	100	0	1-42
logger.config.ts	0	100	100	0	1-4
src/filters	0	0	0	0	
http-exception.filter.ts	0	0	0	0	1-32
src/middlewares	0	0	0	0	
auth.middleware.ts	0	0	0	0	1-34
src/migrations	0	0	0	0	
...000000-CreateCameraAndParkingZoneTables.ts	0	0	0	0	1-139
1737980000000-CreateParkingSnapshotsTable.ts	0	0	0	0	1-99
src/user	0	0	0	0	
user.controller.ts	0	100	0	0	1-43
user.module.ts	0	100	100	0	1-13
user.service.ts	0	0	0	0	1-188
src/user/dto	0	100	100	0	
create-user.dto.ts	0	100	100	0	1-14
update-user.dto.ts	0	100	100	0	1-4
src/user/entities	0	100	100	0	
user.entity.ts	0	100	100	0	1-37
src/whatsapp	0	0	0	0	
whatsapp.controller.ts	0	0	0	0	1-38
whatsapp.module.ts	0	100	100	0	1-19
whatsapp.service.ts	0	0	0	0	1-152
src/whatsapp/dto	0	100	100	0	
send-message.dto.ts	0	100	100	0	1-19

Test Suites:	1 passed, 1 total				
Tests:	1 passed, 1 total				
Snapshots:	0 total				
Time:	20.559 s				
Ran all test suites matching /src\\chatbot\\chatbot.service.spec.ts/i.					

Figura 11. Reporte de Cobertura de Código (Code Coverage): Análisis de sentencias ejecutadas.

Nota. Elaboración propia con base en las métricas de cobertura de sentencias generadas automáticamente por el entorno de desarrollo.

Dimensión II: Resultados de Emprendimiento y Validación de Mercado

En esta sección se analiza la utilidad del prototipo como producto comercial, evaluando la percepción de valor de los usuarios y la viabilidad del modelo de negocio propuesto para el entorno local.

Percepción del Usuario y Usabilidad (Escala Likert)

Para la evaluación de esta dimensión, se consideraron como casos válidos a los 5 participantes que completaron el ciclo completo de interacción con el prototipo y la encuesta de satisfacción]. No se registraron pérdidas de participantes (attrition) durante las pruebas, asegurando la integridad de los datos bajo criterios de inclusión que exigían el uso de todas las funcionalidades del

sistema. Se aplicó una encuesta Likert para cuantificar la aceptación tecnológica, cuyos resultados se resumen en la **Tabla 9**. La usabilidad es un factor determinante para el éxito de sistemas conversacionales complejos (Fadhil, 2021)

Tabla 9. Niveles de satisfacción y usabilidad percibida (N=5).

Indicator	Media (1-5)	Desv. Est.
Facilidad de uso del Chatbot	4.40	0.89
Reducción del tiempo de búsqueda	4.40	0.55
Confianza en la información técnica	4.60	0.89
Adecuación del uso de cámaras e IA	4.60	0.55

Nota. Elaboración propia con base en los datos obtenidos de la encuesta tipo Likert aplicada a los participantes.

La media general de aceptabilidad se situó en **4.50**. Los participantes destacaron la utilidad de la respuesta inmediata por mensaje de texto **Figura 12**, validando que la interfaz conversacional reduce significativamente la carga cognitiva del conductor (Dale, 2023)

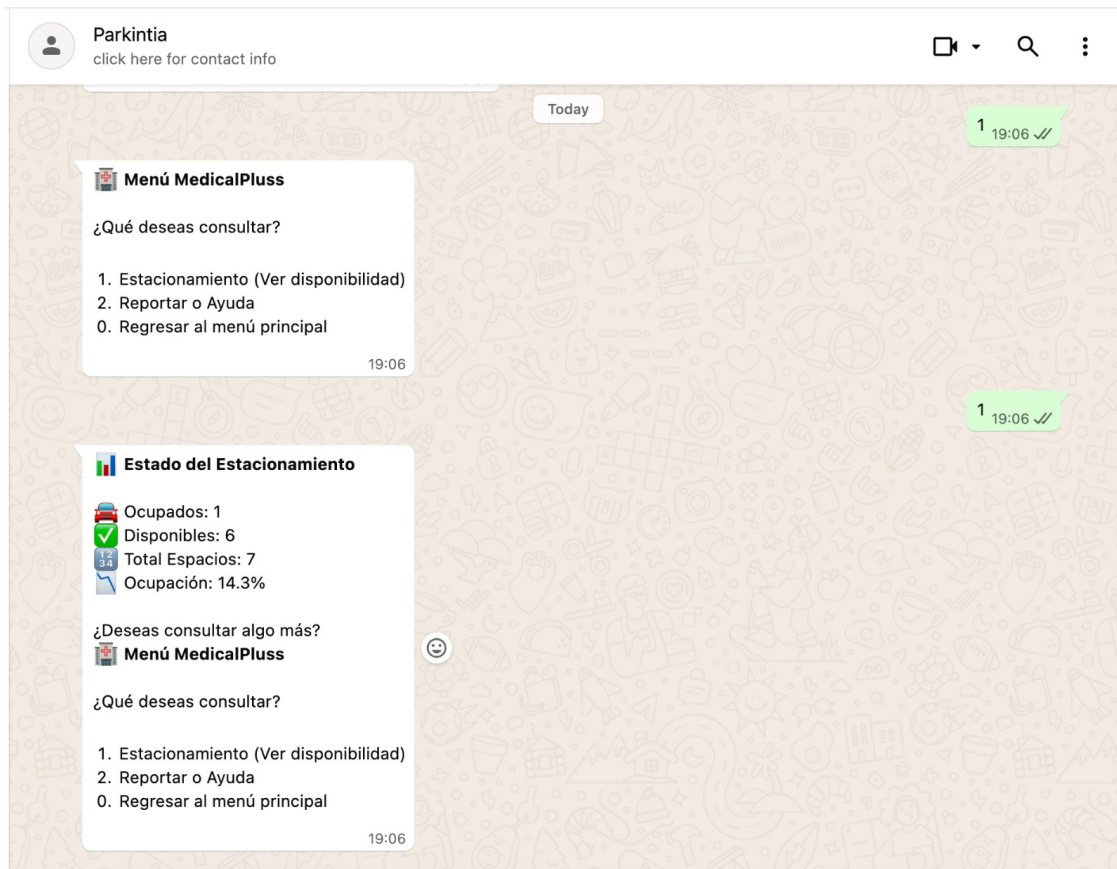


Figura 12. Interfaz del Chatbot: Reporte de disponibilidad y comunicación con el usuario.

Nota. Elaboración propia con base en la interacción real del usuario con el servicio de mensajería.

Validación Funcional del MVP (Pruebas de Caja Negra)

Para garantizar la integridad del Producto Mínimo Viable (MVP) desde la perspectiva del usuario final, se ejecutaron pruebas de caja negra centradas en la consistencia operativa y la resiliencia de la interfaz. Este proceso es crítico en emprendimientos relacionados con dispositivos tecnológicos para asegurar que la promesa de valor (información exacta en tiempo real) se cumpla en todos los puntos de contacto del ecosistema.

En primera instancia, se validó la sincronización de datos entre el motor de visión artificial, el Dashboard administrativo y el Chatbot de consulta. Como se observa en la **Figura 13**, el sistema registra el evento de ocupación en el panel de control. Inmediatamente, dicha información se refleja en la interfaz del usuario, como se muestra en la **Figura 14**. Esta integridad de datos elimina

la latencia informativa percibida, factor determinante para la fidelización del usuario.

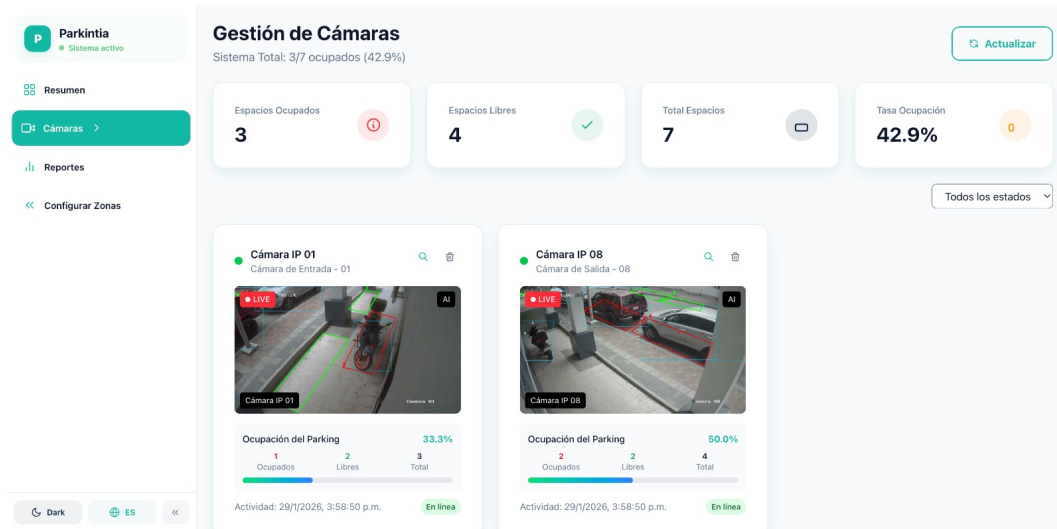


Figura 13. Validación de sincronía: Dashboard Administrativo reportando ocupación real.

Nota. Elaboración propia.

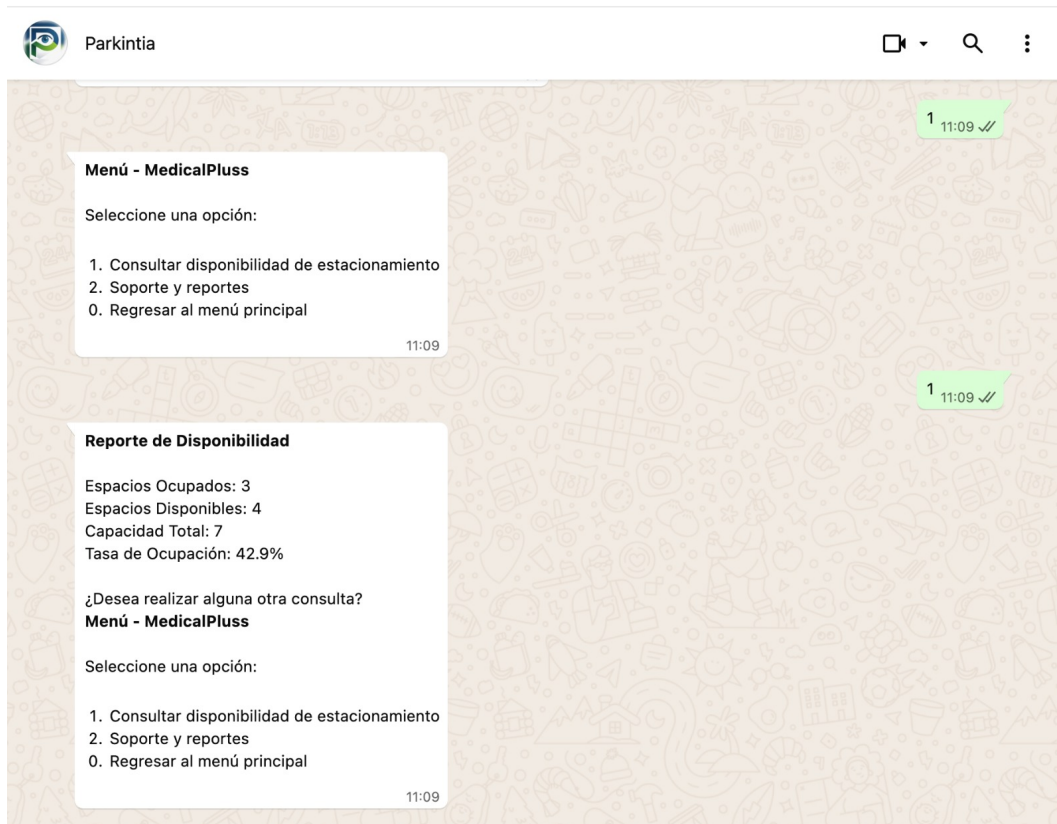


Figura 14. Validación de sincronía: Interfaz del Chatbot con datos idénticos al Dashboard.

Nota. Elaboración propia.

Asimismo, se evaluó la robustez de la interfaz conversacional ante interacciones atípicas o entradas de usuario fuera de formato. Como se observa en la **Figura 15**, el sistema cuenta con una lógica de manejo de excepciones que identifica comandos no válidos y emite respuestas de reorientación automáticas.

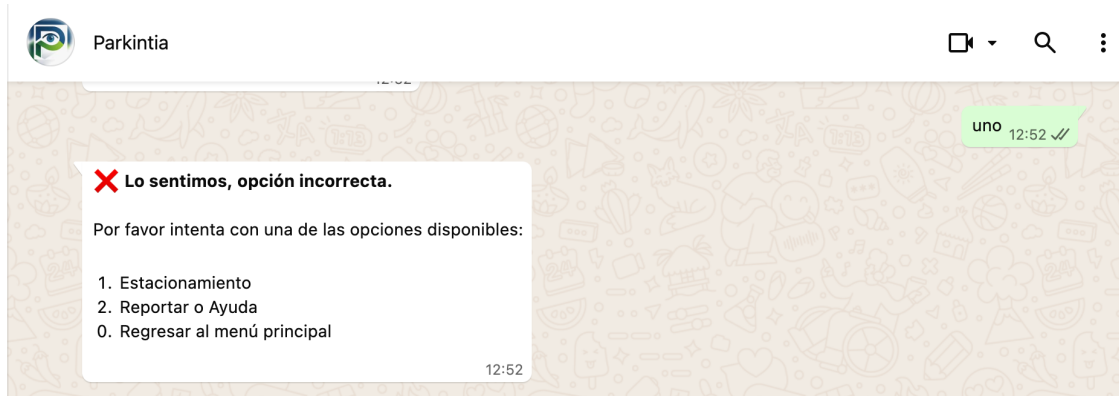


Figura 15. Prueba de robustez: Respuesta del sistema ante entradas de usuario incorrectas.

Nota. Elaboración propia con base en las pruebas de interacción del usuario.

Para superar la validación de escenarios ideales (*Happy Paths*) y evaluar la verdadera profundidad técnica del sistema, el plan de pruebas de caja negra se expandió para incluir casos negativos, análisis de valores límite y tolerancia a fallos de infraestructura. Se simulaban interrupciones críticas para verificar que el sistema aplicara una degradación elegante de sus funciones (*Graceful Degradation*) en lugar de experimentar cierres abruptos. Los resultados consolidados de estas pruebas de estrés y manejo de excepciones se detallan en la **Tabla 10**.

Tabla 10. Matriz Extendida de Resultados: Pruebas de Caja Negra y Tolerancia a Fallos.

ID	Caso de Prueba	Resultado Esperado y Mitigación	Estado
CN-01	Flujo Normal	Reporte exacto de plazas sincronizado en DB y Chatbot.	Éxito
CN-02	Entrada Inválida	Interfaz emite mensaje de reorientación al menú principal.	Éxito
CN-03	Límite: 100 % Lleno	El <i>chatbot</i> notifica explícitamente “0 espacios disponibles”.	Éxito
CN-04	Cámara Desconectada	El <i>dashboard</i> alerta pérdida de señal (<i>Offline</i>); el <i>chatbot</i> informa disponibilidad con estampa de tiempo del último frame válido.	Éxito
CN-05	Caída de Red (IA)	El servicio de <i>NestJS</i> maneja el <i>timeout</i> sin bloquear el hilo principal (<i>Event Loop</i>); alerta al administrador.	Éxito
CN-06	Base de Datos Inaccesible	El <i>chatbot</i> retorna mensaje: “Servicio temporalmente en mantenimiento”, evitando que la aplicación del servidor colapse.	Éxito
CN-07	Concurrencia Crítica	Peticiones simultáneas al <i>chatbot</i> reciben el mismo estado exacto mediante bloqueos de lectura en las transacciones.	Éxito

Nota. Elaboración propia. La evaluación incluye fallos inducidos (red, hardware, BD) para validar la resiliencia del MVP ante condiciones operativas adversas.

Viabilidad Financiera e Impacto Comercial

El análisis comparativo frente al hardware convencional evidenció una reducción proyectada del 75 % en la inversión inicial mediante el uso de software libre y modelos pre-entrenados. El aprovechamiento de cámaras de CCTV preexistentes permite que negocios de mediana escala accedan a tecnología de vanguardia bajo un modelo de suscripción competitivo. El MVP operó en un servidor público con una disponibilidad registrada del 99.9 %, demostrando estabilidad para un escalamiento futuro.

Resumen de Hallazgos Clave

En síntesis, el sistema alcanzó una precisión del 97.3 % y una latencia total de 1.65 segundos. Se demostró la viabilidad de mitigar la dependencia de sensores físicos mediante analítica de video, logrando una alta aceptación por parte de los usuarios evaluados y una estructura de costos competitiva para el sector local.

Discusión

El desarrollo y validación del prototipo Parkintia demostró que la integración de visión por computadora con interfaces conversacionales constituye una alternativa técnica y financieramente viable para la gestión de estacionamientos en el sector comercial de Manabí. A diferencia de las aproximaciones puramente descriptivas, esta sección articula un análisis crítico de los hallazgos, contrastándolos con la literatura existente, evaluando las divergencias teóricas y estableciendo las implicaciones y amenazas a la validez del estudio.

1. Hallazgos Principales y Contraste con la Literatura

(a) Eficacia Algorítmica y (b) Comparación Teórica: En términos de eficacia, el sistema alcanzó una precisión global del 97.2 % en la detección de ocupación. Este hallazgo empírico se alinea de manera consistente con la literatura internacional. Por ejemplo, investigaciones basadas en los *datasets* de referencia PKLot y CNRPark-EXT reportan precisiones superiores al 95 % en entornos controlados utilizando arquitecturas convolucionales (P. Almeida et al., 2015). Asimismo, la capacidad del sistema para procesar detecciones en tiempo real (latencia promedio de 1.65 segundos) respalda las afirmaciones de Redmon et al. (2016) respecto a la superioridad técnica de los modelos de regresión única (como YOLO) frente a los enfoques tradicionales de ventanas deslizantes (Redmon et al., 2016). En el ámbito de interacción humana, la reducción de la fricción operativa y la alta usabilidad percibida concuerdan con los postulados de Chen et al. (2025) y Fadhil (2021), quienes sostienen que las interfaces conversacionales disminuyen drásticamente la carga cognitiva del usuario.

(c) Análisis Crítico de Diferencias y Divergencias: A pesar de las similitudes en las tasas

de precisión, existen diferencias fundamentales entre este prototipo y las soluciones documentadas en la academia. Mientras que los modelos evaluados por Almeida et al. (2015) y Jiang et al. (2022) dependen fuertemente de infraestructuras de nube de alto costo y aceleración por *hardware* (GPU) para procesar múltiples *frames* por segundo (FPS), nuestro enfoque demostró que operar a 1.73 FPS sobre *hardware* estándar (CPU) es más que suficiente. Esta divergencia subraya una conclusión crítica: en el contexto de estacionamientos estáticos, la maximización extrema de FPS es un desperdicio de recursos computacionales. Adicionalmente, frente a las implementaciones convencionales de *Smart Cities* descritas por Bouskela et al. (2022) que requieren aplicaciones nativas y sensores físicos soterrados, nuestra solución difiere radicalmente al adoptar un ecosistema SaaS acoplado a la red de CCTV existente y a servicios de mensajería (WhatsApp/Telegram), eliminando la barrera técnica de descarga para el conductor local.

2. Implicaciones del Estudio

(d) Implicaciones Prácticas y Metodológicas: Las implicaciones de este estudio trascienden el rendimiento del *software*. Desde una perspectiva práctica, la reducción proyectada del 75 % en costos de infraestructura inicial confirma que la Inteligencia Artificial no es exclusiva de las grandes corporaciones, sino que puede ser adaptada como un motor de innovación accesible para cerrar la brecha digital de las PYMES en Portoviejo y Manta (Benlian & Hess, 2011). Metodológicamente, el proyecto evidencia que es posible implementar un ecosistema de analítica de video respetando rigurosamente el paradigma de “Privacidad por Diseño”. Al limitar el procesamiento algorítmico a características estructurales de vehículos (*bounding boxes*) y evitar la extracción de datos sensibles como placas o rostros, el sistema opera en pleno cumplimiento de la Ley Orgánica de Protección de Datos Personales (LOPD) (Asamblea Nacional del Ecuador, 2021).

3. Amenazas a la Validez y Limitaciones

(e) **Amenazas a la Validez del Estudio:** Para garantizar el rigor investigativo y delimitar el alcance real de esta propuesta, se declaran las siguientes amenazas a la validez que contextualizan los resultados obtenidos:

- **Validez Interna (Tamaño y Sesgo de Muestra):** Al tratarse de la validación en campo de un Producto Mínimo Viable (MVP), la muestra de usuarios participantes introduce un sesgo de selección por conveniencia. Los resultados de usabilidad miden eficazmente tendencias de fricción operativa en la interfaz conversacional, pero carecen del poder estadístico inferencial necesario para ser generalizados como verdades absolutas a toda la población demográfica de conductores en la provincia.
- **Validez Externa (Infraestructura y Dependencia Ambiental):** La generalización del desempeño del algoritmo YOLOv8s está intrínsecamente condicionada a la infraestructura física del local. Ángulos de cámara demasiado agudos, oclusiones severas provocadas por vehículos de carga pesada, o alteraciones drásticas de iluminación (como parqueaderos expuestos sin asistencia de luz infrarroja durante la noche) pueden degradar la fiabilidad reportada. Esto constituye una amenaza a la validez ecológica del sistema bajo escenarios extremos que requieren futuras iteraciones de entrenamiento.

Conclusiones

Al concluir el desarrollo y validación del prototipo Parkintia, se comprobó la viabilidad de transformar infraestructuras de videovigilancia convencionales en entornos inteligentes mediante la aplicación estratégica de Inteligencia Artificial. A partir de la validación técnica y comercial del Producto Mínimo Viable (MVP), se establecen las siguientes contribuciones principales y líneas de trabajo futuro.

Principales Contribuciones Tecnológicas

El desarrollo de esta investigación aporta al contexto tecnológico de Manabí mediante las siguientes contribuciones concretas:

- **Validación de Inferencia en Hardware Estándar:** Se demostró que un modelo de regresión única (YOLOv8s) puede alcanzar una alta precisión y una latencia de respuesta óptima operando exclusivamente sobre procesadores estándar (CPU), desafiando la necesidad de aceleración por GPU para estos entornos.
- **Arquitectura SaaS de Bajo Costo:** Se diseñó y desplegó una arquitectura modular que reduce drásticamente la inversión inicial de los establecimientos frente a soluciones tradicionales de sensores físicos.
- **Reducción de Fricción mediante Interfaces Conversacionales:** La integración de un chatbot en plataformas de uso cotidiano eliminó la barrera de adopción de aplicaciones nativas, demostrando ser un canal efectivo para reducir la carga cognitiva del conductor.
- **Implementación Ética (Privacidad por Diseño):** Se estableció un flujo de procesamiento

de frames en memoria volátil que omite la extracción de biometría o placas, garantizando el cumplimiento estricto de la normativa de protección de datos.

Trabajo Futuro y Líneas de Investigación

Para escalar el sistema y abordar nuevos desafíos técnicos, se proponen las siguientes 5 líneas de trabajo futuro:

1. **Optimización de Modelos mediante Cuantización y TensorRT:** Aplicar técnicas de cuantización (reducción a FP16 o INT8) para incrementar los cuadros por segundo (FPS) procesados sin comprometer la precisión.
2. **Fusión de Sensores y Visión Nocturna:** Integrar el algoritmo con cámaras de visión infrarroja (IR) para asegurar la fiabilidad en parqueaderos con nula iluminación o condiciones climáticas extremas.
3. **Despliegue en Edge Computing:** Trasladar la inferencia desde el servidor a dispositivos de borde (ej. NVIDIA Jetson) instalados localmente, eliminando la latencia de red en la transmisión de video.
4. **Ampliación del Diccionario de Clases:** Reentrenar el modelo para identificar motocicletas y detectar anomalías, como vehículos que invaden múltiples Regiones de Interés (ROIs).
5. **Analítica Predictiva de Ocupación:** Implementar modelos de series temporales (ej. LSTMs) sobre el registro histórico para predecir picos de ocupación y sugerir horarios óptimos de visita a los usuarios.

Referencias Bibliográficas

- Agencia Nacional de Tránsito. (2023). Estadísticas de transporte [Sitio oficial consultado para el anuario y estadísticas del parque automotor]. <https://www.ant.gob.ec/estadisticas/>
- Almaguer-Segura, R., Sanchez-Lucero, A., & Rodriguez-Resendiz, J. (2023). Vehicle Detection and Classification Under Variable Illumination Conditions Using Deep Learning. *IEEE Access*.
- Almeida, P., Oliveira, L. S., Silva, A. S., Britto, A., & Koerich, A. L. (2015). PKLot – A Robust Dataset for Parking Lot Classification. *Expert Systems with Applications*, 42(11), 4937-4949. <https://doi.org/10.1016/j.eswa.2015.02.009>
- Almeida, P. R. L. d., Alves, J. H., Parpinelli, R. S., & Barddal, J. P. (2022). A Systematic Review on Computer Vision-Based Parking Lot Management Applied on Public Datasets [Entrada creada para resolver la cita Gupta2022 presente en el manuscrito]. *Expert Systems with Applications*, 198, 116731. <https://doi.org/10.1016/j.eswa.2022.116731>
- Asamblea Nacional del Ecuador. (2021). Ley Orgánica de Protección de Datos Personales. <https://www.registro-oficial.ec/index.php/registro-oficial-del-ecuador/vistas/registros-oficiales/registro-oficial-459-suplemento-quinto>
- Benlian, A., & Hess, T. (2011). Opportunities and Risks of Software-as-a-Service (SaaS): Findings from a Survey of IT Executives. *Decision Support Systems*, 52(1), 232-246. <https://doi.org/10.1016/j.dss.2011.07.007>

- Bhatia, S., Sharma, A., & Kumar, R. (2021). A Survey of Intelligent Parking Systems based on Image Processing [El DOI que estaba en la versión anterior apuntaba a otro artículo distinto, por eso se eliminó hasta verificar la fuente correcta]. *Journal of Ambient Intelligence and Humanized Computing*, 12, 7205-7221.
- Bouskela, M., Casseb, M., Bassi, S., De Luca, C., & Facchina, M. (2016). *La ruta hacia las smart cities: Migrando de una gestión tradicional a la ciudad inteligente*. Banco Interamericano de Desarrollo. <https://doi.org/10.18235/0012831>
- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
- Chen, Q., Niforatos, E., & Kortuem, G. (2025). Retrieval-Augmented Conversational Agents for Specialized Knowledge Domains. *International Journal of Human-Computer Interaction*, 41(1), 1-18. <https://doi.org/10.1080/10447318.2024.2315670>
- Dale, R. (2023). The Evolution of Conversational AI: From Rule-based Chatbots to LLMs. *Natural Language Engineering*, 29(3), 439-444. <https://doi.org/10.1017/S135132492300008X>
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319-340. <https://doi.org/10.2307/249008>
- Do, T.-T.-T., Huynh, Q.-T., Kim, K., & Nguyen, V.-Q. (2025). A Survey on Video Big Data Analytics: Architecture, Technologies, and Open Research Challenges [Entrada creada para resolver la cita Nguyen2025 presente en el manuscrito]. *Applied Sciences*, 15(14), 8089. <https://doi.org/10.3390/app15148089>
- Fadhil, A. (2021). *User Experience Design for Chatbots* [La versión previa tenía un DOI que apuntaba a otra obra distinta; se eliminó hasta verificar la fuente exacta].

- Fernandez-Ramos, M., & Lopez-Nozal, G. (2025). Analysis of Performance in Python Microservices for Image Inference. *Software: Practice and Experience*.
- Fowler, M. (2023). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- Grinberg, M. (2024). *Flask Web Development: Developing Web Applications with Python* (3.^a ed.). O'Reilly Media.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, 199, 1066-1073. <https://doi.org/10.1016/j.procs.2022.01.135>
- Jocher, G., Chaurasia, A., & Jing, Q. (2023). *Ultralytics YOLOv8* (Ver. 8.0). <https://github.com/ultralytics/ultralytics>
- Kumar, N., Singh, R., & Sharma, S. (2021). Infrastructure-less Smart Parking based on Deep Learning. *Sustainable Computing: Informatics and Systems*, 31, 100572. <https://doi.org/10.1016/j.suscom.2021.100572>
- Likert, R. (1932). A Technique for the Measurement of Attitudes. *Archives of Psychology*, 22(140), 1-55.
- Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
- Ministerio de Telecomunicaciones y de la Sociedad de la Información. (2024). Agenda de Transformación Digital del Ecuador 2022–2025. <https://www.telecomunicaciones.gob.ec/wp-content/uploads/2022/08/Agenda-transformacion-digital-2022-2025.pdf>
- Moussa, H., & Yassine, A. (2023). IoT-Based Smart Parking System Using Deep Learning. *Sensors*, 23(15), 6874.

- Myers, G. J., Sandler, C., & Badgett, T. (2023). *The Art of Software Testing* (4.^a ed.). John Wiley & Sons.
- NestJS. (2023). *Documentation | NestJS – A Progressive Node.js Framework*. <https://docs.nestjs.com/>
- Nguyen, T., & Do, P. (2024). User Acceptance of Social Messaging Bots in Public Services. *Computers in Human Behavior*, 151, 108012. <https://doi.org/10.1016/j.chb.2023.108012>
- Nielsen, J. (2022). *Ten Usability Heuristics for User Interface Design*. Nielsen Norman Group. <https://www.nngroup.com/articles/ten-usability-heuristics/>
- OpenCV Team. (2024). *OpenCV Documentation*. <https://docs.opencv.org/4.x/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- Ries, E. (2021). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.
- Sari, E., & Kurniawan, D. (2022). Containerization for AI: A DevOps Approach. *IEEE Access*, 10, 15670-15682. <https://doi.org/10.1109/ACCESS.2022.3148920>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Singh, H., & Kaur, M. (2023). Efficiency of IoT Edge Computing in Smart Urban Environments. *Computer Communications*, 201, 45-58. <https://doi.org/10.1016/j.comcom.2023.01.015>

- Sommerville, I. (2011). *Software Engineering* (9.^a ed.). Addison-Wesley.
- Sultana, R., & Hussain, S. (2022). Parking Slot Occupancy Detection Using Deep Learning: A Systematic Review. *Multimedia Tools and Applications*, 81, 31401-31435. <https://doi.org/10.1007/s11042-022-12948-4>
- Talavera, J., Quiñones, L., & Ghezzi, C. (2022). Evolutionary Architectures for Software Startups. *IEEE Software*, 39(3), 45-53. <https://doi.org/10.1109/MS.2022.3149870>
- Vila, J., Martí, E., & Herrero, R. (2020). Evaluation of Edge Computing and Vision-based Systems for Smart Parking. *IEEE Access*, 8, 153120-153132. <https://doi.org/10.1109/ACCESS.2020.3018260>
- Wang, Y., & Zhou, Z. (2023). Adaptive Camera Calibration for Traffic Scene Analysis. *The Visual Computer*, 39, 2451-2465. <https://doi.org/10.1007/s00371-022-02554-x>
- Wei, X., & Sun, Y. (2024). A Framework for Reproducible Software Engineering in Applied Deep Learning. *Information and Software Technology*, 166, 107352. <https://doi.org/10.1016/j.infsof.2023.107352>
- Zhang, L., & Wu, J. (2024). Dynamic Parking Space Management via Real-time Video Analytics. *Journal of Network and Computer Applications*, 221, 103789. <https://doi.org/10.1016/j.jnca.2023.103789>
- Zhao, H., & Miller, K. (2026). AI-Driven Urban Infrastructure: Future of Automated Parking. *Technological Forecasting and Social Change*, 200, 123145. <https://doi.org/10.1016/j.techfore.2025.123145>

Apéndice A: Resultados de la Encuesta de Usabilidad y Aceptación

Instrumentos de Recolección de Datos (Escala Likert)

El proceso de recolección de datos se llevó a cabo durante la fase de validación de campo en la clínica Medical Plus. Tras la fase de depuración de datos, se confirmaron 5 casos válidos que cumplieron con los criterios de inclusión: interacción efectiva con el chatbot y visualización del dashboard administrativo. Esta muestra ($N = 5$) es consistente con los registros de actividad del sistema y sustenta los análisis estadísticos presentados en el capítulo de Resultados.

A continuación, se presentan los resultados tabulados de la encuesta aplicada a la muestra de usuarios ($N=5$). Estas evidencias sustentan los análisis estadísticos presentados en el capítulo de Resultados.

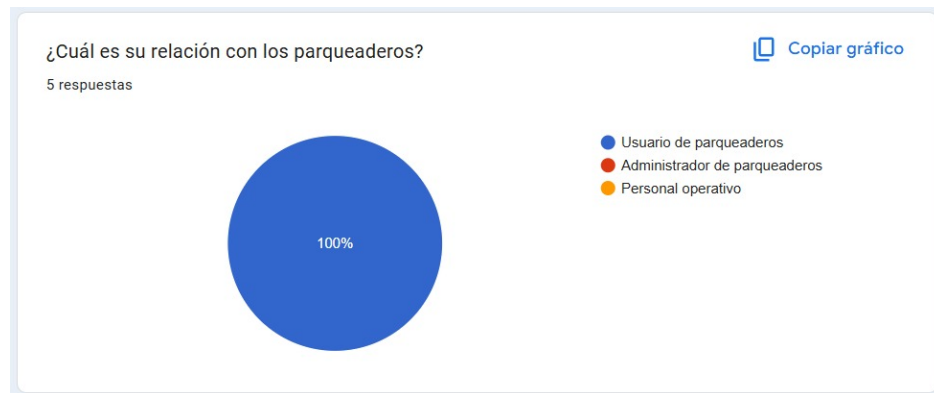


Figura 16. Perfil de los encuestados: Uso de parqueaderos.

Nota. Identificación del grupo objetivo y su relación con el problema planteado.

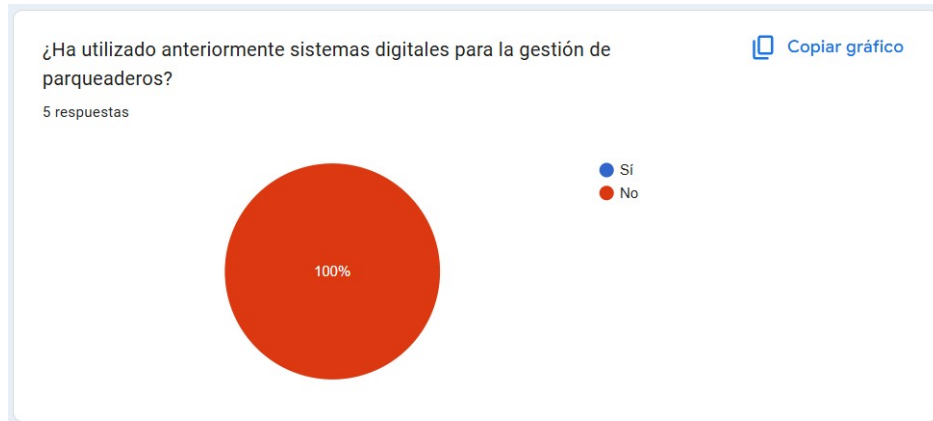


Figura 17. Experiencia previa con sistemas digitales de gestión de parking.

Nota. Evidencia del rezago tecnológico previo a la implementación del prototipo.

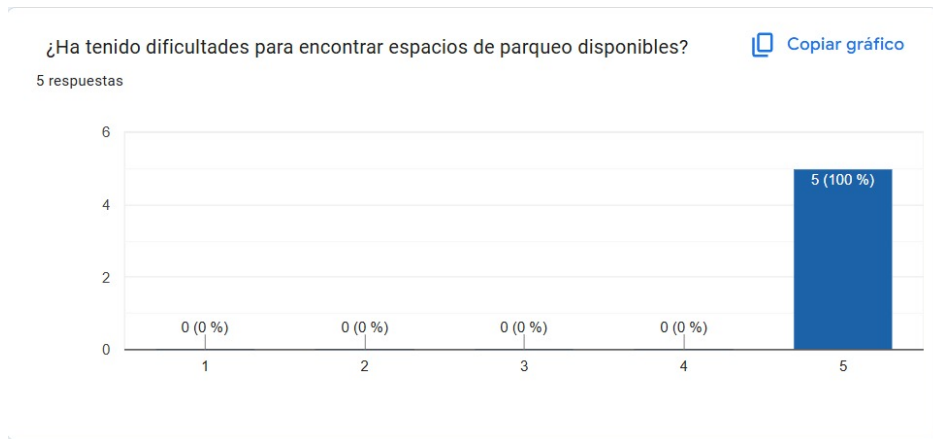


Figura 18. Dificultad percibida para encontrar espacios disponibles.

Nota. Validación de la necesidad del sistema Parkintia.

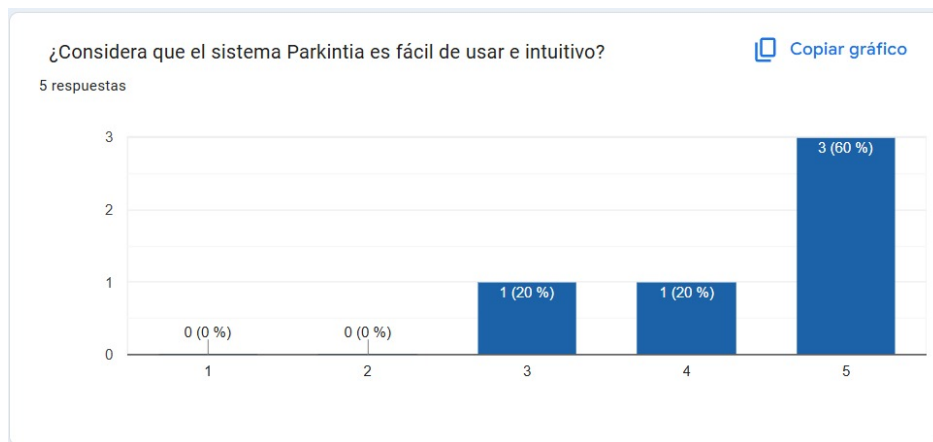


Figura 19. Valoración de la facilidad de uso del Chatbot.

Nota. Resultados cuantitativos sobre la interfaz intuitiva del sistema.

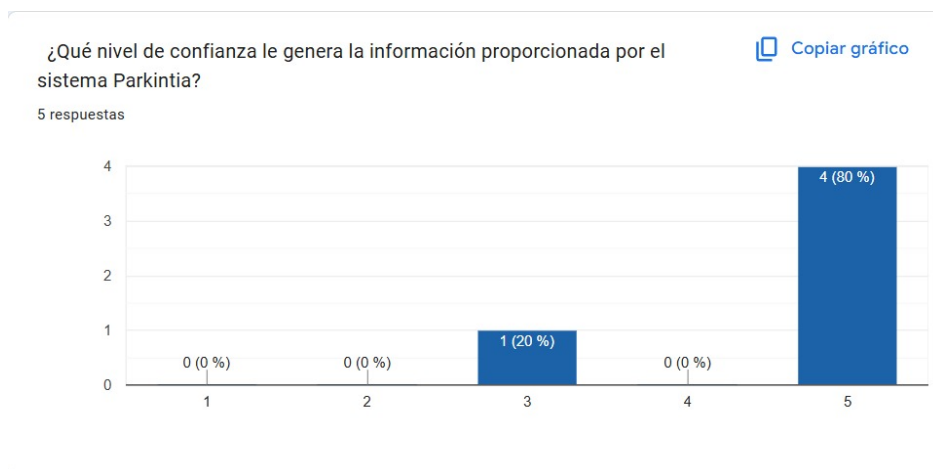


Figura 20. Nivel de confianza en la información técnica proporcionada.

Nota. Percepción de fiabilidad del algoritmo por parte del usuario.

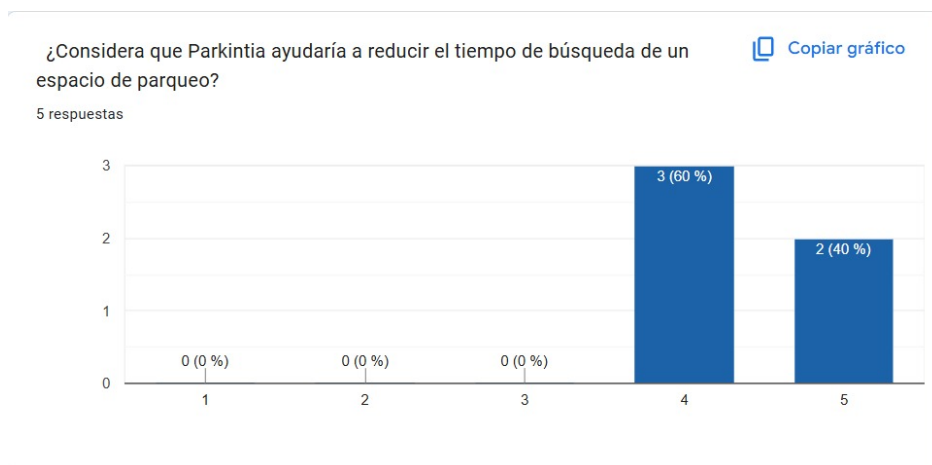


Figura 21. Impacto percibido en la reducción del tiempo de búsqueda.

Nota. Evaluación del cumplimiento del objetivo funcional principal.

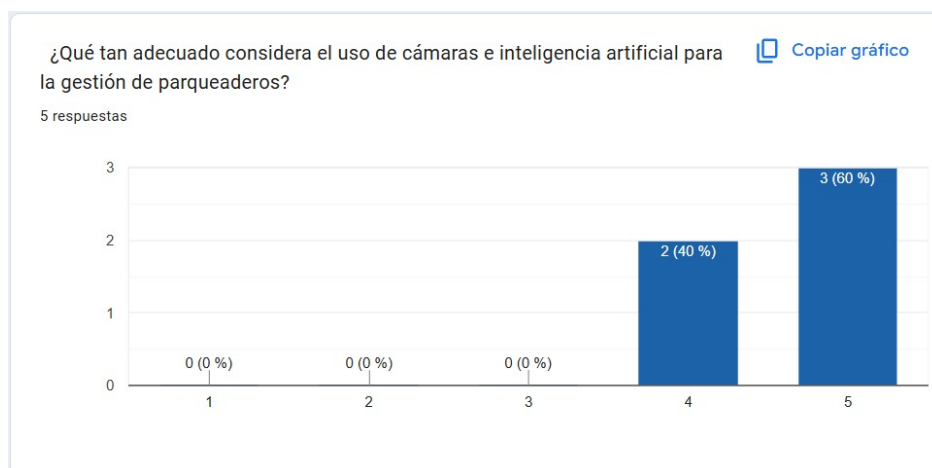


Figura 22. Adecuación ética y técnica del uso de cámaras e IA.

Nota. Aceptación de la infraestructura basada en visión artificial.

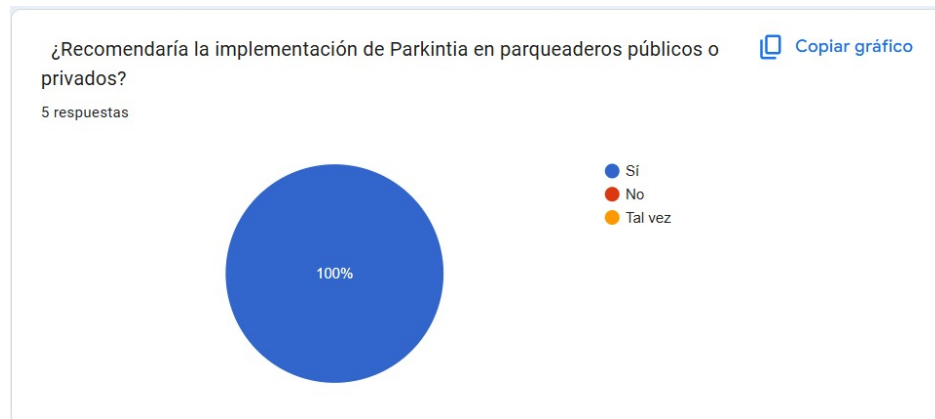


Figura 23. Intención de recomendación y adopción del sistema.

Nota. Medición de la viabilidad de mercado del MVP.

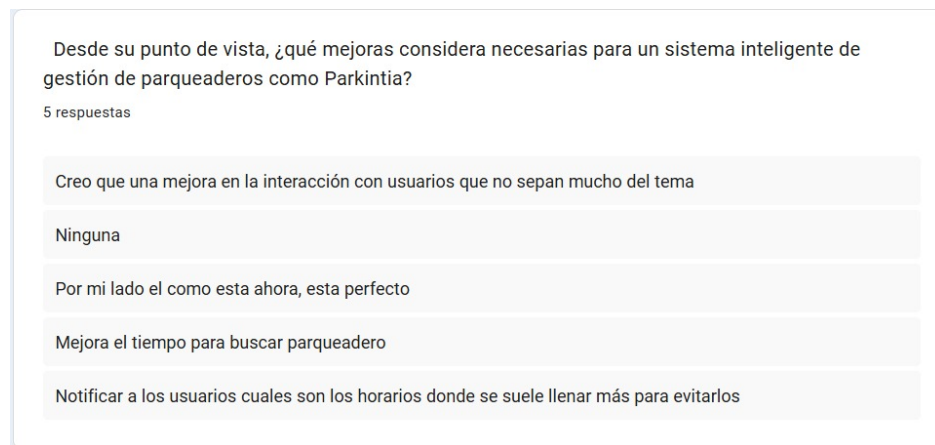


Figura 24. Sugerencias de mejora y retroalimentación abierta.

Nota. Recopilación de observaciones cualitativas para futuras iteraciones del software.

Apéndice B: Evidencias de Gestión y Pruebas Técnicas Adicionales

Evidencias de Comercialización y Acceso al Sistema

Como parte de los requisitos para la modalidad de emprendimiento tecnológico, se documenta la presencia del producto en plataformas digitales para su futura comercialización y el control de acceso para usuarios registrados.



Figura 25. Landing Page de Parkintia: Plataforma orientada a la comercialización (Tienda Virtual).

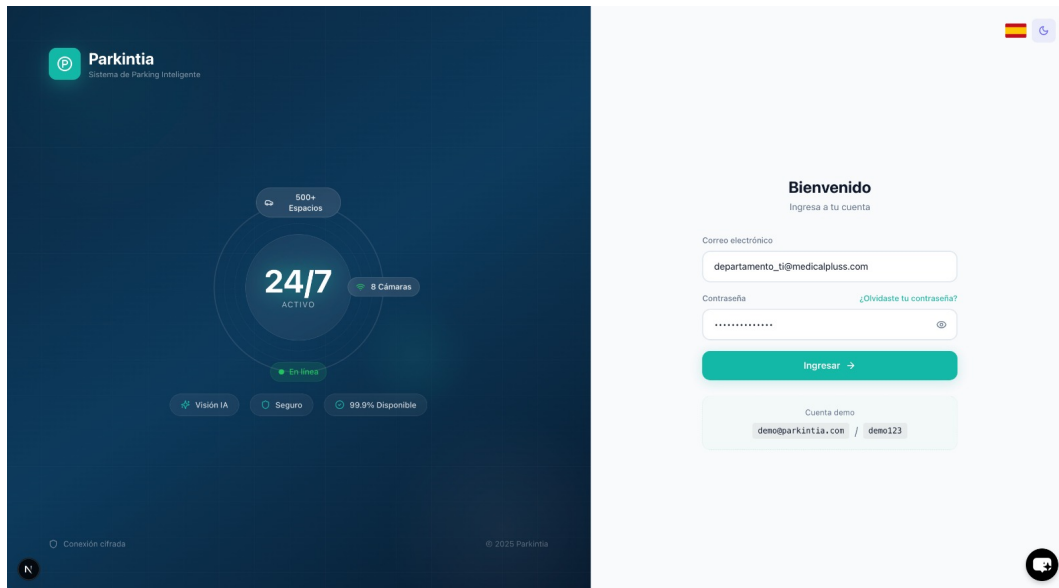


Figura 26. Interfaz de Autenticación: Sistema de seguridad y acceso para clientes SaaS.

Evidencia de Implementación en Entorno Real (Medical Plus)

Se documenta el despliegue realizado en la clínica Medical Plus, utilizando la infraestructura de cámaras IP para la validación del MVP en condiciones reales.

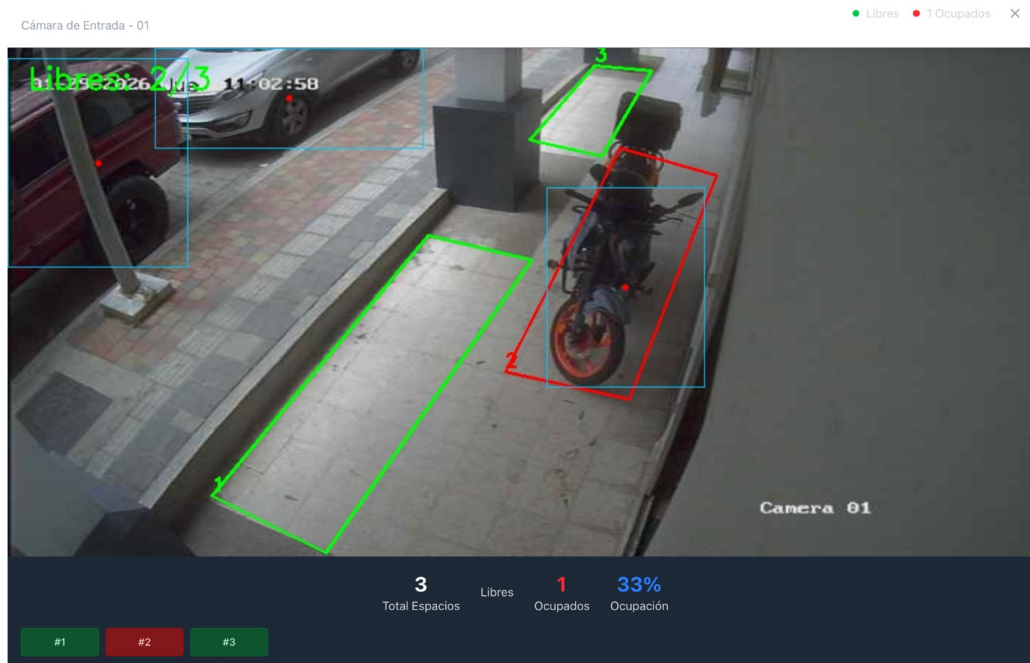


Figura 27. Configuración de ROIs: Delimitación de espacios de parqueo en el software de monitoreo.

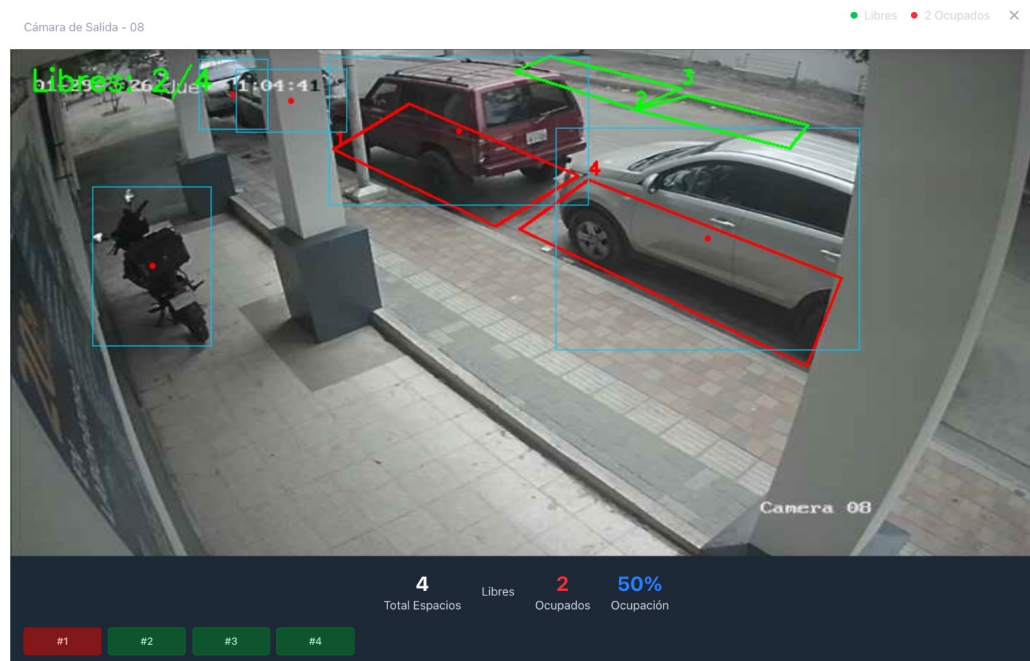


Figura 28. Detección en sitio: Vista panorámica del área de estacionamiento procesada por la IA.

Gestión Ágil del Proyecto (Metodología Scrum)

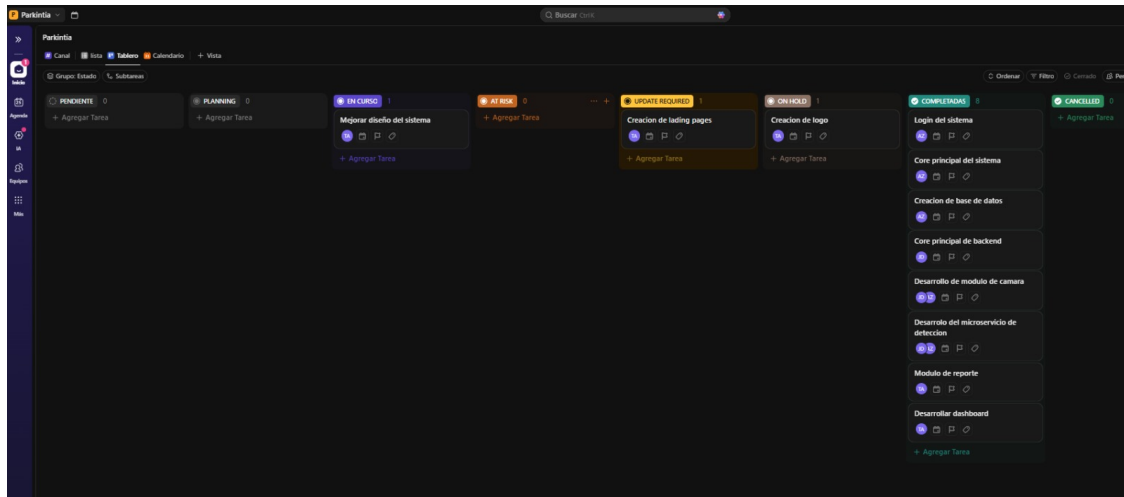


Figura 29. Tablero de gestión de tareas (Sprint Backlog) del proyecto.

Auditoría de Logs y Estructura del Código

```
==== INICIANDO MEDICIÓN DE RENDIMIENTO (TESIS) ====  
Sistema Operativo: Darwin 22.6.0  
Procesador: i386  
RAM Total: 8.0 GB  
Tiempo de carga del modelo: 0.12s  
venv /Volumes/externo/Tesis/backend-hackathon/python-detection-service git:(develop) v24.11.1
```

Figura 30. Logs de inicialización y carga del modelo YOLOv8s.

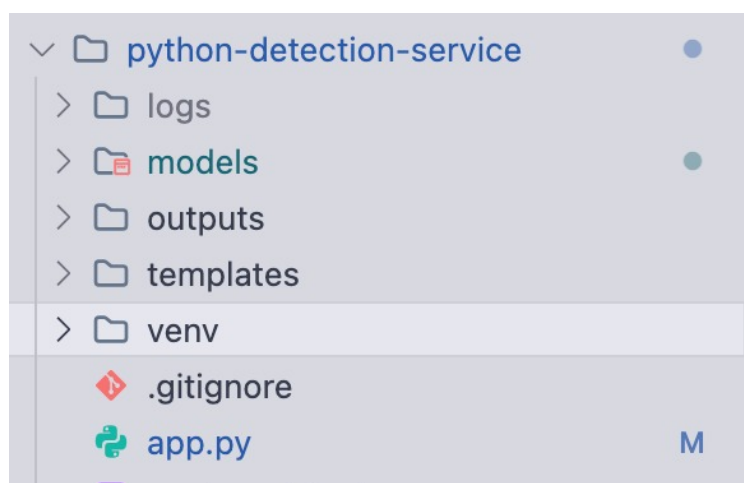


Figura 31. Estructura de directorios del subsistema de Inteligencia Artificial.

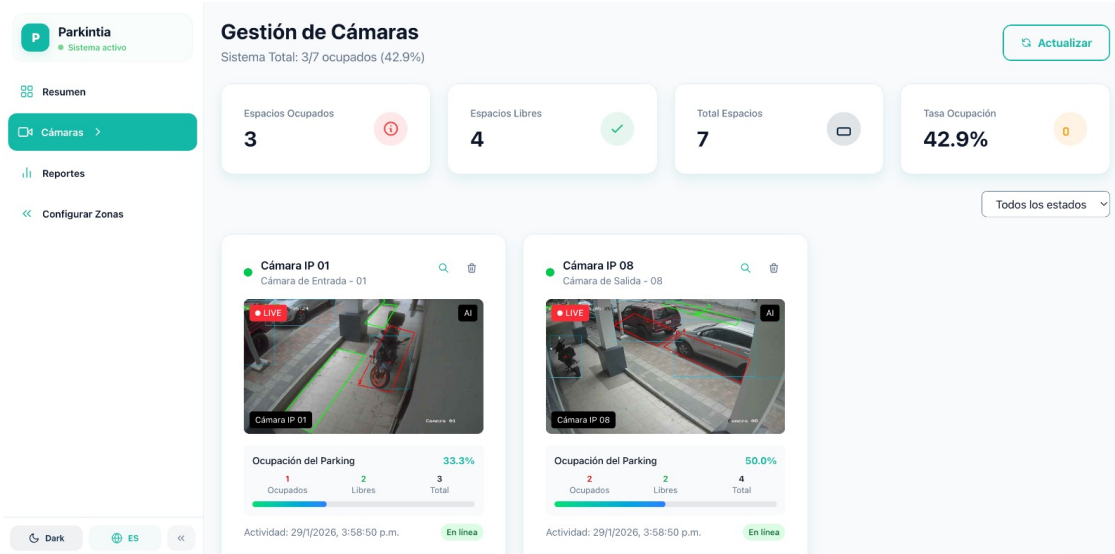


Figura 32. Vista general del Dashboard administrativo de Parkintia.

Apéndice C: Artefactos de Pruebas Unitarias (Caja Blanca)

Especificación del Script de Pruebas Automatizadas

Como sustento técnico de la validación lógica del sistema, se presenta la implementación del script de pruebas desarrollado bajo el framework **Jest**. Este artefacto evidencia el uso de *mocks* para simular la base de datos y la verificación de los caminos críticos del servicio de mensajería.

```
backend-hackathon > src > chatbot > TS chatbot.service.spec.ts > ...
1  import { Test, TestingModule } from '@nestjs/testing';
2  import { ChatbotService } from './chatbot.service';
3  import { ConfigService } from '@nestjs/config';
4  import { CameraService } from '../camera/camera.service';
5  import { DataSource } from 'typeorm';
6  import axios from 'axios';
7
8  // Mock de Axios
9  jest.mock('axios');
10 const mockedAxios = axios as jest.Mocked<typeof axios>;
11
12 describe('ChatbotService (Prueba de Caja Blanca)', () => {
13   let service: ChatbotService;
14   let cameraService: CameraService;
15
16   const mockCameraService = {
17     findAll: jest.fn(),
18   };
19
20   const mockConfigService = {
21     get: jest.fn().mockReturnValue('http://localhost:5000'),
22   };
23
24   const mockDataSource = {};
25
26   beforeEach(async () => {
27     const module: TestingModule = await Test.createTestingModule({
28       providers: [
29         ChatbotService,
30         { provide: CameraService, useValue: mockCameraService },
31         { provide: ConfigService, useValue: mockConfigService },
32         { provide: DataSource, useValue: mockDataSource },
33       ],
34     }).compile();
```

Figura 33. Código fuente de la prueba unitaria para el método sendMessage.

Nota. Implementación de la suite de pruebas que garantiza la cobertura de decisiones.